

1,001 CODING PROMPTS

Automate your Software Development with AI & ChatGPT Vibe Coding

Mammoth Club Official Guide PRO+

- ✓ FREE Online Course
- ✓ FREE Cheatsheet
- ✓ FREE Official Mammoth Club Certificate



MAMMOTH CLUB



Written by Alex Kropf • Edited by John Bura

Cover Design by Jared Matson & John Bura • Contributions by James Dabalus

Powered by  CoursePro.ai

*From the creators of the best-selling Hello Coding: Anyone Can
Learn to Code & more*

Praise for Mammoth Club

I have completed many tutorials. This one is the most outstanding one that I have seen thus far. It is doubtful that it could be topped. This is a superior tutorial. Amazing. —Joseph A., Mammoth Club Student

Exactly what I wanted! Just enough BASIC information without being technically overwhelming and intimidating. —Paul V., Mammoth Club Student

This course so far is by far amazing! The instructor is very encouraging and upbeat, and his instructions are very clear. It's an amazing course. —Moiz S., Mammoth Club Student

It's scary to think that by following these instructional videos I can be equipped with the skills to program Python. —Charles E., Mammoth Club Student

I ended up taking it and it was INCREDIBLE. They set great challenges that build off what was taught in the lecture, but don't directly give you the answer. It asks you to extend your knowledge and refer to the right documentation. So good for learning. —A_Unicycle, Mammoth Club Student

This is AMAZING! I just learned how to code without breaking a sweat, this is really easy and fun! —Shalonda L., Mammoth Club Student

Clear instructions and excellent projects. —Ian F., Mammoth Club Student



MAMMOTH CLUB



Go to [MammothClub.com](https://www.mammothclub.com)

to access 3,000+ online courses and 5,000+ hours of video content!

Mammoth Club is a leading online course provider in everything from learning to code to becoming a YouTube star. Since 2011, Mammoth Club has built a global student community with over 9 million courses sold.



Scan the QR code or go this link to redeem your free course and cheat sheet:

www.mammothclub.com/course/1-hour-coding/UNLOCKED

Mammoth Club books can be purchased at a special discount when ordered in bulk for promotional giveaways, fundraisers, or educational initiatives. Customized editions or selected excerpts can also be produced to meet specific needs. For more information, please reach out to support@mammothinteractive.com.

Portions of this book may be shared promotionally if with direct citation to [MammothClub.com](https://www.mammothclub.com). This book may not be reproduced — mechanically, electronically, or by any other means, including photocopying — without written permission of the publisher.



The publisher does not provide medical, legal, accounting, or other professional services. Readers seeking such expertise should consult a qualified professional. This book is not meant to be used for clinical procedures or medical treatment. To the maximum extent permitted by law, the publisher and editors are not responsible for any harm or damage to individuals or property resulting from the use or misuse of the material presented herein. All rights reserved. This book does not constitute financial, investment, legal, or tax advice. You are solely responsible for your financial decisions. We make no guarantees of income, business outcomes, or investment returns. By using this book, you agree that the author and publisher cannot be held liable for any loss, damage, or results arising from actions you take based on its content.

Written by Alex Kropf • Proofread and Edited by John Bura • Online Course, Exam and Cheatsheet by James Dabalus • Cover Design by Jared Matson and John Bura • Interior Book Design, Outline Development, Layout Design and Interior Illustrations by Alex Kropf • Copyright © 2025 by Mammoth Club

Table of Contents

Praise for Mammoth Club	2
Go to MammothClub.com	3
Welcome, Vibe Coder	5
PART 1	6
801+ LLM PROMPTS FOR SOFTWARE DEVELOPERS	6
PART 2	159
100 CODING INTERVIEW PROMPTS - MASTER SOLUTION PATTERNS	159
How to Use These Prompts	159
Two Pointers Pattern (1-10)	160
Hash Map/Set Patterns (11-20)	161
Sliding Window Pattern (21-30)	163
Binary Search Pattern (31-40)	165
Depth-First Search (DFS) Pattern (41-50)	166
Breadth-First Search (BFS) Pattern (51-60)	168
Dynamic Programming Fundamentals (61-70)	170
Greedy Algorithm Patterns (71-80)	171
Backtracking Pattern (81-90)	173
Tree Traversal Mastery (91-100)	175
PART 3	176
270 RAPID PROMPTS - ULTIMATE PROGRAMMING CHALLENGE LIST	176
Beginner Level	176
Advanced Level	181
Where To Go From Here	190
GET THE FREE ONLINE COURSE & CERTIFICATE	190
About Your Author	191
Note From Your Author	191
Visit MammothClub.com	192

Welcome, Vibe Coder

Welcome to the Mammoth Club experience—where the future of coding meets the power of artificial intelligence. You're not just opening a book. You're stepping into a new era of software development where high-performance thinking, fast problem solving, and creative engineering converge.

What You'll Find Inside

This isn't a random collection of prompts. It's a curated developer resource—a playbook for turning AI into your co-pilot and crushing complex problems with clarity and confidence.

Here's a preview of what's ahead:

- **801+ LLM Prompts for Software Developers**
Organized by theme and purpose—ready for use with any large language model.
- **100 Coding Interview Prompts with Master Solution Patterns**
Master essential algorithm strategies like Two Pointers, DFS, Sliding Window, Binary Search, and Dynamic Programming.
- **270 Rapid Prompts for Daily Challenges**
Break out of your comfort zone and tackle real-world scenarios, from beginner to advanced.

Each section is designed to be practical, repeatable, and impactful. No filler—just tools you can apply immediately.

Why You?

Because you're the kind of developer who knows that growth doesn't happen by accident. You've got the curiosity to explore and the ambition to improve. Whether you're just starting out or deep in your career, the tools in this book will evolve with you.

Use them to brainstorm faster, debug more precisely, explore design patterns, or ace your next interview. Your toolkit starts here.

Get Ready to Build Big

This is your moment. A new generation of development is here—powered by AI, driven by purpose, and fueled by creators like you.

So, welcome to the Mammoth Club. Open your editor. Load up your favorite model. And let's redefine what's possible in software development. Let's begin.

PART 1

801+ LLM Prompts for Software Developers

Each prompt maintains the three-level structure with increasing complexity and includes relevant placeholders for customization!

Build a To-Do List Application

Beginner: How do I build a simple to-do list application?

Intermediate: I want to build a to-do list application. My tech stack is [STACK: e.g., "React and Node.js"]. I need features like [FEATURES: e.g., "add, delete, mark complete"]. I have [YOUR EXPERIENCE LEVEL] and want a step-by-step guide including project structure, key components, and deployment options.

Advanced: Act as a senior full-stack developer with 10+ years of experience. Design a scalable to-do list application following SOLID principles and clean architecture. MY REQUIREMENTS: Tech stack: [FRONTEND/BACKEND] Database: [SQL/NoSQL] Authentication: [METHOD] Features: [LIST FEATURES] User capacity: [EXPECTED USERS] Include:

- Architecture diagrams
- API design with RESTful/GraphQL endpoints
- State management strategy
- Testing approach (unit, integration, e2e)

- CI/CD pipeline setup
- Performance optimization techniques Format as a technical specification document.

Implement Binary Search Algorithm

Beginner: How do I implement a binary search algorithm?

Intermediate: I want to implement binary search in [LANGUAGE: e.g., "Python"]. My use case is [USE CASE: e.g., "searching sorted user IDs"]. I need to handle [EDGE CASES: e.g., "duplicates, empty arrays"] and want both iterative and recursive implementations with time/space complexity analysis.

Advanced: Act as an algorithms expert. Implement a generic binary search framework that handles: Data type: [TYPE] Comparison function: [CUSTOM/DEFAULT] Search variants: [EXACT/LOWER_BOUND/UPPER_BOUND] Performance requirements: [TIME/SPACE CONSTRAINTS] Include:

- Template/generic implementation
- Benchmark comparisons
- Memory optimization techniques
- Parallel search strategies
- Real-world application examples Provide production-ready code with comprehensive error handling.

Create REST API Endpoints

Beginner: How do I create REST API endpoints?

Intermediate: I want to create REST API endpoints for [RESOURCE: e.g., "user management"]. Using [FRAMEWORK: e.g., "Express.js"], I need to implement [OPERATIONS: e.g., "CRUD operations"]. Include validation, error handling, and basic authentication setup.

Advanced: Act as an API architect. Design a RESTful API following OpenAPI 3.0 specification for: Domain: [BUSINESS DOMAIN] Resources: [LIST OF RESOURCES]

Authentication: [OAuth2/JWT/API Keys] Rate limiting: [REQUESTS PER MINUTE]
Versioning strategy: [URL/HEADER] Include:

- HATEOAS implementation
 - Pagination strategies
 - Caching mechanisms
 - API documentation generation
 - Monitoring and analytics
 - GraphQL migration path
- Format as API design document with examples.

Build a Web Scraper

Beginner: How do I build a web scraper?

Intermediate: I want to build a web scraper for [WEBSITE TYPE: e.g., "e-commerce sites"]. Using [LANGUAGE/LIBRARY: e.g., "Python with BeautifulSoup"], I need to extract [DATA: e.g., "product prices and reviews"]. Include handling for pagination and rate limiting.

Advanced: Act as a web scraping specialist. Design a distributed scraping system for: Target sites: [DOMAINS] Data volume: [PAGES/DAY] Compliance: [ROBOTS.TXT/LEGAL] Storage: [DATABASE/CLOUD] Scaling needs: [CONCURRENT REQUESTS] Include:

- Anti-detection techniques
 - Proxy rotation strategy
 - Queue management (Redis/RabbitMQ)
 - Data validation pipeline
 - Monitoring and alerting
 - Legal compliance checklist
- Provide architecture with code examples.

Implement User Authentication

Beginner: How do I implement user authentication?

Intermediate: I want to implement user authentication using [METHOD: e.g., "JWT tokens"]. My stack is [TECH STACK] and I need [FEATURES: e.g., "login, logout, password reset"]. Include security best practices and session management.

Advanced: Act as a security engineer. Design a comprehensive authentication system with: Auth methods: [PASSWORD/OAUTH/2FA/BIOMETRIC] User base: [SIZE] Compliance: [GDPR/HIPAA/SOC2] Infrastructure: [CLOUD/ON-PREMISE] SSO requirements: [SAML/OIDC] Include:

- Zero-trust architecture
- Token rotation strategy
- Audit logging
- Fraud detection
- Account recovery flows
- Performance benchmarks Format as security design document.

Deploy Application to Cloud

Beginner: How do I deploy my application to the cloud?

Intermediate: I want to deploy my [APP TYPE: e.g., "Node.js application"] to [CLOUD PROVIDER: e.g., "AWS"]. My app needs [REQUIREMENTS: e.g., "database, file storage"]. I have [BUDGET] and need a deployment guide with cost optimization tips.

Advanced: Act as a DevOps architect. Design a cloud deployment strategy for: Application: [TYPE/STACK] Traffic: [REQUESTS/SEC] Availability: [SLA PERCENTAGE] Regions: [GEOGRAPHICAL NEEDS] Budget: [MONTHLY LIMIT] Include:

- Infrastructure as Code (Terraform/CloudFormation)
- Container orchestration (K8s/ECS)
- Blue-green deployment
- Disaster recovery plan
- Cost optimization matrix

- Monitoring stack design Provide implementation roadmap.

Write Unit Tests

Beginner: How do I write unit tests for my code?

Intermediate: I want to write unit tests for [CODE TYPE: e.g., "React components"]. Using [TESTING FRAMEWORK: e.g., "Jest"], I need to test [SCENARIOS: e.g., "user interactions, API calls"]. Include mocking strategies and coverage goals.

Advanced: Act as a QA architect. Design a comprehensive testing strategy for: Application type: [WEB/MOBILE/API] Codebase size: [KLOC] Team size: [DEVELOPERS] Coverage target: [PERCENTAGE] CI/CD integration: [TOOL] Include:

- Test pyramid implementation
- Contract testing approach
- Mutation testing setup
- Performance test suite
- Chaos engineering tests
- ROI metrics and reporting Format as QA strategy document.

Optimize Database Queries

Beginner: How do I optimize database queries?

Intermediate: I want to optimize queries for my [DATABASE: e.g., "PostgreSQL"] database. My problematic queries involve [OPERATIONS: e.g., "joins on large tables"]. Current performance is [METRICS] and I need to achieve [TARGET].

Advanced: Act as a database performance expert. Optimize a database system with: Database: [TYPE/VERSION] Data volume: [SIZE] Query patterns: [OLTP/OLAP/MIXED] Concurrency: [USERS] SLA: [RESPONSE TIME] Include:

- Index strategy design
- Partitioning approach

- Query plan analysis
- Connection pooling setup
- Caching layer design
- Monitoring and alerting Provide before/after benchmarks.

Build Real-time Chat Application

Beginner: How do I build a real-time chat application?

Intermediate: I want to build a real-time chat app using [TECHNOLOGY: e.g., "WebSockets"]. Features needed include [FEATURES: e.g., "private messages, typing indicators"]. My stack is [STACK] and I need scalability for [USER COUNT] concurrent users.

Advanced: Act as a real-time systems architect. Design a chat platform supporting: Users: [CONCURRENT] Features: [VIDEO/VOICE/FILE SHARING] Latency: [TARGET MS] Reliability: [UPTIME SLA] Compliance: [E2E ENCRYPTION/GDPR] Include:

- WebRTC implementation
- Message queue design
- Presence system architecture
- Offline message handling
- Load balancing strategy
- Database sharding plan Format as technical architecture document.

Implement Caching Strategy

Beginner: How do I implement caching in my application?

Intermediate: I want to implement caching for [APPLICATION TYPE]. Using [CACHING SOLUTION: e.g., "Redis"], I need to cache [DATA TYPES: e.g., "API responses, user sessions"]. Include cache invalidation strategies and monitoring setup.

Advanced: Act as a caching expert. Design a multi-layer caching strategy for: Application: [TYPE] Traffic: [REQUESTS/SEC] Data volatility: [UPDATE

FREQUENCY] Cache size: [MEMORY BUDGET] Consistency: [EVENTUAL/STRONG] Include:

- CDN configuration
- Application-level caching
- Database query caching
- Cache warming strategies
- Invalidation patterns
- Performance metrics Provide implementation guide with code.

Create Data Visualization Dashboard

Beginner: How do I create a data visualization dashboard?

Intermediate: I want to create a dashboard for [DATA TYPE: e.g., "sales metrics"]. Using [LIBRARY: e.g., "D3.js or Chart.js"], I need to visualize [METRICS: e.g., "trends, comparisons"]. Include responsive design and real-time updates.

Advanced: Act as a data visualization architect. Design an interactive dashboard system for: Data sources: [APIs/DATABASES] Update frequency: [REAL-TIME/BATCH] Users: [CONCURRENT] Visualizations: [TYPES NEEDED] Performance: [LOAD TIME] Include:

- Data pipeline architecture
- WebGL optimization
- Drill-down capabilities
- Export functionality
- Mobile optimization
- A/B testing framework Format as technical specification.

Handle File Uploads

Beginner: How do I handle file uploads in my web application?

Intermediate: I want to handle file uploads for [FILE TYPES: e.g., "images and documents"]. My stack is [TECH STACK] and I need to support files up to [SIZE LIMIT]. Include validation, progress tracking, and storage options.

Advanced: Act as a file handling expert. Design a scalable file upload system supporting: File types: [ALLOWED TYPES] Size limit: [MAX SIZE] Volume: [FILES/DAY] Storage: [S3/CLOUD/LOCAL] Processing: [THUMBNAILS/SCANNING] Include:

- Chunked upload implementation
- Virus scanning integration
- CDN distribution
- Backup strategy
- Compliance requirements
- Cost optimization Provide architecture with code examples.

Implement Search Functionality

Beginner: How do I implement search functionality?

Intermediate: I want to implement search for [CONTENT TYPE: e.g., "products in e-commerce"]. Using [TECHNOLOGY: e.g., "Elasticsearch"], I need features like [FEATURES: e.g., "autocomplete, filters"]. Include relevance tuning and performance optimization.

Advanced: Act as a search engineer. Design a comprehensive search system with: Data volume: [DOCUMENTS] Languages: [SUPPORTED] Features: [FACETS/AUTOCOMPLETE/NLP] Latency: [TARGET MS] Availability: [SLA] Include:

- Index design strategy
- Query optimization
- Relevance tuning
- A/B testing framework
- Analytics integration

- Scaling strategy Format as search architecture document.

Build Microservices Architecture

Beginner: How do I build a microservices architecture?

Intermediate: I want to convert my [MONOLITH TYPE] application to microservices. Planning to use [TECHNOLOGIES: e.g., "Docker, Kubernetes"]. I need to split into [NUMBER] services handling [DOMAINS]. Include communication patterns and deployment strategy.

Advanced: Act as a microservices architect. Design a distributed system with: Services: [COUNT AND PURPOSES] Communication: [SYNC/ASYNCR] Data consistency: [PATTERN] Deployment: [PLATFORM] Team structure: [CONWAY'S LAW] Include:

- Service mesh design
- Saga pattern implementation
- Distributed tracing
- Circuit breaker patterns
- API gateway design
- Migration strategy Provide detailed architecture blueprint.

Implement Data Encryption

Beginner: How do I implement data encryption?

Intermediate: I want to encrypt [DATA TYPE: e.g., "user passwords and personal data"]. Using [LANGUAGE/Framework], I need encryption for [AT REST/IN TRANSIT]. Include key management and compliance considerations.

Advanced: Act as a cryptography expert. Design an encryption system for: Data types: [PII/FINANCIAL/HEALTH] Compliance: [STANDARDS] Performance: [THROUGHPUT] Key management: [HSM/KMS] Algorithms: [APPROVED LIST] Include:

- Encryption at rest strategy

- TLS configuration
- Key rotation automation
- Secure key storage
- Audit logging
- Quantum-resistant planning Format as security implementation guide.

Create CI/CD Pipeline

Beginner: How do I create a CI/CD pipeline?

Intermediate: I want to create a CI/CD pipeline for my [PROJECT TYPE]. Using [TOOLS: e.g., "GitHub Actions, Jenkins"], I need to automate [PROCESSES: e.g., "testing, building, deployment"]. Include environment management and rollback strategies.

Advanced: Act as a DevOps engineer. Design a complete CI/CD pipeline with:
Environments: [DEV/STAGING/PROD] Testing stages: [UNIT/INTEGRATION/E2E]
Deployment: [BLUE-GREEN/CANARY] Monitoring: [TOOLS] Compliance: [AUDIT NEEDS] Include:

- GitOps implementation
- Secret management
- Artifact management
- Performance gates
- Security scanning
- Cost tracking Provide implementation roadmap.

Handle Concurrent Requests

Beginner: How do I handle concurrent requests in my application?

Intermediate: I want to handle concurrent requests in my [APPLICATION TYPE]. Expected load is [REQUESTS/SEC] with [CONCURRENT USERS]. Using [LANGUAGE/Framework], I need strategies for thread safety and resource management.

Advanced: Act as a concurrency expert. Design a high-concurrency system handling:
Request rate: [REQUESTS/SEC] Concurrent users: [COUNT] Resource constraints:
[CPU/MEMORY] Latency requirements: [P99] Error budget: [PERCENTAGE]
Include:

- Thread pool optimization
- Lock-free algorithms
- Backpressure handling
- Resource pooling
- Monitoring strategy
- Load testing plan Format as performance engineering guide.

Build GraphQL API

Beginner: How do I build a GraphQL API?

Intermediate: I want to build a GraphQL API for [DOMAIN: e.g., "e-commerce platform"]. Using [FRAMEWORK: e.g., "Apollo Server"], I need to implement [FEATURES: e.g., "queries, mutations, subscriptions"]. Include schema design and resolver patterns.

Advanced: Act as a GraphQL architect. Design a federated GraphQL system for:
Services: [MICROSERVICES] Schema size: [TYPES] Query complexity: [LIMITS]
Performance: [LATENCY] Security: [AUTHORIZATION] Include:

- Schema stitching strategy
- DataLoader implementation
- Query cost analysis
- Subscription scaling
- Schema versioning
- Migration from REST Provide architecture and code examples.

Implement Logging System

Beginner: How do I implement logging in my application?

Intermediate: I want to implement logging for my [APPLICATION TYPE]. Using [LOGGING LIBRARY: e.g., "Winston, Log4j"], I need to log [EVENT TYPES: e.g., "errors, user actions, performance"]. Include log levels, formatting, and storage strategies.

Advanced: Act as an observability engineer. Design a centralized logging system for: Applications: [COUNT] Volume: [LOGS/SEC] Retention: [DAYS] Compliance: [REQUIREMENTS] Analysis: [USE CASES] Include:

- Log aggregation pipeline
 - Structured logging format
 - Search and analytics
 - Alerting rules
 - Cost optimization
 - Security considerations
- Format as logging architecture guide.

Create Responsive Web Design

Beginner: How do I create a responsive web design?

Intermediate: I want to create a responsive design for [WEBSITE TYPE: e.g., "portfolio site"]. Using [FRAMEWORK: e.g., "Bootstrap, Tailwind CSS"], I need to support [DEVICES: e.g., "mobile, tablet, desktop"]. Include performance optimization and accessibility features.

Advanced: Act as a frontend architect. Design a responsive system supporting: Breakpoints: [CUSTOM SIZES] Performance: [CORE WEB VITALS] Accessibility: [WCAG LEVEL] Browsers: [SUPPORT MATRIX] Features: [OFFLINE/PWA] Include:

- CSS architecture strategy
- Image optimization pipeline
- Font loading strategy

- Animation performance
- Testing methodology
- Build optimization Provide implementation guide with examples.

Handle Background Jobs

Beginner: How do I handle background jobs?

Intermediate: I want to implement background jobs for [TASKS: e.g., "email sending, report generation"]. Using [QUEUE SYSTEM: e.g., "Redis Queue, Celery"], I need to process [VOLUME] jobs with [PRIORITY LEVELS]. Include retry logic and monitoring.

Advanced: Act as a distributed systems engineer. Design a job processing system handling: Job types: [CATEGORIES] Volume: [JOBS/HOUR] Priority levels: [COUNT] Reliability: [SLA] Resources: [WORKERS] Include:

- Queue architecture design
- Worker scaling strategy
- Dead letter queue handling
- Job scheduling system
- Monitoring and alerting
- Cost optimization Format as technical architecture document.

Implement Rate Limiting

Beginner: How do I implement rate limiting?

Intermediate: I want to implement rate limiting for my [API TYPE]. I need to limit [RESOURCE: e.g., "API calls"] to [LIMITS: e.g., "100 requests per minute"]. Using [TECHNOLOGY], include different limiting strategies and user feedback.

Advanced: Act as an API security expert. Design a rate limiting system with: Strategies: [TOKEN BUCKET/SLIDING WINDOW] Limits: [PER USER/IP/ENDPOINT] Distributed: [REDIS/HAZELCAST] Bypass rules: [WHITELIST] Response: [429 HANDLING] Include:

- Algorithm comparison
- Distributed implementation
- Dynamic limit adjustment
- Analytics integration
- DDoS mitigation
- SLA management Provide implementation with benchmarks.

Build Email Service

Beginner: How do I build an email service?

Intermediate: I want to build an email service for [PURPOSE: e.g., "transactional emails"]. Using [SERVICE: e.g., "SendGrid, AWS SES"], I need to send [VOLUME] emails daily with [FEATURES: e.g., "templates, tracking"]. Include deliverability best practices.

Advanced: Act as an email infrastructure expert. Design an email system handling: Volume: [EMAILS/DAY] Types: [TRANSACTIONAL/MARKETING] Compliance: [CAN-SPAM/GDPR] Deliverability: [TARGET RATE] Features: [TRACKING/AB TESTING] Include:

- Multi-provider strategy
- Template management system
- Bounce handling
- Reputation monitoring
- Analytics pipeline
- Cost optimization Format as email architecture guide.

Create Mobile App Backend

Beginner: How do I create a backend for a mobile app?

Intermediate: I want to create a backend for my [MOBILE PLATFORM: e.g., "iOS and Android"] app. Using [BACKEND FRAMEWORK], I need to support [FEATURES: e.g., "push notifications, offline sync"]. Include API design and security considerations.

Advanced: Act as a mobile backend architect. Design a backend system supporting: Platforms: [iOS/ANDROID/WEB] Users: [CONCURRENT] Features: [PUSH/SYNC/ANALYTICS] Offline: [STRATEGY] Security: [REQUIREMENTS] Include:

- BaaS vs custom comparison
- Push notification architecture
- Offline data sync strategy
- App versioning support
- Analytics integration
- Performance optimization Provide architecture blueprint.

Implement WebSocket Communication

Beginner: How do I implement WebSocket communication?

Intermediate: I want to implement WebSocket communication for [USE CASE: e.g., "live notifications"]. Using [LIBRARY: e.g., "Socket.io"], I need to handle [CONNECTIONS] concurrent connections with [FEATURES: e.g., "rooms, broadcasting"]. Include reconnection strategies.

Advanced: Act as a real-time communication expert. Design a WebSocket system supporting: Connections: [CONCURRENT] Messages: [PER SECOND] Reliability: [DELIVERY GUARANTEE] Scaling: [HORIZONTAL/VERTICAL] Features: [PRESENCE/HISTORY] Include:

- Load balancer configuration
- Session affinity handling
- Message ordering guarantees
- Fallback mechanisms
- Performance tuning

- Monitoring strategy Format as technical specification.

Build OAuth Integration

Beginner: How do I build OAuth integration?

Intermediate: I want to integrate OAuth for [PROVIDERS: e.g., "Google, GitHub"]. My application is [TYPE] and I need to implement [FLOW: e.g., "authorization code flow"]. Include token management and security best practices.

Advanced: Act as an identity expert. Design an OAuth system supporting: Providers: [LIST] Flows: [AUTHORIZATION/IMPLICIT/CLIENT] Users: [SCALE] Features: [SSO/FEDERATION] Compliance: [STANDARDS] Include:

- Multi-tenant architecture
- Token lifecycle management
- Consent management
- Account linking strategy
- Security audit checklist
- Migration planning Provide implementation guide.

Build Calendar System

Beginner: How do I build a calendar system?

Intermediate: I want to build a calendar for [APPLICATION TYPE]. Supporting [FEATURES: e.g., "events, recurring, reminders"], I need to handle [USER COUNT] with [TIMEZONE REQUIREMENTS]. Include conflict detection and sharing capabilities.

Advanced: Act as a calendar architect. Design scheduling system with: Users: [SCALE] Events: [VOLUME/COMPLEXITY] Features: [RECURRING/EXCEPTIONS] Timezones: [HANDLING STRATEGY] Integration: [EXTERNAL CALENDARS] Include:

- Recurrence rule engine
- Timezone architecture

- Conflict resolution
- Availability calculation
- Sync strategy
- Performance optimization Format as system design.

Implement IP Geolocation

Beginner: How do I implement IP geolocation?

Intermediate: I want to implement IP-based location for [USE CASE: e.g., "content localization"]. Using [SERVICE/DATABASE: e.g., "MaxMind"], I need to process [REQUEST VOLUME] with [ACCURACY REQUIREMENTS]. Include fallback strategies and caching.

Advanced: Act as a geolocation expert. Design IP location system with: Requests: [VOLUME/SEC] Accuracy: [REQUIREMENTS] Updates: [DATABASE REFRESH] Privacy: [COMPLIANCE] Performance: [LATENCY] Include:

- Database architecture
- Update pipeline
- Accuracy validation
- Privacy compliance
- Caching strategy
- Fallback mechanisms Provide implementation roadmap.

Create Markdown Editor

Beginner: How do I create a markdown editor?

Intermediate: I want to build a markdown editor with [FEATURES: e.g., "live preview, syntax highlighting"]. Using [FRAMEWORK], I need to support [EXTENSIONS: e.g., "tables, diagrams"] with [PERFORMANCE REQUIREMENTS]. Include toolbar and export options.

Advanced: Act as an editor architect. Design markdown system with: Features: [WYSIWYG/EXTENSIONS] Performance: [DOCUMENT SIZE] Collaboration: [REAL-TIME] Export: [FORMATS] Plugins: [ARCHITECTURE] Include:

- Parser architecture
- Extension system
- Performance optimization
- Collaboration engine
- Export pipeline
- Plugin marketplace Format as technical design.

Build Inventory Management

Beginner: How do I build inventory management?

Intermedicate: I want to build inventory tracking for [BUSINESS TYPE]. Managing [PRODUCT COUNT] with [LOCATION COUNT], I need features like [FEATURES: e.g., "stock alerts, transfers"]. Include real-time updates and reporting.

Advanced: Act as an inventory architect. Design inventory system for: SKUs: [COUNT] Locations: [WAREHOUSES/STORES] Transactions: [VOLUME/DAY] Integration: [ERP/POS] Features: [FORECASTING/OPTIMIZATION] Include:

- Multi-location architecture
- Transaction processing
- Stock calculation engine
- Alert system
- Analytics pipeline
- Integration framework Provide system blueprint.

Implement Captcha System

Beginner: How do I implement a captcha system?

Intermediate: I want to implement captcha for [USE CASE: e.g., "form submission"]. Using [SERVICE: e.g., "reCAPTCHA, hCaptcha"], I need to handle [REQUEST VOLUME] with [ACCESSIBILITY REQUIREMENTS]. Include fallback options and analytics.

Advanced: Act as a security expert. Design captcha system with: Challenges: [TYPES] Volume: [REQUESTS/DAY] Accessibility: [REQUIREMENTS] Performance: [IMPACT] Analytics: [METRICS] Include:

- Challenge selection
- Risk scoring
- Accessibility options
- Performance optimization
- Fraud detection
- A/B testing framework Format as security design.

Create Language Detection

Beginner: How do I create language detection?

Intermediate: I want to detect languages in [CONTENT TYPE: e.g., "user input, documents"]. Using [LIBRARY/API], I need to identify [LANGUAGE COUNT] with [ACCURACY REQUIREMENTS]. Include confidence scoring and mixed language handling.

Advanced: Act as an NLP expert. Design language detection for: Languages: [SUPPORTED] Content: [SIZE/TYPE] Accuracy: [REQUIREMENTS] Performance: [LATENCY] Features: [MIXED/CONFIDENCE] Include:

- Detection algorithms
- Model architecture
- Confidence calculation
- Mixed language handling
- Performance optimization

- Continuous improvement Provide NLP system design.

Build QR Code Generator

Beginner: How do I build a QR code generator?

Intermediate: I want to generate QR codes for [USE CASE: e.g., "payments, URLs"]. Supporting [DATA TYPES] with [CUSTOMIZATION: e.g., "logos, colors"], I need to generate [VOLUME] codes with [FEATURES]. Include error correction and tracking.

Advanced: Act as a QR architect. Design QR system supporting: Types: [DATA FORMATS] Volume: [CODES/DAY] Customization: [BRANDING] Tracking: [ANALYTICS] Security: [REQUIREMENTS] Include:

- Generation pipeline
- Customization engine
- Storage strategy
- Analytics integration
- Security measures
- Performance optimization Format as system architecture.

Implement Retry Logic

Beginner: How do I implement retry logic?

Intermediate: I want to implement retry logic for [OPERATION TYPE: e.g., "API calls, database queries"]. Using [PATTERN: e.g., "exponential backoff"], I need to handle [FAILURE SCENARIOS] with [TIMEOUT REQUIREMENTS]. Include circuit breaker pattern.

Advanced: Act as a resilience expert. Design retry system with: Operations: [TYPES] Strategies: [PATTERNS] Timeouts: [CONFIGURATION] Monitoring: [METRICS] Policies: [CUSTOMIZATION] Include:

- Retry architecture
- Backoff algorithms
- Circuit breaker integration

- Timeout handling
- Monitoring framework
- Policy engine Provide resilience guide.

Create Tag System

Beginner: How do I create a tag system?

Intermediate: I want to implement tagging for [CONTENT TYPE]. Supporting [TAG FEATURES: e.g., "hierarchical, auto-complete"], I need to handle [TAG COUNT] with [PERFORMANCE REQUIREMENTS]. Include tag suggestions and analytics.

Advanced: Act as a taxonomy expert. Design tagging system with: Tags: [COUNT/HIERARCHY] Content: [VOLUME] Features: [AUTO-TAG/SUGGEST] Search: [INTEGRATION] Analytics: [INSIGHTS] Include:

- Tag architecture
- Hierarchy management
- Auto-tagging engine
- Search integration
- Analytics pipeline
- Performance optimization Format as system design.

Build Version Control System

Beginner: How do I build a version control system?

Intermediate: I want to build version control for [CONTENT TYPE: e.g., "documents, configurations"]. Supporting [FEATURES: e.g., "branching, merging"], I need to handle [FILE SIZE] with [USER COUNT]. Include diff algorithms and conflict resolution.

Advanced: Act as a VCS architect. Design version control supporting: Files: [SIZE/COUNT] Users: [CONCURRENT] Features: [BRANCHES/TAGS] Storage: [STRATEGY] Performance: [OPERATIONS/SEC] Include:

- Storage architecture
- Diff algorithms
- Merge strategies
- Distributed design
- Performance optimization
- Security model Provide VCS architecture.

Implement IP Blocking

Beginner: How do I implement IP blocking?

Intermediate: I want to implement IP blocking for [SECURITY PURPOSE]. Managing [IP COUNT] with [BLOCK TYPES: e.g., "temporary, permanent"], I need features like [FEATURES: e.g., "whitelist, geo-blocking"]. Include performance considerations.

Advanced: Act as a security architect. Design IP blocking with: IPs: [VOLUME] Rules: [COMPLEXITY] Performance: [LOOKUP TIME] Distribution: [EDGE/CENTRAL] Management: [INTERFACE] Include:

- Storage optimization
- Lookup algorithms
- Rule engine
- Geo-blocking
- Distributed sync
- Management API Format as security design.

Create Polling System

Beginner: How do I create a polling system?

Intermediate: I want to build polls for [PLATFORM TYPE]. Supporting [POLL TYPES: e.g., "single choice, multiple choice"], I need to handle [VOTE VOLUME] with [REAL-TIME REQUIREMENTS]. Include result visualization and fraud prevention.

Advanced: Act as a voting architect. Design polling system with: Polls: [CONCURRENT] Votes: [VOLUME/SEC] Security: [FRAUD PREVENTION] Results: [REAL-TIME/AGGREGATED] Features: [ANONYMOUS/WEIGHTED] Include:

- Vote storage architecture
- Real-time aggregation
- Fraud detection
- Result caching
- Analytics integration
- API design Provide system blueprint.

Build Sitemap Generator

Beginner: How do I build a sitemap generator?

Intermediate: I want to generate sitemaps for [WEBSITE TYPE]. With [PAGE COUNT] and [UPDATE FREQUENCY], I need to create [SITEMAP TYPES: e.g., "XML, image, video"]. Include crawling strategies and submission automation.

Advanced: Act as an SEO architect. Design sitemap system for: Pages: [COUNT/TYPES] Updates: [FREQUENCY] Features: [PRIORITY/LASTMOD] Distribution: [CDN/COMPRESSION] Monitoring: [COVERAGE] Include:

- Crawling architecture
- Generation pipeline
- Update detection
- Compression strategy
- Submission automation
- Monitoring dashboard Format as SEO system design.

Implement Request Signing

Beginner: How do I implement request signing?

Intermediate: I want to implement request signing for [API TYPE]. Using [ALGORITHM: e.g., "HMAC-SHA256"], I need to sign [REQUEST PARTS: e.g., "headers, body"] with [KEY MANAGEMENT]. Include replay attack prevention.

Advanced: Act as an API security expert. Design signing system with: Algorithms: [SUPPORTED] Coverage: [HEADERS/BODY] Keys: [ROTATION STRATEGY] Replay: [PREVENTION] Performance: [OVERHEAD] Include:

- Signature architecture
- Key management
- Replay prevention
- Clock skew handling
- Performance optimization
- Migration strategy Provide security implementation.

Create Leaderboard System

Beginner: How do I create a leaderboard system?

Intermediate: I want to build leaderboards for [GAME/APP TYPE]. Tracking [METRIC TYPES] for [USER COUNT], I need features like [FEATURES: e.g., "real-time updates, time periods"]. Include ranking algorithms and caching.

Advanced: Act as a gaming architect. Design leaderboard system with: Players: [COUNT] Metrics: [TYPES/COMPLEXITY] Updates: [FREQUENCY] Periods: [DAILY/WEEKLY/ALL-TIME] Features: [TOURNAMENTS/SEASONS] Include:

- Storage architecture
- Ranking algorithms
- Real-time updates
- Caching strategy
- Cheat detection
- API design Format as gaming infrastructure.

Build Contact Form

Beginner: How do I build a contact form?

Intermediate: I want to build a contact form with [FIELDS: e.g., "name, email, message"]. Including [FEATURES: e.g., "file attachments, captcha"], I need to handle [SUBMISSION VOLUME] with [SPAM PROTECTION]. Include email delivery and storage.

Advanced: Act as a form architect. Design contact system with: Forms: [TYPES/COUNT] Volume: [SUBMISSIONS/DAY] Validation: [RULES] Delivery: [CHANNELS] Analytics: [TRACKING] Include:

- Form builder architecture
- Validation framework
- Spam detection
- Multi-channel delivery
- Analytics integration
- GDPR compliance Provide form system design.

Implement Data Masking

Beginner: How do I implement data masking?

Intermediate: I want to mask [DATA TYPES: e.g., "PII, credit cards"] for [PURPOSE: e.g., "logs, testing"]. Using [TECHNIQUES: e.g., "tokenization, encryption"], I need to maintain [REQUIREMENTS: e.g., "format, reversibility"]. Include performance impact.

Advanced: Act as a privacy expert. Design masking system for: Data: [TYPES/VOLUME] Techniques: [METHODS] Reversibility: [REQUIREMENTS] Performance: [OVERHEAD] Compliance: [STANDARDS] Include:

- Masking architecture
- Algorithm selection
- Key management

- Performance optimization
- Audit capabilities
- Compliance validation Format as privacy design.

Create Subscription System

Beginner: How do I create a subscription system?

Intermediate: I want to build subscriptions for [SERVICE TYPE]. Supporting [PLAN TYPES] with [BILLING CYCLES], I need features like [FEATURES: e.g., "trials, upgrades, cancellations"]. Include payment integration and dunning.

Advanced: Act as a billing architect. Design subscription platform with: Plans: [COMPLEXITY] Customers: [SCALE] Billing: [PROVIDERS/CYCLES] Features: [TRIALS/ADDONS] Compliance: [REGULATIONS] Include:

- Billing engine design
- Plan management
- Proration logic
- Dunning process
- Revenue recognition
- Analytics pipeline Provide billing architecture.

Build URL Router

Beginner: How do I build a URL router?

Intermediate: I want to build routing for [FRAMEWORK/LANGUAGE]. Supporting [ROUTE PATTERNS: e.g., "parameters, wildcards"], I need to handle [ROUTE COUNT] with [PERFORMANCE REQUIREMENTS]. Include middleware and error handling.

Advanced: Act as a routing expert. Design router supporting: Routes: [COUNT/COMPLEXITY] Patterns: [REGEX/WILDCARDS] Performance: [LOOKUPS/SEC] Features: [MIDDLEWARE/GROUPS] Memory: [CONSTRAINTS] Include:

- Route matching algorithms

- Parameter extraction
- Middleware architecture
- Performance optimization
- Route generation
- Testing framework Format as router design.

Implement Infinite Scroll

Beginner: How do I implement infinite scroll?

Intermediate: I want to implement infinite scroll for [CONTENT TYPE]. Loading [ITEMS PER PAGE] from [DATA SOURCE], I need to handle [TOTAL ITEMS] with [PERFORMANCE REQUIREMENTS]. Include loading states and SEO considerations.

Advanced: Act as a frontend architect. Design infinite scroll with: Items: [TOTAL/PER PAGE] Performance: [RENDER TIME] SEO: [REQUIREMENTS] Accessibility: [STANDARDS] Features: [VIRTUAL SCROLL] Include:

- Virtualization strategy
- Loading optimization
- SEO solutions
- Accessibility features
- State management
- Performance monitoring Provide implementation guide.

Create Barcode Scanner

Beginner: How do I create a barcode scanner?

Intermediate: I want to build barcode scanning for [PLATFORM: e.g., "web, mobile"]. Supporting [BARCODE TYPES] with [CAMERA REQUIREMENTS], I need features like [FEATURES: e.g., "batch scanning, history"]. Include performance optimization.

Advanced: Act as a scanning expert. Design barcode system with: Types: [1D/2D CODES] Performance: [SCANS/SEC] Accuracy: [REQUIREMENTS] Platforms: [WEB/MOBILE] Features: [BATCH/OFFLINE] Include:

- Camera integration
- Decoding pipeline
- Performance tuning
- Offline capability
- History management
- Analytics integration Format as scanning architecture.

Build Status Page

Beginner: How do I build a status page?

Intermediate: I want to create a status page for [SERVICE COUNT] services. Showing [METRICS: e.g., "uptime, response time"], I need features like [FEATURES: e.g., "incidents, maintenance windows"]. Include automated updates and notifications.

Advanced: Act as a reliability architect. Design status platform with: Services: [COUNT/TYPES] Metrics: [COLLECTION/DISPLAY] Incidents: [MANAGEMENT] Subscribers: [SCALE] Integration: [MONITORING TOOLS] Include:

- Architecture design
- Metric aggregation
- Incident workflow
- Notification system
- Historical data
- API design Provide status system blueprint.

Implement Dark Mode

Beginner: How do I implement dark mode?

Intermediate: I want to add dark mode to [APPLICATION TYPE]. Supporting [THEME FEATURES: e.g., "auto-switch, custom colors"], I need to handle [COMPONENT COUNT] with [PERSISTENCE REQUIREMENTS]. Include transition animations and accessibility.

Advanced: Act as a UI architect. Design theming system with: Themes: [COUNT/CUSTOMIZATION] Components: [LIBRARY SIZE] Performance: [SWITCH TIME] Accessibility: [CONTRAST REQUIREMENTS] Persistence: [STRATEGY] Include:

- Theme architecture
- Color system design
- Component integration
- Performance optimization
- Accessibility validation
- Testing strategy Format as theming guide.

Create Export System

Beginner: How do I create an export system?

Intermediate: I want to export [DATA TYPES] to [FORMATS: e.g., "CSV, PDF, Excel"]. Processing [DATA VOLUME] with [FEATURES: e.g., "filters, templates"], I need to handle large exports with progress tracking.

Advanced: Act as a data export expert. Design export system for: Data: [VOLUME/TYPES] Formats: [SUPPORTED] Scale: [CONCURRENT EXPORTS] Features: [TEMPLATES/SCHEDULING] Storage: [TEMPORARY/PERMANENT] Include:

- Export architecture
- Format converters
- Queue management
- Progress tracking
- Template engine
- Storage optimization Provide export platform design.

Build Referral System

Beginner: How do I build a referral system?

Intermediate: I want to create referrals for [BUSINESS TYPE]. With [REWARD STRUCTURE] and [TRACKING REQUIREMENTS], I need features like [FEATURES: e.g., "unique codes, analytics"]. Include fraud prevention and payout management.

Advanced: Act as a growth architect. Design referral platform with: Users: [SCALE] Rewards: [STRUCTURE/TIERS] Tracking: [ATTRIBUTION] Fraud: [PREVENTION] Analytics: [METRICS] Include:

- Referral architecture
- Code generation
- Attribution tracking
- Fraud detection
- Reward engine
- Analytics pipeline Format as growth system design.

Implement Text Search

Beginner: How do I implement text search?

Intermediate: I want to search through [CONTENT TYPE] with [DATA VOLUME]. Using [SEARCH METHOD: e.g., "full-text, fuzzy"], I need features like [FEATURES: e.g., "highlighting, facets"]. Include relevance tuning and performance optimization.

Advanced: Act as a search architect. Design text search for: Content: [VOLUME/TYPES] Queries: [COMPLEXITY/QPS] Features: [STEMMING/SYNONYMS] Languages: [SUPPORTED] Performance: [LATENCY] Include:

- Index architecture
- Query processing
- Relevance scoring
- Language handling

- Performance tuning
- Analytics integration Provide search implementation.

Create Slideshow Component

Beginner: How do I create a slideshow component?

Intermediate: I want to build a slideshow for [CONTENT TYPE: e.g., "images, videos"]. With [FEATURES: e.g., "autoplay, thumbnails, captions"], I need to support [PLATFORM REQUIREMENTS] with touch gestures and keyboard navigation.

Advanced: Act as a component architect. Design slideshow system with: Media: [TYPES/SIZE] Features: [TRANSITIONS/EFFECTS] Performance: [LOAD TIME] Accessibility: [REQUIREMENTS] Platforms: [DESKTOP/MOBILE] Include:

- Component architecture
- Media loading strategy
- Animation engine
- Touch handling
- Accessibility features
- Performance optimization Format as component design.

Build Coupon System

Beginner: How do I build a coupon system?

Intermediate: I want to create coupons for [E-COMMERCE TYPE]. Supporting [DISCOUNT TYPES: e.g., "percentage, fixed, BOGO"], I need to handle [VALIDATION RULES] with [USAGE LIMITS]. Include code generation and tracking.

Advanced: Act as an e-commerce architect. Design coupon platform with: Types: [DISCOUNT STRUCTURES] Rules: [COMPLEXITY] Scale: [REDEMPTIONS/DAY] Validation: [REAL-TIME] Analytics: [TRACKING] Include:

- Coupon architecture
- Rule engine design

- Validation pipeline
- Usage tracking
- Fraud prevention
- Analytics integration Provide coupon system design.

Implement Social Login

Beginner: How do I implement social login?

Intermediate: I want to add social login for [PROVIDERS: e.g., "Google, Facebook, GitHub"]. Handling [USER SCALE] with [FEATURES: e.g., "account linking, data import"], I need to manage multiple providers and user data.

Advanced: Act as an identity architect. Design social auth system with: Providers: [COUNT/TYPES] Users: [SCALE] Features: [LINKING/IMPORT] Privacy: [DATA HANDLING] Compliance: [GDPR/CCPA] Include:

- Auth architecture
- Account linking logic
- Data synchronization
- Privacy controls
- Provider abstraction
- Migration strategy Format as identity design.

Create Bookmark System

Beginner: How do I create a bookmark system?

Intermediate: I want to build bookmarking for [CONTENT TYPE]. Supporting [FEATURES: e.g., "folders, tags, sharing"], I need to handle [BOOKMARK COUNT] per user with [SYNC REQUIREMENTS]. Include search and organization features.

Advanced: Act as a bookmark architect. Design bookmarking platform with: Users: [SCALE] Bookmarks: [PER USER] Features: [COLLECTIONS/TAGS] Sync: [REAL-TIME/DEVICES] Sharing: [PUBLIC/PRIVATE] Include:

- Storage architecture
- Organization system
- Sync strategy
- Search integration
- Sharing mechanism
- Performance optimization Provide bookmark system design.

Build Activity Feed

Beginner: How do I build an activity feed?

Intermediate: I want to create an activity feed for [APPLICATION TYPE]. Showing [EVENT TYPES] for [USER COUNT], I need features like [FEATURES: e.g., "aggregation, filtering"]. Include real-time updates and personalization.

Advanced: Act as a feed architect. Design activity system with: Events: [TYPES/VOLUME] Users: [FOLLOWERS/FOLLOWING] Delivery: [PUSH/PULL] Features: [AGGREGATION/RANKING] Scale: [EVENTS/SEC] Include:

- Feed architecture
- Fanout strategy
- Aggregation rules
- Ranking algorithm
- Caching approach
- Performance optimization Format as feed system design.

Implement Code Syntax Highlighting

Beginner: How do I implement syntax highlighting?

Intermediate: I want to add syntax highlighting for [LANGUAGES: e.g., "JavaScript, Python"]. Using [LIBRARY: e.g., "Prism, Highlight.js"], I need features like [FEATURES: e.g., "line numbers, themes"]. Include performance optimization for large files.

Advanced: Act as a code editor expert. Design highlighting system with: Languages: [COUNT/COMPLEXITY] Performance: [FILE SIZE LIMITS] Features: [THEMES/PLUGINS] Accuracy: [TOKENIZATION] Integration: [EDITORS] Include:

- Parser architecture
 - Theme system
 - Language detection
 - Performance optimization
 - Plugin framework
 - Accessibility features
- Format as highlighting design.

Create Feedback Widget

Beginner: How do I create a feedback widget?

Intermediate: I want to build a feedback widget for [APPLICATION TYPE]. Collecting [FEEDBACK TYPES: e.g., "ratings, comments, bugs"], I need features like [FEATURES: e.g., "screenshots, metadata"]. Include analytics and response management.

Advanced: Act as a feedback architect. Design feedback system with: Types: [FEEDBACK CATEGORIES] Volume: [SUBMISSIONS/DAY] Features: [SCREENSHOTS/VIDEO] Analysis: [SENTIMENT/CATEGORIZATION] Integration: [TICKETING/CRM] Include:

- Widget architecture
 - Collection pipeline
 - Analysis engine
 - Response workflow
 - Analytics dashboard
 - Integration framework
- Provide feedback platform design.

Build Discussion Forum

Beginner: How do I build a discussion forum?

Intermediate: I want to create a forum for [COMMUNITY TYPE]. Supporting [FEATURES: e.g., "categories, threads, voting"], I need to handle [USER COUNT] with [MODERATION REQUIREMENTS]. Include search and notification features.

Advanced: Act as a community architect. Design forum platform with: Users: [ACTIVE/TOTAL] Content: [POSTS/DAY] Features: [VOTING/BADGES/REPUTATION] Moderation: [AUTO/MANUAL] Scale: [CONCURRENT USERS] Include:

- Forum architecture
- Threading system
- Reputation engine
- Moderation pipeline
- Search integration
- Performance optimization Format as community platform design.

Implement Print Styles

Beginner: How do I implement print styles?

Intermediate: I want to create print styles for [DOCUMENT TYPE]. Optimizing [CONTENT ELEMENTS] for printing, I need features like [FEATURES: e.g., "page breaks, headers/footers"]. Include print preview and PDF generation options.

Advanced: Act as a print design expert. Design print system with: Documents: [TYPES/COMPLEXITY] Features: [PAGINATION/LAYOUT] Formats: [PAPER SIZES] Quality: [DPI REQUIREMENTS] Integration: [PDF GENERATION] Include:

- Print architecture
- Layout engine
- Page break logic

- Print preview
- PDF pipeline
- Testing strategy Provide print design guide.

Create Badge System

Beginner: How do I create a badge system?

Intermediate: I want to implement badges for [APPLICATION TYPE]. With [BADGE TYPES: e.g., "achievements, levels, streaks"], I need to track [CRITERIA] and award [USER COUNT]. Include progress tracking and notifications.

Advanced: Act as a gamification architect. Design badge system with: Badges: [COUNT/TYPES] Criteria: [COMPLEXITY] Users: [SCALE] Features: [PROGRESS/RARITY] Analytics: [ENGAGEMENT METRICS] Include:

- Badge architecture
- Achievement engine
- Progress tracking
- Notification system
- Analytics pipeline
- Admin interface Format as gamification design.

Build Time Tracking System

Beginner: How do I build a time tracking system?

Intermediate: I want to track time for [USE CASE: e.g., "projects, tasks"]. Supporting [FEATURES: e.g., "timers, manual entry, reports"], I need to handle [USER COUNT] with [INTEGRATION REQUIREMENTS]. Include billing and export features.

Advanced: Act as a time tracking architect. Design time system with: Users: [SCALE] Projects: [HIERARCHY] Features: [AUTOMATION/IDLE] Reports: [TYPES/COMPLEXITY] Integration: [BILLING/PAYROLL] Include:

- Tracking architecture

- Timer implementation
- Report engine
- Billing integration
- Export pipeline
- Mobile sync Provide time tracking design.

Implement Breadcrumb Navigation

Beginner: How do I implement breadcrumb navigation?

Intermediate: I want to add breadcrumbs to [WEBSITE TYPE]. Supporting [HIERARCHY DEPTH] with [DYNAMIC GENERATION], I need features like [FEATURES: e.g., "schema markup, responsive"]. Include SEO optimization and accessibility.

Advanced: Act as a navigation architect. Design breadcrumb system with: Structure: [DEPTH/COMPLEXITY] Generation: [DYNAMIC/STATIC] Features: [SCHEMA/TRUNCATION] SEO: [STRUCTURED DATA] Accessibility: [ARIA LABELS] Include:

- Breadcrumb architecture
- Generation strategy
- Schema implementation
- Responsive design
- Accessibility features
- Performance optimization Format as navigation guide.

Create Share Buttons

Beginner: How do I create share buttons?

Intermediate: I want to add sharing for [PLATFORMS: e.g., "Twitter, Facebook, LinkedIn"]. With [CONTENT TYPES] and [TRACKING REQUIREMENTS], I need features like [FEATURES: e.g., "custom messages, analytics"]. Include native app handling.

Advanced: Act as a social sharing expert. Design share system with: Platforms: [COUNT/APIs] Content: [TYPES/CUSTOMIZATION] Tracking: [ANALYTICS/UTM] Performance: [LOAD TIME] Privacy: [GDPR COMPLIANCE] Include:

- Share architecture
- Platform integration
- Analytics pipeline
- Native app handling
- Privacy controls
- A/B testing setup Format as sharing design.

Build Changelog System

Beginner: How do I build a changelog system?

Intermediate: I want to create a changelog for [PRODUCT TYPE]. Tracking [CHANGE TYPES: e.g., "features, fixes, breaking changes"], I need features like [FEATURES: e.g., "versioning, subscriptions"]. Include API and formatting options.

Advanced: Act as a product architect. Design changelog platform with: Products: [COUNT/VERSIONS] Changes: [CATEGORIZATION] Distribution: [CHANNELS] Format: [MARKDOWN/API/RSS] Automation: [CI/CD INTEGRATION] Include:

- Changelog architecture
- Version management
- Distribution system
- API design
- Automation pipeline
- Subscriber management Provide changelog system design.

Implement Copy to Clipboard

Beginner: How do I implement copy to clipboard?

Intermediate: I want to add clipboard functionality for [CONTENT TYPES: e.g., "text, code, tables"]. Supporting [BROWSERS/DEVICES] with [FALLBACK OPTIONS], I need features like [FEATURES: e.g., "success feedback, formatting"]. Include security considerations.

Advanced: Act as a clipboard expert. Design copy system with: Content: [TYPES/FORMATS] Browsers: [COMPATIBILITY] Features: [RICH TEXT/IMAGES] Security: [SANITIZATION] Feedback: [UI/ACCESSIBILITY] Include:

- Clipboard architecture
 - Format handling
 - Fallback strategies
 - Security measures
 - Feedback system
 - Testing approach
- Format as clipboard guide.

Create Comparison Table

Beginner: How do I create a comparison table?

Intermediate: I want to build comparison tables for [PRODUCT/SERVICE TYPE]. With [FEATURE COUNT] and [CUSTOMIZATION OPTIONS], I need features like [FEATURES: e.g., "sorting, filtering, highlighting"]. Include responsive design and export options.

Advanced: Act as a comparison architect. Design table system with: Items: [COUNT/COMPLEXITY] Features: [DYNAMIC/STATIC] Interaction: [SORT/FILTER/PIN] Responsive: [MOBILE STRATEGY] Export: [FORMATS] Include:

- Table architecture
- Data management
- Interaction design
- Responsive strategy
- Export pipeline

- Performance optimization Provide comparison design.

Build FAQ System

Beginner: How do I build an FAQ system?

Intermediate: I want to create FAQs for [WEBSITE TYPE]. Managing [QUESTION COUNT] with [CATEGORIES], I need features like [FEATURES: e.g., "search, accordion, anchors"]. Include SEO optimization and analytics tracking.

Advanced: Act as a content architect. Design FAQ platform with: Questions: [COUNT/CATEGORIES] Search: [IMPLEMENTATION] Features: [VOTING/SUGGESTIONS] SEO: [SCHEMA MARKUP] Analytics: [TRACKING] Include:

- FAQ architecture
- Search integration
- Schema implementation
- User feedback system
- Analytics pipeline
- Admin interface Format as FAQ system design.

Implement Avatar Upload

Beginner: How do I implement avatar upload?

Intermediate: I want to add avatar upload for [USER COUNT]. Supporting [IMAGE FORMATS] with [SIZE LIMITS], I need features like [FEATURES: e.g., "crop, resize, filters"]. Include validation and storage optimization.

Advanced: Act as a media architect. Design avatar system with: Users: [SCALE] Storage: [STRATEGY/CDN] Processing: [CROP/RESIZE/FILTER] Formats: [WEBP/AVIF SUPPORT] Performance: [LOAD TIME] Include:

- Upload architecture
- Processing pipeline
- Storage optimization

- CDN integration
- Format selection
- Fallback handling Provide avatar system design.

Create Toast Notifications

Beginner: How do I create toast notifications?

Intermediate: I want to implement toast notifications for [APPLICATION TYPE]. With [NOTIFICATION TYPES: e.g., "success, error, info"] and [FEATURES: e.g., "actions, duration, queue"], I need to handle multiple notifications with animations.

Advanced: Act as a notification architect. Design toast system with: Types: [CATEGORIES/PRIORITIES] Features: [ACTIONS/PERSISTENCE] Queue: [MANAGEMENT STRATEGY] Animation: [ENTER/EXIT] Accessibility: [ARIA/KEYBOARD] Include:

- Toast architecture
- Queue management
- Animation system
- Accessibility features
- State management
- Testing strategy Format as notification design.

Build Landing Page

Beginner: How do I build a landing page?

Intermediate: I want to create a landing page for [PRODUCT/SERVICE]. With [SECTIONS: e.g., "hero, features, testimonials"], I need [CONVERSION GOALS] with [PERFORMANCE REQUIREMENTS]. Include A/B testing and analytics setup.

Advanced: Act as a conversion architect. Design landing page system with: Variants: [A/B TESTS] Performance: [CORE WEB VITALS] Conversion: [TRACKING/GOALS] Personalization: [SEGMENTS] Integration: [CRM/ANALYTICS] Include:

- Page architecture
- Performance optimization
- Conversion tracking
- Testing framework
- Personalization engine
- Analytics integration Format as landing page design.

Implement Keyboard Shortcuts

Beginner: How do I implement keyboard shortcuts?

Intermediate: I want to add keyboard shortcuts to [APPLICATION TYPE]. Supporting [SHORTCUT COUNT] with [CUSTOMIZATION OPTIONS], I need features like [FEATURES: e.g., "help modal, conflict detection"]. Include accessibility considerations.

Advanced: Act as a keyboard architect. Design shortcut system with: Shortcuts: [COUNT/COMPLEXITY] Customization: [USER DEFINED] Conflicts: [DETECTION/RESOLUTION] Discovery: [HELP SYSTEM] Accessibility: [ALTERNATIVES] Include:

- Shortcut architecture
- Binding system
- Conflict detection
- Help interface
- Customization engine
- Testing framework Provide keyboard system design.

Create Progress Tracker

Beginner: How do I create a progress tracker?

Intermediate: I want to build progress tracking for [PROCESS TYPE: e.g., "onboarding, course completion"]. With [STEP COUNT] and [FEATURES: e.g., "persistence, branching"], I need to track [USER COUNT] with analytics integration.

Advanced: Act as a progress architect. Design tracking system with: Processes: [TYPES/COMPLEXITY] Users: [SCALE] Features: [BRANCHING/SKIPPING] Persistence: [STRATEGY] Analytics: [INSIGHTS] Include:

- Tracker architecture
 - State management
 - Branching logic
 - Persistence layer
 - Analytics pipeline
 - Visualization options
- Format as progress design.

Build Related Content System

Beginner: How do I build a related content system?

Intermediate: I want to show related content for [CONTENT TYPE]. Using [ALGORITHM: e.g., "tags, ML, collaborative"], I need to recommend [ITEM COUNT] with [PERFORMANCE REQUIREMENTS]. Include relevance scoring and caching.

Advanced: Act as a recommendation architect. Design related content with: Content: [VOLUME/TYPES] Algorithms: [METHODS] Performance: [LATENCY] Personalization: [LEVEL] Testing: [A/B FRAMEWORK] Include:

- Recommendation architecture
 - Algorithm selection
 - Scoring system
 - Caching strategy
 - Testing framework
 - Performance optimization
- Provide recommendation design.

Implement Cookie Consent

Beginner: How do I implement cookie consent?

Intermediate: I want to add cookie consent for [WEBSITE TYPE]. Supporting [REGULATIONS: e.g., "GDPR, CCPA"], I need features like [FEATURES: e.g., "categories, preferences"]. Include consent storage and analytics blocking.

Advanced: Act as a privacy architect. Design consent system with: Regulations: [COMPLIANCE] Categories: [COOKIE TYPES] Storage: [CONSENT RECORDS] Integration: [TAG MANAGERS] UI: [CUSTOMIZATION] Include:

- Consent architecture
- Category management
- Storage strategy
- Script blocking
- Preference center
- Audit trail Format as privacy design.

Create Tooltip System

Beginner: How do I create a tooltip system?

Intermediate: I want to implement tooltips for [UI ELEMENTS]. With [TRIGGER TYPES: e.g., "hover, click, focus"] and [FEATURES: e.g., "positioning, arrows"], I need to handle [TOOLTIP COUNT] with accessibility support.

Advanced: Act as a UI architect. Design tooltip system with: Triggers: [EVENTS/ DELAYS] Positioning: [SMART/FLIP] Content: [HTML/COMPONENTS] Animation: [TRANSITIONS] Accessibility: [ARIA/KEYBOARD] Include:

- Tooltip architecture
- Positioning engine
- Content management
- Animation system
- Accessibility features
- Performance optimization Provide tooltip design guide.

Build Navigation Menu

Beginner: How do I build a navigation menu?

Intermediate: I want to create a navigation menu for [WEBSITE TYPE]. With [MENU STRUCTURE: e.g., "multi-level, mega menu"] and [FEATURES: e.g., "mobile responsive, sticky"], I need to support [ITEM COUNT] with accessibility features.

Advanced: Act as a UX architect. Design navigation system with: Structure: [LEVELS/COMPLEXITY] Responsive: [MOBILE STRATEGY] Features: [SEARCH/MEGA MENU] Performance: [RENDER TIME] Accessibility: [WCAG COMPLIANCE] Include:

- Menu architecture
 - Mobile adaptation
 - Keyboard navigation
 - Performance optimization
 - Analytics tracking
 - A/B testing setup
- Format as navigation blueprint.

Implement Data Import

Beginner: How do I implement data import functionality?

Intermediate: I want to import [DATA TYPES: e.g., "CSV, JSON, Excel"] into my application. Processing [FILE SIZES] with [VALIDATION REQUIREMENTS], I need features like [FEATURES: e.g., "mapping, preview, error handling"]. Include progress tracking.

Advanced: Act as a data integration expert. Design import system for: Formats: [SUPPORTED TYPES] Volume: [RECORDS/FILE] Validation: [RULES COMPLEXITY] Mapping: [AUTO/MANUAL] Performance: [THROUGHPUT] Include:

- Import architecture
- Parser selection
- Validation pipeline

- Mapping engine
- Error recovery
- Batch processing Provide import system design.

Create Notification Center

Beginner: How do I create a notification center?

Intermediate: I want to build a notification center for [APPLICATION TYPE]. Managing [NOTIFICATION TYPES] for [USER COUNT], I need features like [FEATURES: e.g., "mark as read, filters, settings"]. Include real-time updates and persistence.

Advanced: Act as a notification architect. Design notification hub with: Types: [CATEGORIES/PRIORITY] Volume: [NOTIFICATIONS/USER/DAY] Delivery: [IN-APP/EMAIL/PUSH] Features: [GROUPING/ACTIONS] Retention: [POLICY] Include:

- Hub architecture
- Delivery orchestration
- State management
- Preference engine
- Analytics pipeline
- Performance optimization Format as notification platform.

Build OCR System

Beginner: How do I build an OCR (Optical Character Recognition) system?

Intermediate: I want to implement OCR for [DOCUMENT TYPES: e.g., "receipts, invoices"]. Using [OCR SERVICE/LIBRARY], I need to extract [DATA FIELDS] with [ACCURACY REQUIREMENTS]. Include post-processing and validation.

Advanced: Act as an OCR architect. Design document processing system with: Documents: [TYPES/VOLUME] Languages: [SUPPORTED] Accuracy: [REQUIREMENTS] Fields: [EXTRACTION RULES] Output: [STRUCTURED DATA] Include:

- OCR pipeline design
- Pre-processing steps
- Field extraction logic
- Validation framework
- Human review workflow
- Integration architecture Provide OCR system blueprint.

Implement Drag and Drop

Beginner: How do I implement drag and drop functionality?

Intermediate: I want to add drag and drop for [USE CASE: e.g., "file upload, list reordering"]. Supporting [DRAG SOURCES] to [DROP TARGETS], I need features like [FEATURES: e.g., "preview, multi-select"]. Include touch device support.

Advanced: Act as an interaction designer. Design drag/drop system with: Elements: [TYPES/COUNT] Features: [MULTI-SELECT/SORT] Feedback: [VISUAL/HAPTIC] Performance: [ITEM LIMITS] Accessibility: [KEYBOARD NAV] Include:

- DnD architecture
- State management
- Animation system
- Touch handling
- Accessibility layer
- Performance optimization Format as interaction design.

Create Comment Moderation

Beginner: How do I create comment moderation?

Intermediate: I want to moderate comments for [PLATFORM TYPE]. Processing [COMMENT VOLUME] with [MODERATION RULES: e.g., "spam, toxicity"], I need features like [FEATURES: e.g., "auto-flag, manual review"]. Include AI integration.

Advanced: Act as a content moderation expert. Design moderation system with:
Volume: [COMMENTS/DAY] Rules: [AUTO/MANUAL] AI: [MODELS/SERVICES]
Workflow: [REVIEW STAGES] Metrics: [ACCURACY/SPEED] Include:

- Moderation pipeline
- AI integration
- Review interface
- Appeals process
- Moderator tools
- Performance metrics Provide moderation architecture.

Build Calendar Booking

Beginner: How do I build a calendar booking system?

Intermediate: I want to create booking for [SERVICE TYPE: e.g., "appointments, resources"]. Managing [CALENDAR COUNT] with [AVAILABILITY RULES], I need features like [FEATURES: e.g., "time zones, recurring slots"]. Include conflict prevention.

Advanced: Act as a scheduling architect. Design booking platform with: Resources: [TYPES/COUNT] Rules: [AVAILABILITY/DURATION] Features: [BUFFER/RECURRING] Scale: [BOOKINGS/DAY] Integration: [CALENDAR APIS] Include:

- Booking architecture
- Availability engine
- Conflict resolution
- Timezone handling
- Notification system
- Analytics dashboard Format as booking system design.

Implement Lazy Loading

Beginner: How do I implement lazy loading?

Intermediate: I want to lazy load [CONTENT TYPE: e.g., "images, components, modules"]. With [PAGE SIZE] and [PERFORMANCE TARGETS], I need features like [FEATURES: e.g., "placeholder, progressive enhancement"]. Include SEO considerations.

Advanced: Act as a performance architect. Design lazy loading system with: Content: [TYPES/SIZE] Strategy: [INTERSECTION/SCROLL] Performance: [METRICS] SEO: [REQUIREMENTS] Fallbacks: [NO-JS SUPPORT] Include:

- Loading architecture
- Observer implementation
- Placeholder system
- SEO optimization
- Performance monitoring
- Testing strategy Format as performance guide.

Create Invoice Generator

Beginner: How do I create an invoice generator?

Intermediate: I want to generate invoices for [BUSINESS TYPE]. Including [INVOICE ELEMENTS: e.g., "line items, taxes, discounts"], I need to create [FORMAT TYPES: e.g., "PDF, email"] with [CUSTOMIZATION LEVEL]. Include numbering and storage.

Advanced: Act as a billing architect. Design invoice system with: Templates: [CUSTOMIZATION] Compliance: [TAX LAWS] Formats: [PDF/EMAIL/API] Volume: [INVOICES/MONTH] Integration: [ACCOUNTING/ERP] Include:

- Invoice architecture
- Template engine
- Tax calculation
- Number generation
- Storage strategy

- Compliance checks Provide billing system design.

Build Search Filters

Beginner: How do I build search filters?

Intermediate: I want to implement filters for [SEARCH TYPE: e.g., "product catalog, job listings"]. With [FILTER TYPES: e.g., "range, multi-select, hierarchical"] and [ITEM COUNT], I need real-time updates and URL state management.

Advanced: Act as a search UX expert. Design filter system with: Filters: [TYPES/COUNT] Data: [VOLUME/STRUCTURE] Performance: [UPDATE SPEED] State: [URL/SESSION] Mobile: [ADAPTATION] Include:

- Filter architecture
- State management
- Performance optimization
- Mobile design
- Analytics tracking
- A/B testing setup Format as search UX design.

Implement SSO Integration

Beginner: How do I implement Single Sign-On (SSO)?

Intermediate: I want to integrate SSO for [ORGANIZATION TYPE]. Supporting [PROTOCOLS: e.g., "SAML, OIDC"] with [PROVIDER COUNT], I need features like [FEATURES: e.g., "JIT provisioning, group sync"]. Include user mapping and session management.

Advanced: Act as an identity architect. Design SSO platform with: Protocols: [SAML/OIDC/OAUTH] Providers: [COUNT/TYPES] Provisioning: [JIT/SCIM] Mapping: [ATTRIBUTES/GROUPS] Compliance: [STANDARDS] Include:

- SSO architecture
- Protocol adapters
- User provisioning

- Session management
- Audit logging
- Migration strategy Format as identity blueprint.

Create Report Builder

Beginner: How do I create a report builder?

Intermediate: I want to build reports for [DATA DOMAIN]. With [REPORT TYPES: e.g., "tabular, charts, dashboards"] and [DATA SOURCES], I need features like [FEATURES: e.g., "drag-drop, scheduling, export"]. Include template management.

Advanced: Act as a reporting architect. Design report builder with: Data: [SOURCES/VOLUME] Reports: [COMPLEXITY] Features: [BUILDER/TEMPLATES] Distribution: [CHANNELS] Performance: [GENERATION TIME] Include:

- Builder architecture
- Query engine
- Visualization library
- Template system
- Distribution pipeline
- Caching strategy Provide reporting platform design.

Build Approval Workflow

Beginner: How do I build an approval workflow?

Intermediate: I want to create approval flows for [PROCESS TYPE: e.g., "expense, leave requests"]. With [APPROVAL LEVELS] and [ROUTING RULES], I need features like [FEATURES: e.g., "delegation, escalation"]. Include notifications and audit trail.

Advanced: Act as a workflow architect. Design approval system with: Workflows: [TYPES/COMPLEXITY] Approvers: [HIERARCHY] Rules: [ROUTING/ESCALATION] Features: [DELEGATION/PARALLEL] Integration: [SYSTEMS] Include:

- Workflow engine
- Rule evaluation
- Delegation logic
- Notification system
- Audit framework
- Analytics dashboard Format as workflow design.

Implement Content Versioning

Beginner: How do I implement content versioning?

Intermediate: I want to version [CONTENT TYPE: e.g., "articles, configurations"]. Tracking [CHANGE FREQUENCY] with [STORAGE REQUIREMENTS], I need features like [FEATURES: e.g., "diff view, rollback, branching"]. Include comparison tools.

Advanced: Act as a version control expert. Design versioning system with: Content: [TYPES/SIZE] Changes: [FREQUENCY/VOLUME] Features: [DIFF/MERGE/BRANCH] Storage: [STRATEGY] Performance: [RETRIEVAL TIME] Include:

- Version architecture
- Storage optimization
- Diff algorithms
- Merge strategies
- Branch management
- UI components Provide versioning blueprint.

Create Quiz System

Beginner: How do I create a quiz system?

Intermediate: I want to build quizzes for [PURPOSE: e.g., "education, assessment"]. Supporting [QUESTION TYPES: e.g., "multiple choice, essay"] with [SCORING

RULES], I need features like [FEATURES: e.g., "timer, randomization"]. Include results analytics.

Advanced: Act as an assessment architect. Design quiz platform with: Questions: [TYPES/BANK SIZE] Scoring: [ALGORITHMS] Features: [ADAPTIVE/TIMED] Security: [ANTI-CHEAT] Analytics: [INSIGHTS] Include:

- Quiz architecture
- Question engine
- Scoring system
- Security measures
- Analytics pipeline
- Reporting tools Format as assessment design.

Build Testimonial System

Beginner: How do I build a testimonial system?

Intermediate: I want to collect testimonials for [BUSINESS TYPE]. With [COLLECTION METHODS: e.g., "forms, video"] and [DISPLAY OPTIONS], I need features like [FEATURES: e.g., "moderation, ratings"]. Include social proof elements.

Advanced: Act as a social proof architect. Design testimonial platform with: Types: [TEXT/VIDEO/AUDIO] Collection: [METHODS] Moderation: [WORKFLOW] Display: [WIDGETS/PAGES] Analytics: [METRICS] Include:

- Collection architecture
- Moderation pipeline
- Display system
- Widget framework
- Analytics tracking
- A/B testing setup Provide testimonial design.

Implement Undo/Redo

Beginner: How do I implement undo/redo functionality?

Intermediate: I want to add undo/redo to [APPLICATION TYPE: e.g., "text editor, drawing app"]. Managing [ACTION TYPES] with [HISTORY DEPTH], I need features like [FEATURES: e.g., "selective undo, persistence"]. Include memory optimization.

Advanced: Act as a state management expert. Design undo/redo system with: Actions: [TYPES/COMPLEXITY] History: [DEPTH/PERSISTENCE] Patterns: [COMMAND/MEMENTO] Performance: [MEMORY LIMITS] Features: [SELECTIVE/BRANCH] Include:

- History architecture
 - Command pattern implementation
 - Memory optimization
 - Persistence strategy
 - UI integration
 - Testing approach
- Format as state management guide.

Create Waitlist System

Beginner: How do I create a waitlist system?

Intermediate: I want to build a waitlist for [PRODUCT LAUNCH]. Managing [EXPECTED SIGNUPS] with [FEATURES: e.g., "priority, referrals"], I need to handle notifications and progressive access. Include analytics and segmentation.

Advanced: Act as a launch architect. Design waitlist platform with: Capacity: [SIGNUPS] Features: [REFERRAL/PRIORITY] Access: [ROLLOUT STRATEGY] Communication: [CHANNELS] Analytics: [METRICS] Include:

- Waitlist architecture
- Priority algorithm
- Referral system
- Communication engine

- Analytics dashboard
- Integration options Provide launch system design.

Build Location Picker

Beginner: How do I build a location picker?

Intermediate: I want to create a location picker for [USE CASE: e.g., "delivery address, store finder"]. Using [MAP PROVIDER], I need features like [FEATURES: e.g., "search, current location, markers"]. Include mobile optimization and offline support.

Advanced: Act as a location UX expert. Design picker system with: Providers: [MAPS/GEOCODING] Features: [SEARCH/DRAW/SUGGEST] Accuracy: [REQUIREMENTS] Offline: [CAPABILITIES] Privacy: [LOCATION HANDLING] Include:

- Picker architecture
- Search integration
- Geocoding pipeline
- Offline strategy
- Privacy controls
- Performance optimization Format as location UX design.

Implement Archive System

Beginner: How do I implement an archive system?

Intermediate: I want to archive [DATA TYPE: e.g., "old records, messages"]. With [ARCHIVE CRITERIA] and [ACCESS PATTERNS], I need features like [FEATURES: e.g., "compression, search, restore"]. Include storage optimization and compliance.

Advanced: Act as a data lifecycle expert. Design archive system with: Data: [TYPES/VOLUME] Criteria: [AGE/RULES] Storage: [TIERS/COMPRESSION] Access: [FREQUENCY/SPEED] Compliance: [RETENTION LAWS] Include:

- Archive architecture

- Tiered storage
- Compression strategy
- Search capabilities
- Restore procedures
- Compliance framework Provide lifecycle management design.

Create Sidebar Navigation

Beginner: How do I create a sidebar navigation?

Intermediate: I want to build a sidebar for [APPLICATION TYPE]. With [NAVIGATION ITEMS] and [FEATURES: e.g., "collapsible, nested items, icons"], I need responsive behavior and state persistence. Include accessibility features.

Advanced: Act as a navigation architect. Design sidebar system with: Structure: [DEPTH/ITEMS] Features: [COLLAPSE/PIN/SEARCH] Responsive: [BEHAVIOR] State: [PERSISTENCE] Customization: [USER PREFS] Include:

- Sidebar architecture
- State management
- Animation system
- Responsive strategy
- Accessibility implementation
- Performance optimization Format as navigation blueprint.

Build Metrics Dashboard

Beginner: How do I build a metrics dashboard?

Intermediate: I want to create a dashboard for [METRICS TYPE: e.g., "business KPIs, system health"]. Displaying [METRIC COUNT] with [VISUALIZATION TYPES], I need real-time updates and [FEATURES: e.g., "drill-down, alerts"]. Include data aggregation.

Advanced: Act as a dashboard architect. Design metrics platform with: Metrics: [TYPES/SOURCES] Visualizations: [CHARTS/WIDGETS] Updates: [REAL-TIME/BATCH] Features: [DRILL-DOWN/EXPORT] Performance: [RENDER TIME]
Include:

- Dashboard architecture
- Data pipeline
- Widget framework
- Caching strategy
- Alert system
- Mobile optimization Provide metrics platform design.

Implement Password Reset

Beginner: How do I implement password reset?

Intermediate: I want to add password reset for [APPLICATION TYPE]. Using [METHOD: e.g., "email link, SMS code"], I need to handle [USER VOLUME] with [SECURITY REQUIREMENTS: e.g., "token expiry, rate limiting"]. Include account recovery options.

Advanced: Act as a security architect. Design password reset with: Methods: [EMAIL/SMS/QUESTIONS] Security: [TOKEN/EXPIRY/ATTEMPTS] Scale: [REQUESTS/DAY] Recovery: [ACCOUNT ACCESS] Compliance: [STANDARDS]
Include:

- Reset architecture
- Token generation
- Delivery methods
- Security measures
- Recovery flows
- Audit logging Format as security blueprint.

Create Gantt Chart

Beginner: How do I create a Gantt chart?

Intermediate: I want to build a Gantt chart for [PROJECT TYPE]. Displaying [TASK COUNT] with [FEATURES: e.g., "dependencies, milestones, resources"], I need interactive editing and [TIMELINE SPAN]. Include export options.

Advanced: Act as a project visualization expert. Design Gantt system with: Tasks: [COUNT/HIERARCHY] Features: [DEPENDENCIES/CRITICAL PATH] Interaction: [DRAG/ZOOM/EDIT] Resources: [ALLOCATION/LOAD] Export: [FORMATS] Include:

- Chart architecture
- Rendering engine
- Interaction system
- Data management
- Export pipeline
- Performance optimization Provide project visualization design.

Build Address Autocomplete

Beginner: How do I build address autocomplete?

Intermediate: I want to implement address autocomplete for [USE CASE: e.g., "checkout, registration"]. Using [GEOCODING SERVICE], I need to support [COUNTRY COUNT] with [FEATURES: e.g., "apartment numbers, validation"]. Include fallback for manual entry.

Advanced: Act as an address UX expert. Design autocomplete system with: Coverage: [COUNTRIES/REGIONS] Accuracy: [VALIDATION LEVEL] Features: [APARTMENTS/PO BOX] Performance: [LATENCY] Fallback: [MANUAL ENTRY] Include:

- Autocomplete architecture
- Geocoding integration

- Validation pipeline
- Address formatting
- Fallback UI
- Analytics tracking Format as address UX design.

Implement Content Moderation

Beginner: How do I implement content moderation?

Intermediate: I want to moderate [CONTENT TYPES: e.g., "images, text, video"] for [PLATFORM TYPE]. Using [MODERATION TOOLS/AI], I need to detect [VIOLATION TYPES] with [ACCURACY REQUIREMENTS]. Include human review workflow.

Advanced: Act as a trust and safety architect. Design moderation platform with: Content: [TYPES/VOLUME] Detection: [AI/RULES/HUMAN] Violations: [CATEGORIES] Workflow: [REVIEW STAGES] Appeals: [PROCESS] Include:

- Moderation architecture
- AI integration
- Review interface
- Appeals system
- Reporting tools
- Compliance framework Provide T&S platform design.

Create Batch Operations

Beginner: How do I create batch operations?

Intermediate: I want to implement batch operations for [ENTITY TYPE: e.g., "users, products"]. Supporting [OPERATION TYPES: e.g., "update, delete, export"] on [VOLUME], I need features like [FEATURES: e.g., "progress, rollback"]. Include validation and confirmation.

Advanced: Act as a bulk operations expert. Design batch system with: Entities: [TYPES/VOLUME] Operations: [CRUD/CUSTOM] Validation: [RULES] Execution: [ASYNC/CHUNKS] Recovery: [ROLLBACK/RETRY] Include:

- Batch architecture
- Queue management
- Validation pipeline
- Progress tracking
- Error recovery
- Audit framework Format as batch operations design.

Build Kanban Board

Beginner: How do I build a Kanban board?

Intermediate: I want to create a Kanban board for [USE CASE: e.g., "project management, workflow"]. With [COLUMN TYPES] and [CARD FEATURES: e.g., "labels, assignees"], I need drag-drop functionality and [CUSTOMIZATION LEVEL]. Include WIP limits.

Advanced: Act as an agile tools architect. Design Kanban system with: Boards: [STRUCTURE/PERMISSIONS] Cards: [FIELDS/ATTACHMENTS] Rules: [WIP/AUTOMATION] Views: [FILTERS/GROUPING] Analytics: [METRICS] Include:

- Board architecture
- Card system
- Automation engine
- Permission model
- Analytics pipeline
- API design Provide Kanban platform design.

Implement Smart Search

Beginner: How do I implement smart search?

Intermediate: I want to build smart search for [CONTENT DOMAIN]. Using [SEARCH TECHNOLOGY: e.g., "Elasticsearch, Algolia"], I need features like [FEATURES: e.g., "typo tolerance, synonyms, facets"]. Include personalization and analytics.

Advanced: Act as a search engineer. Design smart search with: Data: [VOLUME/TYPES] Features: [NLP/PERSONALIZATION] Performance: [LATENCY/QPS] Relevance: [TUNING STRATEGY] Analytics: [INSIGHTS/AB TESTING] Include:

- Search architecture
- NLP pipeline
- Personalization engine
- Relevance tuning
- Analytics framework
- A/B testing setup Format as search platform design.

Create Onboarding Flow

Beginner: How do I create an onboarding flow?

Intermediate: I want to build onboarding for [APPLICATION TYPE]. With [STEP COUNT] and [USER SEGMENTS], I need features like [FEATURES: e.g., "progress tracking, skip options, tooltips"]. Include analytics and A/B testing.

Advanced: Act as an onboarding expert. Design onboarding system with: Flows: [SEGMENTS/PATHS] Steps: [INTERACTIVE/EDUCATIONAL] Personalization: [RULES] Analytics: [FUNNEL/DROP-OFF] Optimization: [A/B TESTS] Include:

- Flow architecture
- Segmentation engine
- Progress tracking
- Analytics pipeline
- Testing framework
- Performance metrics Provide onboarding blueprint.

Build Collaborative Editor

Beginner: How do I build a collaborative editor?

Intermediate: I want to create a collaborative editor for [CONTENT TYPE: e.g., "text, code, diagrams"]. Supporting [USER COUNT] simultaneous editors with [FEATURES: e.g., "cursors, presence, comments"], I need conflict resolution and persistence.

Advanced: Act as a collaboration architect. Design editor system with: Users: [CONCURRENT] Features: [CURSORS/PRESENCE/CHAT] Sync: [OT/CRDT] Performance: [LATENCY] Persistence: [VERSIONING] Include:

- Editor architecture
- Sync algorithm selection
- Conflict resolution
- Presence system
- Performance optimization
- Offline support Format as collaboration design.

Implement Geographic Search

Beginner: How do I implement geographic search?

Intermediate: I want to search for [ENTITY TYPE: e.g., "stores, events"] by location. Using [GEOSPATIAL DATABASE/SERVICE], I need to find items within [RADIUS/POLYGON] with [FILTERS]. Include distance calculation and sorting.

Advanced: Act as a geospatial architect. Design geographic search with: Data: [POINTS/POLYGONS] Queries: [RADIUS/POLYGON/KNN] Scale: [QUERIES/SEC] Accuracy: [PRECISION] Features: [ROUTING/ELEVATION] Include:

- Spatial architecture
- Index optimization
- Query strategies
- Caching approach

- Accuracy tradeoffs
- Performance tuning Provide geospatial design.

Create Split Testing

Beginner: How do I create split testing?

Intermediate: I want to implement A/B testing for [TEST TYPES: e.g., "features, UI, algorithms"]. Managing [EXPERIMENT COUNT] with [TRAFFIC ALLOCATION], I need statistical significance calculation and [FEATURES: e.g., "targeting, mutual exclusion"]. Include reporting.

Advanced: Act as an experimentation architect. Design A/B platform with: Experiments: [CONCURRENT] Allocation: [TRAFFIC/SEGMENTS] Statistics: [METHODS/CONFIDENCE] Features: [MULTIVARIATE/BANDIT] Analysis: [REAL-TIME/BATCH] Include:

- Experimentation architecture
- Assignment service
- Statistical engine
- Targeting system
- Analysis pipeline
- Reporting framework Format as experimentation design.

Build Document Scanner

Beginner: How do I build a document scanner?

Intermediate: I want to create a document scanner for [PLATFORM: e.g., "mobile, web"]. Capturing [DOCUMENT TYPES] with [ENHANCEMENT FEATURES: e.g., "crop, rotate, filters"], I need OCR integration and [OUTPUT FORMATS]. Include quality detection.

Advanced: Act as a scanning architect. Design document scanner with: Capture: [CAMERA/UPLOAD] Enhancement: [FILTERS/CROP/PERSPECTIVE] Quality:

[DETECTION/GUIDANCE] OCR: [INTEGRATION/ACCURACY] Output: [PDF/IMAGES/TEXT] Include:

- Scanner architecture
- Image processing pipeline
- Quality detection
- OCR integration
- Export system
- Cloud sync Provide scanner platform design.

Implement Role Management

Beginner: How do I implement role management?

Intermediate: I want to manage roles for [APPLICATION TYPE]. With [ROLE HIERARCHY] and [PERMISSION GRANULARITY], I need features like [FEATURES: e.g., "inheritance, delegation, custom roles"]. Include audit logging and bulk operations.

Advanced: Act as an access control architect. Design RBAC system with: Roles: [HIERARCHY/COUNT] Permissions: [GRANULARITY] Assignment: [RULES/INHERITANCE] Delegation: [TEMPORARY/PERMANENT] Audit: [COMPLIANCE LEVEL] Include:

- RBAC architecture
- Permission engine
- Inheritance model
- Delegation system
- Audit framework
- Migration tools Format as access control design.

Create Timeline Component

Beginner: How do I create a timeline component?

Intermediate: I want to build a timeline for [USE CASE: e.g., "history, events, milestones"]. Displaying [EVENT COUNT] with [LAYOUT: e.g., "vertical, horizontal"], I need features like [FEATURES: e.g., "filtering, zoom, clustering"]. Include responsive design.

Advanced: Act as a visualization architect. Design timeline system with: Events: [COUNT/TYPES] Layout: [VERTICAL/HORIZONTAL/SPIRAL] Features: [ZOOM/FILTER/CLUSTER] Performance: [RENDER LIMIT] Interaction: [DETAILS/NAVIGATION] Include:

- Timeline architecture
- Layout engine
- Clustering algorithm
- Interaction system
- Performance optimization
- Mobile adaptation Provide timeline design guide.

Build Recommendation Feed

Beginner: How do I build a recommendation feed?

Intermediate: I want to create a personalized feed for [CONTENT TYPE]. Using [RECOMMENDATION ALGORITHM], I need to serve [USER COUNT] with [UPDATE FREQUENCY]. Include engagement tracking and diversity controls.

Advanced: Act as a feed architect. Design recommendation feed with: Content: [VOLUME/TYPES] Algorithms: [CF/CONTENT/HYBRID] Personalization: [LEVEL] Diversity: [CONTROLS] Performance: [LATENCY/THROUGHPUT] Include:

- Feed architecture
- ML pipeline
- Ranking system
- Diversity injection
- Caching strategy

- Experimentation framework Format as feed platform design.

Implement Session Recording

Beginner: How do I implement session recording?

Intermediate: I want to record user sessions for [PURPOSE: e.g., "debugging, analytics"]. Capturing [DATA TYPES: e.g., "DOM, clicks, console"] with [PRIVACY REQUIREMENTS], I need playback features and [RETENTION PERIOD]. Include data sanitization.

Advanced: Act as a session replay architect. Design recording system with: Capture: [DOM/EVENTS/NETWORK] Privacy: [PII HANDLING] Storage: [COMPRESSION/RETENTION] Playback: [FEATURES/SPEED] Analytics: [INSIGHTS/SEARCH] Include:

- Recording architecture
- Privacy framework
- Compression strategy
- Playback engine
- Search capabilities
- Integration options Provide session platform design.

Create Price Calculator

Beginner: How do I create a price calculator?

Intermediate: I want to build a pricing calculator for [PRODUCT/SERVICE]. With [PRICING VARIABLES: e.g., "quantity, features, discounts"] and [COMPLEXITY LEVEL], I need real-time updates and [OUTPUT OPTIONS: e.g., "quote, comparison"]. Include currency support.

Advanced: Act as a pricing architect. Design calculator system with: Models: [PRICING STRUCTURES] Variables: [COUNT/DEPENDENCIES] Rules: [DISCOUNTS/BUNDLES] Currency: [MULTI-CURRENCY] Output: [QUOTES/PROPOSALS] Include:

- Calculator architecture

- Pricing engine
- Rule evaluation
- Currency handling
- Quote generation
- A/B testing setup Format as pricing platform design.

Build News Aggregator

Beginner: How do I build a news aggregator?

Intermediate: I want to aggregate news from [SOURCE COUNT] sources. Processing [ARTICLE VOLUME] with [FEATURES: e.g., "categorization, deduplication"], I need personalization and [UPDATE FREQUENCY]. Include content extraction and ranking.

Advanced: Act as a content aggregation expert. Design news platform with: Sources: [RSS/API/SCRAPING] Volume: [ARTICLES/DAY] Processing: [NLP/DEDUP/CATEGORIZATION] Personalization: [ALGORITHMS] Distribution: [CHANNELS] Include:

- Aggregation architecture
- Content pipeline
- Deduplication system
- Ranking algorithm
- Personalization engine
- API design Provide aggregator blueprint.

Implement Spell Checker

Beginner: How do I implement a spell checker?

Intermediate: I want to add spell checking to [APPLICATION TYPE]. Supporting [LANGUAGES] with [DICTIONARY SIZE], I need features like [FEATURES: e.g., "suggestions, custom dictionary, context awareness"]. Include performance optimization.

Advanced: Act as a NLP architect. Design spell check system with: Languages: [SUPPORTED] Dictionary: [SIZE/SOURCES] Algorithms: [EDIT DISTANCE/PHONETIC] Context: [AWARENESS LEVEL] Performance: [LATENCY] Include:

- Spell check architecture
- Dictionary management
- Suggestion algorithms
- Context analysis
- Custom dictionary
- Performance optimization Format as NLP system design.

Create Resource Scheduler

Beginner: How do I create a resource scheduler?

Intermediat: I want to schedule [RESOURCE TYPES: e.g., "rooms, equipment, staff"] for [ORGANIZATION TYPE]. Managing [RESOURCE COUNT] with [BOOKING RULES], I need conflict detection and [FEATURES: e.g., "recurring, waitlist"]. Include availability visualization.

Advanced: Act as a scheduling architect. Design resource system with: Resources: [TYPES/COUNT] Rules: [CONSTRAINTS/PRIORITY] Bookings: [VOLUME/PATTERNS] Features: [WAITLIST/OPTIMIZATION] Visualization: [CALENDAR/GANTT] Include:

- Scheduler architecture
- Constraint engine
- Optimization algorithms
- Conflict resolution
- Visualization system
- Integration framework Provide scheduling platform design.

Build Forum Search

Beginner: How do I build forum search?

Intermediate: I want to implement search for a forum with [POST COUNT]. Searching [CONTENT TYPES: e.g., "titles, posts, users"] with [FEATURES: e.g., "filters, sorting, highlighting"], I need to handle [QUERY VOLUME]. Include relevance tuning.

Advanced: Act as a forum search expert. Design search system with: Content: [POSTS/COMMENTS/USERS] Index: [STRUCTURE/SIZE] Features: [FACETS/BOOST/SYNONYMS] Performance: [QPS/LATENCY] Relevance: [SIGNALS/TUNING] Include:

- Search architecture
- Indexing strategy
- Query processing
- Relevance scoring
- Faceted search
- Performance optimization Format as forum search design.

Implement JWT Refresh

Beginner: How do I implement JWT token refresh?

Intermediate: I want to implement JWT refresh tokens for [APPLICATION TYPE]. With [TOKEN LIFETIMES] and [SECURITY REQUIREMENTS], I need to handle [USER SCALE] with features like [FEATURES: e.g., "rotation, revocation"]. Include device management.

Advanced: Act as a JWT security expert. Design token refresh system with: Tokens: [ACCESS/REFRESH LIFETIMES] Storage: [CLIENT/SERVER] Rotation: [STRATEGY] Revocation: [BLACKLIST/WHITELIST] Devices: [MANAGEMENT] Include:

- Token architecture

- Refresh flow
- Rotation strategy
- Revocation system
- Device tracking
- Security hardening Format as JWT security design.

Create Heatmap Visualization

Beginner: How do I create a heatmap visualization?

Intermediate: I want to build heatmaps for [DATA TYPE: e.g., "click tracking, geographic data"]. Visualizing [DATA POINTS] with [COLOR SCHEMES], I need features like [FEATURES: e.g., "zoom, tooltips, export"]. Include performance optimization for large datasets.

Advanced: Act as a data visualization expert. Design heatmap system with: Data: [VOLUME/DIMENSIONS] Rendering: [CANVAS/SVG/WEBGL] Interaction: [ZOOM/PAN/FILTER] Performance: [POINTS LIMIT] Export: [FORMATS] Include:

- Heatmap architecture
- Rendering strategy
- Data aggregation
- Interaction layer
- Performance optimization
- Export pipeline Format as visualization design.

Build Social Feed

Beginner: How do I build a social feed?

Intermediate: I want to create a social feed for [PLATFORM TYPE]. Displaying [POST TYPES: e.g., "text, images, videos"] from [USER CONNECTIONS], I need features like [FEATURES: e.g., "likes, comments, shares"]. Include real-time updates and personalization.

Advanced: Act as a social platform architect. Design feed system with: Users: [SCALE/CONNECTIONS] Content: [TYPES/VOLUME] Delivery: [PUSH/PULL/HYBRID] Features: [REACTIONS/SHARING] Ranking: [CHRONOLOGICAL/ALGORITHMIC] Include:

- Feed architecture
- Fanout strategy
- Ranking algorithm
- Caching layer
- Real-time updates
- Analytics pipeline Provide social feed design.

Implement Resource Monitor

Beginner: How do I implement resource monitoring?

Intermediate: I want to monitor [RESOURCE TYPES: e.g., "CPU, memory, disk"] for [SYSTEM TYPE]. Collecting metrics at [FREQUENCY] with [ALERTING THRESHOLDS], I need visualization and [FEATURES: e.g., "historical data, predictions"]. Include anomaly detection.

Advanced: Act as a monitoring architect. Design resource monitor with: Resources: [TYPES/GRANULARITY] Collection: [AGENTS/APIS] Storage: [TSDB/RETENTION] Alerting: [RULES/CHANNELS] Analysis: [ANOMALY/FORECAST] Include:

- Monitor architecture
- Collection strategy
- Time series storage
- Alert engine
- Anomaly detection
- Predictive analytics Format as monitoring design.

Create Poll Widget

Beginner: How do I create a poll widget?

Intermediate: I want to build a poll widget for [WEBSITE TYPE]. Supporting [POLL OPTIONS] with [VOTE RESTRICTIONS: e.g., "one per user, IP-based"], I need real-time results and [DISPLAY OPTIONS]. Include embed code generation.

Advanced: Act as a widget architect. Design poll system with: Polls: [TYPES/COMPLEXITY] Voting: [RESTRICTIONS/VALIDATION] Display: [THEMES/RESPONSIVE] Embed: [IFRAME/JS/API] Analytics: [INSIGHTS] Include:

- Widget architecture
- Vote validation
- Real-time updates
- Embed framework
- Analytics tracking
- Fraud prevention Format as widget platform design.

Implement Text Editor

Beginner: How do I implement a text editor?

Intermediate: I want to build a text editor for [PURPOSE: e.g., "blogging, note-taking"]. With [FEATURES: e.g., "formatting, images, tables"] and [EDITOR TYPE: e.g., "WYSIWYG, markdown"], I need autosave and collaboration features.

Advanced: Act as an editor architect. Design text editor with: Features: [RICH TEXT/MARKDOWN/CODE] Collaboration: [REAL-TIME/ASYNC] Storage: [AUTOSAVE/VERSIONS] Plugins: [EXTENSIBILITY] Performance: [DOCUMENT SIZE] Include:

- Editor architecture
- Plugin system
- Collaboration engine
- Storage strategy

- Performance optimization
- Mobile support Provide editor platform design.

Build Weather Widget

Beginner: How do I build a weather widget?

Intermediate: I want to create a weather widget showing [DATA POINTS: e.g., "temperature, forecast, alerts"]. Using [WEATHER API], I need to support [LOCATION OPTIONS] with [UPDATE FREQUENCY]. Include visualization and caching.

Advanced: Act as a weather architect. Design weather system with: Data: [SOURCES/AGGREGATION] Locations: [SEARCH/DETECT/SAVE] Updates: [FREQUENCY/PUSH] Visualization: [CHARTS/MAPS] Alerts: [SEVERE WEATHER] Include:

- Weather architecture
- API aggregation
- Location handling
- Caching strategy
- Alert system
- Offline support Format as weather widget design.

Implement Content Filtering

Beginner: How do I implement content filtering?

Intermediate: I want to filter [CONTENT TYPE] based on [CRITERIA: e.g., "categories, tags, date range"]. Managing [ITEM COUNT] with [FILTER COMPLEXITY], I need features like [FEATURES: e.g., "save filters, quick presets"]. Include URL state management.

Advanced: Act as a filtering expert. Design filter system with: Filters: [TYPES/COMPLEXITY] Data: [VOLUME/STRUCTURE] State: [URL/LOCAL/SERVER] Performance: [FILTER SPEED] UX: [INSTANT/APPLY BUTTON] Include:

- Filter architecture
- State management
- Query optimization
- Preset system
- Performance tuning
- Analytics integration Provide filtering framework design.

Create Stock Tracker

Beginner: How do I create a stock tracker?

Intermediate: I want to build a stock tracker for [MARKET TYPES]. Tracking [SYMBOL COUNT] with [FEATURES: e.g., "real-time quotes, charts, alerts"], I need portfolio management and [DATA REFRESH RATE]. Include historical data visualization.

Advanced: Act as a financial data architect. Design stock tracking platform with: Markets: [EXCHANGES/ASSETS] Data: [REAL-TIME/DELAYED] Features: [PORTFOLIO/WATCHLIST] Charts: [TECHNICAL INDICATORS] Alerts: [PRICE/VOLUME/NEWS] Include:

- Market data architecture
- Streaming pipeline
- Portfolio engine
- Charting system
- Alert framework
- API design Format as fintech platform design.

Build Mood Tracker

Beginner: How do I build a mood tracker?

Intermediate: I want to create a mood tracking app for [PLATFORM]. Recording [DATA TYPES: e.g., "mood, activities, notes"] with [FREQUENCY], I need features like [FEATURES: e.g., "reminders, insights, charts"]. Include data export and privacy.

Advanced: Act as a wellness architect. Design mood tracking system with: Tracking: [METRICS/FREQUENCY] Analysis: [PATTERNS/CORRELATIONS] Privacy: [ENCRYPTION/LOCAL] Insights: [ML/STATISTICS] Export: [FORMATS/COMPLIANCE] Include:

- Tracking architecture
- Analysis engine
- Privacy framework
- Insight generation
- Export system
- Sync strategy Provide wellness platform design.

Implement Code Sandbox

Beginner: How do I implement a code sandbox?

Intermediate: I want to build a code sandbox for [LANGUAGES: e.g., "JavaScript, Python"]. Supporting [FEATURES: e.g., "console, multi-file, packages"], I need secure execution and [SHARING OPTIONS]. Include version control and templates.

Advanced: Act as a sandbox architect. Design code execution platform with: Languages: [SUPPORTED/VERSIONS] Execution: [ISOLATION/SECURITY] Features: [PACKAGES/FILESYSTEM] Sharing: [COLLABORATION/EMBED] Templates: [LIBRARY/CUSTOM] Include:

- Sandbox architecture
- Security isolation
- Package management
- Collaboration features
- Template system

- Performance optimization Format as sandbox platform design.

Create Budget Tracker

Beginner: How do I create a budget tracker?

Intermediate: I want to build a budget tracker for [USE CASE: e.g., "personal, business"]. Tracking [CATEGORY COUNT] with [FEATURES: e.g., "recurring, goals, reports"], I need bank integration and [VISUALIZATION TYPES]. Include forecasting.

Advanced: Act as a financial architect. Design budget platform with: Accounts: [TYPES/INTEGRATION] Categories: [HIERARCHY/RULES] Features: [GOALS/FORECAST] Import: [BANK/MANUAL/OCR] Reports: [INSIGHTS/EXPORT] Include:

- Budget architecture
- Bank integration
- Categorization engine
- Forecasting model
- Report generation
- Security framework Provide budget platform design.

Build Recipe Manager

Beginner: How do I build a recipe manager?

Intermediate: I want to create a recipe manager with [FEATURES: e.g., "ingredients, instructions, photos"]. Managing [RECIPE COUNT] with [ORGANIZATION: e.g., "categories, tags, meal planning"], I need search and sharing capabilities.

Advanced: Act as a culinary platform architect. Design recipe system with: Recipes: [STRUCTURE/MEDIA] Features: [NUTRITION/SCALING] Organization: [TAGS/COLLECTIONS] Planning: [MEALS/SHOPPING] Sharing: [SOCIAL/EXPORT] Include:

- Recipe architecture
- Ingredient parsing

- Nutrition calculation
- Meal planning engine
- Shopping list generator
- Social features Format as recipe platform design.

Implement Zoom Controls

Beginner: How do I implement zoom controls?

Intermediate: I want to add zoom functionality to [CONTENT TYPE: e.g., "images, maps, documents"]. Supporting [ZOOM METHODS: e.g., "pinch, wheel, buttons"] with [ZOOM RANGE], I need smooth animations and [FEATURES: e.g., "pan, reset"]. Include touch support.

Advanced: Act as a zoom interaction expert. Design zoom system with: Content: [TYPES/RESOLUTION] Controls: [TOUCH/MOUSE/KEYBOARD] Performance: [FPS/SMOOTHNESS] Features: [PAN/ROTATE/GESTURES] Constraints: [MIN/MAX/BOUNDARIES] Include:

- Zoom architecture
- Gesture recognition
- Animation engine
- Performance optimization
- Constraint system
- Accessibility features Provide zoom control design.

Create Gift Card System

Beginner: How do I create a gift card system?

Intermediate: I want to build gift cards for [BUSINESS TYPE]. Supporting [CARD TYPES: e.g., "digital, physical"] with [VALUE OPTIONS], I need features like [FEATURES: e.g., "balance check, partial redemption"]. Include fraud prevention.

Advanced: Act as a gift card architect. Design gift card platform with: Types: [DIGITAL/PHYSICAL/BOTH] Values: [FIXED/VARIABLE/RELOAD] Redemption: [PARTIAL/MULTIPLE] Security: [FRAUD PREVENTION] Integration: [POS/ECOMMERCE] Include:

- Card architecture
- Balance management
- Redemption engine
- Security measures
- Integration framework
- Reporting system Format as gift card design.

Build Music Player

Beginner: How do I build a music player?

Intermediate: I want to create a music player for [PLATFORM]. Supporting [AUDIO FORMATS] with [FEATURES: e.g., "playlists, shuffle, equalizer"], I need streaming capability and [STORAGE OPTIONS]. Include offline support and lyrics.

Advanced: Act as an audio platform architect. Design music player with: Formats: [SUPPORTED/CODECS] Features: [EQ/CROSSFADE/GAPLESS] Streaming: [PROTOCOLS/QUALITY] Offline: [DOWNLOAD/SYNC] Social: [SHARING/DISCOVERY] Include:

- Player architecture
- Audio pipeline
- Streaming strategy
- Offline system
- Playlist engine
- Social features Provide music platform design.

Implement Live Ticker

Beginner: How do I implement a live ticker?

Intermediate: I want to create a live ticker for [DATA TYPE: e.g., "news, sports scores, stocks"]. Displaying [UPDATE FREQUENCY] with [SCROLL OPTIONS], I need features like [FEATURES: e.g., "pause, speed control"]. Include data source management.

Advanced: Act as a real-time display expert. Design ticker system with: Data: [SOURCES/VOLUME] Display: [SCROLL/ROTATE/FADE] Updates: [PUSH/POLL/STREAM] Controls: [PAUSE/SPEED/FILTER] Performance: [ITEMS/SECOND] Include:

- Ticker architecture
- Animation engine
- Data pipeline
- Control system
- Performance optimization
- Customization options Format as ticker design guide.

Create Habit Tracker

Beginner: How do I create a habit tracker?

Intermediate: I want to build a habit tracker for [PLATFORM]. Tracking [HABIT COUNT] with [FREQUENCY OPTIONS: e.g., "daily, weekly"], I need features like [FEATURES: e.g., "streaks, reminders, statistics"]. Include goal setting and insights.

Advanced: Act as a behavior tracking architect. Design habit system with: Habits: [TYPES/FREQUENCY] Tracking: [CHECK-IN/AUTOMATIC] Gamification: [STREAKS/REWARDS] Insights: [ANALYTICS/PATTERNS] Reminders: [SMART/ADAPTIVE] Include:

- Habit architecture
- Tracking engine

- Gamification system
- Analytics pipeline
- Reminder algorithm
- Sync framework Provide habit platform design.

Build Floor Plan Designer

Beginner: How do I build a floor plan designer?

Intermediate: I want to create a floor plan designer for [USE CASE: e.g., "home design, office layout"]. With [FEATURES: e.g., "drag-drop furniture, measurements"], I need 2D/3D views and [EXPORT OPTIONS]. Include object library.

Advanced: Act as a CAD architect. Design floor plan system with: Canvas: [2D/3D RENDERING] Objects: [LIBRARY/CUSTOM] Features: [SNAP/MEASURE/LAYERS] Collaboration: [REAL-TIME/ASYNCR] Export: [FORMATS/QUALITY] Include:

- Designer architecture
- Rendering engine
- Object management
- Measurement system
- Collaboration layer
- Export pipeline Format as CAD platform design.

Implement Touch Gestures

Beginner: How do I implement touch gestures?

Intermediate: I want to add touch gestures to [APPLICATION TYPE]. Supporting [GESTURE TYPES: e.g., "swipe, pinch, long press"] on [PLATFORMS], I need gesture recognition and [FEATURES: e.g., "custom gestures"]. Include conflict resolution.

Advanced: Act as a gesture interaction expert. Design touch system with: Gestures: [STANDARD/CUSTOM] Recognition: [ACCURACY/SPEED] Conflicts: [RESOLUTION STRATEGY] Feedback: [VISUAL/HAPTIC] Customization: [USER DEFINED] Include:

- Gesture architecture
- Recognition engine
- Conflict resolver
- Feedback system
- Customization framework
- Testing tools Provide gesture design guide.

Create Expense Report

Beginner: How do I create expense reports?

Intermediate: I want to build expense reporting for [ORGANIZATION TYPE]. Managing [EXPENSE CATEGORIES] with [APPROVAL WORKFLOW], I need features like [FEATURES: e.g., "receipt scanning, mileage"]. Include policy compliance and reimbursement.

Advanced: Act as an expense architect. Design expense system with: Categories: [HIERARCHY/POLICIES] Capture: [RECEIPT OCR/IMPORT] Workflow: [APPROVAL LEVELS] Compliance: [POLICY ENGINE] Integration: [ACCOUNTING/PAYROLL] Include:

- Expense architecture
- OCR pipeline
- Approval engine
- Policy framework
- Integration layer
- Analytics dashboard Format as expense platform design.

Build Virtual Keyboard

Beginner: How do I build a virtual keyboard?

Intermediate: I want to create a virtual keyboard for [USE CASE: e.g., "touch screen, accessibility"]. Supporting [LAYOUT TYPES] with [LANGUAGE COUNT], I need features like [FEATURES: e.g., "predictive text, themes"]. Include custom layouts.

Advanced: Act as a keyboard architect. Design virtual keyboard with: Layouts: [QWERTY/CUSTOM/ADAPTIVE] Languages: [IME/PREDICTION] Features: [SWIPE/VOICE/EMOJI] Accessibility: [SIZE/CONTRAST] Performance: [LATENCY/ACCURACY] Include:

- Keyboard architecture
- Input method engine
- Prediction system
- Layout framework
- Theme engine
- Accessibility features Provide keyboard platform design.

Implement Breadth Search

Beginner: How do I implement breadth-first search?

Intermediate: I want to implement BFS for [GRAPH TYPE: e.g., "social network, maze solving"]. Working with [NODE COUNT] and [EDGE DENSITY], I need to find [SEARCH GOAL: e.g., "shortest path, all reachable nodes"]. Include visualization.

Advanced: Act as a graph algorithms expert. Design BFS system with: Graph: [SIZE/STRUCTURE] Storage: [ADJACENCY LIST/MATRIX] Goals: [PATH/TRAVERSAL/COMPONENT] Optimization: [MEMORY/SPEED] Visualization: [REAL-TIME/STEP] Include:

- BFS architecture
- Graph representation
- Queue optimization
- Path reconstruction
- Visualization engine

- Performance analysis Format as algorithm design guide.

Create Discount Engine

Beginner: How do I create a discount engine?

Intermediate: I want to build a discount engine for [E-COMMERCE TYPE]. Supporting [DISCOUNT TYPES: e.g., "percentage, BOGO, tiered"] with [RULE COMPLEXITY], I need stacking logic and [FEATURES: e.g., "exclusions, limits"]. Include validation.

Advanced: Act as a pricing architect. Design discount engine with: Types: [CATALOG/RULES] Stacking: [LOGIC/PRIORITY] Conditions: [COMPLEX RULES] Limits: [PER USER/GLOBAL] Performance: [CALCULATION SPEED] Include:

- Engine architecture
- Rule evaluation
- Stacking algorithm
- Validation system
- Testing framework
- Analytics integration Provide discount engine design.

Build Team Calendar

Beginner: How do I build a team calendar?

Intermediate: I want to create a shared calendar for [TEAM SIZE]. Supporting [EVENT TYPES: e.g., "meetings, leave, deadlines"] with [VIEW OPTIONS], I need features like [FEATURES: e.g., "availability, conflicts"]. Include timezone handling.

Advanced: Act as a calendar architect. Design team calendar with: Teams: [SIZE/HIERARCHY] Events: [TYPES/RECURRING] Views: [DAY/WEEK/MONTH/CUSTOM] Features: [AVAILABILITY/SCHEDULING] Integration: [EXTERNAL CALENDARS] Include:

- Calendar architecture
- Event engine

- Availability calculation
- Conflict detection
- Sync framework
- Permission model Format as team calendar design.

Implement Form Builder

Beginner: How do I implement a form builder?

Intermediate: I want to build a form builder for [USE CASE]. Supporting [FIELD TYPES] with [FEATURES: e.g., "conditional logic, validation"], I need drag-drop interface and [EXPORT OPTIONS]. Include template library.

Advanced: Act as a form platform architect. Design form builder with: Fields: [TYPES/CUSTOM] Logic: [CONDITIONS/CALCULATIONS] Validation: [RULES/CUSTOM] Templates: [LIBRARY/SHARING] Integration: [APIS/WEBHOOKS] Include:

- Builder architecture
- Field framework
- Logic engine
- Template system
- Integration layer
- Analytics pipeline Provide form platform design.

Create Knowledge Base

Beginner: How do I create a knowledge base?

Intermediate: I want to build a knowledge base for [CONTENT TYPE]. Organizing [ARTICLE COUNT] with [STRUCTURE: e.g., "categories, tags"], I need features like [FEATURES: e.g., "search, feedback, versioning"]. Include authoring tools.

Advanced: Act as a knowledge architect. Design KB system with: Content: [STRUCTURE/VOLUME] Search: [FULL TEXT/SEMANTIC] Authoring:

[MARKDOWN/WYSIWYG] Feedback: [RATINGS/SUGGESTIONS] Analytics: [USAGE/GAPS] Include:

- KB architecture
- Content management
- Search implementation
- Feedback system
- Analytics framework
- API design Format as knowledge platform design.

Build Appointment Reminder

Beginner: How do I build appointment reminders?

Intermediate: I want to create reminders for [APPOINTMENT TYPES]. Sending [REMINDER CHANNELS: e.g., "email, SMS, push"] with [TIMING RULES], I need features like [FEATURES: e.g., "templates, confirmation"]. Include timezone handling.

Advanced: Act as a reminder architect. Design reminder system with: Appointments: [TYPES/VOLUME] Channels: [EMAIL/SMS/VOICE] Timing: [RULES/OPTIMIZATION] Templates: [PERSONALIZATION] Responses: [CONFIRM/RESCHEDULE] Include:

- Reminder architecture
- Scheduling engine
- Channel management
- Template system
- Response handling
- Analytics tracking Provide reminder platform design.

Implement Masonry Layout

Beginner: How do I implement a masonry layout?

Intermediate: I want to create a masonry layout for [CONTENT TYPE: e.g., "images, cards"]. Handling [ITEM COUNT] with [FEATURES: e.g., "responsive columns, animations"], I need dynamic loading and [SORT OPTIONS]. Include performance optimization.

Advanced: Act as a layout architect. Design masonry system with: Items: [COUNT/DIMENSIONS] Columns: [RESPONSIVE RULES] Loading: [LAZY/INFINITE] Animation: [ENTRY/REFLOW] Performance: [RENDER LIMIT] Include:

- Layout architecture
- Positioning algorithm
- Responsive system
- Animation engine
- Performance optimization
- Virtualization strategy Format as layout design guide.

Create Travel Planner

Beginner: How do I create a travel planner?

Intermediate: I want to build a travel planner for [TRIP TYPES]. Managing [ITINERARY ELEMENTS: e.g., "flights, hotels, activities"] with [FEATURES: e.g., "budget tracking, documents"], I need collaboration and offline access.

Advanced: Act as a travel platform architect. Design planner system with: Trips: [COMPLEXITY/DURATION] Elements: [BOOKINGS/ACTIVITIES] Collaboration: [SHARING/EDITING] Integration: [BOOKING APIS] Offline: [SYNC STRATEGY] Include:

- Planner architecture
- Itinerary engine
- Collaboration framework
- API integrations
- Offline system

- Mobile optimization Format as travel platform design.

Build Markdown Parser

Beginner: How do I build a markdown parser?

Intermediate: I want to parse markdown for [USE CASE]. Supporting [MARKDOWN FEATURES: e.g., "tables, footnotes, syntax highlighting"] with [EXTENSIONS], I need HTML output and [SECURITY REQUIREMENTS]. Include custom renderers.

Advanced: Act as a parser architect. Design markdown system with: Syntax: [COMMONMARK/GFM/CUSTOM] Extensions: [PLUGINS/CUSTOM] Output: [HTML/AST/CUSTOM] Security: [SANITIZATION] Performance: [DOCUMENT SIZE] Include:

- Parser architecture
- Tokenizer design
- AST structure
- Renderer framework
- Security layer
- Extension system Provide parser design guide.

Implement Cart Abandonment

Beginner: How do I implement cart abandonment tracking?

Intermediate: I want to track abandoned carts for [E-COMMERCE TYPE]. Monitoring [CART STAGES] with [TRIGGER RULES], I need recovery campaigns and [FEATURES: e.g., "email sequences, discounts"]. Include analytics and A/B testing.

Advanced: Act as an e-commerce recovery expert. Design abandonment system with: Tracking: [EVENTS/STAGES] Triggers: [TIME/BEHAVIOR] Campaigns: [EMAIL/SMS/RETARGETING] Personalization: [CONTENT/OFFERS] Analytics: [RECOVERY RATE/ROI] Include:

- Tracking architecture

- Trigger engine
- Campaign automation
- Personalization system
- Analytics framework
- Testing platform Format as recovery system design.

Create Video Player

Beginner: How do I create a video player?

Intermediate: I want to build a video player for [PLATFORM]. Supporting [VIDEO FORMATS] with [FEATURES: e.g., "captions, quality selection, chapters"], I need adaptive streaming and [PLAYBACK FEATURES]. Include analytics tracking.

Advanced: Act as a video platform architect. Design player system with: Formats: [CODECS/CONTAINERS] Streaming: [HLS/DASH/PROGRESSIVE] Features: [CAPTIONS/CHAPTERS/SPEED] Quality: [ADAPTIVE/MANUAL] Analytics: [ENGAGEMENT/BUFFERING] Include:

- Player architecture
- Streaming strategy
- Feature framework
- Quality adaptation
- Analytics pipeline
- Plugin system Provide video platform design.

Build Contact Manager

Beginner: How do I build a contact manager?

Intermediate: I want to create a contact manager for [USE CASE]. Managing [CONTACT COUNT] with [FEATURES: e.g., "groups, custom fields, import/export"], I need search and [INTEGRATION OPTIONS]. Include duplicate detection.

Advanced: Act as a CRM architect. Design contact system with: Contacts: [SCALE/FIELDS] Organization: [GROUPS/TAGS] Features: [MERGE/HISTORY] Integration: [EMAIL/CALENDAR] Sync: [DEVICES/SERVICES] Include:

- Contact architecture
- Data model design
- Search implementation
- Duplicate detection
- Sync framework
- API design Format as contact platform design.

Implement Password Strength

Beginner: How do I implement password strength checking?

Intermediate: I want to check password strength for [APPLICATION TYPE]. Using [CRITERIA: e.g., "length, complexity, common passwords"] with [SCORING METHOD], I need real-time feedback and [FEATURES: e.g., "suggestions"]. Include policy configuration.

Advanced: Act as a password security expert. Design strength checker with: Criteria: [RULES/SCORING] Databases: [COMMON/BREACHED] Feedback: [REAL-TIME/DETAILED] Policies: [CONFIGURABLE/INDUSTRY] Localization: [LANGUAGES] Include:

- Checker architecture
- Scoring algorithm
- Database integration
- Feedback system
- Policy engine
- Performance optimization Provide password security design.

Create Timeline Editor

Beginner: How do I create a timeline editor?

Intermediate: I want to build a timeline editor for [USE CASE: e.g., "video editing, project planning"]. Managing [TRACK TYPES] with [FEATURES: e.g., "snap, zoom, markers"], I need precise control and [EXPORT OPTIONS]. Include undo/redo.

Advanced: Act as a timeline architect. Design editor system with: Tracks: [TYPES/LAYERS] Elements: [CLIPS/KEYFRAMES] Controls: [SNAP/GRID/ZOOM] Precision: [FRAME/SAMPLE] Export: [FORMATS/QUALITY] Include:

- Timeline architecture
- Track management
- Precision engine
- Interaction system
- Export pipeline
- Plugin framework Format as timeline editor design.

Build Order Tracking

Beginner: How do I build order tracking?

Intermediate: I want to create order tracking for [BUSINESS TYPE]. Tracking [ORDER STAGES] with [UPDATE METHODS], I need features like [FEATURES: e.g., "notifications, map view, delivery proof"]. Include carrier integration.

Advanced: Act as a logistics architect. Design tracking system with: Orders: [VOLUME/COMPLEXITY] Stages: [CUSTOM/STANDARD] Updates: [REAL-TIME/WEBHOOK] Carriers: [INTEGRATION/APIS] Features: [MAPS/PHOTOS/SIGNATURE] Include:

- Tracking architecture
- Status engine
- Carrier integration
- Notification system

- Visualization layer
- Analytics dashboard Provide logistics platform design.

Implement Tab Navigation

Beginner: How do I implement tab navigation?

Intermediate: I want to create tabs for [CONTENT TYPE]. Supporting [TAB COUNT] with [FEATURES: e.g., "closeable, reorderable, overflow"], I need lazy loading and [PERSISTENCE OPTIONS]. Include keyboard navigation.

Advanced: Act as a navigation architect. Design tab system with: Tabs: [COUNT/DYNAMIC] Features: [DRAG/CLOSE/PIN] Content: [LAZY/PRELOAD] State: [PERSISTENCE/RESTORE] Accessibility: [ARIA/KEYBOARD] Include:

- Tab architecture
- State management
- Drag system
- Content loading
- Accessibility layer
- Performance optimization Format as tab navigation design.

Create Loyalty Program

Beginner: How do I create a loyalty program?

Intermediate: I want to build a loyalty program for [BUSINESS TYPE]. With [POINT STRUCTURE] and [REWARD TIERS], I need features like [FEATURES: e.g., "point history, redemption"]. Include member portal and analytics.

Advanced: Act as a loyalty architect. Design program system with: Structure: [POINTS/TIERS/BENEFITS] Earning: [RULES/MULTIPLIERS] Redemption: [CATALOG/RESTRICTIONS] Members: [SCALE/SEGMENTS] Analytics: [BEHAVIOR/ROI] Include:

- Program architecture

- Point engine
- Tier management
- Redemption system
- Member portal
- Analytics framework Provide loyalty platform design.

Build Code Differ

Beginner: How do I build a code differ?

Intermediate: I want to create a code diff tool for [USE CASE]. Supporting [LANGUAGE SYNTAX] with [DIFF ALGORITHMS], I need features like [FEATURES: e.g., "side-by-side, inline, syntax highlighting"]. Include merge capabilities.

Advanced: Act as a diff algorithm expert. Design code differ with: Algorithms: [MYERS/PATIENCE/CUSTOM] Views: [UNIFIED/SPLIT/INLINE] Features: [SYNTAX/SEMANTIC] Performance: [FILE SIZE LIMITS] Merge: [3-WAY/CONFLICT] Include:

- Diff architecture
- Algorithm selection
- Syntax integration
- View rendering
- Merge engine
- Performance optimization Format as diff tool design.

Implement Search Suggestions

Beginner: How do I implement search suggestions?

Intermediate: I want to add search suggestions for [SEARCH TYPE]. Using [DATA SOURCE] with [SUGGESTION COUNT], I need features like [FEATURES: e.g., "popular searches, history, categories"]. Include personalization and caching.

Advanced: Act as a search UX expert. Design suggestion system with: Sources: [POPULAR/HISTORY/PREDICTIVE] Ranking: [PERSONALIZED/GLOBAL] Categories: [GROUPING/ICONS] Performance: [LATENCY TARGET] Privacy: [DATA HANDLING] Include:

- Suggestion architecture
- Ranking algorithm
- Personalization engine
- Cache strategy
- Privacy framework
- A/B testing setup Format as search suggestion design.

Create Referral Tracking

Beginner: How do I create referral tracking?

Intermediate: I want to track referrals for [CAMPAIGN TYPE]. Using [TRACKING METHODS: e.g., "codes, links, cookies"] with [ATTRIBUTION WINDOW], I need features like [FEATURES: e.g., "multi-touch, rewards"]. Include fraud detection.

Advanced: Act as an attribution expert. Design referral tracking with: Methods: [CODES/LINKS/FINGERPRINT] Attribution: [FIRST/LAST/MULTI] Rewards: [STRUCTURE/TIERS] Fraud: [DETECTION/PREVENTION] Analytics: [FUNNEL/COHORT] Include:

- Tracking architecture
- Attribution engine
- Reward system
- Fraud detection
- Analytics pipeline
- Integration framework Provide referral platform design.

Build Slideshow Editor

Beginner: How do I build a slideshow editor?

Intermediate: I want to create a slideshow editor for [PRESENTATION TYPE]. With [SLIDE ELEMENTS: e.g., "text, images, transitions"] and [FEATURES: e.g., "templates, animations"], I need collaboration and export options.

Advanced: Act as a presentation architect. Design slideshow editor with: Elements: [TYPES/LAYOUTS] Animations: [TRANSITIONS/EFFECTS] Templates: [LIBRARY/CUSTOM] Collaboration: [REAL-TIME/COMMENTS] Export: [FORMATS/QUALITY] Include:

- Editor architecture
- Element framework
- Animation engine
- Template system
- Collaboration layer
- Export pipeline Format as presentation platform design.

Implement Scroll Spy

Beginner: How do I implement scroll spy?

Intermediate: I want to add scroll spy for [PAGE TYPE: e.g., "documentation, landing page"]. Tracking [SECTION COUNT] with [FEATURES: e.g., "smooth scroll, offset"], I need active state management and [UPDATE FREQUENCY]. Include performance optimization.

Advanced: Act as a scroll interaction expert. Design scroll spy system with: Sections: [COUNT/NESTED] Detection: [INTERSECTION/SCROLL] Features: [SMOOTH/OFFSET/HISTORY] Performance: [THROTTLE/DEBOUNCE] Accessibility: [FOCUS MANAGEMENT] Include:

- Scroll spy architecture
- Detection strategy

- State management
- Performance optimization
- History integration
- Accessibility features Format as scroll interaction design.

Create Issue Tracker

Beginner: How do I create an issue tracker?

Intermediate: I want to build an issue tracker for [PROJECT TYPE]. Managing [ISSUE TYPES] with [WORKFLOW STATES], I need features like [FEATURES: e.g., "assignment, labels, milestones"]. Include search and reporting.

Advanced: Act as an issue tracking architect. Design tracker system with: Issues: [TYPES/CUSTOM FIELDS] Workflow: [STATES/TRANSITIONS] Features: [BOARDS/SPRINTS] Permissions: [ROLES/PROJECTS] Integration: [VCS/CI/CD] Include:

- Tracker architecture
- Workflow engine
- Board implementation
- Permission system
- Integration framework
- Analytics dashboard Provide issue tracker design.

Build Affiliate System

Beginner: How do I build an affiliate system?

Intermediate: I want to create an affiliate program for [BUSINESS TYPE]. Managing [AFFILIATE COUNT] with [COMMISSION STRUCTURE], I need features like [FEATURES: e.g., "tracking, payouts, materials"]. Include reporting and fraud prevention.

Advanced: Act as an affiliate architect. Design affiliate platform with: Affiliates: [TIERS/SCALE] Tracking: [METHODS/ACCURACY] Commissions: [STRUCTURE/RULES] Payouts: [AUTOMATION/METHODS] Compliance: [TAX/REGULATIONS] Include:

- Affiliate architecture
- Tracking system
- Commission engine
- Payout automation
- Compliance framework
- Analytics platform Format as affiliate system design.

Implement Data Deduplication

Beginner: How do I implement data deduplication?

Intermediate: I want to deduplicate [DATA TYPE: e.g., "customer records, products"] in my database. Working with [RECORD COUNT] using [MATCHING CRITERIA: e.g., "fuzzy matching, exact match"], I need to handle [MERGE STRATEGY] with conflict resolution.

Advanced: Act as a data quality expert. Design deduplication system with: Data: [VOLUME/TYPES] Matching: [ALGORITHMS/THRESHOLD] Merge: [RULES/PRIORITIES] Performance: [PROCESSING TIME] Audit: [TRACKING/ROLLBACK] Include:

- Deduplication architecture
- Matching algorithms
- Merge engine
- Conflict resolution
- Audit trail
- Performance optimization Format as data quality design.

Create Visitor Analytics

Beginner: How do I create visitor analytics?

Intermediate: I want to track visitor analytics for [WEBSITE TYPE]. Collecting [METRICS: e.g., "pageviews, sessions, conversions"] for [VISITOR VOLUME], I need features like [FEATURES: e.g., "real-time, cohorts, funnels"]. Include privacy compliance.

Advanced: Act as an analytics architect. Design visitor tracking with: Metrics: [STANDARD/CUSTOM] Collection: [CLIENT/SERVER] Privacy: [GDPR/CCPA COMPLIANT] Analysis: [REAL-TIME/BATCH] Storage: [RETENTION/AGGREGATION] Include:

- Analytics architecture
- Collection pipeline
- Privacy framework
- Processing engine
- Visualization layer
- Export capabilities Provide analytics platform design.

Build Workflow Automation

Beginner: How do I build workflow automation?

Intermediate: I want to automate workflows for [BUSINESS PROCESS]. Creating [WORKFLOW COUNT] with [TRIGGER TYPES: e.g., "time, event, webhook"], I need features like [FEATURES: e.g., "conditions, loops, error handling"]. Include monitoring dashboard.

Advanced: Act as an automation architect. Design workflow platform with: Workflows: [COMPLEXITY/SCALE] Triggers: [TYPES/SOURCES] Actions: [INTEGRATIONS/CUSTOM] Logic: [CONDITIONS/LOOPS] Monitoring: [LOGS/METRICS] Include:

- Automation architecture

- Trigger system
- Action framework
- Execution engine
- Error handling
- Monitoring dashboard Format as automation platform design.

Implement Tree Structure

Beginner: How do I implement a tree data structure?

Intermediate: I want to implement a tree for [USE CASE: e.g., "file system, org chart"]. Supporting [NODE COUNT] with [OPERATIONS: e.g., "insert, delete, traverse"], I need features like [FEATURES: e.g., "drag-drop, search, expand/collapse"]. Include visualization.

Advanced: Act as a data structure expert. Design tree system with: Structure: [BINARY/N-ARY/BALANCED] Operations: [CRUD/TRAVERSAL] Performance: [TIME COMPLEXITY] Features: [LAZY LOAD/VIRTUAL] Persistence: [STORAGE STRATEGY] Include:

- Tree architecture
- Node management
- Traversal algorithms
- Balancing strategy
- Visualization engine
- Performance optimization Provide tree implementation guide.

Create Meeting Scheduler

Beginner: How do I create a meeting scheduler?

Intermediate: I want to build a meeting scheduler for [ORGANIZATION SIZE]. Finding [SLOT DURATION] across [PARTICIPANT COUNT] calendars, I need features like [FEATURES: e.g., "buffer time, recurring, room booking"]. Include timezone support.

Advanced: Act as a scheduling architect. Design meeting system with: Participants: [SCALE/AVAILABILITY] Constraints: [ROOMS/EQUIPMENT] Optimization: [ALGORITHMS] Features: [VOTING/SUGGESTIONS] Integration: [CALENDAR SYSTEMS] Include:

- Scheduler architecture
- Availability engine
- Optimization algorithms
- Resource management
- Integration layer
- Notification system Format as scheduling platform design.

Build Rich Text Editor

Beginner: How do I build a rich text editor?

Intermediate: I want to create a rich text editor with [FEATURES: e.g., "formatting, links, media"]. Supporting [CONTENT TYPES] with [TOOLBAR OPTIONS], I need collaborative editing and [EXPORT FORMATS]. Include plugin architecture.

Advanced: Act as an editor architect. Design rich text system with: Features: [FORMATTING/MEDIA/TABLES] Collaboration: [OT/CRDT] Plugins: [ARCHITECTURE/API] Performance: [DOCUMENT SIZE] Export: [HTML/MARKDOWN/PDF] Include:

- Editor architecture
- Content model
- Collaboration engine
- Plugin framework
- Export pipeline
- Mobile optimization Provide editor platform design.

Implement Connection Pool

Beginner: How do I implement a connection pool?

Intermediate: I want to create a connection pool for [RESOURCE TYPE: e.g., "database, API"]. Managing [POOL SIZE] connections with [FEATURES: e.g., "health checks, retry logic"], I need to handle [CONCURRENT REQUESTS]. Include monitoring.

Advanced: Act as a connection architect. Design pool system with: Resources: [TYPES/LIMITS] Sizing: [MIN/MAX/DYNAMIC] Health: [CHECKS/RECOVERY] Performance: [LATENCY/THROUGHPUT] Monitoring: [METRICS/ALERTS] Include:

- Pool architecture
 - Connection lifecycle
 - Health monitoring
 - Dynamic sizing
 - Performance tuning
 - Debug capabilities
- Format as connection pool design.

Create Email Designer

Beginner: How do I create an email designer?

Intermediate: I want to build an email designer for [EMAIL TYPES: e.g., "marketing, transactional"]. With [DESIGN FEATURES: e.g., "drag-drop, templates"] and [CUSTOMIZATION LEVEL], I need responsive design and [PREVIEW OPTIONS]. Include code export.

Advanced: Act as an email design architect. Design email builder with: Templates: [LIBRARY/CUSTOM] Elements: [BLOCKS/COMPONENTS] Responsive: [MOBILE OPTIMIZATION] Testing: [CLIENT PREVIEW] Export: [HTML/MJML/AMP] Include:

- Designer architecture
- Component library

- Responsive system
- Preview engine
- Export framework
- Template marketplace Provide email builder design.

Build Seat Map Selector

Beginner: How do I build a seat map selector?

Intermediate: I want to create a seat selector for [VENUE TYPE: e.g., "theater, airplane"]. Managing [SEAT COUNT] with [FEATURES: e.g., "categories, pricing, availability"], I need interactive selection and [BOOKING RULES]. Include accessibility info.

Advanced: Act as a booking UX architect. Design seat selector with: Venues: [TYPES/LAYOUTS] Categories: [PRICING/FEATURES] Selection: [SINGLE/MULTIPLE/GROUP] Rules: [ADJACENCY/RESTRICTIONS] Performance: [RENDER LIMIT] Include:

- Seat map architecture
- Rendering engine
- Selection logic
- Booking rules
- Accessibility layer
- Performance optimization Format as booking UX design.

Implement Batch Import

Beginner: How do I implement batch import?

Intermediate: I want to batch import [DATA TYPE] from [SOURCE FORMATS]. Processing [RECORD VOLUME] with [VALIDATION RULES], I need features like [FEATURES: e.g., "mapping, preview, rollback"]. Include progress tracking and error handling.

Advanced: Act as a data import architect. Design batch import with: Sources: [FORMATS/APIS] Volume: [RECORDS/SIZE] Validation: [RULES/STAGES] Processing: [CHUNKS/PARALLEL] Recovery: [ROLLBACK/RETRY] Include:

- Import architecture
- Validation pipeline
- Chunk processing
- Error recovery
- Progress tracking
- Mapping framework Provide import system design.

Create Learning Path

Beginner: How do I create a learning path system?

Intermediate: I want to build learning paths for [EDUCATION TYPE]. Organizing [CONTENT TYPES: e.g., "courses, quizzes, projects"] with [PROGRESSION RULES], I need features like [FEATURES: e.g., "prerequisites, achievements"]. Include progress tracking.

Advanced: Act as an education architect. Design learning platform with: Content: [TYPES/STRUCTURE] Paths: [LINEAR/ADAPTIVE] Assessment: [METHODS/SCORING] Progress: [TRACKING/GAMIFICATION] Analytics: [LEARNER/CONTENT] Include:

- Path architecture
- Content management
- Assessment engine
- Progress system
- Analytics framework
- Recommendation engine Format as learning platform design.

Build Commenting Thread

Beginner: How do I build threaded comments?

Intermediate: I want to implement threaded comments for [PLATFORM TYPE]. Supporting [NESTING DEPTH] with [FEATURES: e.g., "voting, mentions, markdown"], I need real-time updates and [MODERATION OPTIONS]. Include performance optimization.

Advanced: Act as a discussion architect. Design comment threading with: Structure: [DEPTH/BRANCHING] Features: [VOTING/REACTIONS/EDIT] Performance: [LAZY LOAD/PAGINATION] Moderation: [AUTO/MANUAL/COMMUNITY] Real-time: [UPDATES/NOTIFICATIONS] Include:

- Threading architecture
- Data structure
- Real-time system
- Moderation pipeline
- Performance strategy
- Analytics integration Format as discussion platform design.

Implement Memory Cache

Beginner: How do I implement a memory cache?

Intermediate: I want to create an in-memory cache for [DATA TYPE]. With [CACHE SIZE] and [EVICTION POLICY: e.g., "LRU, LFU, TTL"], I need features like [FEATURES: e.g., "tags, warming, statistics"]. Include thread safety.

Advanced: Act as a caching architect. Design memory cache with: Size: [LIMITS/DYNAMIC] Eviction: [POLICIES/HYBRID] Concurrency: [THREAD SAFETY] Features: [TAGS/NAMESPACES] Monitoring: [STATS/ALERTS] Include:

- Cache architecture
- Eviction algorithms
- Concurrency control

- Tag system
- Monitoring framework
- Performance analysis Provide cache implementation guide.

Create Drawing Canvas

Beginner: How do I create a drawing canvas?

Intermediate: I want to build a drawing canvas for [USE CASE: e.g., "whiteboard, annotations"]. Supporting [TOOLS: e.g., "pen, shapes, text"] with [FEATURES: e.g., "layers, undo"], I need touch support and [EXPORT OPTIONS]. Include collaboration features.

Advanced: Act as a canvas architect. Design drawing system with: Tools: [BRUSHES/SHAPES/TEXT] Features: [LAYERS/BLEND MODES] Input: [MOUSE/TOUCH/STYLUS] Collaboration: [REAL-TIME/CURSORS] Export: [FORMATS/RESOLUTION] Include:

- Canvas architecture
- Tool framework
- Layer system
- Collaboration engine
- Export pipeline
- Performance optimization Format as drawing platform design.

Build Subscription Box

Beginner: How do I build a subscription box service?

Intermediate: I want to create subscription boxes for [PRODUCT TYPE]. Managing [SUBSCRIBER COUNT] with [FEATURES: e.g., "customization, skip, pause"], I need inventory management and [SHIPPING INTEGRATION]. Include customer portal.

Advanced: Act as a subscription architect. Design box platform with: Products: [CURATION/CUSTOMIZATION] Billing: [CYCLES/FLEXIBILITY] Inventory:

[ALLOCATION/FORECASTING] Shipping: [PROVIDERS/TRACKING] Portal:
[PREFERENCES/HISTORY] Include:

- Subscription architecture
- Product curation
- Inventory allocation
- Billing integration
- Portal features
- Analytics dashboard Provide subscription box design.

Implement API Versioning

Beginner: How do I implement API versioning?

Intermediate: I want to version my API for [SERVICE TYPE]. Using [VERSIONING STRATEGY: e.g., "URL, header, content negotiation"], I need to support [VERSION COUNT] with [DEPRECATION POLICY]. Include migration guides.

Advanced: Act as an API architect. Design versioning system with: Strategy: [URL/HEADER/ACCEPT] Versions: [LIFECYCLE/SUPPORT] Breaking changes: [DETECTION/MIGRATION] Documentation: [AUTO-GENERATION] Monitoring: [USAGE/DEPRECATION] Include:

- Versioning architecture
- Routing strategy
- Migration framework
- Documentation system
- Usage analytics
- Sunset process Format as API versioning guide.

Create Photo Gallery

Beginner: How do I create a photo gallery?

Intermediate: I want to build a photo gallery for [USE CASE]. Displaying [PHOTO COUNT] with [FEATURES: e.g., "albums, tags, lightbox"], I need lazy loading and [VIEW OPTIONS: e.g., "grid, masonry"]. Include EXIF data handling.

Advanced: Act as a gallery architect. Design photo system with: Photos: [VOLUME/RESOLUTION] Organization: [ALBUMS/TAGS/FACES] Views: [LAYOUTS/TRANSITIONS] Features: [SEARCH/SHARE/DOWNLOAD] Performance: [LOADING/CACHING] Include:

- Gallery architecture
- Image processing
- Metadata handling
- View engine
- Sharing system
- Performance optimization Provide gallery platform design.

Build Permission System

Beginner: How do I build a permission system?

Intermediate: I want to implement permissions for [APPLICATION TYPE]. Managing [RESOURCE TYPES] with [PERMISSION GRANULARITY], I need features like [FEATURES: e.g., "inheritance, groups, delegation"]. Include audit logging.

Advanced: Act as a security architect. Design permission system with: Model: [RBAC/ABAC/HYBRID] Resources: [HIERARCHY/TYPES] Permissions: [GRANULARITY/INHERITANCE] Performance: [EVALUATION SPEED] Audit: [COMPLIANCE LEVEL] Include:

- Permission architecture
- Evaluation engine
- Inheritance model
- Caching strategy
- Audit framework

- Migration tools Format as permission design guide.

Implement Smart Home

Beginner: How do I implement smart home controls?

Intermediate: I want to control smart home devices for [DEVICE TYPES: e.g., "lights, thermostats, locks"]. Managing [DEVICE COUNT] with [PROTOCOLS: e.g., "Zigbee, Z-Wave, WiFi"], I need automation and [FEATURES: e.g., "scenes, schedules"]. Include voice control.

Advanced: Act as an IoT architect. Design smart home platform with: Devices: [TYPES/PROTOCOLS] Automation: [RULES/TRIGGERS] Integration: [PLATFORMS/APIS] Security: [ENCRYPTION/ACCESS] Interface: [MOBILE/VOICE/WEB] Include:

- IoT architecture
- Device management
- Automation engine
- Security framework
- Interface design
- Local/cloud hybrid Provide smart home design.

Create Tournament Bracket

Beginner: How do I create a tournament bracket?

Intermediate: I want to build tournament brackets for [TOURNAMENT TYPE: e.g., "single elimination, round robin"]. Managing [PARTICIPANT COUNT] with [FEATURES: e.g., "seeding, scheduling"], I need score tracking and [BRACKET FORMATS]. Include live updates.

Advanced: Act as a tournament architect. Design bracket system with: Formats: [SINGLE/DOUBLE/SWISS] Participants: [SEEDING/RANKING] Scheduling: [CONFLICTS/VENUES] Scoring: [SYSTEMS/TIEBREAKERS] Display: [VIEWS/EMBEDS] Include:

- Bracket architecture
- Tournament engine
- Scheduling algorithm
- Score management
- Display framework
- Analytics tracking Format as tournament design.

Build Recipe Search

Beginner: How do I build recipe search?

Intermediate: I want to search recipes by [CRITERIA: e.g., "ingredients, cuisine, dietary"]. Searching [RECIPE COUNT] with [FEATURES: e.g., "filters, substitutions"], I need nutritional data and [RANKING OPTIONS]. Include personalization.

Advanced: Act as a culinary search expert. Design recipe search with: Criteria: [INGREDIENTS/NUTRITION/TAGS] Matching: [EXACT/FUZZY/SEMANTIC] Filters: [DIETARY/TIME/SKILL] Ranking: [RELEVANCE/PERSONAL] Features: [SUBSTITUTIONS/SCALING] Include:

- Search architecture
- Ingredient matching
- Filter system
- Ranking algorithm
- Personalization engine
- Nutrition calculator Provide recipe search design.

Implement Distributed Cache

Beginner: How do I implement a distributed cache?

Intermediate: I want to create a distributed cache for [USE CASE]. Using [TECHNOLOGY: e.g., "Redis Cluster, Hazelcast"], I need to cache [DATA VOLUME]

across [NODE COUNT] with [CONSISTENCY REQUIREMENTS]. Include failover handling.

Advanced: Act as a distributed systems expert. Design cache cluster with: Topology: [SHARDING/REPLICATION] Consistency: [EVENTUAL/STRONG] Partitioning: [HASH/RANGE] Failover: [AUTOMATIC/MANUAL] Monitoring: [HEALTH/PERFORMANCE] Include:

- Cluster architecture
- Sharding strategy
- Replication design
- Failover mechanism
- Monitoring system
- Client libraries Format as distributed cache design.

Create Video Conference

Beginner: How do I create video conferencing?

Intermediate: I want to build video conferencing for [USE CASE]. Supporting [PARTICIPANT COUNT] with [FEATURES: e.g., "screen share, recording, chat"], I need quality adaptation and [PLATFORM SUPPORT]. Include virtual backgrounds.

Advanced: Act as a WebRTC architect. Design conference system with: Scale: [PARTICIPANTS/ROOMS] Media: [VIDEO/AUDIO/SCREEN] Quality: [ADAPTIVE/SIMULCAST] Features: [RECORDING/STREAMING] Infrastructure: [SFU/MCU] Include:

- Conference architecture
- Media server design
- Quality adaptation
- Recording system
- Scaling strategy
- Client optimization Provide video conference design.

Build Asset Manager

Beginner: How do I build an asset manager?

Intermediate: I want to manage digital assets for [ASSET TYPES: e.g., "images, videos, documents"]. Handling [VOLUME] with [FEATURES: e.g., "tagging, versions, permissions"], I need search and [WORKFLOW OPTIONS]. Include CDN integration.

Advanced: Act as a DAM architect. Design asset management with: Assets: [TYPES/VOLUME] Metadata: [EXTRACTION/CUSTOM] Workflow: [APPROVAL/PROCESSING] Distribution: [CDN/TRANSFORMS] Integration: [CREATIVE TOOLS] Include:

- DAM architecture
- Metadata system
- Workflow engine
- Processing pipeline
- CDN strategy
- API framework Format as asset platform design.

Implement Load Testing

Beginner: How do I implement load testing?

Intermediate: I want to load test [APPLICATION TYPE]. Simulating [USER COUNT] with [SCENARIO TYPES: e.g., "ramp-up, spike, sustained"], I need to measure [METRICS: e.g., "response time, throughput"]. Include reporting dashboard.

Advanced: Act as a performance engineer. Design load testing platform with: Scenarios: [PATTERNS/SCRIPTS] Load: [DISTRIBUTION/SCALE] Metrics: [APM/CUSTOM] Analysis: [BOTTLENECKS/TRENDS] Integration: [CI/CD/MONITORING] Include:

- Testing architecture
- Load generation
- Metric collection

- Analysis engine
- Reporting system
- CI/CD integration Provide load testing design.

Create Wiki System

Beginner: How do I create a wiki system?

Intermediate: I want to build a wiki for [CONTENT DOMAIN]. Managing [PAGE COUNT] with [FEATURES: e.g., "versioning, categories, templates"], I need collaborative editing and [PERMISSION LEVELS]. Include full-text search.

Advanced: Act as a wiki architect. Design wiki platform with: Content: [STRUCTURE/VOLUME] Editing: [WYSIWYG/MARKUP] Versioning: [DIFF/HISTORY] Organization: [NAMESPACES/CATEGORIES] Search: [FULL-TEXT/SEMANTIC] Include:

- Wiki architecture
- Content management
- Version control
- Search implementation
- Template engine
- Access control Format as wiki platform design.

Build Fitness Tracker

Beginner: How do I build a fitness tracker?

Intermediate: I want to create a fitness tracker for [ACTIVITY TYPES]. Recording [METRICS: e.g., "workouts, nutrition, sleep"] with [DEVICE INTEGRATION], I need goal setting and [ANALYSIS FEATURES]. Include social features.

Advanced: Act as a fitness platform architect. Design tracker system with: Activities: [TYPES/DETECTION] Metrics: [BIOMETRIC/PERFORMANCE] Devices: [WEARABLES/SENSORS] Analysis: [TRENDS/RECOMMENDATIONS] Social: [CHALLENGES/SHARING] Include:

- Tracker architecture
- Device integration
- Data aggregation
- Analysis engine
- Social framework
- Privacy controls Provide fitness platform design.

Implement Graph Database

Beginner: How do I implement a graph database?

Intermediate: I want to use a graph database for [USE CASE: e.g., "social network, recommendation engine"]. Modeling [NODE TYPES] and [RELATIONSHIP TYPES] with [QUERY PATTERNS], I need traversal optimization and visualization.

Advanced: Act as a graph database expert. Design graph system with: Model: [NODES/EDGES/PROPERTIES] Storage: [NATIVE/INDEXED] Queries: [CYPHER/GREMLIN] Performance: [TRAVERSAL/INDEXING] Visualization: [LAYOUT/INTERACTION] Include:

- Graph architecture
- Storage engine
- Query optimization
- Index strategy
- Visualization framework
- Scaling approach Format as graph database design.

Create Appointment System

Beginner: How do I create an appointment system?

Intermediate: I want to build appointments for [SERVICE TYPE]. Managing [PROVIDER COUNT] with [APPOINTMENT TYPES], I need features like

[FEATURES: e.g., "availability rules, reminders, waitlist"]. Include customer self-service.

Advanced: Act as an appointment architect. Design booking system with: Providers: [SCHEDULES/SERVICES] Availability: [RULES/EXCEPTIONS] Booking: [CHANNELS/CONFIRMATION] Queue: [WAITLIST/OPTIMIZATION] Integration: [CALENDAR/PAYMENT] Include:

- Appointment architecture
- Availability engine
- Booking workflow
- Queue management
- Integration layer
- Analytics dashboard Provide appointment platform design.

Build Code Playground

Beginner: How do I build a code playground?

Intermediate: I want to create a code playground for [LANGUAGES]. Supporting [FEATURES: e.g., "multiple files, console output, sharing"], I need sandboxed execution and [COLLABORATION OPTIONS]. Include code templates.

Advanced: Act as a playground architect. Design code environment with: Languages: [RUNTIMES/VERSIONS] Execution: [SANDBOX/TIMEOUT] Features: [DEBUGGING/SHARING] Storage: [PERSISTENCE/VERSIONS] Collaboration: [LIVE/ASYNC] Include:

- Playground architecture
- Execution sandbox
- File management
- Sharing system
- Template library
- Performance limits Format as playground design.

Implement Spell Correction

Beginner: How do I implement spell correction?

Intermediate: I want to auto-correct spelling for [INPUT TYPE: e.g., "search queries, user content"]. Using [ALGORITHM: e.g., "edit distance, phonetic"], I need to handle [LANGUAGE COUNT] with context awareness and custom dictionaries.

Advanced: Act as a NLP expert. Design spell correction with: Algorithms: [DISTANCE/ML-BASED] Context: [N-GRAM/SEMANTIC] Languages: [MODELS/RULES] Dictionary: [SOURCES/CUSTOM] Performance: [LATENCY/ACCURACY] Include:

- Correction architecture
- Algorithm selection
- Context analysis
- Dictionary management
- Model training
- API design Provide spell correction design.

Create Marketplace Platform

Beginner: How do I create a marketplace platform?

Intermediate: I want to build a marketplace for [PRODUCT/SERVICE TYPE]. Supporting [SELLER COUNT] and [BUYER COUNT] with [FEATURES: e.g., "listings, reviews, messaging"], I need payment processing and dispute resolution.

Advanced: Act as a marketplace architect. Design platform with: Participants: [SELLERS/BUYERS/SCALE] Listings: [TYPES/APPROVAL] Transactions: [ESCROW/SPLIT] Trust: [REVIEWS/VERIFICATION] Disputes: [RESOLUTION/ESCALATION] Include:

- Marketplace architecture
- Listing management
- Payment flow

- Trust system
- Dispute framework
- Analytics platform Format as marketplace design.

Build Weather Alert System

Beginner: How do I build a weather alert system?

Intermediate: I want to create weather alerts for [REGION COVERAGE]. Monitoring [WEATHER TYPES: e.g., "severe, advisory"] from [DATA SOURCES], I need to send [NOTIFICATION CHANNELS] with location-based targeting.

Advanced: Act as an alert system architect. Design weather alerting with: Coverage: [REGIONS/RESOLUTION] Sources: [APIS/AGGREGATION] Conditions: [THRESHOLDS/RULES] Delivery: [CHANNELS/PRIORITY] Targeting: [LOCATION/PREFERENCES] Include:

- Alert architecture
- Data ingestion
- Rule engine
- Delivery system
- Targeting algorithm
- Redundancy design Provide alert platform design.

Implement Query Builder

Beginner: How do I implement a query builder?

Intermediate: I want to build a visual query builder for [DATABASE TYPE]. Supporting [QUERY OPERATIONS: e.g., "filter, join, aggregate"] with drag-drop interface, I need SQL generation and [PREVIEW OPTIONS]. Include query saving.

Advanced: Act as a query interface expert. Design query builder with: Operations: [SQL/NoSQL SUPPORT] Interface: [VISUAL/DRAG-DROP] Validation: [SYNTAX/PERMISSIONS] Optimization: [HINTS/EXPLAIN] Sharing: [SAVE/EXPORT/EMBED] Include:

- Builder architecture
- Operation framework
- SQL generation
- Validation system
- Optimization hints
- Collaboration features Format as query builder design.

Create Inventory Forecast

Beginner: How do I create inventory forecasting?

Intermediate: I want to forecast inventory for [PRODUCT TYPES]. Using [HISTORICAL DATA] with [FORECASTING METHODS: e.g., "moving average, ML models"], I need to predict [TIME HORIZON] with seasonality handling.

Advanced: Act as a supply chain architect. Design forecasting system with: Models: [STATISTICAL/ML/HYBRID] Factors: [SEASONALITY/TRENDS/EVENTS] Accuracy: [METRICS/CONFIDENCE] Integration: [ERP/ORDERING] Scenarios: [WHAT-IF/SIMULATION] Include:

- Forecast architecture
- Model selection
- Data pipeline
- Accuracy tracking
- Scenario engine
- Integration layer Format as forecasting design.

Build Document Editor

Beginner: How do I build a document editor?

Intermediate: I want to create a document editor for [DOCUMENT TYPES]. Supporting [FEATURES: e.g., "formatting, collaboration, comments"] with [FILE FORMATS], I need version control and [EXPORT OPTIONS]. Include templates.

Advanced: Act as a document platform architect. Design editor system with:

Features: [FORMATTING/LAYOUT/STYLES] Collaboration: [REAL-TIME/COMMENTS] Versioning: [HISTORY/BRANCHES] Templates: [LIBRARY/CUSTOM] Export: [FORMATS/FIDELITY] Include:

- Editor architecture
- Document model
- Collaboration engine
- Version system
- Template framework
- Export pipeline Provide document platform design.

Implement Health Monitoring

Beginner: How do I implement health monitoring?

Intermediate: I want to monitor health metrics for [SYSTEM COMPONENTS].

Collecting [METRIC TYPES: e.g., "availability, latency, errors"] at [FREQUENCY], I need alerting and [DASHBOARD FEATURES]. Include dependency tracking.

Advanced: Act as an SRE architect. Design health monitoring with: Metrics:

[GOLDEN SIGNALS/CUSTOM] Collection: [PUSH/PULL/EVENTS] Aggregation:

[TIME SERIES/STATISTICS] Alerting: [RULES/ESCALATION] Visualization:

[DASHBOARDS/REPORTS] Include:

- Monitoring architecture
- Metric collection
- Alert engine
- Dependency mapping
- Dashboard design
- Incident response Format as observability design.

Create Food Delivery

Beginner: How do I create a food delivery system?

Intermediate: I want to build food delivery for [MARKET TYPE]. Managing [RESTAURANT COUNT] and [DRIVER COUNT] with [FEATURES: e.g., "real-time tracking, ratings"], I need order management and route optimization.

Advanced: Act as a delivery platform architect. Design food delivery with: Participants: [RESTAURANTS/DRIVERS/CUSTOMERS] Orders: [FLOW/STATES/TRACKING] Dispatch: [ASSIGNMENT/OPTIMIZATION] Tracking: [REAL-TIME/ETA] Payments: [SPLIT/TIPS/REFUNDS] Include:

- Platform architecture
- Order management
- Dispatch algorithm
- Tracking system
- Payment flow
- Analytics framework Provide delivery platform design.

Build Annotation Tool

Beginner: How do I build an annotation tool?

Intermediate: I want to create annotations for [CONTENT TYPE: e.g., "images, documents, videos"]. Supporting [ANNOTATION TYPES: e.g., "text, shapes, labels"] with [COLLABORATION FEATURES], I need export and version control.

Advanced: Act as an annotation architect. Design annotation system with: Content: [TYPES/FORMATS] Annotations: [SHAPES/METADATA] Collaboration: [REAL-TIME/REVIEW] Workflow: [STAGES/APPROVAL] Export: [FORMATS/STANDARDS] Include:

- Annotation architecture
- Tool framework
- Collaboration engine

- Workflow system
- Export pipeline
- Performance optimization Format as annotation platform design.

Implement Feature Flag

Beginner: How do I implement feature flags?

Intermediate: I want to add feature flags to [APPLICATION TYPE]. Managing [FLAG COUNT] with [TARGETING: e.g., "users, groups, percentage"], I need gradual rollout and [EVALUATION PERFORMANCE]. Include analytics integration.

Advanced: Act as a feature flag architect. Design flag system with: Flags: [TYPES/LIFECYCLE] Targeting: [RULES/SEGMENTS] Evaluation: [CLIENT/SERVER] Rollout: [STRATEGIES/SAFETY] Analytics: [METRICS/EXPERIMENTS] Include:

- Flag architecture
- Rule engine
- SDK design
- Rollout strategies
- Analytics integration
- Emergency controls Provide feature flag design.

Create School Management

Beginner: How do I create a school management system?

Intermediate: I want to build school management for [INSTITUTION TYPE]. Managing [STUDENT COUNT] with [MODULES: e.g., "attendance, grades, schedules"], I need parent portal and [COMMUNICATION FEATURES]. Include reporting.

Advanced: Act as an education architect. Design school system with: Modules: [ACADEMIC/ADMIN/FINANCE] Users: [ROLES/PERMISSIONS] Academic: [GRADES/ATTENDANCE/SCHEDULE] Communication: [ANNOUNCEMENTS/MESSAGING] Reporting: [CARDS/ANALYTICS/COMPLIANCE] Include:

- System architecture
- Module design
- User management
- Academic engine
- Communication platform
- Reporting framework Format as school platform design.

Build Translation Service

Beginner: How do I build a translation service?

Intermediate: I want to create translation for [CONTENT TYPES]. Supporting [LANGUAGE PAIRS] with [FEATURES: e.g., "glossaries, context"], I need API access and [QUALITY OPTIONS: e.g., "machine, human"]. Include version management.

Advanced: Act as a translation architect. Design translation platform with: Languages: [PAIRS/DETECTION] Methods: [MT/HUMAN/HYBRID] Quality: [LEVELS/REVIEW] Glossaries: [MANAGEMENT/CONTEXT] API: [REST/STREAMING/BATCH] Include:

- Translation architecture
- Engine selection
- Quality framework
- Glossary system
- API design
- Workflow management Provide translation platform design.

Optimize Frontend Performance

Beginner: How do I optimize frontend performance?

Intermediate: I want to optimize my [FRAMEWORK: e.g., "React"] application's performance. Current metrics are [METRICS] and target is [GOALS]. Focus on [AREAS: e.g., "bundle size, rendering"]. Include measurement tools and techniques.

Advanced: Act as a performance engineer. Optimize a frontend application with:
Framework: [TYPE/VERSION] Metrics: [CORE WEB VITALS] Budget:
[PERFORMANCE BUDGET] Users: [GEOGRAPHIC DISTRIBUTION] Devices:
[TARGET DEVICES] Include:

- Bundle optimization strategy
- Rendering optimization
- Network optimization
- Caching strategies
- Monitoring setup
- A/B testing framework Provide before/after analysis.

Create PDF Generator

Beginner: How do I create a PDF generator?

Intermediate: I want to generate PDFs for [PURPOSE: e.g., "invoices, reports"]. Using [LIBRARY: e.g., "Puppeteer, PDFKit"], I need to create [DOCUMENT TYPES] with [FEATURES: e.g., "charts, tables"]. Include template management and performance optimization.

Advanced: Act as a document generation expert. Design a PDF system handling:
Volume: [DOCUMENTS/DAY] Complexity: [PAGES/ELEMENTS] Templates:
[COUNT] Performance: [GENERATION TIME] Features: [SIGNATURES/FORMS]
Include:

- Template engine selection
- Asset management
- Batch processing
- Digital signatures
- Archival strategy
- Cost optimization Format as technical architecture.

Implement Blockchain Integration

Beginner: How do I implement blockchain integration?

Intermediate: I want to integrate blockchain for [USE CASE: e.g., "supply chain tracking"]. Using [BLOCKCHAIN: e.g., "Ethereum, Hyperledger"], I need to implement [FEATURES: e.g., "smart contracts, transactions"]. Include gas optimization and security.

Advanced: Act as a blockchain architect. Design a DApp with: Blockchain: [PLATFORM] Consensus: [MECHANISM] Throughput: [TPS] Smart contracts: [COMPLEXITY] Integration: [EXISTING SYSTEMS] Include:

- Contract architecture
- Off-chain storage
- Oracle integration
- Gas optimization
- Security audit plan
- Scaling solutions Provide implementation roadmap.

Build Admin Dashboard

Beginner: How do I build an admin dashboard?

Intermediate: I want to build an admin dashboard for [APPLICATION TYPE]. Using [FRAMEWORK: e.g., "React Admin"], I need features like [FEATURES: e.g., "user management, analytics"]. Include role-based access control and audit logging.

Advanced: Act as a dashboard architect. Design an admin system with: Users: [ADMIN COUNT] Entities: [MANAGED RESOURCES] Permissions: [RBAC/ABAC] Analytics: [METRICS NEEDED] Compliance: [AUDIT REQUIREMENTS] Include:

- Multi-tenant architecture
- Permission system design
- Audit trail implementation
- Real-time monitoring

- Bulk operations
- Extension framework Format as technical specification.

Create API Documentation

Beginner: How do I create API documentation?

Intermediate: I want to create documentation for my [API TYPE] API. Using [TOOL: e.g., "Swagger, Postman"], I need to document [ENDPOINTS COUNT] endpoints with [FEATURES: e.g., "examples, testing"]. Include versioning and automation strategies.

Advanced: Act as an API documentation expert. Create a documentation system with: APIs: [COUNT/VERSIONS] Format: [OPENAPI/ASYNCAPI] Testing: [INTEGRATION] Localization: [LANGUAGES] Hosting: [PLATFORM] Include:

- Documentation as code
- Interactive examples
- SDK generation
- Change tracking
- Review workflow
- Analytics integration Provide implementation guide.

Handle Data Migration

Beginner: How do I handle data migration?

Intermediate: I want to migrate data from [SOURCE] to [DESTINATION]. The data volume is [SIZE] with [COMPLEXITY: e.g., "schema changes, transformations"]. I need strategies for validation, rollback, and minimal downtime.

Advanced: Act as a data migration expert. Design a migration strategy for: Data volume: [SIZE] Downtime: [TOLERANCE] Transformation: [COMPLEXITY] Validation: [REQUIREMENTS] Rollback: [STRATEGY] Include:

- ETL pipeline design
- Zero-downtime approach

- Data validation framework
- Performance optimization
- Audit trail
- Disaster recovery Format as migration plan.

Build Recommendation System

Beginner: How do I build a recommendation system?

Intermediate: I want to build a recommendation system for [DOMAIN: e.g., "e-commerce products"]. Using [APPROACH: e.g., "collaborative filtering"], I need to recommend [ITEMS] to [USERS]. Include evaluation metrics and cold start handling.

Advanced: Act as a ML engineer. Design a recommendation system with: Users: [SCALE] Items: [CATALOG SIZE] Features: [DATA TYPES] Latency: [REQUIREMENTS] Algorithms: [ENSEMBLE] Include:

- Feature engineering pipeline
- Model serving architecture
- A/B testing framework
- Explanation generation
- Bias mitigation
- Real-time updates Provide ML system design.

Implement Two-Factor Authentication

Beginner: How do I implement two-factor authentication?

Intermediate: I want to implement 2FA for my [APPLICATION TYPE]. Supporting [METHODS: e.g., "SMS, TOTP, backup codes"], I need to handle [USER COUNT] users. Include recovery flows and security considerations.

Advanced: Act as a security architect. Design a 2FA system supporting: Methods: [SMS/TOTP/FIDO2/BIOMETRIC] Users: [SCALE] Recovery: [OPTIONS] Compliance: [STANDARDS] Integration: [EXISTING AUTH] Include:

- Multi-factor architecture
- Backup code generation
- Risk-based authentication
- Account recovery
- Audit logging
- Phishing resistance Format as security design document.

Create Data Pipeline

Beginner: How do I create a data pipeline?

Intermediate: I want to create a data pipeline for [PURPOSE: e.g., "ETL processing"]. Using [TOOLS: e.g., "Apache Airflow"], I need to process [DATA VOLUME] from [SOURCES]. Include error handling and monitoring strategies.

Advanced: Act as a data engineer. Design a data pipeline handling: Sources: [SYSTEMS/FORMATS] Volume: [GB/DAY] Latency: [BATCH/STREAMING] Transformations: [COMPLEXITY] Quality: [SLA] Include:

- Orchestration design
- Schema evolution
- Data quality checks
- Lineage tracking
- Cost optimization
- Disaster recovery Provide architecture blueprint.

Build Video Streaming Service

Beginner: How do I build a video streaming service?

Intermediate: I want to build video streaming for [USE CASE: e.g., "educational content"]. Supporting [QUALITY LEVELS] with [CONCURRENT VIEWERS], I need features like [FEATURES: e.g., "adaptive bitrate, subtitles"]. Include CDN setup and bandwidth optimization.

Advanced: Act as a video platform architect. Design a streaming system supporting: Viewers: [CONCURRENT] Content: [HOURS/BITRATES] Latency: [LIVE/VOD] Regions: [GLOBAL DISTRIBUTION] Features: [DRM/ANALYTICS] Include:

- Transcoding pipeline
 - CDN strategy
 - Adaptive streaming
 - DRM implementation
 - Analytics pipeline
 - Cost optimization
- Format as technical architecture.

Implement Message Queue

Beginner: How do I implement a message queue?

Intermediate: I want to implement message queuing for [USE CASE]. Using [TECHNOLOGY: e.g., "RabbitMQ, Kafka"], I need to handle [MESSAGE VOLUME] with [RELIABILITY REQUIREMENTS]. Include dead letter queues and monitoring.

Advanced: Act as a messaging expert. Design a message queue system with: Throughput: [MESSAGES/SEC] Latency: [REQUIREMENTS] Durability: [GUARANTEES] Ordering: [REQUIREMENTS] Topology: [PATTERNS] Include:

- Broker selection criteria
 - Partitioning strategy
 - Consumer group design
 - Error handling patterns
 - Monitoring stack
 - Capacity planning
- Provide implementation guide.

Create Monitoring Dashboard

Beginner: How do I create a monitoring dashboard?

Intermediate: I want to create monitoring for my [SYSTEM TYPE]. Using [TOOLS: e.g., "Prometheus, Grafana"], I need to track [METRICS: e.g., "performance, errors, business KPIs"]. Include alerting rules and visualization best practices.

Advanced: Act as an observability expert. Design a monitoring system for: Services: [COUNT] Metrics: [VOLUME/SEC] Retention: [PERIOD] Dashboards: [AUDIENCE] Compliance: [REQUIREMENTS] Include:

- Metric collection strategy
- Dashboard hierarchy
- Alert fatigue prevention
- SLI/SLO definition
- Capacity planning
- Cost optimization Format as observability guide.

Build Payment Integration

Beginner: How do I build payment integration?

Intermediate: I want to integrate payments using [PROVIDER: e.g., "Stripe, PayPal"]. Supporting [PAYMENT METHODS] for [BUSINESS MODEL], I need features like [FEATURES: e.g., "subscriptions, refunds"]. Include PCI compliance and error handling.

Advanced: Act as a payments architect. Design a payment system handling: Providers: [MULTIPLE] Methods: [CARDS/WALLETS/CRYPTO] Volume: [TRANSACTIONS/DAY] Compliance: [PCI/REGIONAL] Features: [SUBSCRIPTIONS/MARKETPLACE] Include:

- Payment orchestration
- Fraud detection
- Reconciliation system
- Compliance checklist
- Failover strategy

- Settlement handling Provide architecture document.

Implement Server-Side Rendering

Beginner: How do I implement server-side rendering?

Intermediate: I want to implement SSR for my [FRAMEWORK: e.g., "React, Vue"] application. I need to optimize for [GOALS: e.g., "SEO, performance"] while maintaining [FEATURES: e.g., "dynamic content, user sessions"]. Include caching strategies.

Advanced: Act as a SSR expert. Design a server-side rendering system with: Framework: [TYPE] Traffic: [REQUESTS/SEC] Performance: [TTFB TARGET] SEO: [REQUIREMENTS] Infrastructure: [EDGE/TRADITIONAL] Include:

- Rendering strategy (SSR/SSG/ISR)
- Edge computing setup
- Cache invalidation
- State hydration
- Performance monitoring
- Cost analysis Format as implementation guide.

Create Load Balancer Configuration

Beginner: How do I create a load balancer configuration?

Intermediate: I want to configure load balancing for [APPLICATION TYPE]. Using [LOAD BALANCER: e.g., "Nginx, HAProxy"], I need to distribute [TRAFFIC VOLUME] across [SERVER COUNT] servers. Include health checks and SSL termination.

Advanced: Act as a infrastructure architect. Design load balancing for: Traffic: [REQUESTS/SEC] Backends: [COUNT/TYPES] Algorithms: [ROUND ROBIN/WEIGHTED] Features: [SSL/WAF/CACHING] Availability: [SLA] Include:

- Multi-tier architecture
- Geographic distribution

- DDoS mitigation
- Blue-green deployment
- Monitoring integration
- Disaster recovery Provide configuration templates.

Build Notification System

Beginner: How do I build a notification system?

Intermediate: I want to build notifications for [APPLICATION TYPE]. Supporting [CHANNELS: e.g., "email, push, SMS"], I need to send [VOLUME] notifications with [FEATURES: e.g., "preferences, scheduling"]. Include delivery tracking and user preferences.

Advanced: Act as a notification architect. Design a system handling: Channels: [EMAIL/PUSH/SMS/IN-APP] Volume: [NOTIFICATIONS/DAY] Latency: [REQUIREMENTS] Personalization: [LEVEL] Compliance: [REGULATIONS] Include:

- Channel abstraction layer
- Template management
- Preference center
- Delivery optimization
- Analytics pipeline
- Cost optimization Format as system design document.

Implement Elasticsearch

Beginner: How do I implement Elasticsearch?

Intermediate: I want to implement Elasticsearch for [USE CASE: e.g., "full-text search"]. Indexing [DATA VOLUME] with [UPDATE FREQUENCY], I need features like [FEATURES: e.g., "aggregations, suggestions"]. Include cluster setup and optimization.

Advanced: Act as a search engineer. Design an Elasticsearch cluster for: Data: [VOLUME/GROWTH] Queries: [QPS] Latency: [P99 TARGET] Features: [REQUIREMENTS] Availability: [SLA] Include:

- Cluster topology design
- Index lifecycle management
- Query optimization
- Monitoring setup
- Backup strategy
- Cost optimization Provide deployment guide.

Create Backup Strategy

Beginner: How do I create a backup strategy?

Intermediate: I want to create backups for my [SYSTEM TYPE]. With [DATA VOLUME] and [RPO/RTO REQUIREMENTS], I need to backup [COMPONENTS: e.g., "database, files, configurations"]. Include testing and restoration procedures.

Advanced: Act as a disaster recovery expert. Design a backup system with: Data: [VOLUME/TYPES] RPO: [MINUTES/HOURS] RTO: [REQUIREMENTS] Compliance: [REGULATIONS] Budget: [CONSTRAINTS] Include:

- 3-2-1 backup strategy
- Automated testing
- Encryption approach
- Cross-region replication
- Recovery procedures
- Cost optimization Format as DR planning document.

Build A/B Testing Framework

Beginner: How do I build an A/B testing framework?

Intermediate: I want to implement A/B testing for [APPLICATION TYPE]. Testing [FEATURES: e.g., "UI changes, algorithms"], I need to handle [USER BASE] with [STATISTICAL REQUIREMENTS]. Include experiment design and analysis tools.

Advanced: Act as an experimentation expert. Design an A/B testing platform with: Users: [VOLUME] Experiments: [CONCURRENT] Segments: [TARGETING] Metrics: [PRIMARY/SECONDARY] Analysis: [STATISTICAL RIGOR] Include:

- Feature flag system
- Traffic allocation
- Statistical engine
- Metric collection
- Result visualization
- Experiment governance Provide platform architecture.

Implement WebRTC

Beginner: How do I implement WebRTC?

Intermediate: I want to implement WebRTC for [USE CASE: e.g., "video calls"]. Supporting [PARTICIPANTS] per session with [QUALITY REQUIREMENTS], I need features like [FEATURES: e.g., "screen sharing, recording"]. Include STUN/TURN setup.

Advanced: Act as a WebRTC expert. Design a real-time communication system with: Scale: [CONCURRENT SESSIONS] Participants: [PER SESSION] Quality: [RESOLUTION/BITRATE] Features: [RECORDING/STREAMING] Infrastructure: [GLOBAL/REGIONAL] Include:

- Signaling server design
- Media server selection
- STUN/TURN deployment
- Quality adaptation
- Recording architecture

- Cost optimization Format as technical blueprint.

Create Machine Learning Pipeline

Beginner: How do I create a machine learning pipeline?

Intermediate: I want to create an ML pipeline for [USE CASE: e.g., "fraud detection"]. Using [FRAMEWORK: e.g., "TensorFlow, PyTorch"], I need to process [DATA VOLUME] with [MODEL COMPLEXITY]. Include training, validation, and deployment stages.

Advanced: Act as an ML engineer. Design an ML platform supporting: Models: [TYPES/COUNT] Data: [VOLUME/VELOCITY] Training: [FREQUENCY/RESOURCES] Serving: [LATENCY/THROUGHPUT] Monitoring: [DRIFT/PERFORMANCE] Include:

- Feature store design
- Training orchestration
- Model registry
- A/B testing integration
- Monitoring pipeline
- Cost optimization Provide MLOps architecture.

Build Content Management System

Beginner: How do I build a content management system?

Intermediate: I want to build a CMS for [CONTENT TYPE: e.g., "blog, documentation"]. Supporting [USER ROLES] with [FEATURES: e.g., "versioning, workflow"], I need to manage [CONTENT VOLUME]. Include API design and plugin architecture.

Advanced: Act as a CMS architect. Design a content platform with: Content types: [VARIETY] Users: [EDITORS/VIEWERS] Workflow: [APPROVAL STAGES] Localization: [LANGUAGES] Integration: [SYSTEMS] Include:

- Headless architecture

- Content modeling
- Workflow engine
- Plugin system
- CDN integration
- Migration tools Format as platform design document.

Implement Geolocation Features

Beginner: How do I implement geolocation features?

Intermediate: I want to implement geolocation for [USE CASE: e.g., "store locator"]. Using [TECHNOLOGY: e.g., "Google Maps API"], I need features like [FEATURES: e.g., "proximity search, routing"]. Include mobile considerations and offline support.

Advanced: Act as a location services expert. Design a geolocation system with: Scale: [USERS/QUERIES] Accuracy: [REQUIREMENTS] Features: [TRACKING/GEOFENCING] Privacy: [COMPLIANCE] Offline: [CAPABILITIES] Include:

- Spatial database design
- Geocoding pipeline
- Privacy framework
- Offline strategy
- Battery optimization
- Cost management Provide implementation guide.

Create Event-Driven Architecture

Beginner: How do I create an event-driven architecture?

Intermediate: I want to implement event-driven architecture for [SYSTEM TYPE]. Using [MESSAGE BROKER], I need to handle [EVENT TYPES] with [THROUGHPUT]. Include event sourcing patterns and error handling strategies.

Advanced: Act as an event-driven architect. Design an event system with: Events: [TYPES/VOLUME] Producers: [SERVICES] Consumers: [COUNT/PATTERNS] Storage: [RETENTION] Guarantees: [DELIVERY] Include:

- Event schema registry
- CQRS implementation
- Saga orchestration
- Event replay capability
- Monitoring strategy
- Testing approach Format as architecture blueprint.

Build API Gateway

Beginner: How do I build an API gateway?

Intermediate: I want to build an API gateway for [MICROSERVICES COUNT] services. Using [TECHNOLOGY: e.g., "Kong, API Gateway"], I need features like [FEATURES: e.g., "routing, authentication, rate limiting"]. Include monitoring and versioning.

Advanced: Act as an API gateway expert. Design a gateway handling: Services: [COUNT] Traffic: [REQUESTS/SEC] Features: [SECURITY/TRANSFORMATION] Protocols: [REST/GRAPHQL/GRPC] Deployment: [EDGE/REGIONAL] Include:

- Traffic management
- Security policies
- Request transformation
- Service discovery
- Circuit breaking
- Cost analysis Provide implementation roadmap.

Implement Container Orchestration

Beginner: How do I implement container orchestration?

Intermediate: I want to orchestrate containers for my [APPLICATION TYPE]. Using [PLATFORM: e.g., "Kubernetes, Docker Swarm"], I need to manage [CONTAINER COUNT] with [SCALING REQUIREMENTS]. Include deployment strategies and monitoring.

Advanced: Act as a container platform architect. Design orchestration for:
Workloads: [TYPES/COUNT] Nodes: [CLUSTER SIZE] Storage: [REQUIREMENTS]
Networking: [COMPLEXITY] Compliance: [STANDARDS] Include:

- Cluster architecture
- Auto-scaling policies
- Service mesh integration
- Storage strategy
- Security hardening
- Cost optimization Format as platform design.

Create Serverless Functions

Beginner: How do I create serverless functions?

Intermediate: I want to create serverless functions for [USE CASE]. Using [PLATFORM: e.g., "AWS Lambda, Vercel"], I need to handle [REQUEST VOLUME] with [PERFORMANCE REQUIREMENTS]. Include cold start optimization and monitoring.

Advanced: Act as a serverless architect. Design a function platform with: Functions: [COUNT/PURPOSE] Events: [TRIGGERS] Performance: [COLD START/LATENCY]
Scale: [CONCURRENT EXECUTIONS] Cost: [BUDGET] Include:

- Function composition patterns
- State management
- Event routing
- Monitoring strategy
- Cost optimization

- Multi-region deployment Provide serverless architecture.

Build Code Review System

Beginner: How do I build a code review system?

Intermediate: I want to build code review for [TEAM SIZE] developers. Supporting [VCS: e.g., "Git"], I need features like [FEATURES: e.g., "comments, approvals, CI integration"]. Include workflow automation and metrics tracking.

Advanced: Act as a DevOps architect. Design a code review platform with: Repositories: [COUNT] Developers: [TEAM SIZE] Workflow: [BRANCHING STRATEGY] Integrations: [CI/CD/TOOLS] Compliance: [AUDIT NEEDS] Include:

- Review workflow engine
- Automated checks
- Metrics collection
- Integration framework
- Security scanning
- Knowledge sharing Format as platform specification.

Implement Data Warehouse

Beginner: How do I implement a data warehouse?

Intermediate: I want to implement a data warehouse for [BUSINESS DOMAIN]. Using [TECHNOLOGY: e.g., "Snowflake, BigQuery"], I need to store [DATA VOLUME] with [QUERY PATTERNS]. Include ETL design and performance optimization.

Advanced: Act as a data architect. Design a warehouse supporting: Data: [VOLUME/VELOCITY] Sources: [SYSTEMS] Users: [ANALYSTS/TOOLS] Queries: [COMPLEXITY] Compliance: [REGULATIONS] Include:

- Dimensional modeling
- ETL/ELT architecture
- Real-time integration

- Query optimization
- Cost management
- Governance framework Provide implementation guide.

Create Image Processing Service

Beginner: How do I create an image processing service?

Intermediate: I want to process images for [USE CASE: e.g., "thumbnails, filters"]. Using [LIBRARY: e.g., "Sharp, ImageMagick"], I need to handle [VOLUME] images with [OPERATIONS: e.g., "resize, crop, filter"]. Include caching and CDN integration.

Advanced: Act as an image processing expert. Design a service handling: Volume: [IMAGES/DAY] Operations: [TRANSFORMATIONS] Formats: [INPUT/OUTPUT] Performance: [LATENCY] Storage: [STRATEGY] Include:

- Processing pipeline
- Queue management
- CDN integration
- Format optimization
- Cost optimization
- ML integration Format as service architecture.

Build Feature Toggle System

Beginner: How do I build a feature toggle system?

Intermediate: I want to implement feature toggles for [APPLICATION TYPE]. Supporting [TOGGLE TYPES: e.g., "release, experiment, ops"], I need to manage [FEATURE COUNT] with [TARGETING: e.g., "user segments"]. Include SDK design and analytics.

Advanced: Act as a feature management expert. Design a toggle system with: Features: [COUNT] Users: [SCALE] Targeting: [RULES COMPLEXITY] Performance: [EVALUATION LATENCY] Compliance: [AUDIT NEEDS] Include:

- Toggle architecture
- Rule evaluation engine
- SDK design patterns
- Analytics integration
- Gradual rollout
- Emergency controls Provide platform design.

Implement Time Series Database

Beginner: How do I implement a time series database?

Intermediate: I want to store time series data for [USE CASE: e.g., "IoT sensors, metrics"]. Using [DATABASE: e.g., "InfluxDB, TimescaleDB"], I need to handle [DATA POINTS/SEC] with [RETENTION PERIOD]. Include aggregation and visualization.

Advanced: Act as a time series expert. Design a TSDB system for: Metrics: [COUNT] Ingestion: [POINTS/SEC] Retention: [POLICY] Queries: [PATTERNS] Availability: [SLA] Include:

- Schema design
- Compression strategy
- Aggregation rules
- Query optimization
- Clustering approach
- Cost optimization Format as database architecture.

Create Multi-Tenant Architecture

Beginner: How do I create a multi-tenant architecture?

Intermediate: I want to build multi-tenant support for [APPLICATION TYPE]. Supporting [TENANT COUNT] with [ISOLATION LEVEL: e.g., "database, schema, row"], I need features like [FEATURES: e.g., "customization, billing"]. Include security considerations.

Advanced: Act as a SaaS architect. Design multi-tenancy supporting: Tenants: [COUNT/GROWTH] Isolation: [LEVEL] Customization: [REQUIREMENTS] Scale: [PER-TENANT LIMITS] Compliance: [STANDARDS] Include:

- Tenant isolation strategy
- Resource allocation
- Customization framework
- Billing integration
- Migration approach
- Operations tooling Provide architecture guide.

Build URL Shortener

Beginner: How do I build a URL shortener?

Intermediate: I want to build a URL shortener handling [VOLUME] URLs. Using [TECH STACK], I need features like [FEATURES: e.g., "custom aliases, analytics, QR codes"]. Include scaling strategy and abuse prevention.

Advanced: Act as a distributed systems expert. Design a URL shortener with: Scale: [URLS/DAY] Traffic: [REDIRECTS/SEC] Features: [ANALYTICS/CUSTOM] Availability: [SLA] Security: [ABUSE PREVENTION] Include:

- ID generation strategy
- Caching architecture
- Analytics pipeline
- Abuse detection
- Geographic distribution
- Cost optimization Format as system design.

Implement Session Management

Beginner: How do I implement session management?

Intermediate: I want to implement sessions for [APPLICATION TYPE]. Using [STORAGE: e.g., "Redis, database"], I need to handle [CONCURRENT SESSIONS] with [FEATURES: e.g., "remember me, device tracking"]. Include security best practices.

Advanced: Act as a security architect. Design session management for: Users: [CONCURRENT] Devices: [PER USER] Storage: [BACKEND] Features: [SSO/FEDERATION] Compliance: [REQUIREMENTS] Include:

- Session architecture
- Token strategy
- Device fingerprinting
- Concurrent session handling
- Revocation mechanism
- Audit logging Provide security design.

Create Search Autocomplete

Beginner: How do I create search autocomplete?

Intermediate: I want to build autocomplete for [SEARCH TYPE: e.g., "product search"]. With [DATASET SIZE] items and [LATENCY REQUIREMENT], I need features like [FEATURES: e.g., "fuzzy matching, ranking"]. Include caching and personalization.

Advanced: Act as a search engineer. Design autocomplete supporting: Items: [CATALOG SIZE] Queries: [QPS] Latency: [TARGET MS] Features: [FUZZY/NLP/PERSONALIZATION] Languages: [SUPPORTED] Include:

- Data structure selection
- Ranking algorithm
- Caching strategy
- Personalization engine
- A/B testing setup

- Performance optimization Format as technical design.

Build Error Tracking System

Beginner: How do I build an error tracking system?

Intermediate: I want to track errors for [APPLICATION TYPE]. Using [TOOL: e.g., "Sentry, custom solution"], I need to capture [ERROR TYPES] with [FEATURES: e.g., "grouping, alerts, source maps"]. Include integration and dashboard design.

Advanced: Act as an observability expert. Design error tracking for: Applications: [COUNT/TYPES] Errors: [VOLUME/DAY] Storage: [RETENTION] Features: [GROUPING/TRENDS] Integration: [TOOLS] Include:

- Collection architecture
- Grouping algorithms
- Alert correlation
- Source map handling
- Privacy compliance
- Cost optimization Provide platform architecture.

Implement GraphQL Subscriptions

Beginner: How do I implement GraphQL subscriptions?

Intermediate: I want to add subscriptions to my GraphQL API for [USE CASE: e.g., "real-time updates"]. Using [TECHNOLOGY: e.g., "Apollo, GraphQL Yoga"], I need to handle [CONCURRENT SUBSCRIPTIONS] with [UPDATE FREQUENCY]. Include scaling considerations.

Advanced: Act as a GraphQL expert. Design subscription system with: Subscriptions: [TYPES] Clients: [CONCURRENT] Updates: [FREQUENCY] Transport: [WEBSOCKET/SSE] Scale: [HORIZONTAL] Include:

- PubSub architecture
- Connection management

- Authorization strategy
- Subscription lifecycle
- Performance optimization
- Monitoring approach Format as implementation guide.

Create Webhook System

Beginner: How do I create a webhook system?

Intermediate: I want to build webhooks for [API TYPE]. Supporting [EVENT TYPES] with [DELIVERY REQUIREMENTS], I need features like [FEATURES: e.g., "retries, signatures, filtering"]. Include reliability and monitoring strategies.

Advanced: Act as an API architect. Design webhook platform with: Events: [TYPES/VOLUME] Endpoints: [COUNT] Delivery: [SLA] Features: [REPLAY/FILTERING] Security: [SIGNING METHOD] Include:

- Event routing architecture
- Retry strategy
- Dead letter handling
- Signature verification
- Monitoring dashboard
- Developer experience Provide platform design.

Build Comment System

Beginner: How do I build a comment system?

Intermediate: I want to build commenting for [PLATFORM TYPE]. Supporting [FEATURES: e.g., "threading, reactions, mentions"], I need to handle [COMMENT VOLUME] with [MODERATION REQUIREMENTS]. Include real-time updates and spam detection.

Advanced: Act as a social platform architect. Design commenting system with: Scale: [COMMENTS/DAY] Features: [THREADING/REACTIONS/RICH TEXT]

Moderation: [AI/MANUAL/COMMUNITY] Performance: [LOAD TIME] Compliance: [REGULATIONS] Include:

- Data model design
- Real-time architecture
- Moderation pipeline
- Spam detection
- Analytics integration
- API design Format as system architecture.

Implement Service Mesh

Beginner: How do I implement a service mesh?

Intermediate: I want to implement service mesh for [MICROSERVICES COUNT] services. Using [TECHNOLOGY: e.g., "Istio, Linkerd"], I need features like [FEATURES: e.g., "traffic management, security"]. Include observability and deployment strategy.

Advanced: Act as a service mesh expert. Design mesh architecture for: Services: [COUNT/TYPES] Traffic: [PATTERNS] Security: [mTLS/POLICIES] Observability: [REQUIREMENTS] Performance: [OVERHEAD BUDGET] Include:

- Control plane design
- Traffic policies
- Security architecture
- Observability integration
- Progressive rollout
- Performance tuning Provide implementation roadmap.

Create Batch Processing System

Beginner: How do I create a batch processing system?

Intermediate: I want to process [DATA TYPE] in batches. Using [FRAMEWORK: e.g., "Apache Spark, Spring Batch"], I need to process [VOLUME] with [SCHEDULE: e.g., "hourly, daily"]. Include error handling and monitoring strategies.

Advanced: Act as a data engineer. Design batch processing for: Jobs: [TYPES/COUNT] Data: [VOLUME/FORMAT] Schedule: [COMPLEXITY] Dependencies: [JOB GRAPH] Resources: [COMPUTE/MEMORY] Include:

- Job orchestration
 - Resource allocation
 - Failure recovery
 - Data validation
 - Performance tuning
 - Cost optimization
- Format as processing architecture.

Build Analytics Platform

Beginner: How do I build an analytics platform?

Intermediate: I want to build analytics for [BUSINESS DOMAIN]. Tracking [METRICS TYPES] from [DATA SOURCES], I need features like [FEATURES: e.g., "dashboards, exports, alerts"]. Include data pipeline and visualization design.

Advanced: Act as an analytics architect. Design platform supporting: Data: [VOLUME/VELOCITY] Users: [ANALYSTS/CONSUMERS] Metrics: [COMPLEXITY] Latency: [REAL-TIME/BATCH] Compliance: [PRIVACY LAWS] Include:

- Data lake architecture
 - Processing pipeline
 - Semantic layer
 - Visualization strategy
 - Self-service analytics
 - Governance framework
- Provide platform blueprint.

Implement Voice Assistant

Beginner: How do I implement a voice assistant?

Intermediate: I want to build voice features for [APPLICATION TYPE]. Using [SERVICE: e.g., "Google Speech, Amazon Transcribe"], I need to support [COMMANDS] with [LANGUAGES]. Include NLU processing and response generation.

Advanced: Act as a voice AI expert. Design voice system with: Commands: [INTENTS/ENTITIES] Languages: [SUPPORTED] Accuracy: [TARGET] Latency: [REQUIREMENTS] Integration: [PLATFORMS] Include:

- Speech pipeline design
 - NLU architecture
 - Dialog management
 - Context handling
 - Personalization
 - Privacy considerations
- Format as AI system design.

Create Health Check System

Beginner: How do I create a health check system?

Intermediate: I want to implement health checks for [SERVICE COUNT] services. Monitoring [CHECK TYPES: e.g., "liveness, readiness, dependencies"], I need to aggregate status with [REPORTING REQUIREMENTS]. Include alerting and dashboard design.

Advanced: Act as a reliability engineer. Design health monitoring for: Services: [COUNT/TYPES] Checks: [DEPTH/FREQUENCY] Dependencies: [GRAPH COMPLEXITY] SLA: [REQUIREMENTS] Integration: [TOOLS] Include:

- Check framework design
- Dependency mapping
- Cascade failure prevention

- Status aggregation
- Alert correlation
- Runbook automation Provide monitoring architecture.

Build Recommendation Engine

Beginner: How do I build a recommendation engine?

Intermediate: I want to build recommendations for [DOMAIN: e.g., "content, products"]. Using [ALGORITHM: e.g., "collaborative filtering"], I need to serve [USER COUNT] with [LATENCY REQUIREMENTS]. Include A/B testing and evaluation metrics.

Advanced: Act as a ML architect. Design recommendation system with: Users: [SCALE] Items: [CATALOG SIZE] Algorithms: [ENSEMBLE] Features: [REAL-TIME/BATCH] Performance: [LATENCY/ACCURACY] Include:

- Feature engineering
- Model serving architecture
- Experimentation platform
- Bias detection
- Explanation generation
- Cold start handling Format as ML system design.

Implement Distributed Locking

Beginner: How do I implement distributed locking?

Intermediate: I want to implement distributed locks for [USE CASE]. Using [TECHNOLOGY: e.g., "Redis, Zookeeper"], I need to handle [LOCK SCENARIOS] with [RELIABILITY REQUIREMENTS]. Include deadlock prevention and monitoring.

Advanced: Act as a distributed systems expert. Design locking system for: Locks: [COUNT/TYPES] Holders: [CONCURRENT] Duration: [TIMEOUTS] Reliability: [GUARANTEES] Performance: [ACQUISITION TIME] Include:

- Lock algorithm selection

- Fencing token design
- Deadlock detection
- Fair queuing
- Monitoring strategy
- Testing approach Provide implementation guide.

Create Configuration Management

Beginner: How do I create configuration management?

Intermediate: I want to manage configuration for [APPLICATION COUNT] apps. Using [TOOL: e.g., "Consul, etcd"], I need features like [FEATURES: e.g., "hot reload, versioning, encryption"]. Include deployment and rollback strategies.

Advanced: Act as a platform architect. Design configuration system for: Applications: [COUNT/TYPES] Configs: [COMPLEXITY] Changes: [FREQUENCY] Security: [ENCRYPTION/ACCESS] Compliance: [AUDIT NEEDS] Include:

- Storage architecture
- Change propagation
- Version control
- Secret management
- Audit logging
- Disaster recovery Format as platform design.

Build Form Validation System

Beginner: How do I build form validation?

Intermediate: I want to build validation for [FORM TYPES: e.g., "registration, checkout"]. Using [FRAMEWORK: e.g., "React Hook Form"], I need to validate [FIELD TYPES] with [REQUIREMENTS: e.g., "real-time, custom rules"]. Include error handling and UX considerations.

Advanced: Act as a frontend architect. Design validation system with: Forms: [COMPLEXITY] Rules: [CUSTOM/STANDARD] Performance: [REAL-TIME NEEDS] Localization: [LANGUAGES] Accessibility: [WCAG LEVEL] Include:

- Validation architecture
- Rule engine design
- Error state management
- Custom validator framework
- Performance optimization
- Testing strategy Provide implementation guide.

Implement CDN Strategy

Beginner: How do I implement a CDN strategy?

Intermediate: I want to use CDN for [CONTENT TYPES: e.g., "static assets, API responses"]. With [GEOGRAPHIC DISTRIBUTION] and [TRAFFIC VOLUME], I need features like [FEATURES: e.g., "purging, headers"]. Include cost optimization and monitoring.

Advanced: Act as a CDN architect. Design CDN strategy for: Content: [TYPES/SIZE] Regions: [COVERAGE] Traffic: [BANDWIDTH] Features: [EDGE COMPUTE/WAF] Budget: [CONSTRAINTS] Include:

- Multi-CDN strategy
- Cache key design
- Purge automation
- Edge computing
- Security integration
- Cost optimization Format as CDN architecture.

Create Audit Logging System

Beginner: How do I create audit logging?

Intermediate: I want to implement audit logging for [APPLICATION TYPE]. Tracking [EVENT TYPES: e.g., "user actions, data changes"], I need to store [LOG VOLUME] with [RETENTION PERIOD]. Include search capabilities and compliance features.

Advanced: Act as a compliance architect. Design audit system for: Events: [TYPES/VOLUME] Retention: [LEGAL REQUIREMENTS] Search: [QUERY NEEDS] Integrity: [TAMPER PROOF] Compliance: [STANDARDS] Include:

- Event schema design
- Storage architecture
- Immutability guarantees
- Search optimization
- Export capabilities
- Access controls Provide compliance guide.

Build Shopping Cart System

Beginner: How do I build a shopping cart system?

Intermediate: I want to build a shopping cart for [E-COMMERCE TYPE]. Supporting [FEATURES: e.g., "guest checkout, saved items"], I need to handle [USER SCALE] with [PERSISTENCE REQUIREMENTS]. Include pricing calculations and inventory checks.

Advanced: Act as an e-commerce architect. Design cart system supporting: Users: [SCALE] Items: [CATALOG SIZE] Features: [PROMOTIONS/BUNDLES] Persistence: [STRATEGY] Performance: [LATENCY] Include:

- Cart state management
- Pricing engine
- Inventory integration
- Abandonment recovery
- Analytics tracking

- Scaling strategy Format as system design.

Implement Data Validation

Beginner: How do I implement data validation?

Intermediate: I want to validate [DATA TYPES: e.g., "user input, API requests"]. Using [LANGUAGE/Framework], I need to implement [VALIDATION RULES] with [ERROR HANDLING STRATEGY]. Include schema validation and custom rules.

Advanced: Act as a data quality expert. Design validation framework for: Data: [TYPES/SOURCES] Rules: [COMPLEXITY] Performance: [THROUGHPUT] Errors: [HANDLING STRATEGY] Evolution: [SCHEMA CHANGES] Include:

- Validation architecture
- Rule engine design
- Error aggregation
- Schema evolution
- Performance optimization
- Testing framework Provide implementation guide.

Create Workflow Engine

Beginner: How do I create a workflow engine?

Intermediate: I want to build workflow automation for [USE CASE: e.g., "approval processes"]. Supporting [WORKFLOW TYPES] with [STEP COUNT], I need features like [FEATURES: e.g., "conditionals, parallel execution"]. Include state management and monitoring.

Advanced: Act as a workflow architect. Design workflow engine supporting: Workflows: [TYPES/COMPLEXITY] Execution: [VOLUME/DAY] Features: [BPMN/CUSTOM] Persistence: [REQUIREMENTS] Integration: [SYSTEMS] Include:

- Execution engine design
- State persistence
- Compensation logic

- Visual designer
- Monitoring dashboard
- Testing framework Format as engine architecture.

Build RSS Feed Reader

Beginner: How do I build an RSS feed reader?

Intermediate: I want to build an RSS reader for [PLATFORM: e.g., "web, mobile"]. Supporting [FEED COUNT] feeds with [UPDATE FREQUENCY], I need features like [FEATURES: e.g., "categories, read status"]. Include feed parsing and storage design.

Advanced: Act as a content aggregation expert. Design RSS platform with: Feeds: [COUNT] Users: [SCALE] Updates: [FREQUENCY] Features: [RECOMMENDATIONS/SEARCH] Performance: [LATENCY] Include:

- Feed crawler architecture
- Content deduplication
- Personalization engine
- Full-text search
- Mobile sync strategy
- Scalability approach Provide platform design.

Implement Zero-Downtime Deployment

Beginner: How do I implement zero-downtime deployment?

Intermediate: I want zero-downtime deployments for [APPLICATION TYPE]. Using [DEPLOYMENT METHOD: e.g., "blue-green, canary"], I need to handle [TRAFFIC VOLUME] with [ROLLBACK TIME]. Include health checks and traffic management.

Advanced: Act as a deployment architect. Design zero-downtime strategy for: Services: [COUNT/TYPES] Traffic: [VOLUME] Dependencies: [COMPLEXITY] Rollback: [RTO] Validation: [AUTOMATED TESTS] Include:

- Deployment patterns

- Database migrations
- Traffic shifting
- Feature flags integration
- Automated validation
- Rollback automation Format as deployment guide.

Create Password Manager

Beginner: How do I create a password manager?

Intermediate: I want to build a password manager with [FEATURES: e.g., "encryption, auto-fill"]. Using [ENCRYPTION METHOD], I need to support [PLATFORMS] with [SYNC REQUIREMENTS]. Include security architecture and browser integration.

Advanced: Act as a security architect. Design password manager with: Encryption: [ALGORITHM/KEY DERIVATION] Platforms: [SUPPORTED] Features: [2FA/SHARING/AUDIT] Sync: [STRATEGY] Compliance: [STANDARDS] Include:

- Cryptographic design
- Zero-knowledge architecture
- Secure sharing
- Browser integration
- Mobile security
- Audit capabilities Format as security design.

Build Link Preview System

Beginner: How do I build a link preview system?

Intermediate: I want to generate link previews for [CONTENT TYPES]. Processing [URL VOLUME] with [CACHE STRATEGY], I need to extract [METADATA: e.g., "title, description, image"]. Include error handling and performance optimization.

Advanced: Act as a content processing expert. Design preview system for: URLs: [VOLUME/DAY] Sources: [DIVERSITY] Cache: [STRATEGY/TTL] Features: [RICH PREVIEWS/OG TAGS] Performance: [GENERATION TIME] Include:

- Scraping architecture
- Metadata extraction
- Image processing
- Cache invalidation
- Abuse prevention
- CDN integration Format as system architecture.

Implement Fuzzy Search

Beginner: How do I implement fuzzy search?

Intermediate: I want to implement fuzzy search for [DATA TYPE: e.g., "user names, addresses"]. Using [ALGORITHM: e.g., "Levenshtein distance"], I need to search [DATASET SIZE] with [PERFORMANCE REQUIREMENTS]. Include ranking and optimization strategies.

Advanced: Act as a search expert. Design fuzzy search system with: Data: [SIZE/TYPE] Algorithms: [COMBINATION] Performance: [LATENCY/ACCURACY] Languages: [SUPPORTED] Features: [PHONETIC/TYPOS] Include:

- Algorithm selection
- Index optimization
- Ranking strategy
- Language handling
- Performance tuning
- A/B testing setup Provide implementation guide.

PART 2

100 Coding Interview Prompts - Master Solution Patterns

Essential patterns and techniques to build your coding interview foundation!

How to Use These Prompts

Each prompt is designed to help you:

1. **Understand the Pattern:** Recognize when to apply specific techniques
2. **Master the Implementation:** Practice the standard template for each pattern
3. **Explain the Logic:** Articulate why the approach works
4. **Handle Edge Cases:** Consider boundary conditions and special cases
5. **Analyze Complexity:** Understand time and space trade-offs

Practice Strategy:

- Start with prompts 1-10 (Two Pointers) to build foundation
- Progress through each pattern systematically
- Practice explaining your solution out loud
- Time yourself to simulate interview conditions
- Focus on clean, readable code with proper variable names

Remember: The goal is not just to solve the problem, but to demonstrate clear thinking, systematic approach, and strong communication skills during your interview!

Two Pointers Pattern (1-10)

1. Two Sum in Sorted Array

Prompt: Given a sorted array of integers and a target sum, implement a function that finds two numbers that add up to the target. Use the two-pointer technique where you place one pointer at the start and one at the end, moving them based on whether the current sum is greater or less than the target. Practice explaining why this works only on sorted arrays and how it achieves $O(n)$ time complexity.

2. Palindrome Validation

Prompt: Write a function to check if a string is a palindrome using two pointers. Place one pointer at the beginning and one at the end, comparing characters while moving inward. Handle edge cases like empty strings and single characters. Practice explaining how this approach is more efficient than reversing the string.

3. Remove Duplicates from Sorted Array

Prompt: Given a sorted array, remove duplicates in-place using two pointers. Use a slow pointer to track the position of unique elements and a fast pointer to scan through the array. Practice explaining why this technique works and how it maintains the relative order of elements.

4. Three Sum Problem

Prompt: Find all unique triplets in an array that sum to zero. First sort the array, then for each element, use two pointers to find pairs that complete the triplet. Practice handling duplicate triplets and explaining the $O(n^2)$ time complexity.

5. Container With Most Water

Prompt: Given an array representing heights of vertical lines, find two lines that form a container holding the most water. Use two pointers starting from both ends, always moving the pointer with the smaller height inward. Practice explaining why this greedy approach works.

6. Valid Palindrome with Alphanumeric Only

Prompt: Check if a string is a palindrome considering only alphanumeric characters and ignoring case. Use two pointers, skipping non-alphanumeric characters. Practice handling character conversion and explaining the filtering logic.

7. Move Zeros to End

Prompt: Move all zeros in an array to the end while maintaining the relative order of non-zero elements. Use two pointers: one to track the position for non-zero elements and one to scan the array. Practice explaining why this is better than using extra space.

8. Intersection of Two Sorted Arrays

Prompt: Find the intersection of two sorted arrays using two pointers. Compare elements at current positions and advance the appropriate pointer. Practice handling duplicates and explaining the $O(m + n)$ time complexity.

9. Merge Two Sorted Arrays

Prompt: Merge two sorted arrays in-place where the first array has enough space. Use two pointers starting from the end of both arrays, filling the first array from right to left. Practice explaining why filling from right to left prevents overwriting.

10. Reverse String In-Place

Prompt: Reverse a string represented as a character array using two pointers. Swap characters at the start and end pointers, moving them toward the center. Practice explaining the $O(n/2) = O(n)$ time complexity and $O(1)$ space complexity.

Hash Map/Set Patterns (11-20)

11. Two Sum with Hash Map

Prompt: Solve the two sum problem using a hash map to achieve $O(n)$ time complexity. For each number, check if its complement exists in the hash map. Practice explaining why this is more efficient than the brute force $O(n^2)$ approach.

12. First Non-Repeating Character

Prompt: Find the first character in a string that doesn't repeat. Use a hash map to count character frequencies, then scan the string again to find the first character with count 1. Practice explaining the two-pass approach and why it's necessary.

13. Group Anagrams

Prompt: Group strings that are anagrams of each other. Use a hash map where the key is the sorted string and the value is a list of anagrams. Practice explaining why sorting works as a key and the time complexity implications.

14. Contains Duplicate

Prompt: Check if an array contains any duplicate values using a hash set. Add elements to the set and return true if an element already exists. Practice explaining why this is $O(n)$ time and $O(n)$ space.

15. Intersection of Two Arrays

Prompt: Find the intersection of two arrays using hash sets. Convert one array to a set, then iterate through the second array checking membership. Practice handling duplicates in the result and explaining set operations.

16. Valid Anagram

Prompt: Check if two strings are anagrams using character frequency counting. Use a hash map to count characters in the first string, then decrement counts for the second string. Practice explaining why this works and handling edge cases.

17. Majority Element

Prompt: Find the element that appears more than $n/2$ times in an array. Use a hash map to count frequencies, then find the element with count $> n/2$. Practice explaining why the majority element is guaranteed to exist in this problem.

18. Happy Number

Prompt: Determine if a number is "happy" (repeatedly replacing it with the sum of squares of its digits eventually leads to 1). Use a hash set to detect cycles. Practice explaining cycle detection and the mathematical insight behind the solution.

19. Isomorphic Strings

Prompt: Check if two strings are isomorphic (characters can be mapped one-to-one). Use two hash maps to ensure bijective mapping. Practice explaining why you need mapping in both directions and handling edge cases.

20. Word Pattern

Prompt: Check if a string follows a given pattern where each character maps to exactly one word. Use hash maps to maintain bidirectional mapping. Practice explaining the importance of checking both directions and handling space-separated words.

Sliding Window Pattern (21-30)

21. Maximum Sum Subarray of Size K

Prompt: Find the maximum sum of any contiguous subarray of size k . Use sliding window: calculate sum of first k elements, then slide the window by removing the leftmost element and adding the new rightmost element. Practice explaining why this is $O(n)$ instead of $O(n*k)$.

22. Longest Substring Without Repeating Characters

Prompt: Find the length of the longest substring without repeating characters. Use a sliding window with a hash set to track characters in the current window. Expand the window by adding characters and contract when duplicates are found. Practice explaining the two-pointer movement.

23. Minimum Window Substring

Prompt: Find the minimum window in string s that contains all characters of string t . Use sliding window with frequency counting. Expand the window until all characters are covered, then contract from the left while maintaining coverage. Practice explaining the complex window validity check.

24. Permutation in String

Prompt: Check if any permutation of string s_1 exists as a substring in s_2 . Use sliding window of size $\text{len}(s_1)$ and compare character frequencies. Practice explaining why frequency comparison works for permutation detection.

25. Find All Anagrams in String

Prompt: Find all start indices where anagrams of pattern occur in text. Use sliding window with frequency counting, comparing window frequency with pattern frequency. Practice explaining how to efficiently update frequencies when sliding the window.

26. Longest Repeating Character Replacement

Prompt: Find the length of the longest substring containing the same letter after changing at most k characters. Use sliding window with frequency counting of the most frequent character. Practice explaining the key insight: $\text{window_size} - \text{max_frequency} \leq k$.

27. Max Consecutive Ones III

Prompt: Find the maximum number of consecutive 1s after flipping at most k zeros. Use sliding window, treating zeros as "flips used". Expand the window and contract when flips exceed k . Practice explaining how this is similar to the character replacement problem.

28. Substring with Concatenation of All Words

Prompt: Find all starting indices where a concatenation of all words appears in the string. Use sliding window with word frequency counting. Practice explaining the complexity of handling fixed-length word concatenations.

29. Minimum Size Subarray Sum

Prompt: Find the minimum length of a contiguous subarray whose sum is greater than or equal to target. Use sliding window: expand by adding elements until $\text{sum} \geq \text{target}$, then contract from left while maintaining the condition. Practice explaining the greedy contraction strategy.

30. Fruits Into Baskets

Prompt: Find the maximum number of fruits you can collect with at most 2 types of fruits. Use sliding window with frequency counting, maintaining at most 2 distinct elements. Practice explaining how this generalizes to "at most k distinct elements" problems.

Binary Search Pattern (31-40)

31. Classic Binary Search

Prompt: Implement binary search to find a target value in a sorted array. Use the template: while (left <= right), calculate $\text{mid} = \text{left} + (\text{right} - \text{left}) / 2$ to avoid overflow. Practice explaining why this is $O(\log n)$ and handling the cases where target is not found.

32. First and Last Position of Element

Prompt: Find the first and last position of a target value in a sorted array. Implement two binary searches: one to find the leftmost occurrence and one for the rightmost. Practice explaining how to modify the standard binary search for boundary finding.

33. Search Insert Position

Prompt: Find the index where a target should be inserted in a sorted array to maintain sorted order. Use binary search, and when the loop ends, left will be the insertion position. Practice explaining why left gives the correct insertion point.

34. Search in Rotated Sorted Array

Prompt: Search for a target in a rotated sorted array. Use binary search by determining which half is sorted, then checking if target is in that half. Practice explaining how to identify the sorted half and the logic for target range checking.

35. Find Peak Element

Prompt: Find a peak element in an array where $\text{nums}[i] > \text{nums}[i-1]$ and $\text{nums}[i] > \text{nums}[i+1]$. Use binary search by comparing mid with mid+1 to determine which side has the peak. Practice explaining why this works and the guarantee of finding a peak.

36. Square Root Using Binary Search

Prompt: Implement integer square root using binary search. Search in the range $[0, x]$ for the largest integer whose square is $\leq x$. Practice explaining the search space and how to handle the precision for integer results.

37. Find Minimum in Rotated Sorted Array

Prompt: Find the minimum element in a rotated sorted array. Use binary search by comparing mid with right to determine which half contains the minimum. Practice explaining how rotation affects the binary search logic.

38. Search a 2D Matrix

Prompt: Search for a target in a 2D matrix where each row is sorted and the first element of each row is greater than the last element of the previous row. Use binary search by treating the matrix as a flattened sorted array. Practice explaining the index conversion between 1D and 2D.

39. Find Bad Version

Prompt: Find the first bad version using a binary search API. All versions after a bad version are also bad. Use binary search to find the first occurrence of true from the API. Practice explaining the boundary condition for first occurrence.

40. Capacity to Ship Packages Within D Days

Prompt: Find the minimum ship capacity to ship all packages within d days. Use binary search on the capacity value between $\max(\text{weights})$ and $\sum(\text{weights})$. Practice explaining how to verify if a capacity works and why this search space is valid.

Depth-First Search (DFS) Pattern (41-50)

41. Binary Tree Inorder Traversal

Prompt: Implement inorder traversal of a binary tree using recursion. Follow the pattern: process left subtree, visit root, process right subtree. Practice explaining the call stack behavior and how recursion handles the traversal state.

42. Binary Tree Maximum Depth

Prompt: Find the maximum depth of a binary tree using DFS. Use recursion where $\text{depth} = 1 + \max(\text{left_depth}, \text{right_depth})$. Practice explaining the base case (null node returns 0) and how the recursive calls build up the depth.

43. Path Sum

Prompt: Check if there's a root-to-leaf path with a given sum using DFS. Recursively check left and right subtrees with updated target ($\text{target} - \text{current_value}$). Practice explaining the leaf node check and backtracking behavior.

44. Symmetric Tree

Prompt: Check if a binary tree is symmetric using DFS. Create a helper function that compares left and right subtrees recursively. Practice explaining the symmetry conditions and how to structure the recursive comparison.

45. Number of Islands

Prompt: Count the number of islands in a 2D grid using DFS. For each unvisited land cell, start a DFS to mark all connected land cells as visited. Practice explaining the 4-directional movement and how DFS explores connected components.

46. Clone Graph

Prompt: Clone an undirected graph using DFS. Use a hash map to store mapping from original to cloned nodes. Practice explaining how DFS handles the recursive cloning and cycle detection through the visited map.

47. Validate Binary Search Tree

Prompt: Check if a binary tree is a valid BST using DFS. Pass down min and max bounds for each node and validate the BST property. Practice explaining why inorder traversal alone isn't sufficient and how bounds propagate.

48. Lowest Common Ancestor

Prompt: Find the lowest common ancestor of two nodes in a binary tree using DFS. Use recursion to search for both nodes and return the LCA when found. Practice explaining the three cases: both in left, both in right, or one in each subtree.

49. Binary Tree Right Side View

Prompt: Find the right side view of a binary tree using DFS. Use level-order traversal or DFS with level tracking to get the rightmost node at each level. Practice explaining how to track levels in DFS and ensure you get the rightmost node.

50. Flatten Binary Tree to Linked List

Prompt: Flatten a binary tree to a linked list in-place using DFS. Use preorder traversal and reconstruct the tree structure. Practice explaining the in-place modification and how to handle the right pointer connections.

Breadth-First Search (BFS) Pattern (51-60)

51. Binary Tree Level Order Traversal

Prompt: Implement level-order traversal of a binary tree using BFS. Use a queue to process nodes level by level. Practice explaining how the queue maintains the level order and how to separate levels if needed.

52. Minimum Depth of Binary Tree

Prompt: Find the minimum depth of a binary tree using BFS. Process nodes level by level and return the depth when the first leaf is found. Practice explaining why BFS is more efficient than DFS for this problem.

53. Rotting Oranges

Prompt: Find the minimum time for all oranges to rot using BFS. Start BFS from all initially rotten oranges simultaneously (multi-source BFS). Practice explaining the time simulation and how to handle the grid boundaries.

54. Word Ladder

Prompt: Find the shortest transformation sequence from beginWord to endWord using BFS. Treat each word as a node and transformations as edges. Practice explaining the graph construction and why BFS guarantees the shortest path.

55. Open the Lock

Prompt: Find the minimum number of turns to open a lock using BFS. Each lock state is a node, and turning a digit creates an edge. Practice explaining the state space representation and how to generate neighboring states.

56. Perfect Squares

Prompt: Find the minimum number of perfect squares that sum to n using BFS. Treat each number as a node and adding a perfect square as an edge. Practice explaining why BFS finds the minimum number of squares.

57. Shortest Path in Binary Matrix

Prompt: Find the shortest path in a binary matrix using BFS. Use 8-directional movement and BFS to find the shortest path. Practice explaining the path reconstruction and handling the 8-directional movement.

58. Walls and Gates

Prompt: Fill empty rooms with distance to nearest gate using BFS. Start multi-source BFS from all gates simultaneously. Practice explaining how multi-source BFS works and why it gives the correct distances.

59. Course Schedule

Prompt: Determine if all courses can be finished using BFS (topological sort). Build a graph of prerequisites and use BFS to detect cycles. Practice explaining the indegree concept and how BFS processes nodes with indegree 0.

60. Snakes and Ladders

Prompt: Find the minimum number of moves to reach the end of a snakes and ladders board using BFS. Convert board position to coordinates and handle snakes/ladders. Practice explaining the coordinate conversion and state representation.

Dynamic Programming Fundamentals (61-70)

61. Fibonacci Sequence

Prompt: Calculate the n th Fibonacci number using dynamic programming. Start with the recursive solution, then optimize using memoization (top-down) or tabulation (bottom-up). Practice explaining the overlapping subproblems and optimal substructure.

62. Climbing Stairs

Prompt: Count the number of ways to climb n stairs (1 or 2 steps at a time) using DP. Recognize that $ways[i] = ways[i-1] + ways[i-2]$. Practice explaining why this is similar to Fibonacci and the recurrence relation.

63. House Robber

Prompt: Find the maximum money you can rob without robbing adjacent houses using DP. For each house, decide whether to rob it or not: $dp[i] = \max(dp[i-1], dp[i-2] + nums[i])$. Practice explaining the decision-making process and state definition.

64. Maximum Subarray (Kadane's Algorithm)

Prompt: Find the maximum sum of a contiguous subarray using DP. At each position, decide whether to extend the current subarray or start a new one. Practice explaining why this greedy approach works and the $O(n)$ time complexity.

65. Coin Change

Prompt: Find the minimum number of coins needed to make a target amount using DP. For each amount, try all coin denominations and take the minimum. Practice explaining the state definition and why this gives the optimal solution.

66. Longest Increasing Subsequence

Prompt: Find the length of the longest increasing subsequence using DP. For each element, find the longest subsequence ending at that element. Practice explaining the $O(n^2)$ solution and the state transition.

67. Unique Paths

Prompt: Count unique paths from top-left to bottom-right in a grid using DP. You can only move right or down. Practice explaining the recurrence relation: $\text{paths}[i][j] = \text{paths}[i-1][j] + \text{paths}[i][j-1]$.

68. Edit Distance

Prompt: Find the minimum edit distance between two strings using DP. Consider three operations: insert, delete, replace. Practice explaining the 2D DP table and how each cell represents the minimum operations needed.

69. Decode Ways

Prompt: Count the number of ways to decode a numeric string using DP. Consider single digit (1-9) and two-digit (10-26) decodings. Practice explaining the state definition and handling edge cases like leading zeros.

70. Word Break

Prompt: Determine if a string can be segmented into dictionary words using DP. For each position, check if any word ending at that position can be formed. Practice explaining the state definition and the checking process.

Greedy Algorithm Patterns (71-80)

71. Best Time to Buy and Sell Stock

Prompt: Find the maximum profit from buying and selling a stock once using a greedy approach. Keep track of the minimum price seen so far and calculate profit at each step. Practice explaining why this greedy approach works and gives the optimal solution.

72. Jump Game

Prompt: Determine if you can reach the last index using a greedy approach. Keep track of the farthest position reachable and check if the current position is within reach. Practice explaining why greedy works and the invariant maintained.

73. Meeting Rooms II

Prompt: Find the minimum number of meeting rooms needed using a greedy approach. Sort meetings by start time and use a min-heap to track end times. Practice explaining the greedy choice and why sorting by start time is optimal.

74. Non-overlapping Intervals

Prompt: Find the minimum number of intervals to remove to make them non-overlapping using a greedy approach. Sort by end time and greedily select intervals. Practice explaining why sorting by end time leads to the optimal solution.

75. Partition Labels

Prompt: Partition a string into as many parts as possible using a greedy approach. Find the last occurrence of each character and extend the current partition accordingly. Practice explaining the greedy strategy and how it ensures valid partitions.

76. Task Scheduler

Prompt: Find the minimum time to execute all tasks with cooling period using a greedy approach. Count task frequencies and schedule the most frequent tasks first. Practice explaining the mathematical insight and the greedy scheduling strategy.

77. Gas Station

Prompt: Find the starting gas station to complete a circular tour using a greedy approach. Keep track of current tank and total surplus. Practice explaining why there's a unique solution and how the greedy approach finds it.

78. Candy

Prompt: Distribute minimum candies to children based on ratings using a greedy approach. Make two passes: left to right for increasing ratings, right to left for decreasing ratings. Practice explaining why two passes are necessary.

79. Queue Reconstruction by Height

Prompt: Reconstruct a queue based on height and number of people in front using a greedy approach. Sort by height (descending) and k value (ascending), then insert by k position. Practice explaining the greedy insertion strategy.

80. Minimum Number of Arrows

Prompt: Find the minimum number of arrows to burst all balloons using a greedy approach. Sort intervals by end position and greedily place arrows. Practice explaining why this greedy choice is optimal and the interval scheduling pattern.

Backtracking Pattern (81-90)

81. Generate Parentheses

Prompt: Generate all valid parentheses combinations using backtracking. Keep track of open and close parentheses counts, making choices based on validity rules. Practice explaining the pruning conditions and the backtracking template.

82. Permutations

Prompt: Generate all permutations of a list using backtracking. Use a visited array to track used elements and backtrack after each recursive call. Practice explaining the choice-making process and the backtracking mechanism.

83. Subsets

Prompt: Generate all subsets of a list using backtracking. For each element, make the choice to include it or not. Practice explaining the binary choice nature and how backtracking explores all possibilities.

84. Combination Sum

Prompt: Find all combinations that sum to a target using backtracking. Allow reuse of numbers and backtrack when the sum exceeds the target. Practice explaining the pruning strategy and how to avoid infinite recursion.

85. Word Search

Prompt: Find if a word exists in a 2D grid using backtracking. Explore all 4 directions from each cell and backtrack when the path doesn't lead to the word. Practice explaining the visited state management and backtracking.

86. N-Queens

Prompt: Place N queens on an $N \times N$ chessboard using backtracking. Check for conflicts in rows, columns, and diagonals before placing each queen. Practice explaining the constraint checking and the backtracking search space.

87. Sudoku Solver

Prompt: Solve a Sudoku puzzle using backtracking. For each empty cell, try digits 1-9 and check validity. Practice explaining the constraint validation and the systematic search approach.

88. Letter Combinations of Phone Number

Prompt: Generate all possible letter combinations from a phone number using backtracking. Build combinations character by character and backtrack to try different choices. Practice explaining the mapping and the recursive structure.

89. Palindrome Partitioning

Prompt: Partition a string into palindromes using backtracking. For each position, try all possible palindromic substrings and backtrack. Practice explaining the palindrome checking and the partitioning strategy.

90. Restore IP Addresses

Prompt: Generate all valid IP addresses from a string using backtracking. Try all possible segment lengths and validate each segment. Practice explaining the IP validation rules and the backtracking search.

Tree Traversal Mastery (91-100)

91. Binary Tree Preorder Traversal (Iterative)

Prompt: Implement iterative preorder traversal using a stack. Push right child first, then left child to maintain correct order. Practice explaining how the stack simulates the recursive call stack and the order of operations.

92. Binary Tree Inorder Traversal (Iterative)

Prompt: Implement iterative inorder traversal using a stack. Go left as far as possible, then process current node and move right. Practice explaining the left-root-right pattern and how the stack maintains the traversal state.

93. Binary Tree Postorder Traversal (Iterative)

Prompt: Implement iterative postorder traversal using a stack. Use two stacks or modify preorder to get reverse postorder. Practice explaining the left-right-root pattern and the complexity of iterative postorder.

94. Binary Tree Level Order Traversal (with Levels)

Prompt: Implement level-order traversal that groups nodes by level using BFS. Track the number of nodes at each level and process them together. Practice explaining how to separate levels and the queue size tracking.

95. Binary Tree Zigzag Level Order Traversal

Prompt: Implement zigzag level-order traversal using BFS with direction alternation. Use a deque or reverse alternate levels. Practice explaining the direction tracking and how to achieve the zigzag pattern.

96. Vertical Order Traversal

Prompt: Implement vertical order traversal by assigning column indices to nodes. Use a hash map to group nodes by column and sort by row when needed. Practice explaining the coordinate system and the sorting requirements.

97. Binary Tree Boundary Traversal

Prompt: Implement boundary traversal (left boundary, leaves, right boundary) using multiple traversals. Combine different traversal strategies for different parts. Practice explaining the boundary definition and the traversal combination.

98. Serialize and Deserialize Binary Tree

Prompt: Implement serialization and deserialization using preorder traversal. Use a delimiter for null nodes and reconstruct the tree using the same order. Practice explaining the encoding format and the reconstruction process.

99. Lowest Common Ancestor in Binary Tree

Prompt: Find the LCA of two nodes using DFS. Return the node where both targets are found in different subtrees. Practice explaining the recursive search and the three cases for LCA determination.

100. Binary Tree Maximum Path Sum

Prompt: Find the maximum path sum in a binary tree using DFS. For each node, calculate the maximum path through that node and update the global maximum. Practice explaining the path definition and how to handle negative values.

PART 3

270 Rapid Prompts - Ultimate Programming Challenge List

Beginner Level

Basic Data Structures & Algorithms

1. Write a function to reverse a string
2. Create a program that checks if a number is prime

3. Implement a function to find the factorial of a number
4. Write a program to check if a string is a palindrome
5. Create a function to find the largest number in an array
6. Implement a basic calculator with +, -, *, / operations
7. Write a program to count vowels in a string
8. Create a function to remove duplicates from an array
9. Implement a program to find the second largest number in an array
10. Write a function to check if two strings are anagrams

Simple Games & Interactive Programs

11. Create a "Guess the Number" game
12. Build a simple Rock, Paper, Scissors game
13. Implement a Tic-Tac-Toe game
14. Create a word guessing game (Hangman)
15. Build a simple dice rolling simulator
16. Create a coin flip simulator
17. Implement a simple quiz application
18. Build a password generator
19. Create a random quote generator
20. Implement a simple to-do list application

String Manipulation

21. Write a function to count words in a sentence
22. Create a program to capitalize the first letter of each word
23. Implement a function to find and replace text in a string
24. Write a program to check if a string contains only digits

25. Create a function to extract numbers from a string
26. Implement a program to count character frequency in a string
27. Write a function to remove whitespace from a string
28. Create a program to check if parentheses are balanced
29. Implement a function to find the longest word in a sentence
30. Write a program to reverse words in a sentence

Mathematical Operations

31. Create a program to find the GCD of two numbers
32. Implement a function to calculate the LCM of two numbers
33. Write a program to check if a number is perfect
34. Create a function to generate Fibonacci sequence
35. Implement a program to find sum of digits in a number
36. Write a function to check if a number is Armstrong number
37. Create a program to convert decimal to binary
38. Implement a function to calculate compound interest
39. Write a program to find area of different geometric shapes
40. Create a temperature converter (Celsius/Fahrenheit/Kelvin)

Array Operations

41. Write a function to find the sum of all elements in an array
42. Create a program to find the average of numbers in an array
43. Implement a function to merge two sorted arrays
44. Write a program to rotate an array left or right
45. Create a function to find intersection of two arrays
46. Implement a program to find missing number in sequence

47. Write a function to sort an array using bubble sort
48. Create a program to find the median of an array
49. Implement a function to check if array is sorted
50. Write a program to find pairs that sum to a target value

File Operations

51. Create a program to read and display file contents
52. Write a function to count lines in a text file
53. Implement a program to copy contents from one file to another
54. Create a function to find and replace text in a file
55. Write a program to count word frequency in a text file
56. Implement a function to merge multiple text files
57. Create a program to split a large file into smaller chunks
58. Write a function to find the largest line in a file
59. Implement a program to reverse the contents of a file
60. Create a function to remove blank lines from a file

Data Validation

61. Write a function to validate email addresses
62. Create a program to check if a password is strong
63. Implement a function to validate phone numbers
64. Write a program to check if a date is valid
65. Create a function to validate credit card numbers
66. Implement a program to check if a URL is valid
67. Write a function to validate postal codes
68. Create a program to check if input is a valid number

- 69. Implement a function to validate IP addresses
- 70. Write a program to check if a string is a valid identifier

Time & Date Operations

- 71. Create a program to calculate age from birthdate
- 72. Write a function to find day of the week for any date
- 73. Implement a program to calculate time difference between dates
- 74. Create a function to add/subtract days from a date
- 75. Write a program to find the next business day
- 76. Implement a function to check if a year is a leap year
- 77. Create a program to display current time in different time zones
- 78. Write a function to format dates in different formats
- 79. Implement a program to calculate working days between dates
- 80. Create a function to find the last day of any month

Basic Web Development

- 81. Create a simple HTML page with basic styling
- 82. Write a JavaScript function to change page background color
- 83. Implement a form with client-side validation
- 84. Create a responsive navigation menu
- 85. Write a script to display current date and time on webpage
- 86. Implement a simple image gallery
- 87. Create a contact form with validation
- 88. Write a function to toggle between light and dark themes
- 89. Implement a simple accordion component
- 90. Create a basic shopping cart interface

Miscellaneous

91. Write a program to generate random passwords
92. Create a function to encode/decode Caesar cipher
93. Implement a simple unit converter
94. Write a program to calculate BMI
95. Create a function to generate multiplication tables
96. Implement a program to find anagrams in a word list
97. Write a function to calculate loan payments
98. Create a program to simulate a simple ATM
99. Implement a function to generate QR codes
100. Write a program to create ASCII art from text

Advanced Level

Data Structures Implementation

101. Implement a stack data structure from scratch
102. Create a queue data structure with enqueue/dequeue operations
103. Build a linked list with insert, delete, and search operations
104. Implement a binary search tree
105. Create a hash table with collision handling
106. Build a heap data structure (min/max heap)
107. Implement a graph using adjacency list
108. Create a trie data structure for string storage
109. Build a circular buffer
110. Implement a priority queue

Advanced Algorithms

- 111. Implement quicksort algorithm
- 112. Create a merge sort implementation
- 113. Build a breadth-first search algorithm
- 114. Implement depth-first search for graphs
- 115. Create Dijkstra's shortest path algorithm
- 116. Build a binary search algorithm
- 117. Implement selection sort algorithm
- 118. Create insertion sort algorithm
- 119. Build a radix sort implementation
- 120. Implement A* pathfinding algorithm

Database Operations

- 121. Create a simple database connection manager
- 122. Write SQL queries for complex data retrieval
- 123. Implement a basic ORM (Object-Relational Mapping)
- 124. Create database migration scripts
- 125. Build a query builder for dynamic SQL
- 126. Implement database connection pooling
- 127. Create stored procedures for common operations
- 128. Build a database backup and restore system
- 129. Implement database indexing strategies
- 130. Create a simple database cache layer

Web Development Projects

- 131. Build a RESTful API with CRUD operations

- 132. Create a user authentication system
- 133. Implement session management
- 134. Build a content management system
- 135. Create a blog platform with admin panel
- 136. Implement a shopping cart with payment integration
- 137. Build a social media feed
- 138. Create a file upload and management system
- 139. Implement a search functionality with filters
- 140. Build a commenting system

GUI Applications

- 141. Create a desktop calculator application
- 142. Build a text editor with save/load functionality
- 143. Implement a media player application
- 144. Create a file manager with folder navigation
- 145. Build a calendar application
- 146. Implement a drawing application
- 147. Create a password manager
- 148. Build a weather application
- 149. Implement a chat application
- 150. Create a task management application

Network Programming

- 151. Build a simple HTTP server
- 152. Create a TCP client-server application
- 153. Implement a file transfer protocol

- 154. Build a real-time chat server
- 155. Create a web scraper
- 156. Implement email sending functionality
- 157. Build a simple proxy server
- 158. Create a URL shortener service
- 159. Implement a basic load balancer
- 160. Build a DNS resolver

Security & Encryption

- 161. Implement RSA encryption/decryption
- 162. Create a password hashing system
- 163. Build a secure session management system
- 164. Implement JWT token authentication
- 165. Create a simple firewall application
- 166. Build a file encryption/decryption tool
- 167. Implement secure random number generation
- 168. Create a digital signature verification system
- 169. Build a simple intrusion detection system
- 170. Implement secure communication protocols

Game Development

- 171. Create a Snake game
- 172. Build a Tetris clone
- 173. Implement a Pac-Man style game
- 174. Create a simple platformer game
- 175. Build a puzzle game (like 2048)

- 176. Implement a card game (like Blackjack)
- 177. Create a memory matching game
- 178. Build a simple RPG battle system
- 179. Implement a tower defense game
- 180. Create a space invaders clone

Data Processing

- 181. Build a CSV parser and processor
- 182. Create a JSON data validator
- 183. Implement XML parsing and manipulation
- 184. Build a data export system
- 185. Create a report generator
- 186. Implement data cleansing algorithms
- 187. Build a data transformation pipeline
- 188. Create a simple ETL (Extract, Transform, Load) system
- 189. Implement data compression algorithms
- 190. Build a data visualization tool

System Programming

- 191. Create a simple operating system kernel
- 192. Build a file system implementation
- 193. Implement memory management algorithms
- 194. Create a process scheduler
- 195. Build a device driver
- 196. Implement inter-process communication
- 197. Create a simple shell interpreter

- 198. Build a system monitor
- 199. Implement a virtual machine
- 200. Create a debugger tool

Mobile Development

- 201. Build a native mobile app for contacts management
- 202. Create a weather app with GPS integration
- 203. Implement a photo gallery app
- 204. Build a note-taking application
- 205. Create a fitness tracking app
- 206. Implement a location-based reminder app
- 207. Build a simple game for mobile
- 208. Create a barcode scanner app
- 209. Implement a voice recording app
- 210. Build a mobile-friendly web application

API Development

- 211. Create a GraphQL API
- 212. Build a WebSocket real-time API
- 213. Implement API rate limiting
- 214. Create API documentation generator
- 215. Build API versioning system
- 216. Implement API authentication middleware
- 217. Create API logging and monitoring
- 218. Build API testing framework
- 219. Implement API caching strategies

220. Create API gateway

Performance Optimization

221. Implement code profiling tools

222. Create memory leak detection

223. Build performance benchmarking suite

224. Implement database query optimization

225. Create caching strategies

226. Build load testing tools

227. Implement code minification

228. Create image optimization tools

229. Build CDN integration

230. Implement lazy loading mechanisms

Testing & Quality Assurance

231. Create unit testing framework

232. Build integration testing suite

233. Implement automated testing pipeline

234. Create test data generation tools

235. Build code coverage analysis

236. Implement mock object framework

237. Create performance testing tools

238. Build security testing suite

239. Implement accessibility testing

240. Create cross-browser testing framework

DevOps & Deployment

- 241. Build CI/CD pipeline
- 242. Create Docker containerization
- 243. Implement infrastructure as code
- 244. Build monitoring and alerting system
- 245. Create log aggregation system
- 246. Implement blue-green deployment
- 247. Build configuration management
- 248. Create backup and recovery system
- 249. Implement service discovery
- 250. Build auto-scaling system

Machine Learning Basics

- 251. Implement linear regression from scratch
- 252. Create a k-means clustering algorithm
- 253. Build a naive Bayes classifier
- 254. Implement decision tree algorithm
- 255. Create a neural network from scratch
- 256. Build recommendation system
- 257. Implement text classification
- 258. Create image recognition system
- 259. Build sentiment analysis tool
- 260. Implement anomaly detection

Compiler & Interpreter

- 261. Build a simple lexer

- 262. Create a parser for mathematical expressions
- 263. Implement an interpreter for basic language
- 264. Build a compiler for simple language
- 265. Create syntax highlighting tool
- 266. Implement code formatter
- 267. Build abstract syntax tree generator
- 268. Create symbol table implementation
- 269. Implement type checker
- 270. Build code generator

Where To Go From Here

Get the FREE Online Course & Certificate

With this book in hand, you're unlocking a world of opportunity — including instant lifetime access to a FREE online course on Mammoth Club!

Scan this QR code to get a 100% off coupon code for an exclusive online course and cheat sheet!
Or go this link to redeem your coupon code:



www.mammothclub.com/course/1-hour-coding/UNLOCKED

You'll also earn a Certificate of Achievement for completing the online course. Join our thriving community of learners spanning 160+ countries, and be part of the 9 million+ courses sold around the globe.

Adding this book, its online course, and your certificate of achievement to your resume, Github, social media and LinkedIn profiles is a powerful way to showcase your dedication to professional growth and your commitment to staying ahead in your field.

Not only does it demonstrate that you have invested time and effort to acquire up-to-date knowledge and practical skills, but it also signals to employers and peers that you are proactive and serious about your career development.

Featuring these achievements makes your profile more competitive and credible, helping you stand out in a crowded job market.

Moreover, sharing them on LinkedIn can attract new opportunities by expanding your visibility within your industry and connecting you with like-minded professionals who value continuous learning and expertise.

See You at the Top

This isn't goodbye — it's your launch pad. I'll see you in Mammoth Club!

About Your Author



Alex Kropf is Mammoth Club's CLO, public speaker, consultant, IT author and Senior Software Developer. Alex has produced 1,000+ best-selling courses, books and workshops for Mammoth Club, Course Pro and clients worldwide.

Mammoth Club is a leading online course provider in everything from learning to code to becoming a YouTube star. Since 2011, Mammoth Club has built a global student community with over 9 million courses sold.

John Bura is Founder and CEO of global tech giant Mammoth Club and viral app Course Pro, the #1 AI-powered Learning Management System for course and content development, training and evaluation.



Trademarks

The "OpenAI" name, the OpenAI logo, the "ChatGPT" and "GPT" brands, and other OpenAI trademarks, are property of OpenAI.

Note From Your Author

Neither the author or publisher of this book nor AI itself can be held responsible if you accidentally step on any copyright toes, overspend your API billing limits, or send confidential data to a compromised chatbot. By flipping through these pages and using AI, you agree to hold yourself entirely responsible for what you do with your AI-powered creations.



This book is brought to you by Mammoth Club — it's not connected to or sponsored by any other company. Everything here is just the author's perspective, not any company's official view.

Visit MammothClub.com

for free online video courses, ebooks, source code, customer support and MORE!

To build and sell your own courses, presentations, books and videos: visit #1 course creation platform:

CoursePro.ai



MAMMOTH *CLUB*