🔵 High-Level System Design Roadmap with C# and .NET

🔴 Fundamentals of High-Level System Design

| ├── 🟢 Understanding System Requirements (Functional & Non-Functional)
| ├── 🟢 Architectural Patterns (Monolith, Microservices, SOA, Event-Driven)
| ├── 🟢 Trade-offs in System Design (Scalability vs Consistency vs Availability)

🔵 Scalability & Performance Considerations

| ├── 🟢 Horizontal vs Vertical Scaling
| ├── 🟢 Caching Strategies (CDN, Redis, Database Caching)
| ├── 🟢 Asynchronous Processing (Message Queues, Event-Driven Architecture)
| ├── 🟢 Rate Limiting & Load Balancing (Reverse Proxy, API Gateway)

🔵 Microservices & Distributed Systems

| ├── 🟢 Monolith to Microservices Migration
| ├── 🟢 Service Discovery & API Gateway (Ocelot, YARP)
| ├── 🟢 Inter-Service Communication (gRPC, REST, Message Queues)
| ├── 🟢 Distributed Transactions & Saga Pattern
| ├── 🟢 Event-Driven Architecture (Kafka, RabbitMQ)

🔵 System Reliability & Fault Tolerance

| ├── 🟢 Circuit Breaker & Retry Policies (Polly in .NET)
| ├── 🟢 Replication & Failover Strategies
| ├── 🟢 Health Checks & Monitoring
| ├── 🟢 Chaos Engineering & Fault Injection

🔵 Data Storage & Database Design

| ├── 🟢 SQL vs NoSQL (Relational vs Document Stores)
| ├── 🟢 Database Partitioning & Sharding
| ├── 🟢 Event Sourcing & CQRS for Scalable Data Handling
| ├── 🟢 Multi-Region Database Architecture

🔵 Security & Compliance in HLD

| ├── 🟢 Authentication & Authorization (OAuth2, OpenID Connect, JWT)
| ├── 🟢 Zero Trust Security Model
| ├── 🟢 Data Encryption at Rest & In Transit

| ├── 🟢 API Security Best Practices (Rate Limiting, IP Whitelisting)

🔵 Cloud-Native System Design

| ├── 🟢 Containerization & Orchestration (Docker, Kubernetes)
| ├── 🟢 Serverless Computing (Azure Functions, AWS Lambda)
| ├── 🟢 Multi-Cloud & Hybrid Cloud Strategies
| ├── 🟢 Infrastructure as Code (Terraform, Bicep, Pulumi)

🔵 CI/CD & DevOps for Scalable Systems

| ├── 🟢 CI/CD Pipelines (GitHub Actions, Azure DevOps, Jenkins)
| ├── 🟢 Blue-Green & Canary Deployments
| ├── 🟢 Observability & Logging (OpenTelemetry, Prometheus, Grafana)
| ├── 🟢 Automated Testing & Deployment Strategies

🔵 Soft Skills & Decision Making

| ├── 🟢 System Thinking & Trade-Off Analysis
| ├── 🟢 Design Reviews & Technical Documentation
| ├── 🟢 Handling Business & Engineering Constraints
| ├── 🟢 Collaboration Between Engineers & Product Teams

Master these concepts to design scalable, reliable, and high-performance systems using C# and .NET!

Follow Abhinn Mishra for more deep dives into .NET & System Architecture!
#CSharp #HighLevelDesign #DotNet #SoftwareArchitecture