My Approach to System Designs - A Comprehensive Guide

Are you struggling with system design interviews? If yes, then to be honest you're not alone and with so many concepts to learn, it's very easy to feel overwhelmed.

BUT here's the good news, you don't actually need to know everything to ace the interview. In fact, **80% of system design interviews focus on just 20% of the topics** (I know, hard to believe).

My guide (which I am very proud of) zeroes in on those key areas, helping you focus your efforts on what really matters. With a clear roadmap, you'll be better prepared and increase your chances of success!

What's inside?????

PART 1: 4-Step Roadmap (If you're Interview Is in 15 Days)

No need to spend hours and hours studying without a clear plan! This 4-step roadmap gives you a simple and structured way to learn system design in a short time. Even with just 15 days, you'll build a pretty strong foundation and prepare effectively for your interview.

1. How to Approach a System Design Interview
2. 15 Fundamental System Design Concepts
3. 3-Step Framework for System Design Interviews
4. Real-World System Design Examples

PART 2: Understand System Design from Basics to Advanced

Learn system design step by step with clear explanations, real-world examples, and code snippets. Whether you're just starting or already have experience, this guide will help you build a strong understanding and boost your confidence.

System Design Topics Covered:

1. **Introduction to System Design**
    a. What is system design?
    b. How to approach system design?
    c. How to know if your system is well-designed?
2. **Scalability and Performance**
3. **Latency and Throughput**
4. **Architectural Patterns**
5. **Core System Design Principles**
    a. Availability and Availability Patterns
    b. Replication
    c. Consistency and Consistency Patterns
    d. CAP Theorem
    e. PACELC Theorem
6. **Database and Storage**
    a. Relational Databases
    b. Database Isolation Levels
    c. Scaling Databases
    d. Sharding and Partitioning
    e. Non-Relational Databases
    f. Choosing the Right Database

7. **Caching**
8. **Asynchronous Processing**
    a. Message Queues (Kafka, RabbitMQ)
    b. Message Streams and Kafka Essentials
9. **Monolithic vs. Microservices Architecture**
10. **Event-Driven Architecture**
11. **API Gateway and Backend for Frontend (BFF)**
12. **REST, GraphQL, and gRPC**
13. **Long Polling, WebSockets, Server-Sent Events (SSE)**
14. **Design Patterns**
15. **Resiliency**
    a. Designing for Resiliency
    b. Load Balancers
    c. Circuit Breakers
    d. Data Redundancy and Recovery
16. **System Essentials**
    a. Essentials at Scale
    b. Bloom Filters
    c. Consistent Hashing
    d. Client-Server and Communication Protocols
    e. Blob Storage and S3
17. **Networking and Communication**
18. **Real-World Architectures & Engineering Blogs**

PART 3: Ultimate Cheat Sheet

This quick-reference cheat sheet I made covers the most important system design concepts, making it easy to review key points right before your interview.

PART 4: 50 Detailed Q&A for LLD and HLD

50 **carefully selected interview questions**, covering both **Low-Level Design (LLD)** and **High-Level Design (HLD)**. Each question comes with:

- ✔ Clear explanations
- ✔ Diagrams for better understanding
- ✔ Real-world examples
- ✔ Practical insights to simplify complex topics