

System Design Course

- Introduction to System Designs
- Importance of learning
- Why to learn this course?
- Who all gets benefited?
- Tips and tricks for interviews.

Concepts:

1. Monolith Vs Microservice
2. Various types of databases and its use cases (RDBMS, NoSQL, Elastic Search etc)
3. Database concepts – ACID, BASE properties, Replication strategies
4. Partitioning , Sharding , Federation etc.
5. Consistent Hashing and CAP Theorem
6. Consistency, availability and Reliability
7. Load balancers – Types uses
8. CDNs
9. DNS
10. Http/Https – SSL, Certificates
11. What is cloud?
12. How to store the password – a classic system design problem?
13. DDOS attack, SQL injections, Cross origin - ways to handle it.
14. Websockets with Demo – why it's so important and how does it work.
15. Restful APIs – Best Practices, SOAP, gRPC
16. Core Kafka –concepts (Kafka, Zookeeper)
17. Kafka – Hands on, CLI and Java Code
18. All About Cache – Different types and strategies and policies.
19. TCP/UDP – when and where to use.
20. 7- by design principles.
21. Performance/Scalability/Latency – Horizontal vs vertical
22. VM vs Dockers
23. Docker containers- hands on
24. Fault tolerance, DR, Zonal, Availability
25. Long polling alternatives
26. Event based design – Event driven programming – examples.
27. MVC architecture
28. Building a tech stack – Ingestion, processing and analytics (using Kafka, Flink, Druid/Power BI)

Uses cases – System design:

1. TinyURL – A URL shortner service
2. Twitter
3. API Rate limiter
4. File Storage- DropBox or equivalent
5. Notification System
6. WhatsApp or equivalent

7. Few random design problems
8. Live Mock design sessions(subject to time availability)

LLD:

1. OOPS
2. SOLID principles
3. Aggregations/Compositions
4. Design patterns

Use cases:

1. Elevator
2. MLCP
3. BookMyShow
4. Few random design topics
- 5.

Generative AI(Add on session based on time availability):

1. Introduction and some examples.