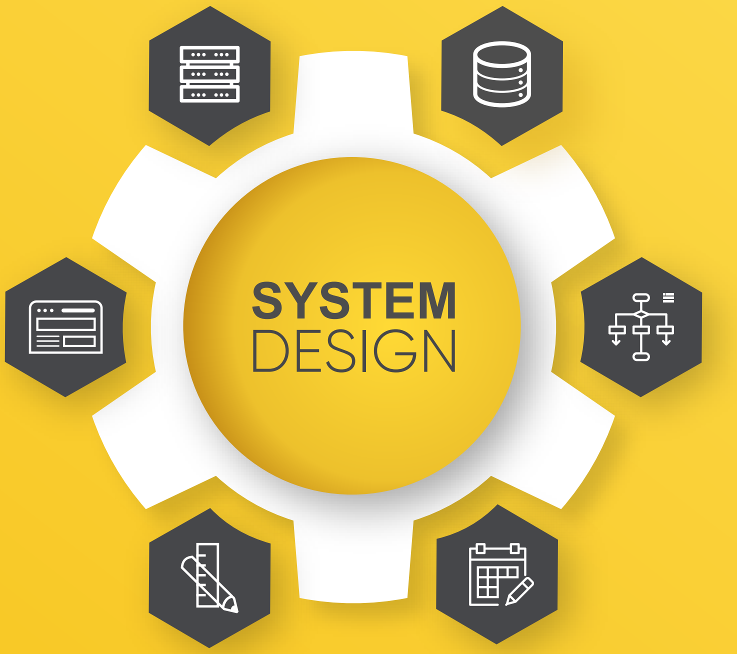
GeeksforGeeks

System Design
Live



Detailed
Course Syllabus

Lecture 1

a) **What exactly is a System Design Interview?**

b) **Expectations from Interviewee**

   i) Breadth Vs Depth

   ii) Should you know everything about everything?

c) **Types of Jobs to target from the market**

d) **System Design Process (Motivating Example: Design UBER.)**

   i) Common Mistakes

   ii) Chaotic Approach

   iii) Systematic Approach

e) **Design problem focussed on Requirement Analysis and Data Modelling**

Lecture 2

a) **Generic Components design diagram of a large-scale system**

b) **Trade-offs in a large-scale system (Motivating Example: Design TWITTER.)**

   i) Performance Vs Scalability

   ii) Latency Vs Throughput

   iii) Availability Vs Consistency (CAP Theorem)

c) **Design problem focused on Requirement Analysis and Data Modelling**

Lecture 3

a) **Load Balancers**

   i) Why?

   ii) Algorithms

   iii) Benefits

b) **SSL Termination and SSL Passthrough**

c) **Reverse Proxy**

   i) When to use?
   ii) Benefits

d) **Design problem-focused using reverse proxy and LB**

- ## Lecture 4

a) **Web Sockets**
   i) Why Web sockets?
   ii) Establishing a web socket connection

b) **Monoliths and Microservices**
   i) Understanding the misconceptions.
   ii) What are they?
   iii) When to use what?
   iv) Advantages of one over the other.

c) **Facebook Messenger Design using Web sockets**

d) **O Auth 2.0**
   i) What is it?
   ii) Understanding with an example.

e) **JWT tokens**
   i) What is it?
   ii) Demo

f) **Design problem focused on writing microservices**

- ## Lecture 5

a) **CDN**
   i) What is it?
   ii) Why and when?

b) **Caching**
   i) Why caching?
   ii) Implementation types.
   iii) When to use which implementation.
   iv) Eviction policies
   v) Redis Intro

c) **Design Problem focused on using CDNs**

Lecture 6

- a) **API Gateway**
    - i) Why?
    - ii) Features
- b) **DNS**
- c) **HTTPS working**
    - i) HTTPS vs HTTP
- d) **Zookeeper**
- e) **Design problems focused on zookeepers.**

Lecture 7

- a) **Distributing data in a large-scale system**
- b) **Linear Hashing**
    - i) What is it?
    - ii) Why?
    - iii) Issues?
    - iv) Solution?
- c) **Consistent hashing**
    - i) What is it?
    - ii) How it is better than linear hashing
- d) **Design problem based on consistent hashing.**

Lecture 8

- a) **How to handle massive data?**
    - i) What are the issues if the data set is huge?
- b) **Indexing**
    - i) Primary
    - ii) Secondary
    - iii) Multilevel
- c) **Data partitioning**
    - i) Vertical partitioning

   ii) Horizontal partitioning

d) **Sharding**
   i) What is data sharding?
   ii) Sharding Techniques

e) **Replication and Mirroring**

f) **Design problem based on data modeling.**

Lecture 9

a) **Push vs Pull Mechanism.**
   i) What are they?
   ii) When to choose what?

b) **Tips on System Design**
   i) How to present end-to-end design during interviews?
   ii) Some direct tips from top companies.

c) **Applying end-to-end learning on designing a System.**

d) **Event Driven design of a problem.**

Lecture 10

a) **Attacks on a service**
   i) Understanding attacking using DDOS as an example
   ii) DDOS working.
   iii) How to identify an attack
   iv) How to mitigate

b) **Encryption Mechanism**
   i) Symmetric
   ii) Asymmetric
   iii) Hashing
   iv) Digital Signature

c) **API Designs**
   i) Tips on designing APIs.

d) **Revisiting Databases**
   i) SQL vs No SQL
   ii) Scalability of SQL vs NoSQL
   iii) ACID properties
   iv) BASE

e) **Applying end-to-end learning on designing a System.**