Learnbay

"System Design Interview Cheat Sheet"

www.learnbay.co

Learnbay

Introduction

System design

interviews test your ability to create scalable, efficient, and maintainable systems. This cheat sheet covers frameworks, API design, scalability, caching, and real-world use cases to help you succeed

www.learnbay.co

Learnbay

Interview Framework

Stage	Description	Example
Understand Problem	Clarify requirements, constraints, and expectations	For a URL Shortener: Ask if custom URLs are needed
High-Level Design	Sketch out system components and APIs.	For File Upload Service: Define components like storage, load balancer
Deep Dive	Explore key components in detail and evaluate trade-offs.	Database schema for a messaging app.
Improve Design	Summarize design and explain trade-offs and alternatives	Caching popular posts in a newsfeed
Wrap-Up	Summarize design and explain trade-offs and alternatives.	Highlight trade-offs between cost and performance.

Learnbay

API Design Choices

API Type	Properties	Data	Use Cases
REST	Stateless, simple, widely adopted	JSON, XML, YAML	Web apps, developer API
RPC	Action-oriented, high performance	JSON, XML, Thrift, Protobuf	Microservices
GraphQL	Flexible, single endpoint	JSON	Mobile apps, complex interactions

Key Example: For an e-commerce platform

Use REST for product catalog APIs.

Use GraphQL for dynamic user dashboards.

Learnbay

Scalability Strategies

Replication

Type	Description	Example
Active-Active	All nodes handle requests simultaneously	Multi-region database setup
Active-Passive	One active node handles requests; others are backups.	Disaster recovery systems

Partitioning

Type	Description	Example
Horizontal	Split rows into different tables	For a URL Shortener: Ask if custom URLs are needed
Vertical	Split columns into different tables	Separate user credentials and preferences

Learnbay

Load Balancing

Type	Description	Example
Round-Robin	Distributes traffic evenly across servers.	Simple web server load
Least Connections	Directs traffic to the server with fewer requests	Real-time chat application

Caching Techniques

Caching Type	Key Features	Example Use Case
In-Memory Cache	Fast access, short-lived storage	Caching session data.
Distributed Cache	Scalable, shared across nodes	Caching search results.

Learnbay

Eviction Policies

Policy	Description
LRU (Least Recently Used)	Split rows into different tables
FIFO (First In, First Out)	Removes the oldest data first.

Tools Comparison

Tool	Features	Best Use Case
Redis	In-memory, low latency	Real-time leaderboards
Memcached	Lightweight, simple	Session management.

Learnbay

Real-World Use Cases

System	Key Components	Challenges
Video Streaming	CDN, transcoding service, storage	High bandwidth, latency.
Social Media Feed	APIs, caching, user preferences	Personalization, scalability.
Chat Application	Message queues, real-time updates, persistent storage	High availability

Example: Chat Application

Message Queue: Handles real-time delivery of messages.
Database: Stores chat history with fast retrieval.
API Design: REST APIs for sending/receiving messages.

Learnbay

Key Diagrams

Diagram 1: **Load Balancer Flow**

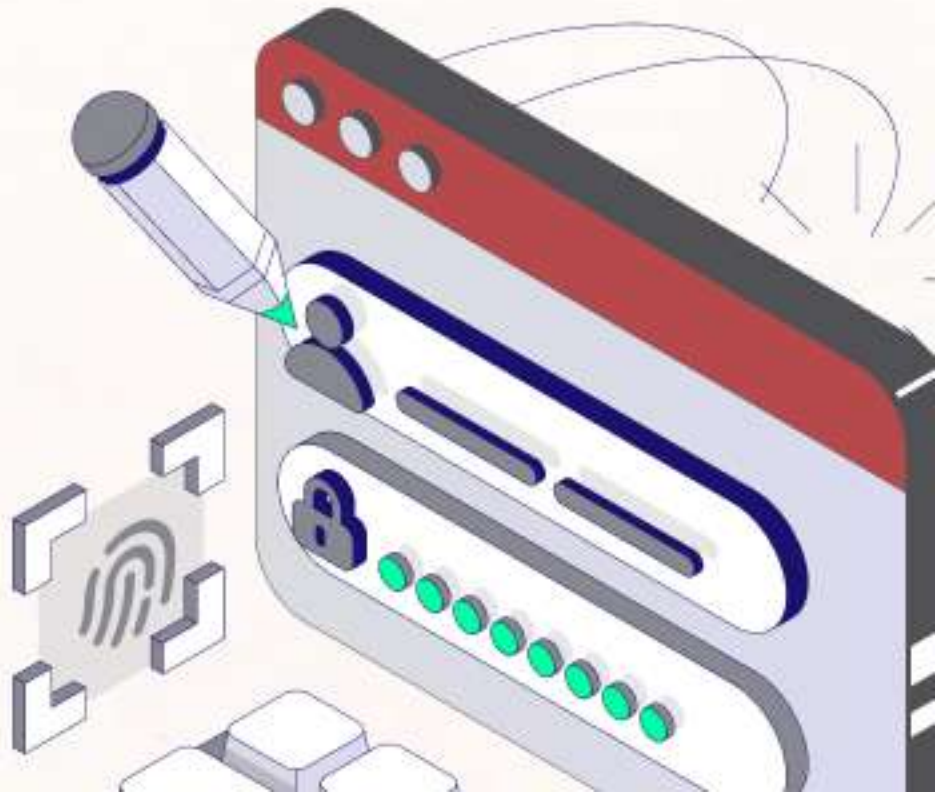
Visualize how incoming requests are distributed to multiple servers

Diagram 2: **Partitioning**

Show horizontal and vertical partitioning with labeled data flows.

Diagram 3: **Caching Strategy**

Demonstrate client-server interactions with cache hits/misses



Learnbay

Final Tips

- **Be Methodical**: Start with clarifying requirements and constraints.
- **Draw Diagrams:** Use visuals to explain architecture effectively.
- **Consider Trade-Offs:** Always explain cost, scalability, and performance implications.
- **Practice Regularly**: Work on real-world design scenarios to build confidence.

www.learnbay.co

Learnbay

Take Your Skills to the Next Level!

VISIT

www.learnbay.co

EXPLORE OUR COURSES

DATA SCIENCE & AI PROGRAMS

SOFTWARE DEVELOPMENT PROGRAMS

Get Certification from :

IBM | IIT Guwahati | WOOLF UNIVERSITY | Microsoft