Here’s a list of **commonly asked system design questions** in interviews (ranging from mid-level to senior engineer roles):

### 🔹 ****Scalable Services****

Design **URL Shortener** (like Bitly)

Design **Rate Limiter** (API throttling)

Design **Notification System** (push/SMS/email)

Design **News Feed System** (like Facebook/Twitter)

Design **Chat/Messaging System** (like WhatsApp/Slack)

Design **Video Streaming Service** (like YouTube/Netflix)

Design **Search Autocomplete** (like Google Search bar)

Design **File Storage System** (like Dropbox/Google Drive)

Design **Payment System** (like PayPal/Stripe/UPI)

Design **Food Delivery System** (like Swiggy/Zomato/Uber Eats)

### 🔹 ****High Throughput Systems****

Design **Distributed Queue System** (like Kafka/RabbitMQ)

Design **Real-Time Analytics System** (like Google Analytics)

Design **Ad Click Tracking System**

Design **Logging & Monitoring System** (like ELK/Datadog)

Design **Leaderboard/Ranking System** (like online games)

### 🔹 ****Content-heavy Applications****

Design **E-commerce System** (Amazon/Flipkart)

Design **Recommendation System** (Netflix/Amazon)

Design **Search Engine** (basic Google)

Design **Online Book Store**

Design **Collaborative Document Editor** (like Google Docs)

### 🔹 ****Specialized Problems****

Design **Distributed Cache** (like Redis/Memcached)

Design **Distributed Locking Service**

Design **CAP Theorem trade-offs in a system**

Design **Distributed Key-Value Store** (like DynamoDB/Cassandra)

Design **Job Scheduling System** (like Cron / Airflow)

### 🔹 ****System Design Concepts Interviewers Often Test****

**Scalability** → How system grows (vertical vs horizontal scaling).

**High Availability & Fault Tolerance** → How to avoid single points of failure.

**Consistency vs Availability (CAP theorem, PACELC tradeoffs)**.

**Data Partitioning & Sharding** → How to split data across nodes.

**Caching Strategies** → CDN, Redis, database caching.

**Database Choices** → SQL vs NoSQL (when and why).

**Load Balancing** → Different algorithms (round robin, consistent hashing).

**Concurrency** → Handling race conditions, distributed locks.

**Monitoring & Logging** → Observability in large-scale systems.

👉 For **Senior Software Engineer interviews**, the most frequent ones are:

**URL Shortener**, **Chat System**, **Food Delivery App**, **Notification System**, **E-commerce backend**, and **Video Streaming Service**.