28-Day System Design Prep Plan (Detailed)

Day 1

Concepts to be Covered: Intro to System Design, scalability, availability, latency vs throughput.

System: URL Shortener (TinyURL).

Tech: Hashing, SQL vs NoSQL, REST APIs.

Extra Notes: Understand trade-offs in DB choice.

Daily Practice Plan: Sketch API design & DB schema for short URL service.

Day 2

Concepts to be Covered: Load Balancing – horizontal vs vertical scaling, reverse proxies.

System: Scalable File/Image Server.

Tech: Nginx, HAProxy, CDN basics.

Extra Notes: Difference between LB algorithms (round robin, consistent hashing).

Daily Practice Plan: Design file upload/download system with LB.

Day 3

Concepts to be Covered: Caching strategies – read-through, write-through, write-back.

System: News Feed system.

Tech: Redis, Memcached.

Extra Notes: Cache invalidation is the hardest problem.

Daily Practice Plan: Implement Redis cache for a sample API.

Day 4

Concepts to be Covered: Database design – SQL vs NoSQL, sharding, replication.

System: E-commerce Product Catalog.

Tech: MySQL, Cassandra, MongoDB.

Extra Notes: Think about consistency vs availability.

Daily Practice Plan: Model DB schema for product search & filter.

Day 5

Concepts to be Covered: Indexing, search, query optimization.

System: Search Autocomplete.

Tech: ElasticSearch, Lucene.

Extra Notes: Focus on prefix search efficiency.

Daily Practice Plan: Design autocomplete system with trie + ES.

Day 6

Concepts to be Covered: Consistency & Availability – CAP theorem.

System: Banking Transaction System.

Tech: ACID DB, message queues.

Extra Notes: When to choose CP vs AP.

Daily Practice Plan: Write flow for money transfer ensuring consistency.

Day 7

Concepts to be Covered: Reliability & Fault tolerance.

System: WhatsApp Chat Backup System.

Tech: Replication, checkpointing, Kafka.

Extra Notes: Discuss leader-follower replication.

Daily Practice Plan: Design backup recovery plan for messages.

Day 8

Concepts to be Covered: Message Queues & Event-driven design.

System: Order Processing System.

Tech: Kafka, RabbitMQ.

Extra Notes: Decouple producer/consumer.

Daily Practice Plan: Design async order handling pipeline.

Day 9

Concepts to be Covered: API Design – REST vs gRPC vs GraphQL.

System: Ride-hailing System (Uber).

Tech: gRPC, REST APIs, API Gateway.

Extra Notes: Focus on API versioning & backward compatibility.

Daily Practice Plan: Write API endpoints for driver/rider matching.

Day 10

Concepts to be Covered: Rate Limiting & Throttling.

System: Twitter API.

Tech: Token Bucket, Leaky Bucket.

Extra Notes: Protect APIs from abuse.

Daily Practice Plan: Implement basic rate limiter in code.

Day 11

Concepts to be Covered: Storage Systems – block, file, object storage.

System: Google Drive/Dropbox.

Tech: S3, GFS, HDFS.

Extra Notes: Focus on durability & replication.

Daily Practice Plan: Sketch design for file upload & sync.

Day 12

Concepts to be Covered: Data Partitioning & Sharding.

System: YouTube Video Metadata System.

Tech: Consistent Hashing, Range Sharding.

Extra Notes: Avoid hotspots with sharding keys.

Daily Practice Plan: Partition user video metadata by region.

Day 13

Concepts to be Covered: CDNs & Edge Computing.

System: Netflix Video Streaming.

Tech: Akamai, Cloudflare.

Extra Notes: Reduce latency using edge nodes.

Daily Practice Plan: Design CDN for static content delivery.

Day 14

Concepts to be Covered: Monitoring, Logging, Alerting.

System: System Health Dashboard.

Tech: Prometheus, Grafana, ELK Stack.

Extra Notes: SLAs, SLOs, SLIs importance.

Daily Practice Plan: Create metrics dashboard for API latency.

Day 15

Concepts to be Covered: Security & Authentication.

System: Secure Payment Gateway.

Tech: OAuth2, JWT, TLS.

Extra Notes: Prevent replay attacks & SQL injection.

Daily Practice Plan: Design login + payment auth flow.

Day 16

Concepts to be Covered: Search Systems.

System: Twitter Search.

Tech: ElasticSearch, Solr.

Extra Notes: Indexing and ranking techniques.

Daily Practice Plan: Implement keyword-based search system.

Day 17

Concepts to be Covered: Distributed Transactions.

System: Booking System (Flights).

Tech: Two-Phase Commit, Sagas.

Extra Notes: Compare eventual vs strong consistency.

Daily Practice Plan: Model DB flow for flight booking with rollback.

Day 18

Concepts to be Covered: Pub/Sub Systems.

System: Notification Service.

Tech: Kafka, Redis Streams.

Extra Notes: Ensure idempotency in event delivery.

Daily Practice Plan: Build push notification service design.

Day 19

Concepts to be Covered: Graph Systems.

System: Social Network Graph (Facebook).

Tech: Neo4j, GraphQL.

Extra Notes: Friend recommendations via graph traversal.

Daily Practice Plan: Design friends-of-friends suggestion engine.

Day 20

Concepts to be Covered: Analytics Systems (OLAP).

System: Ad Click Tracking.

Tech: Snowflake, BigQuery, Spark.

Extra Notes: ETL pipelines & batch processing.

Daily Practice Plan: Sketch high-level analytics pipeline.

Day 21

Concepts to be Covered: Real-time Processing.

System: Stock Trading System.

Tech: Apache Kafka, Flink, Storm.

Extra Notes: Low latency critical.

Daily Practice Plan: Design order matching engine.

Day 22

Concepts to be Covered: WebSockets & Real-time APIs.

System: Chat App (Slack/WhatsApp).

Tech: WebSockets, STOMP, SignalR.

Extra Notes: Focus on delivery guarantees.

Daily Practice Plan: Design message sync with WebSockets.

Day 23

Concepts to be Covered: Payment Systems.

System: UPI/PayPal.

Tech: Idempotent APIs, secure DB.

Extra Notes: Reconcile failures safely.

Daily Practice Plan: Flow for double-spend prevention.

Day 24

Concepts to be Covered: Search Ranking & Personalization.

System: E-commerce Recommendations.

Tech: ML models + ElasticSearch.

Extra Notes: Hybrid search + personalization.

Daily Practice Plan: Sketch recommendation engine architecture.

Day 25

Concepts to be Covered: Data Warehousing.

System: ETL Data Lake.

Tech: Hive, Snowflake, Redshift.

Extra Notes: Batch vs stream ETL.

Daily Practice Plan: Design data ingestion + warehouse model.

Day 26

Concepts to be Covered: Microservices Architecture.

System: Amazon Shopping System.

Tech: Service Mesh, gRPC, Kafka.

Extra Notes: Avoid monolith scaling issues.

Daily Practice Plan: Break down monolith into services.

Day 27

Concepts to be Covered: Design at Scale.

System: Google Maps.

Tech: Distributed DBs, sharding, caching.

Extra Notes: Multi-region deployment strategies.

Daily Practice Plan: Design global map tile service.

Day 28

Concepts to be Covered: Mock Interview & Consolidation.

System: End-to-End System Design.

Tech: All tools covered.

Extra Notes: Focus on trade-offs & clarity.

Daily Practice Plan: Do 2 mock interviews on any system (e.g. Instagram, YouTube).