Day 1 — Scaling Basics

- Vertical vs Horizontal Scaling
- Availability vs Reliability
- Latency vs Throughput
- SPOF (Single Point of Failure)
  ✅ Practice: Know why horizontal scaling is preferred.

Day 2 — Networking Basics

- HTTP, HTTPS
- REST vs RPC vs WebSockets
- Load Balancer Basics
  ✅ Practice: Draw simple client → server → LB flow

Day 3 — Databases (SQL & NoSQL)

- ACID vs BASE
- SQL vs NoSQL
- CAP Theorem
  ✅ Practice: Choose DB for Instagram feed.

Day 4 — Caching

- Redis, Memcached
- Cache eviction: LRU, LFU
- Cache write strategies: write-through, write-back
  ✅ Practice: Design cache for product details (Amazon).

Day 5 — Indexing + Query Optimization

- B-tree, Hash index
- Why indexing matters
  ✅ Practice: Explain how index works in MySQL.

Day 6 — Storage & CDN

- Object storage (S3)

- CDN basics
- Replication
  ✅ Practice: Why YouTube uses CDN.

Day 7 — Revision + Mini Project

✅ Mini Design: **TinyURL (URL Shortener)**

- Hashing
- Database choice
- Cache
- High-level diagram

✅ WEEK 2 — Distributed Systems (Days 8–14)

🎯 Goal: Understand how large-scale systems behave.

Day 8 — Replication

- Master–Slave
- Leader–Follower
- Multi-master
  ✅ Choose replication for WhatsApp messages.

Day 9 — Sharding

- Range, Hash, Consistent Hashing
  ✅ How to shard 1 billion users.

Day 10 — Message Queues

- Kafka, RabbitMQ
- Pub/Sub

- Event-driven design
  ✅ Why Uber uses Kafka.

Day 11 — Consistency Models

- Strong consistency
- Eventual consistency
- Read-after-write
  ✅ Choose consistency for Instagram likes.

Day 12 — Rate Limiting

- Token bucket
- Leaky bucket
- Redis rate limiter
  ✅ Design rate limiter for API Gateway.

Day 13 — Concurrency & Locks

- Optimistic vs Pessimistic locking
- Deadlocks
  ✅ Why banking uses optimistic lock.

Day 14 — Revision + Mini Project

✅ Mini Design: **Chat System (WhatsApp basic)**

- WebSockets
- Messaging queue
- Read receipts
- Storage model

✅ WEEK 3 — System Design Patterns (Days 15–21)

🎯 Goal: Learn tools used by microservices & modern systems.

Day 15 — Microservices

- Pros & Cons
- Service discovery
- API Gateway
  ✅ Draw simple microservice architecture.

Day 16 — API Gateways

- Load balancing
- Authentication
- Routing
  ✅ Kong vs NGINX vs AWS API Gateway.

Day 17 — Circuit Breaker & Failover

- Hystrix pattern
- Retry, backoff
  ✅ Why Netflix invented Hystrix.

Day 18 — Distributed Transactions

- 2PC
- Sagas
  ✅ Saga example: booking ticket + payment.

Day 19 — Logging & Monitoring

- Prometheus
- Grafana
- ELK / EFK
  ✅ Why logs must be centralized.

Day 20 — Security

- OAuth2
- JWT
- Encryption
  ✅ Why JWT is stateless.

Day 21 — Revision + Mini Project

✅ Mini Design: **Notification Service** (Push + Email + SMS)

✅ WEEK 4 — Full High-Level Designs (Days 22–30)

🎯 Goal: Build 7 complete system design solutions.

Day 22 — Design Instagram Feed

- Fan-out
- Cache
- Media storage
- Ranking algorithm

Day 23 — Design Twitter

- Timeline service

- Tweet service
- ElasticSearch
- Follow graph

Day 24 — Design YouTube / Netflix

- Video upload
- Transcoding
- CDN
- Streaming protocol

Day 25 — Design Uber / Ola

- Location service
- Real-time matching
- Maps
- Surge pricing

Day 26 — Design E-commerce (Amazon)

- Product service
- Order service
- Inventory
- Search service

Day 27 — Design Google Docs (Real-time collaboration)

- Operational transform
- WebSocket communication
- Locking logic

Day 28 — Design Dropbox / Google Drive

- File storage
- Versioning
- Chunking
- Sync client

Day 29 — Mock Interview (Self-Practice)

Pick 1 random system:
✅ Ride sharing
✅ Instagram
✅ Ticket booking
✅ Chat app
✅ URL Shortener

Answer in 45 minutes.

Day 30 — Final Revision

- Review patterns
- Review consistency
- Check scalability tradeoffs
- Practice diagrams
