Notes: Frontend ↔ Backend Communication & Testing Concepts

■ Cloud Technology:
- Provides computing services (servers, storage, networking, databases, software) over the internet.
- Models:
- IaaS (Infrastructure as a Service): Rent servers, storage. Ex: AWS EC2, Google Compute Engine.
- PaaS (Platform as a Service): Ready platform to build/run apps. Ex: Heroku, Google App Engine.
- SaaS (Software as a Service): Ready-to-use apps over internet. Ex: Gmail, Google Drive, Zoom.

■ Databases Examples:
- Relational (SQL): MySQL, PostgreSQL, Oracle, SQL Server, SQLite.
- NoSQL: MongoDB, Cassandra, Redis, CouchDB, DynamoDB.
- Cloud Databases: Firebase Firestore, AWS RDS, Snowflake, Amazon Redshift.

■ Manual vs Automation Testing:
- Manual Testing: Human tester executes test cases without tools. Best for exploratory, usability testing.
- Automation Testing: Scripts/tools execute tests automatically. Best for regression, performance testing.
- Manual = slow but flexible. Automation = fast and repeatable.

■ HLD vs LLD:
- HLD (High-Level Design): Big picture design, system architecture, technologies, modules, data flow.
- LLD (Low-Level Design): Detailed logic of components, class diagrams, database schema, algorithms.

■ Data Structures & Algorithms:
- Data Structures: Ways to organize data (arrays, linked lists, stacks, queues, trees, graphs, hash tables).
- Algorithms: Step-by-step procedure to solve problems (sorting, searching, graph traversal).

■ Frontend vs Backend:
- Frontend: User interface (HTML, CSS, JavaScript, React, Angular).
- Backend: Server logic (Node.js, Java, Python, .NET) + database handling.

■ Communication Between Frontend & Backend:
1. REST API:
- Request-response model (HTTP methods: GET, POST, PUT, DELETE).
- Data in JSON format. Most common.
2. GraphQL:
- Client specifies exactly what data it needs via queries.
- Prevents over-fetching and under-fetching.
3. WebSockets:
- Persistent, two-way communication channel.
- Real-time updates (chat apps, stock updates, multiplayer games).
4. gRPC:
- High-performance protocol using binary format (Protocol Buffers).
- Common in microservices and backend-to-backend communication.

■ REST vs WebSockets (Analogy):

- REST: Like sending letters ■ (every request is a new message with envelope & address).
- WebSocket: Like a live phone call ■ (open pipe, both can talk anytime without reconnecting).