Web and Mobile Application Development

Khaula Molapo

20240001

Week: 3

1.

A food delivery mobile application uses APIs to connect the mobile interface with a remote server. When a user opens the app, it sends a GET request to the server's REST API to retrieve restaurant data in JSON format. The JSON response includes restaurant names, menus, prices, and ratings. When the user places an order, the app sends a POST request containing order details such as items selected, quantity, delivery address, and payment method. The server validates the JSON data, processes payment, and updates the database before returning a confirmation response. To improve performance, the app caches frequently accessed JSON responses such as popular restaurants. Authentication tokens are also included in API requests to ensure secure communication between the mobile app and server. If the API fails or returns an error status, the application displays an error message and retries the request. This API-driven approach ensures real-time updates, efficient data exchange, and scalability across different devices.

2.

REST statelessness means that every API request from a client must contain all the information needed for the server to understand and process it. The server does not store client session data between requests, which makes systems more scalable and easier to maintain. Each request includes authentication tokens, parameters, and necessary data so the server treats every interaction independently. This reduces server memory usage and allows load balancing across multiple servers because any server can handle any request. Statelessness also improves reliability since failures do not affect stored sessions. However, clients must manage their own state and send complete information each time.

3.

While testing an API using Postman, I sent GET and POST requests to check whether endpoints returned correct JSON responses. I verified HTTP status codes such as 200 for success and 400 for bad requests. Postman's built-in JSON viewer helped confirm the structure and formatting of returned data. I also tested invalid inputs to observe how the API handled errors and validation rules. Using headers, I included authentication tokens to simulate real user requests. Through this process, I learned the importance of validating JSON structure and ensuring consistent field names and data types. Testing APIs early helps developers identify problems quickly, improve reliability, and ensure smooth communication between mobile apps and backend services.

4.

An API workflow diagram illustrates how a client application communicates with a server through structured requests and responses. The process begins with the mobile or web client sending an HTTP request to an API endpoint. The API gateway receives the request, performs authentication checks, and forwards it to the application server. The server processes business logic, interacts with the database if needed, and generates a JSON response. The response is then returned through the API back to the client interface for display. Workflow diagrams help developers visualize data flow, identify potential bottlenecks, and understand how different components interact. They also assist in planning security measures and improving efficiency during development and testing.