Module 7: Networking and API Integration

Theory Assignments:

1. What is a RESTful API and its Importance in Mobile Applications:

A RESTful API (Representational State Transfer) is a web service that allows communication between a client (like a mobile app) and a server over HTTP using standard methods (GET, POST, PUT, DELETE). It is based on stateless communication, meaning that each request from the client to the server is independent and does not rely on previous requests.

- **Importance in Mobile Applications:**
  - Data Communication: RESTful APIs enable mobile apps to communicate with servers and fetch or send data, like user profiles, posts, or product information.
  - Scalability: They allow the backend system to scale easily by serving multiple clients, including mobile apps, web apps, and other platforms.
  - Standardization: REST is widely used, meaning that mobile developers can easily integrate third-party services and APIs, such as payment gateways or authentication systems.
  - Flexibility: REST APIs work over HTTP, which is the most common protocol for the web, making them easy to use and implement in mobile apps.

2. How JSON Data is Parsed and Used in Flutter:

JSON (JavaScript Object Notation) is a lightweight data format commonly used to send and receive data from a server in web and mobile applications.

- Parsing JSON in Flutter:
  - Decoding JSON: In Flutter, you can use the dart:convert library to parse JSON data into Dart objects. The jsonDecode() function is used to convert a JSON string into a Dart map or list.
  - Using Dart Models: To manage the parsed data, you often map the JSON response into Dart classes (models) by creating a model class with fields that match the JSON structure.
  - **Example Flow:**
    1. Fetch JSON data from an API using the http package.
    2. Decode the JSON into a Dart object.
    3. Convert the Dart object into a usable model with fields.

3. Purpose of HTTP Methods (GET, POST, PUT, DELETE) and When to Use Each:

- **GET**:
  - Purpose: Retrieves data from the server.
  - When to Use: Use GET when you want to request data, such as fetching user details, lists of items, or other resources.
- **POST**:
  - Purpose: Sends data to the server to create a new resource.
  - When to Use: Use POST when submitting form data, creating new records, or uploading files (e.g., creating a new user or adding a new post).
- **PUT**:
  - Purpose: Updates an existing resource on the server with new data.
  - When to Use: Use PUT when you want to update a resource completely (e.g., updating a user profile or modifying an item).
- **DELETE:**
  - Purpose: Deletes a resource from the server.
  - When to Use: Use DELETE when removing data from the server, like deleting a user account or removing an item from a list.