MODULE-7 (THEORY ASSIGNMENT)

>Networking and API Integration<

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 using HTTP methods. It enables mobile apps to fetch, send, update, or delete data over the internet. RESTful APIs are important because they help mobile apps interact with databases or cloud services in real time.

2. **How is JSON data parsed and used in Flutter?**

➔ In Flutter, JSON data is parsed using the dart:convert library. The jsonDecode() function is used to convert a JSON string into a Dart map or list. This data can then be used to create Dart objects. Parsing is commonly done when receiving data from RESTful APIs.

```
import 'dart:convert';

var jsonData = '{"name": "John", "age": 25}';
var parsedData = jsonDecode(jsonData); // parsedData is a Map
```

3. **Purpose of HTTP methods (GET, POST, PUT, DELETE) and when to use each:**

- GET: Used to fetch data from a server (e.g., get user details).
- POST: Used to send new data to the server (e.g., submit a form).
- PUT: Used to update existing data on the server (e.g., edit profile info).
- DELETE: Used to remove data from the server (e.g., delete a task).

MODULE- 9 (THEORY ASSIGNMENT)

>Animations and Transitions<

1. Difference between Implicit and Explicit Animations in Flutter:

- Implicit animations are easy to use and require minimal code. Flutter handles the animation automatically when a property changes.
  Example*:* AnimatedContainer*,* AnimatedOpacity*.*
- Explicit animations give full control over the animation using classes like AnimationController. They are used for complex or custom animations.
  Example*:* using AnimationController *with* Tween*.*

2. Purpose of AnimationController and its usage:

-> AnimationController is used in Flutter to control animations explicitly. It defines the duration, start, stop, and repeat behavior of an animation. It must be used inside a StatefulWidget and needs a TickerProvider.

```
_animationController = AnimationController(
  vsync: this,
  duration: Duration(seconds: 2), );
_animationController.forward();
```

3. Concept of Hero Animations in Flutter:

-> Hero animations create smooth transitions between two screens by animating a shared widget. It uses the Hero widget with the same tag on both screens. It gives a visual effect that the widget is "flying" from one screen to another.

```
Hero(tag: 'profilePic',
child: Image.asset('assets/profile.png'),);
```

MODULE- 10 (THEORY ASSIGNMENT)

>Firebase Integration<

1. **Purpose of Firebase and its core services:**

➔ Firebase is a Backend-as-a-Service (BaaS) platform by Google that helps developers build and manage mobile and web apps quickly without managing servers. It offers tools for authentication, database, analytics, and more.

- Core services include:

  - Firebase Authentication
  - Cloud Firestore (NoSQL database)
  - Firebase Realtime Database
  - Firebase Cloud Messaging (FCM)
  - Firebase Storage
  - Firebase Analytics
  - Firebase Hosting

2. **Firebase Authentication and its use cases in Flutter:**

➔ Firebase Authentication provides ready-to-use methods for user sign-in and sign-up using email/password, phone number, Google, Facebook, etc.

➔ Use cases in Flutter:

  - User login/sign-up flows
  - Social media sign-ins
  - Secure user access management
  - Anonymous authentication for guest users

3. How Firestore differs from traditional SQL databases:

- Firestore is a NoSQL, document-based database, storing data in collections and documents.
- SQL databases are relational and store data in structured tables with rows and columns.

Key differences:

- Firestore is schema-less; SQL has fixed schemas.
- Firestore supports real-time syncing; SQL usually requires manual querying.
- Firestore is more flexible and scalable for mobile apps, while SQL is better for complex relational queries.