



Vivekanand Education Society's Institute of Technology

(An Autonomous Institute Affiliated to University of Mumbai)

Department of Information Technology

A.Y. 24-25

Mobile App Development and Progressive Web App Lab

| | |
|----------------|----------------|
| Experiment No. | Assignment-1 |
| Title. | |
| Roll No. | 57 |
| Name | Hitesh Tanwani |
| Class | D15 B |
| Subject | MAD & PWA |
| Grade: | |

(03)
Q
A

ASSIGNMENT - 1

Explain the key features and advantages of using Flutter for mobile app development.

Flutter is a popular open source UI toolkit developed by Google for building natively compiled applications for mobile (iOS & Android) web and desktop from a single code base.

Key Features of Flutter:

1. Single codebase: Write once, run on multiple platforms (iOS, Android, web)

2. Dart Programming: Uses Dart, which is optimized for fast performance and ahead-of-time compilation

3. Hot Reload: Instantly reflects changes in the app without restarting, making development faster and more efficient

4. Rich Widget Library: Provides a vast collection of customizable widgets that support Material design and Cupertino styles for a native look and feel.

Advantages of Using Flutter:

1. Faster Development Time
2. Cost Effective
3. Consistent UI

Q.1(b) Discuss how the Flutter framework differs from traditional approaches and why it has gained popularity in the developer community?

→ Flutter uses a single database for multiple platforms, unlike native development that requires separate code for iOS (Swift) & Android (Kotlin). It does not rely on platform-specific UI components instead render everything using its own Skia graphic engine, ensuring consistency. Unlike React Native, which uses JavaScript bridge, Flutter compiles directly to native ARM code, offering better performance. Its hot reload feature allows development to see changes instantly making development faster & more efficient.

Flutter has gained popularity due to its faster development cost efficiency & cross platform support. Business prefers it as it reduces development time & cost while delivering high performance apps.

- 2) Describe the concept of the widget tree in Flutter. Explain the widget composition is used to build complex UI
- In flutter everything is a widget (button, text, layout, etc). These widgets are arranged in hierarchical structure known as the widget tree. The widget tree determines the UI.

~~Widget composition to build complex UI:~~

- Flutter ~~encourages~~ a composition-based approach rather than inheritance
- Instead of creating large, monolithic widget, developers build small, reusable widget that are combined to form complex UIs.

e.g. A column widget can hold multiple Text & Button widget creating a structured layout.

Q. 2b Provide example of commonly used widgets & their role in creating a widget tree

→ 1) Structural widget

- scaffold :- Provide basic structure of screen.
- container:- used for layout styling.
- column & row:- used for vertical & Horizontal layout.

2) Interactive widget

- text field:- for user input
- elevated button:- clickable buttons

3) Styling Widget

- padding :- Add spacing around widget
- align center:- Adjust alignment

4) List & scrollable widget

- list view:- scrollable list
- grid view:- provide display items in a grid.

```
on simple widget tree  
scaffold ( appBar : AppBar (title : Text ("Flutter App"))  
body : Column ( children : [  
Text ("Welcome to flutter !")  
Elevated Button ( onPressed : () {  
child : Text ("Click Me")  
}],  
),  
);
```

Q) Discuss the importance of state management in flutter application.

Importance of State Management in flutter application state management refers to handling dynamic data that changes overtime.

In flutter, the UI rebuilds where the state changes, ensuring the app remains interactive & responsive. Proper state management helps in performance optimization, code maintainability & better UI behaviour.

ex. simple widget tree

Q.3b compare and contrast the different state management approach in flutter approaches such as Set State, provider Riverpod. Provide scenarios where each approach is suitable.

→ 1. Set State

• ~~SetState~~

~~setState~~ is the simplest state management approach in flutter. It's built into the framework and is ideal for managing small, local state within a single widget.

Pros - easy, no dependencies

Cons - limited, not scalable.

Best for: small apps, UI state (eg toggle buttons).

2. Provider

~~Provider~~ is a popular state management package in flutter, designed to share state across widgets while separating business logic from UI code.

Pros: separates UI from logic, reactive

Cons: verbose for simple cases

Best For: Medium apps (eg Auth, shopping cart).

3.

3. Riverpod

Riverpod builds on the concept of Provider but offers more advanced features like better testability, no reliance on the widget tree.

Pros: Decouples from widget tree, better testability

Cons: Steeper learning curve

Best for: Large apps with complex state, async management, or advanced features (eg. fetching data from API's)

4

i) Explain the process of integrating firebase with a flutter application. Discuss the benefits of using ~~firebase~~ as a backend solution.

Integrating firebase with flutter & its Benifits.

→ Integrating process

- Set up firebase console.
- Create a firebase project.
- Register the App for Android & ios
- Download & add google-services.json or Google Services - info.plist (ios)

dependencies:

- firebase_core : latest version
- firebase_auth : latest version.
- yaml
- Initialize Firebase in Flutter

dart

```
void main() async {  
    WidgetsFlutterBinding.ensureInitialized();  
    await Firebase.initializeApp();  
    runApp(MyApp());  
}
```

Benefits:-

No need to manage servers. Provide authentication, database & cloud function, scalable & cost-effective.

b) highlight the firebase services commonly used in flutter development & provide brief overview of how data synchronization is achieved.

→ Firebase offers several powerful services commonly used in flutter development enabling easy integration for backend functionalities

1. ~~Firebase Authentication~~

Simplifies user authentication with methods like email / password, google sign in and anonymous login.

In flutter it is integrated using the firebase-auth plugin, enabling quick sign ups, log-ins

2 Cloud Firestore

It is a NoSQL document-based database that supports real-time syncing.

The cloud-firestore plugin allows reading and writing documents, setting up - real time listeners.

3. Firebase Realtime Database.

Unlike firestore, the realtime database uses a JSON tree structure for storing key values pairs.

In flutter, the Firebase database plugin helps us set up listeners for real-time updates, making it easy to build apps like Chat applications.