

NAME: _____

STD.: _____ DIV.: _____

DATE: 10/10/2023

PAGE: 69

Name: SHREYAS NAIK

Rollno: 38

Div: D15B

MPL Assignment 01

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

i) Key features of flutter

- 1) Single codebase - Write one code for both Android and iOS
- 2) Fast Performance - Uses dart language and high performance rendering engine.
- 3) Hot reload: See change instantly without restarting the app
- 4) Rich UI components - comes with customizable widgets for smooth UI design.
- 5) Native like experience - Provides ~~with~~ quality animations and fast execution.

Advantages of Using Flutter:

- 1) Saves time & effort - Single codebase for multiple platform
- 2) High speed Development - Hot reload feature speeds up coding
- 3) Cost effective - Reduces development cost & time
- 4) Attractive UI - Provides beautiful and customizable widgets
- 5) Good Performance - Uses dart and Skia for fast and smooth rendering.

- ⑥ Discuss how the Flutter framework differs from traditional approaches and why it has gained popularity in the dev community.

Soln: How flutter differs from Traditional Approaches

- 1) Single Codebase - Traditional methods need separate code for Android and iOS, but flutter uses one code for both
- 2) Hot Reload: - Traditional apps requires full restart after changes, but flutter updates instantly.
- 3) UI Rendering: Traditional apps use native components, flutter has its own rendering engine (Skia) for faster performance.
- 4) Performance - flutter compiles directly to native machine making it faster than frameworks that use a bridge
- 5) Customization - Traditional UI design depends on platform specific components, but flutter provides fully customizable widgets.

Why flutter is popular among Developers

- 1) Fast Development - Hot Reload and single codebase save time
- 2) Cross-Platform Support - Works on mobile, web and desktop
- 3) Beautiful UI - Rich, customizable widgets for modern design
- 4) High performance: Runs smoothly without a bridge like Native.
- 5) Active Community & Google Support - Regular updates and strong community help developers

NAME: _____

STD.: _____ DIV.: _____

DATE: _____

PAGE: _____

Describe the concept of the widget tree in flutter. Explain how widget composition is used to build complex user interfaces.

Concept of Widget Tree in flutter

In flutter, everything is a widget. Widgets are arranged in a tree structure called the widget tree. This tree represents the UI of the app, where parent widgets contain child widgets.

For example: a scaffold widget can have a column widget which contains Text and Button widgets. Changes in widgets update the tree dynamically.

Widgets Composition for complex UI

Flutter uses small, reusable widgets to build complex UI instead of creating a single large UI block, developers combine multiple small widgets like rows, column, containers, etc.

For example:

- A ListView can contain multiple Card widgets
- A Column can hold Text, Images and Buttons.

This modular approach makes the UI flexible, readable and easy to manage.

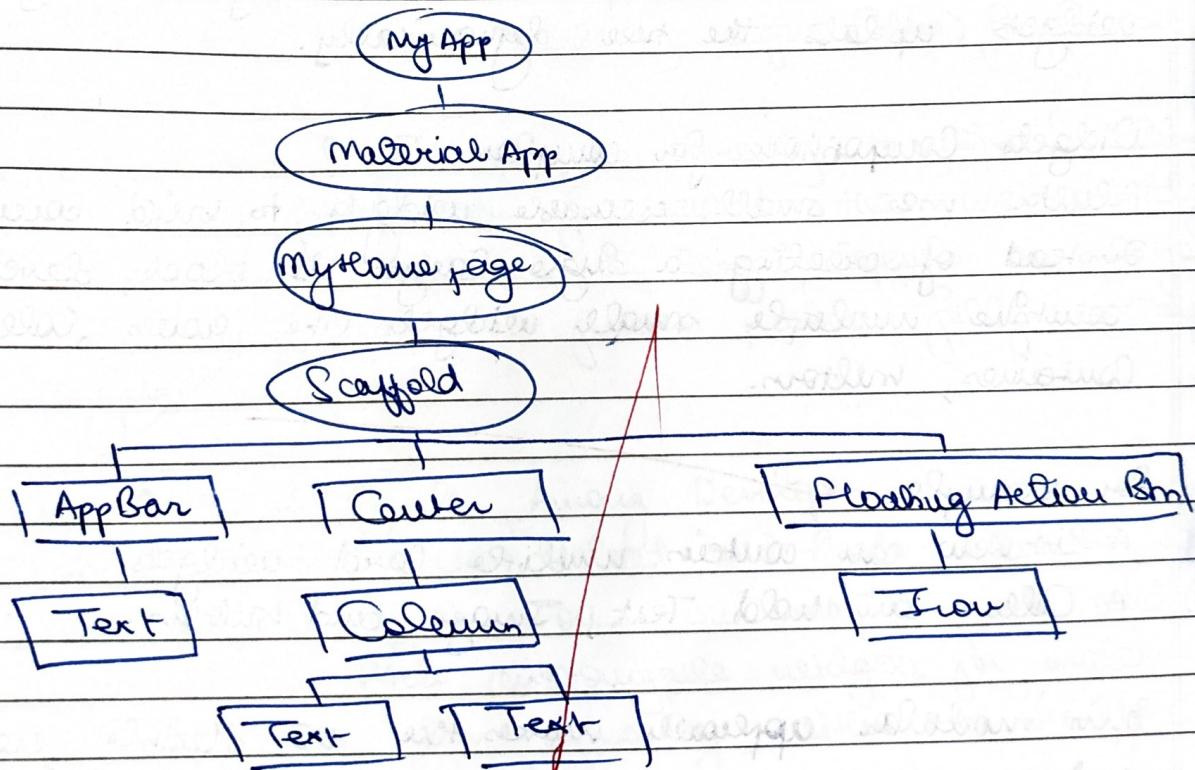
P.T.O



- b) Provide examples of commonly used widgets and their roles in creating a widgets tree.

Soln: Commonly Used Widgets and Their Roles in a Widget tree

- 1) Scaffold: Provides the basic layout structure (AppBar, body)
- 2) AppBar: Displays the top navigation bar with a title
- 3) Text: Displays simple text or a well
- 4) Image: Shows Images from assets or URLs
- 5) Container: Used for styling (Bg color, padding, margin)
- 6) Row: Arranges child widgets horizontally.



Discuss the importance of state management in Flutter application.

Importance of state management in Flutter Application

State management is important because it controls how the app stores, updates and displays data when user interacts with it.

Why State Management is Needed?

Keeps UI Updated - Ensures that the app reflects change (e.g. button clicks, text inputs)

Improves Performance - Updates only necessary parts of the UI instead of reloading everything.

Manages complex data - Helps handle user inputs, API data and navigation efficiently.

Types of state in Flutter:

- i) Local State - Managed within a single widget using StatefulWidget
- ii) Global State - Shared across multiple screens using Provider, Riverpod, Bloc or Redux.

Compare and contrast the different state management approaches available in Flutter such as `useState`, `Provider` and `Riverpod`. Provide scenarios where each approach is suitable.

P.T.O



Soln

Approach

How it Works

When to Use

setState

Updates UI by calling
setState() in a
statefulWidget

Best for small apps
managing state within
single widget ex: Top
bar

Provider

Uses InheritedWidget
to share state across
widgets efficiently

Suitable for medium
apps where data needs
to be shared b/w
multiple widgets. Ex:
Managing user authentia-

Riverpod

An improved version of Provider with better
performance and simpler management with
signals.

Injection ex: Handling
data and app-wide
themes.

Choosing the Right Approach

- Use setState for simple UI updates
- Use Provider for moderate scale sharing across widgets
- Use Riverpod for scalably well-architected applications

(Q. 4) (a)

Explain the process of integrating Firebase with flutter application. Discuss the benefits using Firebase backend soln.

Process of Integrating Firebase with Flutter Application

1. Create a Firebase project - Go to [Firebase Console]

2. <https://console.firebaseio.google.com/>, create a new project.

3. Add Firebase to flutter app - Register the app and download the google services.json or GoogleService-info.plist

4. Install Firebase Packages - Add dependencies like

firebase-core and firebase-auth in pubspec.yaml

5. Initialize Firebase - Import Firebase in main.dart and call `Firebase.initializeApp()`.

6. Use Firebase Services - Implement authentication, database or cloud functions as needed.

7. Benefits of using Firebase as a Backend Solution

8. Real-time database - Syncs data instantly across devices

9. Authentication - Provides ready-to-use sign-in options (Google, Email etc)

Cloud Firestore - Stores structured data efficiently

Scalability - Handles large user bases without managing servers.

10. Highlight the Firebase services commonly used in flutter development and provide a brief overview of how data synchronization is achieved.

Common Firebase Services Used in flutter Development

- 1) Firebase Authentication - Provides user sign in methods (Google, Email, Facebook, etc)
- 2) Cloud Firestore: A NoSQL database that stores and organizes data.
- 3) Firebase Realtime Database - Stores and updates data instantly across all connected devices
- 4) Firebase Cloud Messaging (FCM) - sends push notifications to users
- 5) Firebase Hosting - Deploy web apps with fast and secure hosting.

How Data Synchronization is Achieved

- 1) Real-time Updates - Firestore and Realtime Database sync data across devices
 - 2) Listeners & Stream - Widgets listen for changes and update the UI automatically
 - 3) Offline Support - Firebase caches data, allowing apps to work offline & sync when online.
- 