Nishtha Joshi
D15B 25
MAD Assignment-1

Q1] a) Explain the key features and advantages of using Flutter for mobile app development.

→ 1] Single codebase :
- Flutter allows developers to write a single codebase for both IOS and Android platforms, reducing development time and effort.

2] Hot Reload :
- One of the standout features of Flutter is hot reload, which enables developers to instantly see the changes.

3] Rich widget library :
- Flutter comes with an interesting set of customizable widgets that allow developers to create complex UIs.

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

→ UI Rendering :-
Traditional approaches often use a native

UI toolkit for each platform, resulting in different UIs and behaviour for IOS and Android.

2] Development Speed:
The hot reload feature in Flutter significantly speeds up development by allowing developers to see changes in real-time.

3] Code Reusability:
Flutter's single codebase allows for a high degree of code reusability.

Q2] a) Describe the concept of the widget tree in flutter. Explain how widget composition is used to build complex user interfaces.

→ Widget Tree in Flutter: In flutter, the widget tree is a hierarchial structure where each component.

Widget composition:
Widget composition is the process of combining simple widgets to create more complex user interfaces.

Parent widgets: These are widgets that contain other widgets. They define structure and layout of the UI.

child widgets: These are the widgets placed inside parent widgets.

Q2] b) Provide examples of commonly used widgets and their roles in creating a widget tree.

→ 1) Container :
Role : A basic box model that can contain other widgets. It is often used for layout purposes, such as padding, margin and decoration.

```
Container (
padding : EdgeInsets. all (16,0),
margin : EdgeInsets. symmetric (vertical : 8.0),
 decoration : BoxDecoration (
 color: Colors. Blue,
 borderRadius: BorderRadius. circular (10.0),
),
child : Text (' Hello, Flutter !'),
)
```

2) Row and column :
Role : These widgets allow you to arrange child widgets horizontally ('Row') or vertically ('column'). They are fundamental for structuring the layout.

```
Row (
 children [
 Icon ( Icons . star),
 Text (' 5 stars'),
 ],
)
```

Q3] a) Importance of state management in Flutter applications.

→ State management is a crucial aspect of Flutter development, and it involves handling the data and UI state of an application.

1) <u>User Interface changes</u> :
- Flutter applications often need to respond to user interactions such as button clicks, form submissions, etc.

2) <u>Reactivity</u> :
- User interfaces need to react to changes in data or external events. Efficient state management allows developers to update the UI in real-time when the underlying data changes, provides a smooth and responsive user Experience.

3) <u>Maintainability</u> :
- well-organized state management helps in writing maintainable code.

4) <u>Code separation</u> :
- Separating UI logic from business logic is a good practice.

Q3] b) Compare and contrast the different state management approaches available in Flutter such as setstate, Provider and Riverpod. Provide scenarios where each approach is suitable.

→ setstate :

Pros :- simple and built in.
- suitable for small apps and quick prototyping.

cons :- limited to widget's subtree.
- can lead to code duplication.

Suitability : 'setstate' is suitable for small to medium sized

2] Provider :

The provider package is a popular state management soln in Flutter.

Pros :
- Easy to use and setup

cons :
might require additional packages.

3] Riverpod :

Riverpod is an extension of the 'provider' package with additional features.

Pros :-
- Based on a simplified syntax and design, making it easy to understand and use.

- cons :
- Requires learning a new API

Scenarios :-

- use 'set state' when :
  - Dealing with simple state changes within a single widget.
  - The application is small and the overhead of other state management sol^n is unnecessary.

- use 'Provider' when :
- Building medium sized applications.
- Dependency injection is a key requirement.

- use 'Riverpod' when :-
- Building large and complex applications with many widgets and screens.
- A more modular and scalabe approach is desired.

Q4] a) Explain the process of integrating Firebase with a Flutter application.

→ Integrating Firebase with a flutter app involves creating a Firebase project, registering the app, adding dependencies and initializing Firebase.

- Key benefits include a real time NoSQL database (Firestore), easy authentication, cloud functions and storage.
- Firebase ensures scalability, handles security and provides analytics tool.

Page No.
Date

Nishtha Joshi
[illegible] Roll no: 25

Q4)b) Highlight the Firebase services commonly used in Flutter development and provide a brief overview of how data synchronization is achieved.

→ In Flutter development essential Firebase services include Firestore, Firebase authentication and Firebase cloud Functions. Firestore, a NoSQL database organizes data into collections and documents.

Firebase Authentication ensures user logins with various methods and its real time synchronization enables dynamic responses to user authentication events like logins or logouts.