Dhairyaan Jain
01513
~~2~~ 22

A+ 03

Date
Page

Assignment-1

Q.1) a) Explain the key features and advantages of using flutter for mobile app development

→ Advantages: and Features:
i) Single Database: write once and run on both android and iOS

ii) Fast development: Instantly see changes

iii) UI with customization: Uses its own rendering agent to create highly customizable UI

iv) Performance: Flutter compiles to native code, ensuring smooth animations

v) Widget library: pre-designed material and cupertino widget for android & iOS

b) How flutter differs from traditional approaches and its popularity
→ Traditional mobile development requires seperate codebase for android and iOS, which increases complexity

Flutter differs by:
i) Using widgets insted of native UI components Everything in flutter is a widget making UI development consistent across platform

Teacher's Sign.:

ii) Rendering Engine: Flutter draws everything using directly using ~~8~~ Skia, boosting performance

iii) Faster development with hot reload: No need to restart the app after making changes

Also flutter reduces development effort with a single code base, high performance & rich UI customization.

Q.2
a) Describe the concept of the widget tree in flutter. Explain how widget composition is used to build complex UI.

-> In flutter everything is a widget. Widgets are organized in a hierarchical widget tree to structure UI.

Widget compositions:
Flutter follows a composition-based approach where UI is built by nesting widgets inside widgets.
Complex UIs are built using simple widget combined together

Example:

```
class MyApp extends StatelessWidget {
  @override
  Widget build(Bulild context) {
     return MaterialApp (
        home: Scaffold (
```

app

Teacher's Sign.:

```
appBar: AppBar(title: Text('Hello')),
body: Center(
   child: Colum(
      children: [
         Text('Hello')
          ElevatedButton(onPressed: ()
      ]
   )
  )
 )
)
}
}
```

b) Provide examples of commonly used widget and their roles in creating a widget tree

-> Commonly used widgets are:

i) Structural widgets:
Container, Column, Row, stack

ii) Interactive widgets:
Elevated button, text field,

iii) ~~Aligning and~~ Styling & Display widgets:
Text, image, card, icon

iv) Stateful widgets:
Stateful widget.

Roles:

Strutural -> Define UI

Interactive -> Capture input

Styling -> Enhance UI

Stateful -> maintain states.

Teacher's Sign.:

Date
Page

Q.3
a) Discuss the importance of state management
-> State management controls UI updates.
Without it UI would not reflect real-time updates (eg. updating text)
Good state management ensures
- improved maintainability
- efficient UI updates
- scalability

b) Compare and contrast the different state management approaches available in flutter such as setState, Provider and Riverpod. provide scenarios
-> i) ~~sett~~ setState (built-in local state management)
~~Built~~ Best for simple state changes

ii) Provides:
Efficient state handling by rebuilding only necessary parts of UI
Best suited for medium sized apps

iii) Riverpod (advanced):
Eliminates dependency injection boiler plate and provider. Ensures better testability and seperation of concerns
Suited for large scale applications

Teacher's Sign.:

Comparison

setState	Provider	Riverpod
Simple, builtin	efficient UI rebuild	Scalable, testable
efficient	requires addition setup	learning curve
Small, local change	Medium size apps	large scale app.

Q4)

a) Explain process of integrating firebase with a flutter application. Discuss the benefits

→ Steps:

i) create a firebase project
ii) add an iOS and/android app and download the json file
iii) Install firebase packages in pubspec.yml
iv) Initialize firebase
v) Use ~~fibe~~ firebase service.

Benefits:
- No need for managing server
- ~~xx~~ Real time database synchronization
- scalable and secure auth
- Integrate with other goggle services.

Q.4)
b) Highlights the firebase services commonly used in flutter development and provide a brief overview of how data synchronization is achieved.
-) Common firebase services
i) firebase authentication: Manages user login
ii) Cloud firestor - NoSQL real-time database for storing and syncing structured data
iii) Firebase storage: store images, videos and files
iv) Firebase messaging - sends notification
v) Analytics - tracks user interaction and app usage

Data synchronization
example:
~~STR~~
```
StreamBuilder (
    stream: FirebaseFireStore,
    builder: (context, snapshot) {
                if (snapshot.hasData)
                    return var message=snapshot.data
                return
          },
      };
  }
```

Teacher's Sign.: ____________