Name: Urvashi Changlani
D15B/04

MAD - Assignment 1

Date
Page 04

A1

1) Explain the key features and advantages of using flutter for mobile app development.
Discuss how the flutter framework differs from traditional approaches and why it has gained popularity in the developer community.

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 codebase.

Key features of flutter :-
1) Single codebase: Write one, run on multiple platform (iOS, Android web desktop).
2) Dart Programming language: Uses Dart, which is optimized for fast performance and ahead of time (AOT) 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 supports styles for a native look and feel.

Advantages of using flutter:
Flutter - Faster development time: Hot reload and a single codebase reduce development effort and time.

Teacher's Sign.:

(2) Cost effective : Since developers write one codebase for multiple platforms, it reduces costs associated with maintaining separate teams for iOS & Android.
(3) Consistent UI : Flutter sends everything to the own engine, ensuring a uniform look across devices.

→ Q1b. Flutter uses a single codebase for multiple platforms, unlike traditional native development that requires separate code for iOS (Swift) & Android (Kotlin). It does not rely on platform-specific UI components but instead renders everything using its own Skia graphic engine, ensuring consistency. Unlike React native, which uses a Javascript bridge, flutter compiles directly to native ARM code, offering better performance. Its hot reload feature allows developers to see changes instantly making development faster & more efficient.

Flutter has gained popularity due to its fast development cost efficiency & cross-platform support. Business prefer it as it reduces development apps. Its customizable widget system ensures a smooth native like experience.

Q2 a Describe the concept of widget tree in flutter. Explain the widget composition is used to build complex UI

Teacher's Sign.: _____

In flutter everything is a widget (button, text, layout etc) these widgets are arranged in hierarchical structure known as the widget tree. The widget 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 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
eg A column widget can hold multiple text & button widget, creating a structured layout.

b. Provide ex of commonly used widgets & their roles in creating a widget tree.

1) Structural widget:
Scaffold: Provide basic structure of a screen
Container: Used for layout styling.
Column & Row: Used for vertical & horizontal layout.

2) Interactive widget
- Text field for user input.

Teacher's Sign.:

Elevated Button : clickable Buttons.

(3) Styling widget:
Padding : Adds spacing around widget.
Align centre - Adjust alignment.

(4) List : Scrollable widget.
List view : Scrollable list
Grid view : Provide / display items in Grid.

ex Simple widget Tree.

```
Scaffold {
app Bar : AppBar (Title : Text ("Flutter App")),
  body : column (
    children : [
      text ("Welcome to flutter")
      Elevated Button (On Pressed () {} child
        Text ("click me")
    ),
  ),
);
```

Q3a 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 over In flutter, the UI rebuilds when the state changes ensuring the app remains interactive

Teacher's Sign.: _____

and responsive proper state management helps in performance optimization code maintainability & better UI behaviour.

compare & contrast the different state management in flutter approaches available in flutter, such as Set State, provider & riverpod provide scenarios where each approach is suitable.

Comparison of state management approaches in flutter approach description suitable Set State Basic state management by calling setState() to update UI small apps, simple UI updates (eg tagging a switch) Provides use inherited widget to ~~efficiently~~ manage state across the widget tree. Medium-sized apps needing global state showing (eg, user authentication). Riverpod More Scalable that provides with improved dependency injection & state handling, larger, complex apps requiring modular & Scalable state management (eg e-commerce apps).

a 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 benefits
Integration Process.
Setup Firebase console.

Teacher's Sign.:

Register the App for Android & iOS.
Download & app google-services.json (Android) or
Install firebase dependencies in yaml.
dependencies:
firebase_core: latest_version
firebase_auth: latest_version
cloud_firestore: latest_version
Initialize firebase in flutter.

dart
void main() async {
  widgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(MyApp());
}

3.

Benefits :-
No need to manage servers (Backend-as-a-service)
Provide authentication database & cloud functions
Scalable & cost-effective.

4 Highlight the firebase services commonly used in flutter development & provide brief overview of how data synchronization is achieved

→ Commonly used firebase services in flutter for data synchronization service functionality.
firebase authentication NoSQL database for realtime data syncing. firebase storage to upload & manage files (images, videos) cloud messaging push notifications, firebase

Teacher's Sign.:

Analytics app usage analytics.

a. Data Synchronization in firebase:
Firebase allows real-time data syncing using snapshot listner.

eg of real-time listener in firebase:
dart

```
firebase firestore instance collection('message').
  snapshots()
  listen((snapshot) {
      for (var doc in snapshot.docs) {
          print(doc.data());
  }
});
```