Deepa Bohrani
D15B - 05

MPL ASSIGNMENT 1

AH 04

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

→ Key Features of flutter:

(1) Single codebase: flutter allows developers to write a single codebase for both android and iOS reducing development effort.

(2) Fast development with hot reload: This feature enables developers to see changes instantly without restarting the entire app.

(3) UI with widgets: flutter provides customizable and pre-designed widgets to get visually appealing user interface.

(4) High performance: Uses the dart programming language which compiles into native ARM code for optimized performance

(5) Support for web and desktop: Besides mobile apps, flutter also supports web and desktop development.

(6) Access to Native features: Developers can use platform-specific APIs & third party plugins to integrate native functionalities

(7) Open Source: free and open source allowing developers to customize and enhance to framework.

(8) Backed by google: Ensures long-term support and continuous improvements.

ADVANTAGES OF USING FLUTTER

- Faster development due to code reusability and reloads
- Cost efficiency as a single codebase reduces development and maintenance costs
- Consistent UI across multiple platforms.
- Better performance compared to hybrid frameworks as it compiles directly to native code.
- Strong integration with firebase for backend services like authentication & database management.

Q1(b) Discuss how the flutter framework differs from tra approaches and why it has gained popularity in the de community?

→ Flutter uses single codebases for multiple platforms traditional native development that requires separate ios (swift) & android (kotlin). It doesnot rely on platfo specific UI components but instead render everything its own skia graphic engine, ensuring consistency. Unl React native, which users a Javascript bridge, flutter compiles directly to native ARM code, offering better pe ce. Its hot reload feature allows developer to see changes instantly, making development faster & mor efficient.

Flutter has gained popularity due to its faster dev cost efficiency, & cross platform support. Business p it as it reduces development time & while delive ensures a smooth native like experience.

Q2 a) Describe the concept of the widget tree in Flutte Explain the widget composition is used to build com UI.

→ In Flutter, everything is a widget (button, text, layouts These widgets are arranged in a hierarchical structu known as the widget tree. The widget tree determ the UI.

Widget Composition to build complex UI:

- Flutter encourages a composition-based approach ra than inheritance
- Instead of creating large, monolithic widget, develop

build small, reusable widget that are combined to form complex UIs.

ex: A column widget can hold multiple text & Button widget, creating a structured layout.

Provide example 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

⇒ TextField: For user input

⇒ ElevetedButton: Clickable buttons.

3) Styling widget

⇒ Padding :- Adds spacing around widget

⇒ Align, centre :- Adjust alignment.

4) List & scrollable widget

⇒ ListView :- Scrollable list

⇒ GridView :- Provide / Display items in Grid.

ex: Simple widget Tree

```
Scaffold (
appBar : AppBar (title : Text ("Flutter App"));
body : Column (
  children : [
  Text ("Welcome to flutter!"),
```

ElevatedButton (onPressed: () {}, child: Text ("Click")
),
),
);

Q3 a) Discuss the importance of state management in flu application.

⟹ Importance of state management in Flutter Applicati
state management refers to handling dynamic data changes overtime
In flutter, the UI rebuilds when the state changes, ensuring the app remains interactive & responsive. proper state management helps in performance optimization code maintainability & better UI beha
Some key reasons why state management is importan flutter:

1] Ensures Data consistency - Flutter applications often dynamic UI updates based on user interactions.

2] Improves Performance - Efficient state management h reduce unnecessary widget rebuilds, improving ap performance.

3] Enhances Code Maintainability - With a structur state management helps reduce unnecessary widge rebuilds, improving app performance.

4] Manages Complex UI interactions:- In complex ap ns, multiple widgets need to communicate and data.

5] Reduces code Duplication: Using state management eliminates redundant logic & allows reusable co

Compare and constrast the different state management in flutter approaches available in flutter such as setstate, Provider, & Riverpod. Provide scenarios where each approach is suitable.

→ comparison of state management Approaches in flutter

Approache Description suitable Scenarios setstate Basic state management by calling setstate() to update UI. Small apps, simple UI updates (eg. toggling a switch) Provider user inherited widget to efficiently manage state across the widget tree. Medium sized apps needing global state sharing (eg user authentication) Riverpodmore scalable than provider with improved dependency injection & state handling, large complex apps requiring modular & scalable state management (eg: e-commerce apps)

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:

Create a firebase project

Register the App for android & ios

Download & add google-services.json (Android) or Google Service Info.plist (iOS)

Install firebase Dependencies

yaml

dependencies :
Firebase_core : latest_version
firebase_auth : latest_version
cloud_firestore : latest_version
Initialize firebase in flutter.

dart

void main() async {
widgets flutter Binding ensure Initialized();
await firebase.initialize App();
runApp (MyApp());
}

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

(CHb) Highlight the firebase service commonly used in flu development & provide brief overview of how data synchronization is achieved.

⟹ Commonly used firebase services in flutter & Data Synchronization service functionality
Firebase Authentication user sign-in (email, google facebook) cloud firebase storage upload & manage files (images, videos) cloud messaging push notifica

Firebase, Analytics App usage analytics

Data Synchronization in firestore:
firestore allows real time data syncing using snanshot listner.

ex: of real time listner in firestore:
dart.

~~firebase~~

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