

03/02/24

Assignment 1.

Q.1.

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

→

- Flutter is an open source UI software development toolkit created by Google for building natively compiled applications for mobile, web, and desktop from a single codebase.

Key features of flutter

- Easy learning curve :- Learning the dart programming language that flutter uses is pretty easy. Developers with minimum coding knowledge can easily develop apps.
- Hot Reload :- crafting interactive and captivating UIs, incorporating great in-app features.
- Rich Widgets
- Single Code Base :- Flutter needs a single code base to be written by the developer.

Advantages of flutter:

1. Fast
2. Creates cross platform application.

3. Open source
4. Easy debugging
5. Automated testing.

b. Discusses how the flutter framework differs from traditional approaches and why it has gained popularity in the developer's community.



i. Single codebase for multiple Platform

- Traditional Approach : Separate codebase required for each platform.

In flutter a single codebase can be used to develop apps for iOS and Android.

ii. Hot Reload

- Traditional Approach : Recompiling and redeploying is time consuming.

Flutter instantly see the effects of code changes without restarting.

iii. Rich Animation Library :

- Traditional approach : Implementing complex animation is difficult.

Flutter powerful animation library for creating intricate animations.

iv. Widget - Based UI development.

Q2:

- a: Describe the concept of the widget tree in flutter. Explain how widget is used to build complete interaction.

→

- The widget tree is a hierarchical structure of widgets that represents the structure and appearance of a flutter app.
- This tree is created to and managed by flutter to efficiently and dynamically update the UI in response to changes.
- Widgets as basic building blocks

Hierarchical structure

Reusable and modular code

Dynamic UI update

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

→

- i) container : The 'Container' widget is a base model that can contain other widgets.

example : Container (

width : 100,

height : 100,

color : Color.blue

child : Text ('Hello');

)

ii Row & column : These widgets allow you to arrange child widgets horizontally or vertically

Example : Row (

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

Q.3: Discuss the importance of state management in flutter applications.

- Reactive and UI updates : Properly organized state management ensure that when data changes, the UI is automatically updated to reflect these changes.
- Performance optimization : Flutter allows developer to optimize performance by rebuilding only the widgets that depend on changed style.
- Maintainability and code organisation : Properly organized state managed allows developers to separate concerns.
- User Input Handling : Many application rely on user interaction such as button presses and gestures.

- b. compare setState , Provider , and Riverpod
Provider scenarios.



setState	Provider	Riverpod
• simple and built-in	Relatively easy	simpler to provider
• Logic can be mixed with UI code	Encourages separation through providers	Promotes clear separation and modularity.
• Not built for dependency	Built in DI	strong support for dependency injection
• limited global access	global access through centralized provider	global access with focus on modularity.

Scenarios for using :

- a. setState : Suitable for small apps with a limited number of widgets and simple state management.
- b. Provider : Well suited for medium sized app where centralized state is needed.
- c. Riverpod : Ideal for large and complex application where modularity, dependency injection are essential.

Q.4.

- a. Explain the process of integrating firebase with flutter application. Discuss the benefits of using firebase as backend solution.

→

Steps of integrating firebase:

- i Create firebase project.
- ii Register ur app with fb
- iii download ard. add config files
- iv Add dependencies
- v Initialize firebase in your app
- vi Use firebase service in your app.

Benefits of using firebase as backend:

- i Realtime database
- ii Authentication
- iii Cloud functions
- iv Cloud firestore
- v Hosting
- vi Auth & security
- vii Analytics
- viii Scalability & Reliability

- b. Highlight the firebase security services commonly used in flutter development and provide a brief overview of data synchronization



In flutter development, Firebase services commonly used include:

- i. **Firebase Authentication** :- Provides backend services and ready made libraries to authenticate users.
- ii. **Cloud Firestore** : A flexible database for web, mobile app.
- iii. **Firebase Realtime Database** : A cloud based NoSQL database that stores and sync data.
- iv. **Firebase Storage** :- Cost effective object storage service that lets secure storage.
- v. **Firebase Analytics** :- Helps to understand user behaviour, measure and grow your app.

Data Synchronization : It is achieved through realtime listeners and the firebase Realtime Database firebase sends data updates to you app in realtime. Offline support is provided.