Niyati .V Gaonkar
D15B
Roll no: 17.

MAD ASSIGNMENT - 1

Q1 Explain the features and advantages of using flutter for app development.

Ans → Flutter is a cross platform UI toolkit developed by google for building native compiled language for mobile, web, desktop

Features:

i) Hot reload : Enables developers to instantly view changes without restarting the app.
ii) Widget - based architecture : UI components in flutter are widgets making the development modular and customized.
iii) Expressive UI : Flutter provides a rich set of customisable widgets for creating visually appealing interfaces.
iv) Single codebase : Develop once, deploy everywhere.
v) Strong community support : A large and active community supports in the development and updation of this open source project.

b) Discuss how flutter framework differs from other traditional approaches.

→ i) Flutter uses a reactive framework whereas others use imperative
ii) Flutter offers a consistent UI across platforms ensuring a native look
iii) It uses Dart language for coding.
iv) Widget based approach enhances developer productivity.
v) Popularity arises from the efficient development process, performance and vibrant community.

Q2.

a) Describe widget - tree in flutter. Explain how widget competition is used to build complex UI.

→ i) In flutter, the widget is a fundamental concept that represents the heirarchy of a user interface elements in an application. Everything in flutter is a widget. Wheather it is a button, text, image or even the entire application
ii) The widget tree is composed of various types of widgets each serving a different purpose. They are of two types: stateless and stateful.

iii) Stateless widgets are immutable. stateful widgets can change their internal state during their lifetime.

b) Provide examples of commonly used widgets.

Ans → i) Material App: Define basic structure of the app
ii) Scaffold: Represents the basic visual structure of the app including the app bar and body.
iii) Container: A box model can contain other widgets, providing layout and styling.
iv) Rows and columns: Arrange child widgets horizontally or vertically.
v) List View: Displays a scrolling list of widgets

Q3
a) Discuss the importance of statemanagement in flutter

Ans → State management is the crucial aspect of building robust and efficient flutter application. In flutter, 'state' refers to the data that influences the appearance and behaviour of widgets. Managing state is important for creating responsive, dynamic and scalable applications.

i) User Interface updates
ii) Performance optimization
iii) Code maintainability
iv) Reusability and modularity
v) Persistance and navigation
vi) Stateful widgets limitations.
vii) Concurrency and asynchronous operations.

b) Compare and contrast the different state management approaches available in flutter such as setState, provide and river

Ans →

i) setState :

Pros :

- Simplicity: 'setState' is the most straight forward way to manage state in flutter. It is built into the framework and is easy to understand for beginners.
- Simple UI design : Simple and strait forward code for designing UI. where state changes are localized.

Cons :

- Limited to the widget tree.
- Over rebuilding widgets.

ii) Provider

Pros:

- Scoped State management
  Allows scoped and localized state management reducing the need for prop drilling
- Easy integration
  It is easy to integrate into flutter apps. between simplicity and flexibility.
- Large community support.

Cons:

- Learning Curve: not so Easy to learn.
- Global Scope:

iii) Riverpod

Pros:

- Scoped and flexible
- Provider Inheritance
- Immutable and reactive

Cons:

- Learning curve is difficult.
- Advanced features.

Usecase:

Large and complex applications.

Q4

a). Explain the process of integration of firebase with a flutter app. Discuss the benifits of using flutter.

i) Create a firebase project.
- Goto firebase website
- login and get started
- Create an new project.

~~restart~~

ii) Add firebase to flutter project
- In flutter project, add firebase SDK and dependencies in '.yaml' file.

iii) Initialize firebase.
- Import the firebase packages and initialize firebase in 'main.dart' file.

iv) Configure firebase services:
Start firebase services like authentication, fireshore etc.

v) Use these services in the app.

Benifits of using firebase
1. Real time database
2. Authentication
3. cloud function
4. Cloud fireshore
5. Firebase Storage.
6. Hosting and analytics
7. Easy setup and integration.

b) Highlight the firebase services commonly used in flutter development and provide a breif overview of how data synchronization is achieved.

Ans → Common Firebase services in Flutter Development are:

i) Authentication : Firebase Authentication for user signin
ii) Firestore : NoSQL database
iii) Firebase Cloud messaging (FCM)

Data Synchronization

i) Listeners and streams : Firebase services use listeners and streams extensively. Flutter developers can use stream-based APIs.
ii) Reactively updating UI : Flutter 'StreamBuilder' widget is commonly used to reactively update UI components based on the changes.
iii) Offline Support : Firebase services provide built in offline support. Flutter apps can work seamlessly.