Mohit Shadija
D15B
50

A+ 03

MPL Assignment 1

Q.1 a] Explain Key features and advantages of using flutter for Mobile App development.

⇒ Key features of Flutter :-
1) Single Codebase :- Write one code for both Android & ios.
2) Fast Performance :- Uses Dart language and a high Performance rendering engine
3) Hot Reload :- See changes instantly without restarting the app.
4) Rich UI components :- Comes with Customizable widgets for smooth UI design
5) Native like Experience :- Provides high quality animations & fast execution.

Advantages of using flutter :-
1] Saves & Effort :- Single Codebase for multiple platforms
2] Cost-Effective :- Reduces development Cost & time
3] High speed development :- Hot Reload feature speeds up coding.
4] Attractive UI :- Provides beautiful & Customizable widgets.
5] Good Performance :- Use Dart and Skia for fast & smooth rendering.

b] Discuss how the flutter framework differs from traditional approaches and why it has gained Popularity in developer community.

⇒ How Flutter Differs from Traditional Approaches :-

1] Single Codebase :- Traditional methods need sperate code for Android & ios. but flutter uses one code for both.

2) Hot Reload :- Traditional Apps require full restart after changes, but flutter updates instantly.

3] UI Rendering :- Traditional apps use native components, while flutter has its own rendering engine for faster performance.

4] Performance :- Flutter compiles directly to native machine code, making it faster than framework that uses a bridge.

Why flutter is Popular Among Developers:

1) Fast Devellopment :- Hot Reload & Single Codebase Save time.

2] Cross - Platform Support :- Works on mobile, web & desktop.

3] Beautiful UI :- Rich, Customizable widgets for modern design.

4] High Performance :- Runs smoothly without a bridge like React Native.

Q2]

a] Describe Concept of the widget tree in Flutter Explain how widget Composition is used to build Complex user into interface.

Concept of Widget Tree in Flutter :-
In flutter everything is a widget. ~~are~~ Widgets are arranged in a tree structure, called widget tree. This tree ~~itself~~ represents UI of the app, where parent widget contain child widget.

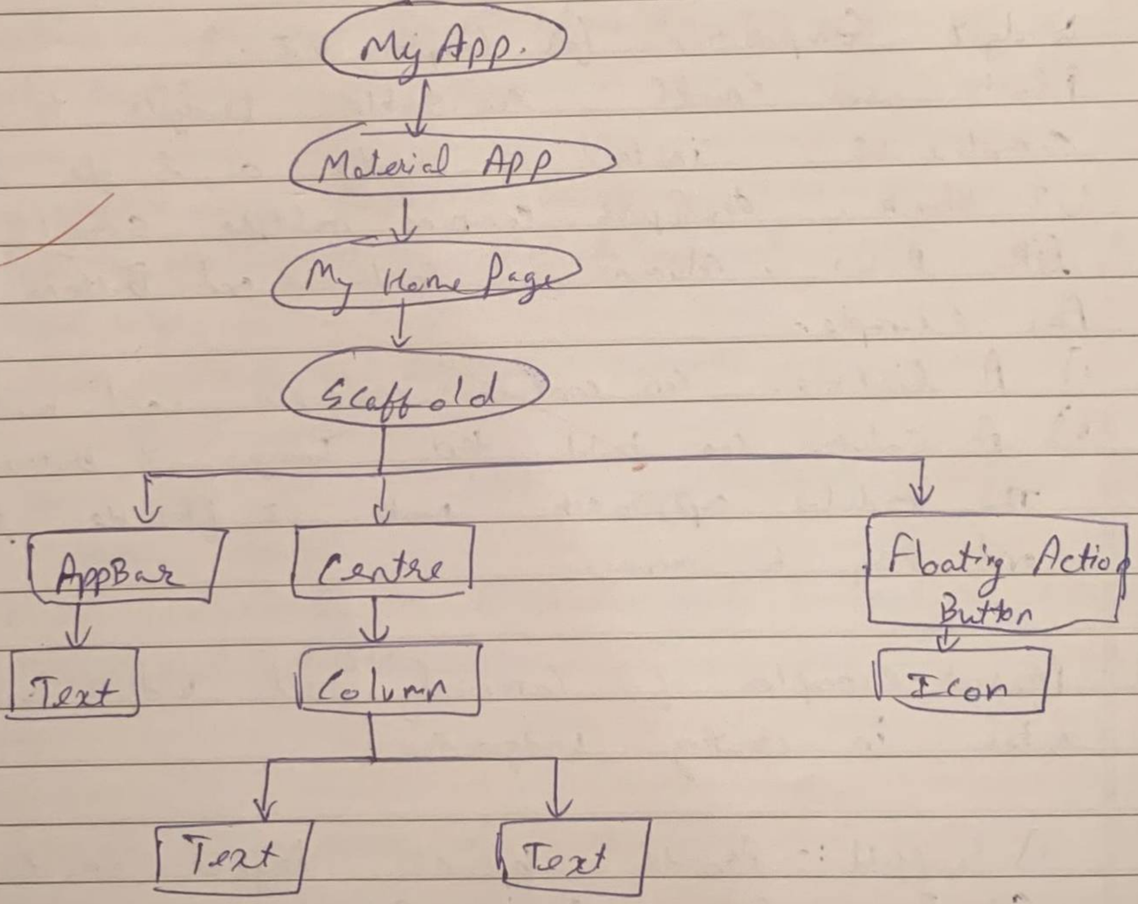
For example, a Scaffold widget can have a Column widget, which contains Text and button widgets.

Widget Composition for Complex UI :-
Flutter uses small reusable widgets to build complex UI, instead of creating a single large UI block, developers combine multiple small widgets like Rows, Columns, Containers and Buttons
For example:-
1) A listview can contain multiple card widgets.
2) A Column can hold text, Images & buttons.
The modular approach make UI flexible, readable and easy to manage.

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

⇒ 1) Scaffold :- Provides basics layout structure (App Bar, Body, Floating Button)
2) AppBar :- Displays top navigation bar with title.
3) Text :- Displays simple text on the screen

4] Image :- shows images from assets on URL's.
5) Container :- used for Styling (Background color, padding, margin)
6) Row :- Arranges child widgets horizontally
7) Column :- Arranges child widgets vertically
8] list view :- Displays ~~scr~~ scrollable lists.
9] Elevated Button :- A clickable button with elivation
10] Text field :- used for user input
Example widget Tree :-



Q3] a] Discuss the importance of State management in Flutter application.

→ Importance of State Management in Flutter Application :-

State Management is ~~impont~~ important because it controls how the app stores, updates and displays data when the user interacts with it.

Why state Management is Needed?
1] Keeps UI updated:- Ensures that the app reflects changes (eg: button clicks, text inputs)
2] Improves Performance :- Update only necessary parts of UI instead of reloading everything.
3) Manages Complex data :- Helps handle user Inputs, API data and navigation efficiently.
4) Ensure smooth user ~~Ex~~ Experience :- Keeps the app responsive & interactive.

Types of state in Flutter :-
1] local state :- Managed within a single widget using stateful ~~w~~ widget.
2] Global state :- Shared across multiple screens using Provider, Riverpod, Bloc or Redux.

b] Compare ~~&~~ and contrast the different state management approaches available in Flutter such as set state, Provider & River pod.

FOR EDUCATIONAL USE

Sundaram

⇒ Approach	How it works	when to use.
Setstate	Updates UI by calling Setstate() in a stateful Widget.	Best for small Apps or managing state within a Single widget. Example:- Togging abutton color.
Provider	Uses Inherited widget to share state across widgets efficiently	Suitable for medium sized apps where data needs to be shared between multple widgets. Example:- Managing user Authentication.
Riverpod.	An improved Version of provider with better performance & simpler Syntax	Best for large Apps that need complex state management with dependency injection Example :- Handling API data & App-wide themes

Choosing Right Approach:-
- Use Setstate for simple UI updates
- Use Provider for moderate state sharing across widgets.
- use Rivapod for scalable, Well-structured Applications.

Q4] a) Explain the Process of integrating Firebase with Flutter application, Discuss the benifits of using Firebase as a backend solution

⇒ Process of integrating Firebase with a Flutter Application :-

1] Create a Firebase Project :- Go to [Firebase Console] (https://console.firebase.google.com/)
create a new Project.
2] Add Firebase to Flutter App :- Register the App. (Add Android / ios) and ~~development~~ download the google-services.json or Google service-Info.plist (ios)
3] Install Firebase Packages :- Add dependencies like 'firebase-core' and 'firebase-auth' in 'Pubspec.yml'.
4] Initialize Firebase
5] Use ~~I~~ Firebase Services :- Implement authentication database, or cloud functions as needed.

Benefits of Using Firebase as a Backend solution :-
1] Real-time Database :- Syncs data instantly across devices.
2] Authentication :- Provides ready to use Sign-in options (Google, Email, etc)
3] cloud Firestore :- Stores structured data efficiently.
4] Hosting & storage :- Hosts web apps & Stores files securely.
5] Scalability :- Handles large user bases without managing servers.

b] Highlight the firebase services Commonly used in Flutter development & provide a brief overview of how data Synchronization is achieved.

⇒ Common firebase Services used in flutter Development:-

1) Firebase Authentication :- Provides user sign-in methods (Google, Email, Facebook)
2) Cloud Firestore :- A NoSQL database that stores & syncs data in real time.
3] Firebase Realtime Database :- Stores & ~~&~~ updates data instantly across all connected devices.
4] Firebase cloud storage :- used for storing & retrieving files like images & videos.
5) Firebase Analytics :- Tracks user behaviour & app performance.

How data Synchronization is Achieved :-
1] Real-time updates :- Firestore & Realtime Database sync data across devices instantly
2] listeners & streams :- Widgets listen for Changes & update the UI autonomically
3] offline support :- Firebase Caches data, allowing apps to work offline & syncs when online.

This ensures fast smooth & automatic data updates in flutter apps