Name: Aastha Jajoo

Division: D15B

Roll no : 23



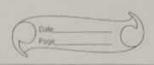
MAD ASSIGNMENT ND. 1

- e.110) Feutter is a popular framework for building vrous-platform mobile applications developed by Google
 - single Codeback for multiple platforms: flutter allows developers to write code once and deploy it on both ios and Android platforms.
 - fast Development and Hot Reload: feutler office a feature called Hot Reload, which allows developers to instantly see the changes they make to the code reflected in the app's interface.
 - so Beautiful and customizable U1: Feutter comes with a nich est of pre-designed widgets that help developers create visually etunning U1.
- High performance: Feutler appe are compiled to directly to native machine code, which eliminates the need for a JavaScript bradge.
 - 5> Strong community and support: Flutter has a vibrant community of developers who actively contribute to its growth and provide support through forums, documentation, and third-party packages.

Teacher's Sign.: __



- (16) feutler differe from traditional approaches to mobile app development in several key ways:
 - unere exparate codebases are negd for ios and Android development.
 - 2> UI sundering: Instead of using notive us components provided by each platform, Futter uses its own set of customizable widgets to render the UI.
 - Ferformance: Flutter appe are compiled directly to native machine code, resulting in high performance and emooth animations.
 - 47 Hot Reload: Flutter's Hot Reload feature allows developers to instantly see changes made to the code reflected in the app's interface without losing app's state.
 - 5> strong community: Hutter has ite strong community support, extensive documentation, and growing exceptem of third-party packages.



Q-2 (a) In Flutter, the widget tree is a hierarchial etructure representing the components of a U.

At the nost of the free is the widget represent the entire ecreen, and each subsequent widget represent expresents a component or element of the U.

euch as buttons, text fields, or containers.

widget composition is the process of combining multiple amaller widgets to create more complex and feature rich all elements. This is achieved by nesting widgets within each other, forming a tree-like structure. These children can further have their own children, we oting a nested hierarch

for example, to create a button with a label and an icon, you would compose it by nesting widgets. The button widget would be the parent, containing a text widget for the label and an icon widget for the icon. This nesting allows for the creation of complex we from eimple building blocks.

6.5 (P)

" Container widget:

Role: It is a versatile widget used to contain other widgets and apply various styling properties such as padding, margine, borders, etc.

Example: A container might serve as the root

example. A container might serve as the root widget of a screen, containing other widgets like text, images, or buttons.

Teacher's Sign .: _



2) column widget:

Aou: it arranges ite children widgets vertically, stacking them from top to bottom.

Example: Within a 'container', a 'column' could be used to organize multiple widgets in a vertical layout, such as a list.

Row Widget

Role: It arranges its children widgets horizontally, placing them side by side.

Example: Inside a 'column' or another 'Rom' a

'kow' widget might be used to display items norizontally, such as now of buttons.

47 enrage widget:

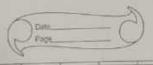
euch as asserts, files.

Example: 3t night be useful to display a logo, icon, or photo within the UI.

5> Yext widget:

Role: It displays a string of text with customizable styling properties such as font size, color, etc. Example: 'Text' widget could be used to display a title, label, or paragraph of text.

Teacher's Sign.: ____



Importance of etate management in flutter applications: ensures that changes in the app's state reflect immediately in the UI. maintainability: Proper state management technique help in organizing and maintaining code, making it easier to understand, debug and modify. se surformance optimization: It ensures that only necessary parts of us are updated when state changes improving app performance. features, nobust etate management becomes crucial for ecalability. o developers to build appe for multiple platforms from a single codebase. 6-3 (P) approach: It is the eimplest form, where the state is stored within the Stateful Widget and updated suitable scenarios: Devitable for state that doesn't need to be shared across multiple widgets or managed globally.



e unful for learning purposes or quick prototyping due to its simplicity. · Approach: 'Provider' offere a way to propagate

data down the widget tree efficiently and manage application state without using 'setState'

Suitable scenarios: O suitable for mid sized to
large applications where multiple widgets need

access to shored detate.

I deal for managing local étate within spécific parte of the widget tree or for inaring étate between eibling widgete.

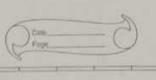
37 Riverpod:

· Approach: "Riverpod builds on top of "Provider and offere additional features like dependency injection, lazy loading and improved testability. suitable scenarios: O 'Ideal for managing

global application state, handling eide effects, and exparating business logic from UI.

a weful for team's working on collaborative projects where modularity, testability and maintainability are crucial.

Teacher's Sign.: _



Q.4(a) create Firebase project - go to Finebase console eneate new project and negister your further app with Firebase by providing package. Add. Finebase SDK - Add Finebase SDK dependencies to your Flutter project by including necessary parkages in your pubspec, yand file. configure Finebase services: - configure Firebase sorbices you want to use, Finebase authentica for wer adthentication, firebase cloud messaging. set up necessary configurations and rules. 4> Initialize Finebase - Initialize Firebase in your futter app by calling Firebase initialize Appl) in your main function. 5) un Finebasi services - one finebase is initialized, you can start using Firebase. services in your Flutter app. Benefite of using Finebase as backend, Real time database - Firebase provides realtime database solutions like finestone, which allow seamless data synchronization. Authentication - Finebase Authentication offere easy- to- use authentication method including making it eimple to implement user autuntication in Jeutler appr. coud functions - Finebase allows you to

extend your app is functionality with cloud scalability - Firebase is scalable platform that can I handle large no . of users and data Teacher's Sign .: _



5) offline support - Firebase offers offline support for firebase and real time database.

Q.4 (b)

Finebase Authentication - Provides user authentication and authorization using email/password, social logins.

database for storing and syncing data b/w cliente in real-time.

3 Finebase kealtime Database: A NOSQL cloud database that synchronizes data in real-time across multiple cliente.

Finebase could messaging: Enables sending push notifications to user's devices.

5) Finebase could Functions: surveiless function that run in response to Finebase events of HTTPS regulate. HTTPS requests.

· Real-Jime Database: Firebase Realtime database synchronizes data across cliente in real time using

Mond Finestore: Firestore uses a more explisticated synchronization mechanism based on a distributed database eystern