Date
Page 1

Name :- Ajinkya Nilesh Jadhav
Div :- D15B Roll no 21

* MAD Assignment - 1 *

Q.1) a) Explain the key features and advantages of using Flutter for mobile app development.

→ 1) Flutter is a free and open-source mobile application development framework created by google.
2) It is used for developing high performance, visually attractive and responsive apps for ios, Android and web platforms.
3) Flutter is based on the Dart programming language and uses the Skia graphics library to render its components.

Key features & Advantages :-
1) Hot reload :- It makes it possible to see the changes in the code instantly reflected on the UI, which speeds up the process to work on the outlook of the application.
2) Cross-platform development :- Flutter enables developers to write code that works on different platforms. which means, two different applications on two devices can use the same codebase eg. samsung & iphone have different OS but the same app will work on both devices.
3) Widget-based Architecture :- UI components in flutter are widgets, making the development modular & customizable.
4) Open source :- Flutter is an open-source platform, is free of cost & has detailed documentation & communities available online.

Teacher's Sign.: _____

Ajinkya Jadhav
D15B 21

Date 2
Page

Advantages :

Q.1) B) Discuss how the Flutter framework differs from traditional approaches and why it has gained popularity in the developer community.

1) Instead of maintaining separate codebases for iOS & Android, Flutter allows developers to write a single codebase that works on both platforms.
2) Flutter uses a reactive programming model, where the UI automatically updates when the underlying data changes.
3) Flutter UI is built using widgets, which are composable and customizable building blocks, which makes it easier to create complex UIs.
4) One popular feature which makes flutter standout is hot reload.
5) Flutter's performance is commendable due to its use of Dart programming language & a compiled, Ahead-of-time (AOT) execution.
6) Flutter is not limited to mobile development; It can be used to build applications for desktop, web and embedded systems.

Teacher's Sign.: _____

Ajinkya Jadhav
D15B 21.

Q.2 a) Describe the concept of the widget tree in Flutter. Explain how widget composition is used to build complex user interfaces.

1) In flutter a widget tree is a hierarchial structure that represents the user interface of an application. It consists of individual elements called widgets, which are the building blocks of UI.
2) The widget tree describes the composition and organization of these widgets, defining how they nest and interact with each other.
3) Widgets are objects in Flutter that define the structural & visual elements of the UI. Widgets are either i) Stateless widgets or ii) Stateful Widget.
4) The widget tree has a hierarchial structure, similar to DOM in web development. At the root of the tree is the top-level widget, typically a materialApp. As we go down the tree, widgets encapsulate & compose other widgets, forming a parent-child relationship.
5) This results in composability, to create more complex UI. eg. Row widget can contain multiple children like text, image & Icon widgets.
6) Stateless widgets are immutable and dont change over time, while stateful widgets can hold mutable state, which is crucial for managing dynamic elements in the UI.
7) Stateful Widgets have lifecycle that includes methods like 'initState', 'build' and 'dispose'.

Teacher's Sign.: _____

Ajinkya Jadhar
D15B 21

Date
Page 4

Q.2 b) Provide examples of commonly used widgets & their roles in creating a widget tree.

1) Container : It is a versatile box model that can contain other widgets, providing padding, margin & decoration. Usually a parent component for many children inside it.

2) Row & Columns : They Arrange child widgets in a horizontal or vertical line.

3) Listview : create scrollable list of widgets.

eg. Listview (
children: [
ListTile (title: Text ('Item 1')),
],
)

4) Stack : They overlap the widgets on top of each other

eg. stack (
children: [
Positioned (
top: 10.0,
left: 10.0,
child: Text ('Top left'),
),
],
)

5) AppBar : Usually Represents the top app bar that typically contains the app's title & navigation.

AppBar (
title: Text ('My App'),
),
),

Teacher's Sign.: _____

Ajinkya Jadhav
D15B 21

Date
Page 5

Q.3 a) Discuss the importance of state management in Flutter Applications.

1) State management is a crucial aspect of building robust and efficient flutter applications. In Flutter, "state" refers to the data that influences the appearance and behaviour of widgets. Managing state effectively is essential for creating responsive, dynamic and scalable applications.

2) Importance:-

i) State management allows for dynamic updates to the user interface based on changes in the application's state.

ii) User Interactions, such as button clicks or text input often lead to changes in the application state.

iii) Proper state management ensures that critical information persists, preventing the loss of data & providing a consistent experience throughout the app.

iv) State management is essential for handling asynchronous operations, such as network requests. State management helps co-ordinate these operations, ensuring the UI reflects correct state

v) State management supports scalability & extensibility as the application grows in complexity.

Teacher's Sign.: _____

Ajinkya Jadhav
D15B 21

Date
Page 6

Q.3 b) Compare and contrast the different state management approaches available in Flutter, such as setState, Provider and Riverpod. Provide scenarios where each approach is suitable.

1) ~~setState : it is a Built-In method in flutter's 'StatefulWidget' class for manging local stat~~
~~Advantage~~ : 1) It is si

setState	Provider	Riverpod.
1) Built in method in 'StatefulWidget' class for managing local state.	1) External package that implifies state management by providing a way to propagate changes through widget tree.	1) Provider based state management sol :- that extends 'Provider' & adds features like global providers & improved testability.
2) Scenario :-		
1) Moderate to large scale applications 2) where simplicity and integration are important.	1) large & complex apps requiring fine grained control over dependencies & state	1) large & complex apps requiring fine grained control over dependencies & state.
3) It is simple to use.	4) It is lightweight & easy to use	4) Provides high level of control & organization.

Teacher's Sign.: _____

Ajinkya Jadhav
D15B 21

Date
Page 7

Q. 4 a) Explain the process of integrating firebase with a flutter application. Discuss the benefits of using firebase as a backend solution.

Steps for integrating firebase with flutter:

1) Create a firebase project:- Go to firebase console & create a new project

2) Add configure firebase in flutter:- Add the firebase configuration files to your flutter project.
Configure Firebase SDK by adding necessary dependencies in your app's pubspec.yaml file

3) Initialize firebase in the main.dart file.

Benefits of using firebase as a backend solution:-

1) Realtime database:- Firebase offers real-time synchronization for databases, enabling instant updates to connected clients.

2) Authentication:- Firebase provides easy to use authentication methods.

3) Cloud firestore:- It is a noSQL document database that scales your app.

4) Cloud storage:- Allows to store for secure & scalable file uploads & downloads supporting images, videos & other media.

5) Analytics & crash reporting:-

6) Cross platform support

Teacher's Sign.: _____

Ajinkya Jadhav
D15B 21.

Page 8

10.4b) Highlight the Firebase services commonly used in Flutter development & provide a brief overview of how data synchronization is achieved.

Common firebase services in flutter development are:-
1) Authentication:- firebase provides authentication services to users.
2) Firestore:- It is a noSQL document database that scales with the app & provides powerful querying.
3) Cloud storage:- It allows for secure & scalable file uploads & downloads, supporting images, videos, etc.
4) Cloud Functions:- Serverless computing with cloud functions allows you to run backend code in response to events triggered by firebase feature.

* Data synchronization:-
1) Firestore database & Realtime database both support real-time data synchronization.
2) When data changes on the server, the changes are immediately pushed to connected clients.
3) In flutter, this is achieved by using streams. Widgets can subscribe to streams of data, and when the data changes in the database, the changes are automatically reflected in the UI.
4) Firebase realtime synchronization is based on WebSocket connections, ensuring efficient & low-latency communication between the client & the server.
5) When a change occurs on the server, it's immediately propagated to all connected clients, providing a seamless and reactive user experience in flutter applications.

Teacher's Sign.: _____