Vivekanand Education Society's
Institute of Technology

Name- Girish Chougule
Class- D15B
Roll No.- 10

Q1 Explain the key feature and advantages of using flutter for mobile App development

A) (i) Single codebase for multiple platforms. With one codebase for both Android and IOS Reducing development effort and maintenance

(ii) Hot Reload - Instantly see changes in the app without Restarting

(iii) Fast Performance - Use the dart language and a compiler approach for smooth and high performance apps

Advantages
1) Faster development
2) Cost Effective
3) Reduce performance issues

[B] Discuss how the flutter framework differs from traditional approaches and why it has gained popularity in the developer community

1) Single codebase vs separate codebase
Traditional Approach : Developers need to write seperate code for Android and ios

2) Rendering Ergine vs Native UI components

Teacher's Sign

Traditional Approach: Relies on platform native UI component which can lead to inconsistent and performance issues Flutter uses the Skia Rendering engine to draw Everything from scratch ensuring a consistent UI across devices

1) Why flutter gained popularity.
1) Fasted development with hot Reload's can instantly UI changes without Restarting the app making the iteration process much quick

2) Cross platform Efficiency
3) consistent UI Across devices since it does not Rely on native components

4) Improved performance - No compilation and direct Access to GPU

---

1) Describe the concept of widget tree in flutter.
Explain how widget composition is used to build complex user interface.

2) In flutter the widget tree is the fundamental structure that Represent the UI of an application

Teacher's Sign.

ElevatedButton(
onPressed: () {
print("Button Pressed");
};
child: Text("submit")
);
};
)

(A) Discuss the importance state managment in flutter Application

→ The flutter state refers to data that can change the appearance of an application this includes user input, UI changes, network changes

THERE are 2 types of states

1) Ephemeral state - small UI specific state that does not affect the whole app

2) App with wide status - Data needed across multiple widgets, importance of state managment

(B) compare and contrast different state managment approaches available in flutter such as set state

Teacher's Sign.:

Set state - local state
Pro - simple built in easy to use
Cons - Not state causes unnecessary Re-build
Best use cases - small update

Provider - App wide state
Pros - lightweight recommend by flutter
Cons - Boilerplate code for nested providers
Best use cases - Medium Scale Apps
authentication themes API data

Q4 Explain the Process of integrating Firebase with a flutter application. Discuss the benefits of using firebase as backend solution

- Fire base provide a powerful backend solution for flutter application offering service like communication Real time database cloud function, storage and more step to integrate

Steps - 1]
Create - a firebase project
- Go to firebase console
- "Add" Project and enter a project name
- configure google analytics if need

widgnt composition refers to building complex UI by combining smaller reusable widgnt. Instead of creating large monolithic UI component into small mergeable widgnt that can be reused

```
class ProfileCard extends StatelessWidget {

  final string name;
  final string component;

ProfileCard ({required this.name, req
        this.image});

@ override
  widgnt build (Build context) {

    return card(
      child column (
        children. I

    Image network (image URL')

    sizedBox (height : 10)
```

3) Provide Examples of commonly used widgnt type

- structural widgnt
  These widget act as foundation for building the UI

Teacher's Sign.

material App - The Root widget of flutter app that provide essential configuration

container - A versatile widget used for styling margin and background customization

```
materialApp (
    home: Scaffold (
    appBar: AppBar (title: Text ('Flutter
                                widget')
    body : container (
        padding : EdgeInsetsall (100)
        child: Text ("Hello, flutter!")
        )
    );
);
```

2) Input and Validation widget
Textfield Accept text input from user
validation button . A button with elevation
gesture detection

Detect gestures like taps swipes and long presses

Ex- coloumn (
    children : [
    Textfield (decoration : InputDecora

[step-2]
Register the flutter app with firebase
In the firebase project dashboard click "Add App". For Android : ENTER Android Package Name and download google service.json file.

[Step-3] Install firebase dependencies
Add firebase dependicies in pubspec.yaml.
firebase core
firebase auth
Download & install

[Step-4] - configure firebase for Android and IOS
1) open Android/build.gradle and ensure the following dependencies com.google.gms google services.

2) open Android/app/build.gradle and at the bottom Apply plugin: com.google.gms.google.services

[Step 5] Initialize Firebase in flutter
Void main() async{
widgets flutter Binding ensure Initialized();
await Firebase.initializeApp()

runApp MyApp();
Run App (MyApp());

Teacher's Sign.:

Benifits of using Firebase

Firebase is Backend as a service (Baas) that simplifies backend development for flutter Apps. Here are some key benefits

1) Easy to setup and scale
   No need to manage backend infrastructure scales automatically based on usage

2) Authentication - Provides Email/password google, Facebook and phone authentication

3) cloud storage
   secure file storage for image videos and documents

4) Highlight the firebase service commonly used in flutter development and provide the brief overview of how data synchronization is achieved.

1) Firebase authentication
   Enables secure Authentication using email/password phone number and third party providers like google facebook and Apple.

) cloud Firestore
   store and sync's data in Real time across device support structure queries and offline Access

Teacher's Sign.:

Ex - Firebase, Firestore instance, collection ("users")
.add ({ 'name': 'JohnDoe'
'email': JohnDoe@example.com'

});

3) Realtime Database
A Realtime - JSON database that automatically updates data across devices.
Ex - Database; Reference Ref = Firebase Database Instance
Let ({"text": "Hello, Firebase"})

4) Firebase Cloud Messaging (FCM)
Enables Push notification and messaging between users. Ex Firebase Messaging instance, subscribe To topic ("news")

5) Firebase hosting
Deploys and serves web App securing with Automatic.
Data Synchronization in firebase
Firebase ensures Real time data Synchronization across multiple devices and platforms using Firestore and Realtime Database

Runtime Database Sync Mechanism
uses persistent websocket connection for live updates.

Teacher's Sign:

Ex - Database Reference

```
Ref = firebase.database().ref("message");
ref.on("value", (snapshot) => {
    print(snapshot.val());
});
```

5) Offline Data Sync.
Firebase caches data locally and Sync storage when the developers online

Ex - FirebaseFirestore.setting.

4) Cloud function for automated updates automatic backend to trigger when data changes

7.

Teacher's Sign