NAME: Bhushan Malpani DIV: D15C ROLL NO: 33

Experiment 02 : To design Flutter UI by including common widgets.

Theory:

Flutter is a popular open-source UI toolkit developed by Google that allows developers to create beautiful, natively compiled applications for mobile, web, and desktop from a single codebase. The heart of Flutter lies in its widget system. Everything in Flutter is a widget—from the entire app layout to individual components such as buttons or text.

In the context of a Quiz App, using Flutter widgets effectively allows developers to design smooth, interactive, and visually appealing interfaces that enhance the user experience. Below are the common widgets used in Flutter and how they are utilized in a Quiz App:

1. **Scaffold**
   Scaffold is the basic structure that provides the framework for implementing material design layout in the app. It supports components like AppBar, Drawer, BottomNavigationBar, and FloatingActionButton. In a quiz app, it holds the overall layout for each screen.

2. **AppBar**
   AppBar is a widget provided by Scaffold that appears at the top of the screen. It is generally used to show the title of the screen, navigation icons, or action buttons. In a quiz app, it usually displays the quiz title or the current question number.

3. **Text**
   The Text widget is used to display strings of text. In a quiz app, it is commonly used to show questions, instructions, scores, and feedback messages like “Correct Answer” or “Try Again.”

4. **Column and Row**
   Column arranges child widgets vertically, while Row arranges them horizontally. They are the foundation for building structured layouts in Flutter. For example, options in a quiz can be displayed in a Column, and control buttons like “Next” and “Previous” can be placed in a Row.

5. **Container and Card**
   Container is a versatile widget that can contain any other widget and apply padding, margin, alignment, background color, or decoration. Card is a special type of Container with elevation and rounded corners, making it suitable for displaying quiz options in a clean, elevated block.

6. **Buttons (ElevatedButton, TextButton, etc.)**
   Buttons are essential for interaction. ElevatedButton is used for primary actions like submitting an answer or moving to the next question. TextButton is used for secondary actions like “Retry” or “Back.”

7. **ListTile**
   ListTile provides a simple way to create tappable options with a title, subtitle, leading or trailing icons. In a quiz app, it is ideal for displaying each option that the user can select.

8. **Image**
   The Image widget is used to show images, which can be loaded from local assets or the internet. It can be used for branding, question images, or icons in the quiz.

9. **AlertDialog**
   This widget is used to show a popup dialog with a message, title, and action buttons. It is helpful in showing the result of the quiz, confirmation to exit the quiz, or wrong answer feedback.

10. **LinearProgressIndicator**
    This is a horizontal progress bar that visually indicates progress. In a quiz app, it is used to show how many questions have been attempted or completed.

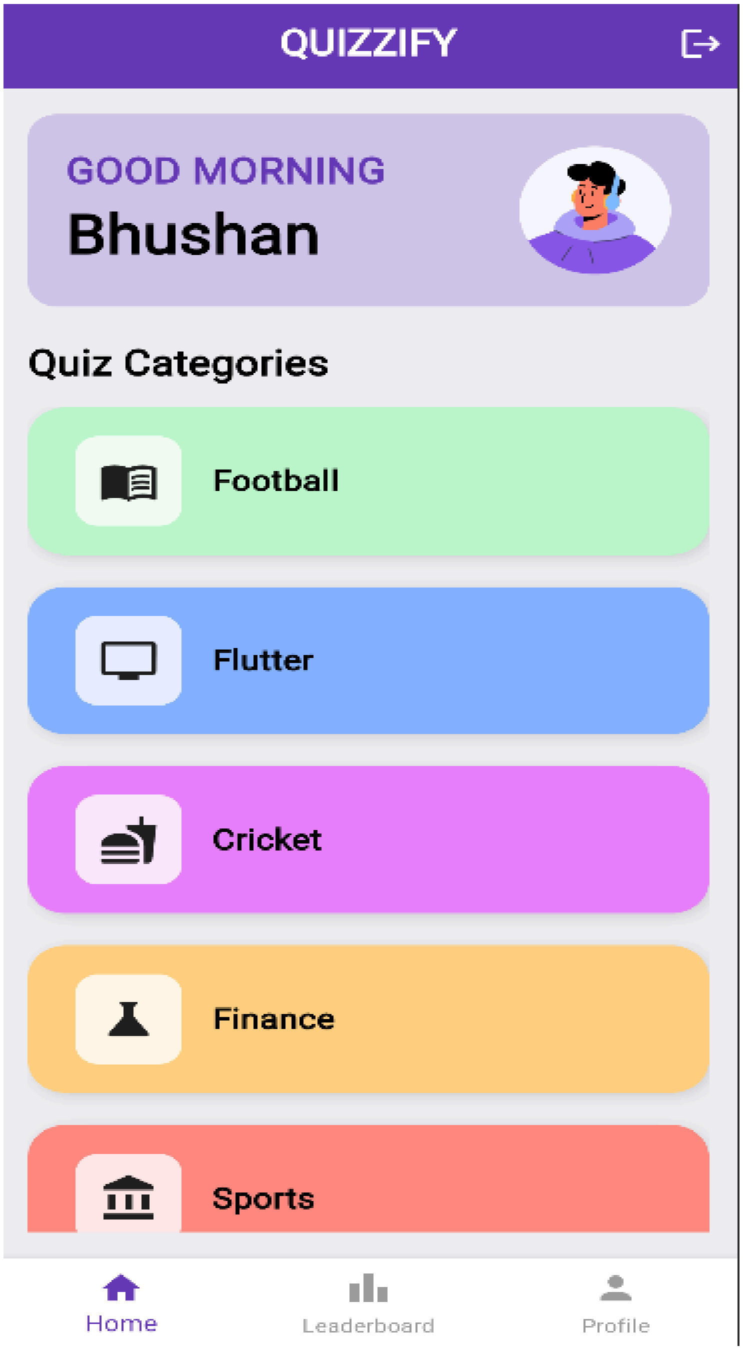
11. **Navigator**
    Navigator is used for screen navigation. It allows moving from one screen to another, such as from the home screen to the quiz screen, or from the quiz screen to the result screen.

- By combining these widgets, developers can design a responsive and well-structured quiz interface with clean layout, user interaction, and real-time feedback.
- Scaffold is used to build the main structure of the app. It holds everything including the AppBar and the body of the screen. AppBar displays the title of the quiz and helps users understand where they are in the app.
- Text widget displays the actual quiz content, like questions, options, or scores. Column helps organize the question followed by multiple options vertically. Row is used when two widgets need to be placed side-by-side, such as "Next" and "Submit" buttons.
- Container and Card are used to design the layout with styling, padding, margins, background color, and elevation. Card is often used to display each quiz option nicely separated from others.
- ElevatedButton is commonly used for interaction when users want to submit their answer or go to the next question. TextButton is useful for less prominent actions like "Skip" or "Retry."
- ListTile is a simple but powerful widget to display one quiz option. It has a built-in touch handler, so when a user taps an option, you can easily process it.
- Image is used to make the app visually appealing, such as placing the app logo on the home page or showing an image-based question.
- AlertDialog is shown after an answer is selected to give feedback or to confirm actions such as exiting the quiz or completing the quiz.
- LinearProgressIndicator shows how far along the user is in the quiz. For example, if the quiz has 10 questions, you can use this widget to indicate if the user has completed 3 out of 10.
- Navigator is used to move between different screens of the quiz app. From the home screen, it can navigate to the quiz screen, and then to the result screen after completion.

These widgets collectively help to build an interactive and functional quiz app with clean navigation and smooth user experience.

```
import 'dart:math';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'leaderboard_page.dart';
// import 'signin_page.dart';
import 'login.dart';
import 'quiz_page.dart';
import 'profile_page.dart';

class HomePage extends StatefulWidget {
  const HomePage({Key? key}) : super(key: key);

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  String username = "User";
  String profilePicUrl =
      "https://avatar.iran.liara.run/public"; // Random avatar placeholder
  final FirebaseAuth _auth = FirebaseAuth.instance;
  int _selectedIndex = 0;

  final List<Color> categoryColors = [
    Colors.greenAccent.shade100,
    Colors.blueAccent.shade100,
    Colors.purpleAccent.shade100,
    Colors.orangeAccent.shade100,
    Colors.redAccent.shade100,
  ];

  final List<IconData> categoryIcons = [
    Icons.menu_book,
    Icons.tv,
    Icons.fastfood,
    Icons.science,
    Icons.account_balance,
    Icons.category,
    Icons.sports_esports,
    Icons.music_note,
    Icons.flight,
    Icons.computer,
  ];

  @override
  void initState() {
    super.initState();
    fetchUserData();
  }

  Future<void> fetchUserData() async {
    User? user = _auth.currentUser;
    if (user != null) {
      DocumentSnapshot userDoc = await FirebaseFirestore.instance
          .collection('users')
          .doc(user.uid)
          .get();
      if (userDoc.exists) {
        setState(() {
          username = userDoc['username'];
          int randomNumber =
              Random().nextInt(50) + 1; // Generate a number between 1-50
          profilePicUrl =
              "https://avatar.iran.liara.run/public/$randomNumber"; // Random avatar
        });
      }
    }
  }

  Stream<QuerySnapshot> fetchCategories() {
    return FirebaseFirestore.instance.collection('categories').snapshots();
  }

  void _onItemTapped(int index) {
    if (index == _selectedIndex) return;

    setState(() {
      _selectedIndex = index;
    });

    User? user = _auth.currentUser; // Get the current user

    switch (index) {
      case 0:
        Navigator.pushReplacement(
          context,
          MaterialPageRoute(builder: (context) => const HomePage()),
        );
        break;
      case 1:
        Navigator.pushReplacement(
          context,
          MaterialPageRoute(
              builder: (context) => const LeaderboardPage(category: 'Science')),
        );
        break;
      case 2:
        if (user != null) {
          Navigator.pushReplacement(
            context,
            MaterialPageRoute(
                builder: (context) =>
                    ProfilePage(userId: user.uid)), // Pass user ID
          );
        }
        break;
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: const Color.fromARGB(255, 237, 237, 241),
      appBar: AppBar(
        title: const Text(
          "QUIZZIFY",
          style: TextStyle(
              fontSize: 22, fontWeight: FontWeight.bold, color: Colors.white),
        ),
        centerTitle: true,
        backgroundColor: Colors.deepPurple,
        actions: [
          IconButton(
            icon: const Icon(Icons.logout, color: Colors.white),
            onPressed: () async {
              await _auth.signOut();
              Navigator.pushReplacement(
                context,
                MaterialPageRoute(builder: (context) => const SignInPage()),
              );
            },
          ),
        ],
      ),
      body: SafeArea(
        child: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              // Good Morning Section with Background Box
              Container(
                padding: const EdgeInsets.all(20),
                decoration: BoxDecoration(
                  color: Colors.deepPurple.shade100, // Background color
                  borderRadius: BorderRadius.circular(15), // Rounded corners
                ),
                child: Row(
                  mainAxisAlignment: MainAxisAlignment.spaceBetween,
                  children: [
                    Column(
                      crossAxisAlignment: CrossAxisAlignment.start,
                      children: [
                        const Text(
                          "GOOD MORNING",
            style: TextStyle(
              fontSize: 22, // Increased font size
              fontWeight: FontWeight.bold,
              color: Colors.deepPurple),
           ),
           Text(
            username,
            style: const TextStyle(
              fontSize: 34, // Increased font size
              fontWeight: FontWeight.bold,
              color: Colors.black),
           ),
          ],
         ),
         // Profile Picture
         CircleAvatar(
          radius: 40,
          backgroundImage: NetworkImage(profilePicUrl),
          backgroundColor: Colors.grey.shade300,
         ),
        ],
       ),
      ),
      const SizedBox(height: 20),

      // Quiz Categories Section
```

```
const Text(
  "Quiz Categories",
  style: TextStyle(
      fontSize: 22,
      fontWeight: FontWeight.bold,
      color: Colors.black),
),
const SizedBox(height: 12),

Expanded(
  child: StreamBuilder<QuerySnapshot>(
    stream: fetchCategories(),
    builder: (context, snapshot) {
      if (!snapshot.hasData) {
        return const Center(child: CircularProgressIndicator());
      }
      var categories = snapshot.data!.docs;
      return Scrollbar(
        thickness: 6,
        radius: const Radius.circular(10),
        child: SingleChildScrollView(
          child: Column(
            children: List.generate(categories.length, (index) {
              var data = categories[index].data()
                  as Map<String, dynamic>;
              Color boxColor =
                  categoryColors[index % categoryColors.length];
              IconData icon =
                  categoryIcons[index % categoryIcons.length];

              return GestureDetector(
                onTap: () {
                  Navigator.push(
                    context,
                    MaterialPageRoute(
                      builder: (context) =>
                          QuizPage(category: data['name']),
                    ),
                  );
                },
                child: Container(
                  margin: const EdgeInsets.only(bottom: 20),
                  padding: const EdgeInsets.symmetric(
                      vertical: 18, horizontal: 25),
                  decoration: BoxDecoration(
                    color: boxColor,
                    borderRadius: BorderRadius.circular(20),
                    boxShadow: [
                      BoxShadow(
                          color: Colors.black12,
                          blurRadius: 6,
                          offset: Offset(2, 2)),
                    ],
                  ),
                  child: Row(
                    children: [
                      Container(
                        padding: const EdgeInsets.all(12),
                        decoration: BoxDecoration(
                          color: Colors.white.withOpacity(0.8),
                          borderRadius: BorderRadius.circular(12),
                        ),
                        child: Icon(
                          icon,
                          color: Colors.black87,
                          size: 32,
                        ),
                      ),
                      const SizedBox(width: 16),
                      Expanded(
                        child: Text(
                          data['name'],
                          style: const TextStyle(
                              fontSize: 18,
                              fontWeight: FontWeight.bold,
                              color: Colors.black),
                        ),
                      ),
              ],
             ),
            ),
           );
          }),
         ),
        ),
       );
      },
     ),
    ),
   ],
  ),
 ),
),

// Bottom Navigation Bar
bottomNavigationBar: BottomNavigationBar(
 currentIndex: _selectedIndex,
 onTap: _onItemTapped,
 selectedItemColor: Colors.deepPurple,
 unselectedItemColor: Colors.grey,
 backgroundColor: Colors.white,
 elevation: 10,
 showUnselectedLabels: true,
 items: const [
          BottomNavigationBarItem(icon: Icon(Icons.home), label: "Home"),
          BottomNavigationBarItem(
              icon: Icon(Icons.leaderboard), label: "Leaderboard"),
          BottomNavigationBarItem(icon: Icon(Icons.person), label: "Profile"),
        ],
      ),
    );
  }
}
```



Conclusion:

Using common widgets helps in building a responsive, maintainable, and scalable Flutter application. In my Quiz App, these widgets form the core building blocks of the UI. By understanding how to combine and customize them, you ensure a smooth user experience and professional-looking design.