Name: Mayuresh S. Pednekar Class:D15B Roll no: 53

EXPERIMENT NO. 2

Aim: To design Flutter UI by including common widgets.

Theory:

What are Flutter Widgets?
In Flutter, widgets are the basic building blocks of the user interface (UI). A widget is an immutable declaration of part of a user interface, which includes structural elements, visual elements, and layout constraints. Widgets can be combined to create complex UIs, and Flutter's reactive framework ensures that the UI automatically updates when the underlying data changes.

Flutter widgets can be categorized into two main types:

1. Stateless Widgets:
StatelessWidget: Represents a widget that doesn't depend on mutable state. Once created, the properties of a stateless widget cannot change. Stateless widgets are useful for parts of the UI that remain static.

2. Stateful Widgets:
StatefulWidget: This represents a widget that can change dynamically based on a mutable state. These widgets can be rebuilt during the lifetime of the application.
State: A separate, mutable object that holds the runtime configuration of a StatefulWidget.

List of common Flutter widgets:

1. Container:
   A box model that can contain other widgets and apply styling.
2. Row and Column:
   Used for arranging widgets horizontally (Row) or vertically (Column).
3. ListView:
   A scrollable list of widgets. There are also specialized versions like ListView.builder for efficiently building large lists.
4. Stack:
   Overlapping widgets on top of each other.
5. GridView:
   A scrollable 2D array of widgets.
6. Scaffold:
   Implements the basic material design visual layout structure.
7. AppBar:
   Represents the top app bar that usually contains titles, icons, and actions.
8. Drawer:
   A slide-in menu that is typically used for navigation.

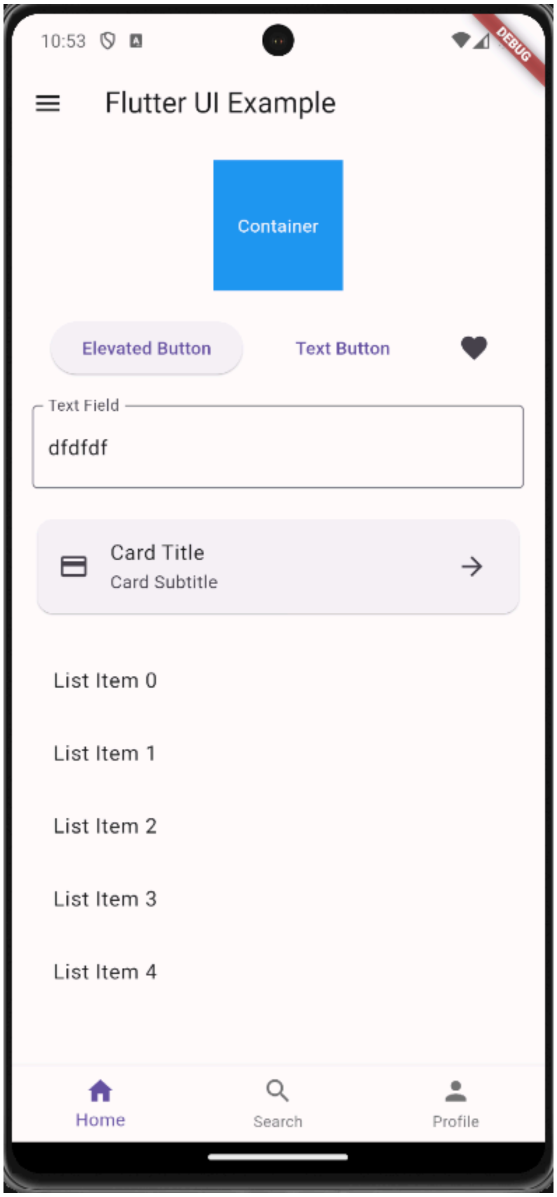
9. BottomNavigationBar:
   A bottom navigation bar for switching between different views or tabs.
10. TabBar and TabBarView:
    Used for implementing tabbed navigation.
11. TextField:
    Allows users to input text.
12. Button Widgets (ElevatedButton, TextButton, IconButton):
    For handling user interactions.
13. Image:
    Displays images.
14. Icon:
    Displays icons from the Material or Cupertino icon sets.
15. Card:
    A material design card. Useful for displaying information in a contained format.
16. AlertDialog:
    A pop-up dialog that interrupts the user's workflow to get a response.
17. Snackbar:
    A brief message is at the bottom of the screen.
18. BottomSheet:
    A sheet that slides up from the bottom.
19. Spacer:
    A widget that takes up available space.
20. Expanded:
    A widget that expands a child of a Row, Column, or Flex to fill the available space along the main axis.

Code:

```
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Flutter UI Example'),
),
drawer: Drawer(
  child: ListView(
    children: [
      ListTile(
        title: const Text('Home'),
        onTap: () {
          // Handle drawer item 1 tap
          Navigator.pop(context); // Close the drawer
        },
      ),
      ListTile(
        title: const Text('Profile'),
        onTap: () {
          // Handle drawer item 2 tap
          Navigator.pop(context); // Close the drawer
        },
      ),
    ],
  ),
),
body: Padding(
  padding: const EdgeInsets.all(16.0),
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      Container(
        color: Colors.blue,
        height: 100,
        width: 100,
        child: const Center(
          child: Text(
            'New App',
            style: TextStyle(color: Colors.white),
          ),
        ),
      ),
      const SizedBox(height: 20),
      Row(
        mainAxisAlignment: MainAxisAlignment.spaceAround,
        children: [
          ElevatedButton(
            onPressed: () {
              // Handle ElevatedButton press
      },
      child: const Text('Click here'),
    ),
    TextButton(
      onPressed: () {
        // Handle TextButton press
      },
      child: const Text('Text Button'),
    ),
    IconButton(
      onPressed: () {
        // Handle IconButton press
      },
      icon: const Icon(Icons.favorite),
    ),
  ],
),
const SizedBox(height: 20),
const TextField(
  decoration: InputDecoration(
    labelText: 'Text Field',
    border: OutlineInputBorder(),
  ),
),
const SizedBox(height: 20),
Card(
  child: ListTile(
    title: const Text('Title'),
    subtitle: const Text('Subtitle'),
    leading: const Icon(Icons.credit_card),
    trailing: const Icon(Icons.arrow_forward),
    onTap: () {
      // Handle card tap
    },
  ),
),
const SizedBox(height: 20),
Expanded(
  child: ListView.builder(
    itemCount: 5,
    itemBuilder: (context, index) {
      return ListTile(
        title: Text('List Item $index'),
        onTap: () {
```

```
                        // Handle list item tap
                      },
                    );
                  },
                ),
              ),
            ],
          ),
        ),
        bottomNavigationBar: BottomNavigationBar(
          items: const [
            BottomNavigationBarItem(
              icon: Icon(Icons.home),
              label: 'Home',
            ),
            BottomNavigationBarItem(
              icon: Icon(Icons.search),
              label: 'Search',
            ),
            BottomNavigationBarItem(
              icon: Icon(Icons.person),
              label: 'Profile',
            ),
          ],
          onTap: (index) {
            // Handle bottom navigation item tap
          },
        ),
      ),
    );
  }
}
```

Output:



Text Field
dfdfdf

Conclusion: We have understood the use and benefits of using different types of widgets like a scaffold, container, listview, row, and columns while building flutter applications and made a seamless UI using it.