Name:Priyank Kulkarni Roll no.: 32 Class: D15B

EXPERIMENT 2

Aim : To design Flutter UI by including common widgets.

Theory :
Flutter Widgets : Flutter widgets are built using a modern framework that takes inspiration from React. The central idea is that you build your UI out of widgets. Widgets describe what their view should look like given their current configuration and state. Widgets in Flutter are categorized into two main types:

StatelessWidget: Widgets that are immutable, meaning their properties (such as color, size, text) cannot change once they are built. Examples include Text, Icon, Container, and Image.

StatefulWidget: Widgets that maintain state, meaning their properties can change over time, causing the UI to update accordingly. Examples include Checkbox, Slider, TextField, and ListView.

Flutter provides a rich library of pre-built widgets that developers can use to construct their app's UI. Additionally, developers can create their own custom widgets by composing existing widgets or extending StatelessWidget or StatefulWidget classes.

Widgets in Flutter follow a declarative programming paradigm, where you describe how the UI should look based on the current state of the app. Flutter's framework then takes care of efficiently updating the UI to reflect any changes in state. This approach allows for fast, expressive, and flexible UI development.

Types of Widgets :
Flutter offers a wide range of widgets for building user interfaces. Here’s a list of some common widgets along with brief explanations and examples:

1. Text: Used for displaying text.
```
Text('Hello, Flutter!')
```

2. Image: Displays an image.
```
Image.network('https://example.com/image.jpg')
```

3. Container: A container that can have a background color, padding, and more.

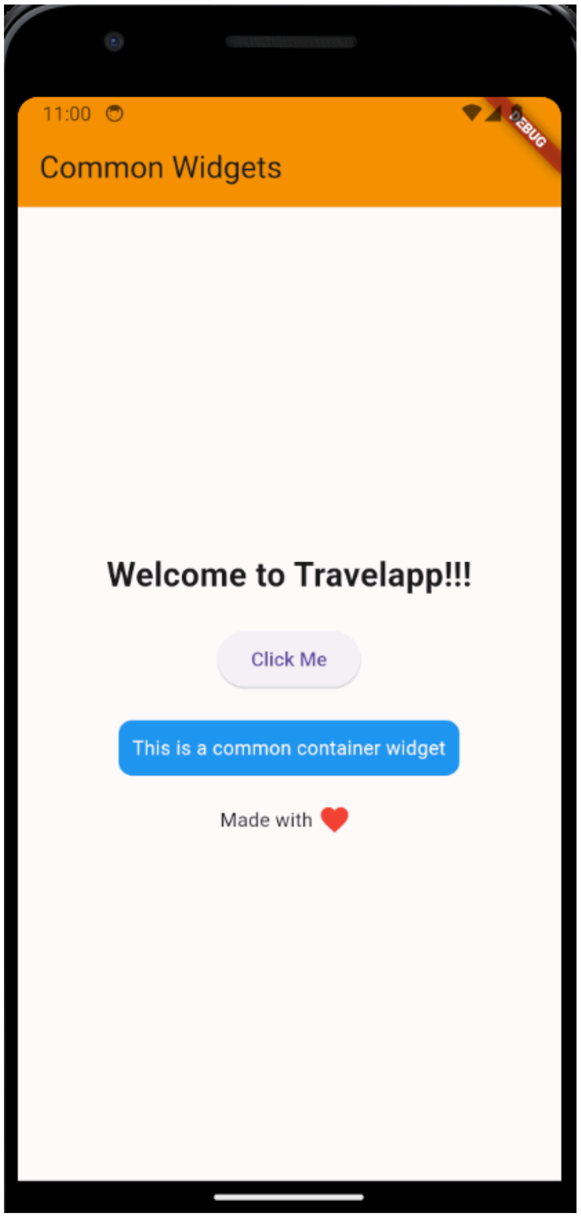
Container(color: Colors.blue,
padding: EdgeInsets.all(16.0),
child: Text('Container Example'),
)

Main.dart

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


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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Common Widgets',
      home: Scaffold(
        appBar: AppBar(
          backgroundColor: Color.fromARGB(255, 246, 148, 0),
          title: Text('Common Widgets'),
        ),
        body: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
              Text(
                'Welcome to Travelapp!!!',
                style: TextStyle(
                  fontSize: 24.0,
                  fontWeight: FontWeight.bold,
                ),
              ),
              SizedBox(height: 20.0),


              ElevatedButton(
                onPressed: () {
                  // Add button functionality here
```

```dart
                },
                child: Text('Click Me'),
              ),
              SizedBox(height: 20.0),
              Container(
                padding: EdgeInsets.all(10.0),
                decoration: BoxDecoration(
                  color: Colors.blue,
                  borderRadius: BorderRadius.circular(10.0),
                ),
                child: Text(
                  'This is a common container widget',
                  style: TextStyle(color: Colors.white),
                ),
              ),
              SizedBox(height: 20.0),
              Row(
                mainAxisAlignment: MainAxisAlignment.center,
                children: <Widget>[
                  Text('Made with '),
                  Icon(Icons.favorite, color: Colors.red),
                  SizedBox(width: 5.0),

                ],
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

Output:



Conclusion:
From this experiment, we have understood what are widgets and how to use it to create a basic UI. Also,how to orient the widgets to make a UI interactive and creative to the user.We have used various types of widgets like row,column to align and text as well.