Experiment no.2
Name: Soham A. Phalke
Div: D15B
Roll no.: 55

Aim: To design Flutter UI by including common widgets.

Theory:
Flutter widgets are the building blocks of Flutter applications. They are elements of the user interface (UI) that determine the layout, appearance, and behavior of your app. Widgets can range from simple components like buttons and text fields to complex layouts and animations. Flutter provides a rich set of built-in widgets that you can use to create highly customized and interactive user interfaces.

Widgets in Flutter are categorized into two main types:

> Stateless Widgets: These widgets are immutable and do not have any internal state. They are typically used for UI components that do not change over time. Stateless widgets are created by extending the StatelessWidget class and overriding the build method to define the UI. Examples include text, buttons, icons, and images.
> Stateful Widgets: These widgets maintain state that can change over time. They are used for UI components that need to update dynamically in response to user interactions, data changes, or other events. Stateful widgets are created by extending the StatefulWidget class, which consists of two classes: the widget itself and a corresponding state class (usually named with a State suffix). The state class manages the mutable state of the widget and is responsible for rebuilding the widget when the state changes.

Here's a list of some commonly used Flutter widgets:

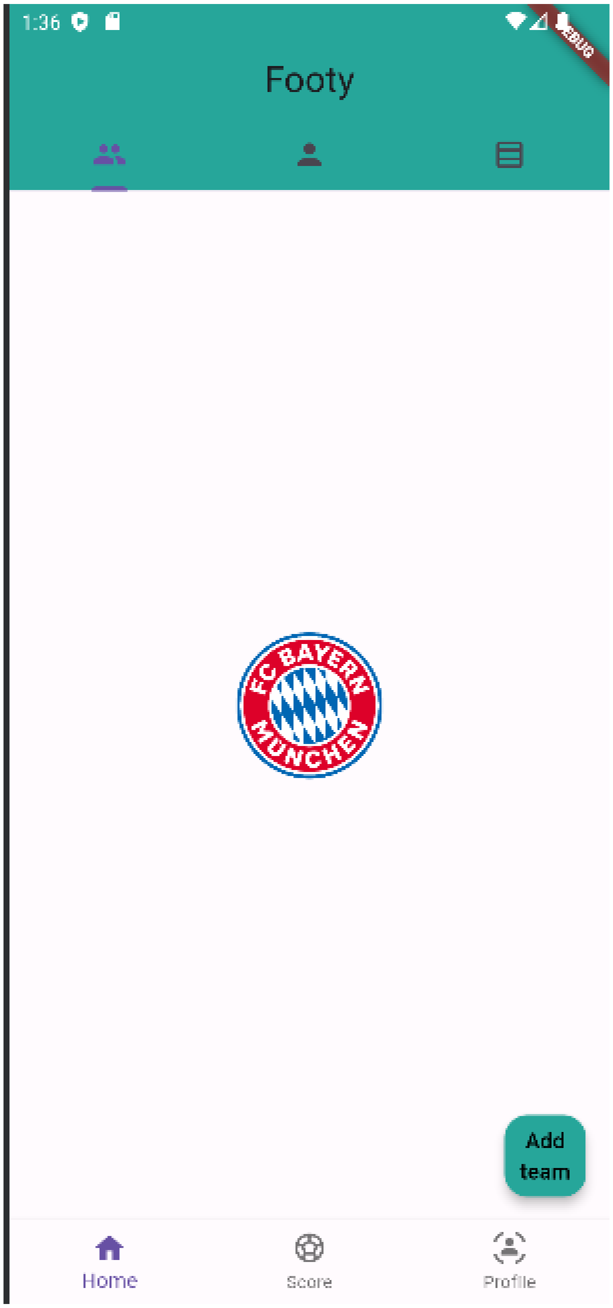
> Text: Displays a string of text.
> Image: Displays an image.
> Container: A versatile widget for creating layout and styling.
> Row: A widget that arranges its children widgets horizontally.
> Column: A widget that arranges its children widgets vertically.
> Stack: A widget that overlays its children widgets.

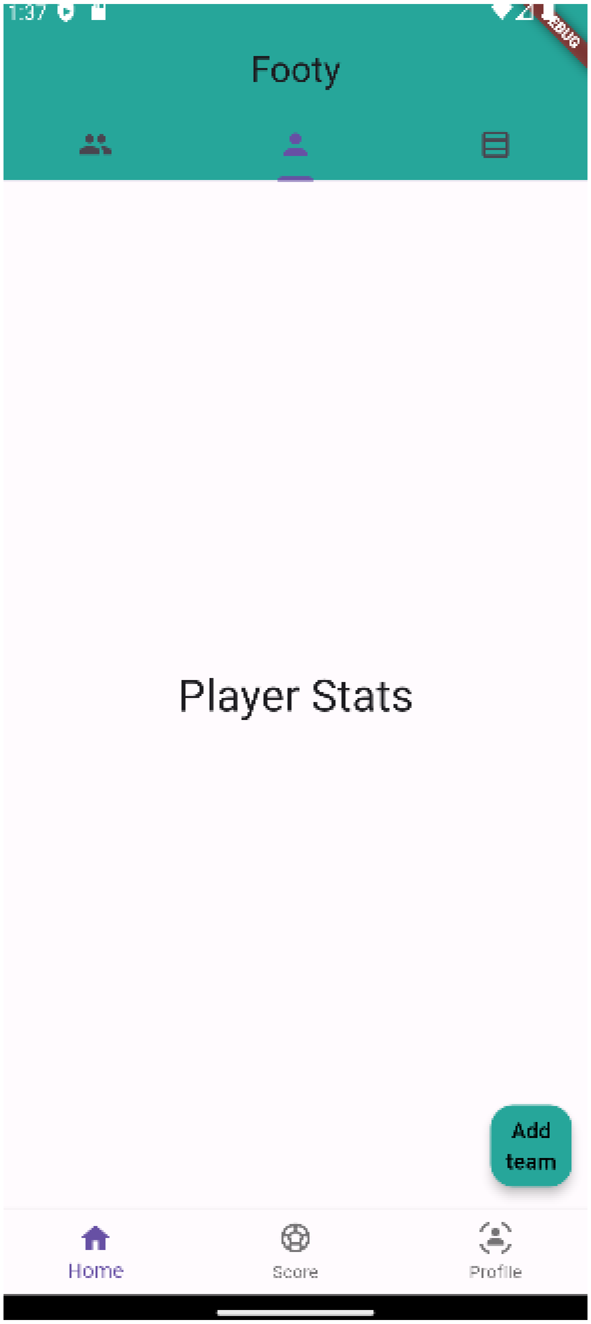
ListView: Displays a scrollable list of widgets.
GridView: Displays a scrollable grid of widgets.
AppBar: A material design app bar.
Scaffold: Implements the basic material design layout structure.
TextField: Allows users to input text.
FlatButton, RaisedButton, IconButton: Various types of buttons.
AlertDialog, BottomSheet: Dialog and modal widgets.
GestureDetector: Detects gestures such as taps, drags, and swipes.
Opacity: Makes its child partially transparent.
AnimatedContainer, AnimatedOpacity, AnimatedBuilder: Widgets for creating animations.
ClipRRect: Clips its child widget with rounded corners.
AspectRatio: Sizes its child widget to a specific aspect ratio.
Spacer: A widget that fills available space in a Row or Column.

These are just a few examples, and Flutter provides many more widgets for various purposes. You can also create custom widgets by composing existing ones or by subclassing StatelessWidget or StatefulWidget.

Code:

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

void main()=> runApp(MaterialApp(
 home: Home(),
 theme: ThemeData(
   fontFamily: 'Teko',
 ),
));


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

 @override
 Widget build(BuildContext context) {
   return DefaultTabController(
     initialIndex: 0,
     length: 3,
     child: Scaffold(
appBar: AppBar(
  title:Text("Footy",
    style: TextStyle(
        fontSize: 25
    ),
  ),
  centerTitle: true,
  backgroundColor: Colors.teal[400],
  bottom: TabBar(
    tabs: [
      Tab(
        icon: Icon(Icons.people),
      ),
      Tab(
        icon: Icon(Icons.person),
      ),
      Tab(
        icon: Icon(Icons.table_rows_outlined),
      ),
    ],
  ),
),
body:
TabBarView(
  children: <Widget>[
    Center(
      child: Image.asset('Assets/bayern.png',
        width: 100,
        height: 100,
      )
    ),
    Center(
      child: Text("Player Stats",
        style: TextStyle(
          fontSize: 30,
          fontFamily: 'Teko',
        ),
      ),
    ),
    Center(
              child: Text("Points table",
                style: TextStyle(
                  fontSize: 30,
                  fontFamily: 'Teko',
                ),
              ),
            ),

          ],

        ),





        bottomNavigationBar:Bottombar(),
        floatingActionButton:FloatingActionButton(
          onPressed: () {},
          child:Text("Add team",
            textAlign: TextAlign.center,
            style: TextStyle(
              fontFamily: 'Teko',
              color: Colors.black,
              fontSize: 15,
            ),),
          backgroundColor: Colors.teal[400],

        ),
      ),
    );
  }
}

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

  @override
  Widget build(BuildContext context) {
    return BottomNavigationBar(
      items: [
        BottomNavigationBarItem(
          icon: Icon(Icons.home),
          label: 'Home',
          backgroundColor: Colors.deepOrange,
        ),
        BottomNavigationBarItem(
          icon: Icon(Icons.sports_soccer),
          label: 'Score',
          backgroundColor: Colors.green,
        ),
        BottomNavigationBarItem(
          icon: Icon(Icons.sensor_occupied),
          label: 'Profile',
          backgroundColor: Colors.green,
        ),


      ],
    );
  }
}
```





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.