# Experiment 2

Name:Piyush Tilokani
Div: D15A
Roll no: 63

Aim: To design Flutter UI by including common widgets.

Theory: In Flutter, widgets are the building blocks of the user interface, and several common widgets play crucial roles in creating engaging and interactive applications. Here's a brief overview of some fundamental Flutter widgets:

1. Container: The most basic building block, a container is a box model that can contain other widgets, allowing you to customize its dimensions, padding, and decoration.

2. Row and Column: These widgets help organize children widgets horizontally (Row) or vertically (Column), facilitating the creation of flexible and responsive layouts.

3. AppBar: AppBar is a material design widget providing a top app bar that typically includes the app's title, leading and trailing icons, and actions.

4. ListView: Used to create scrollable lists of widgets, ListView is versatile for displaying a large number of items efficiently.

5. TextField: Enables users to input text, providing a text editing interface with options for validation, styling, and interaction.

6. ElevatedButton is a Flutter widget used to create a button with a raised appearance. It typically represents the primary action in a user interface. The button has a background color, elevation, and responds to user interactions with visual feedback

7. Image: The Image widget displays images from various sources, supporting both local and network images.

8. Scaffold: A top-level container for an app's visual elements, Scaffold provides a structure that includes an AppBar, body, and other optional features like drawers and bottom navigation.

9. Card: Representing a material design card, this widget displays information in a compact and visually appealing format, often used for grouping related content.

10. GestureDetector: Allows detection of various gestures like taps, drags, and long presses, enabling interactive responses to user input.

11. Stack: A widget that allows children widgets to be overlaid, facilitating complex UI designs by layering widgets on top of each other.

12. FutureBuilder: Ideal for handling asynchronous operations, FutureBuilder simplifies the management of UI updates based on the completion of a Future, making it valuable for fetching and displaying data.

These are just a few of the many widgets available in Flutter, each serving a unique purpose in crafting dynamic and user-friendly interfaces.

Code:

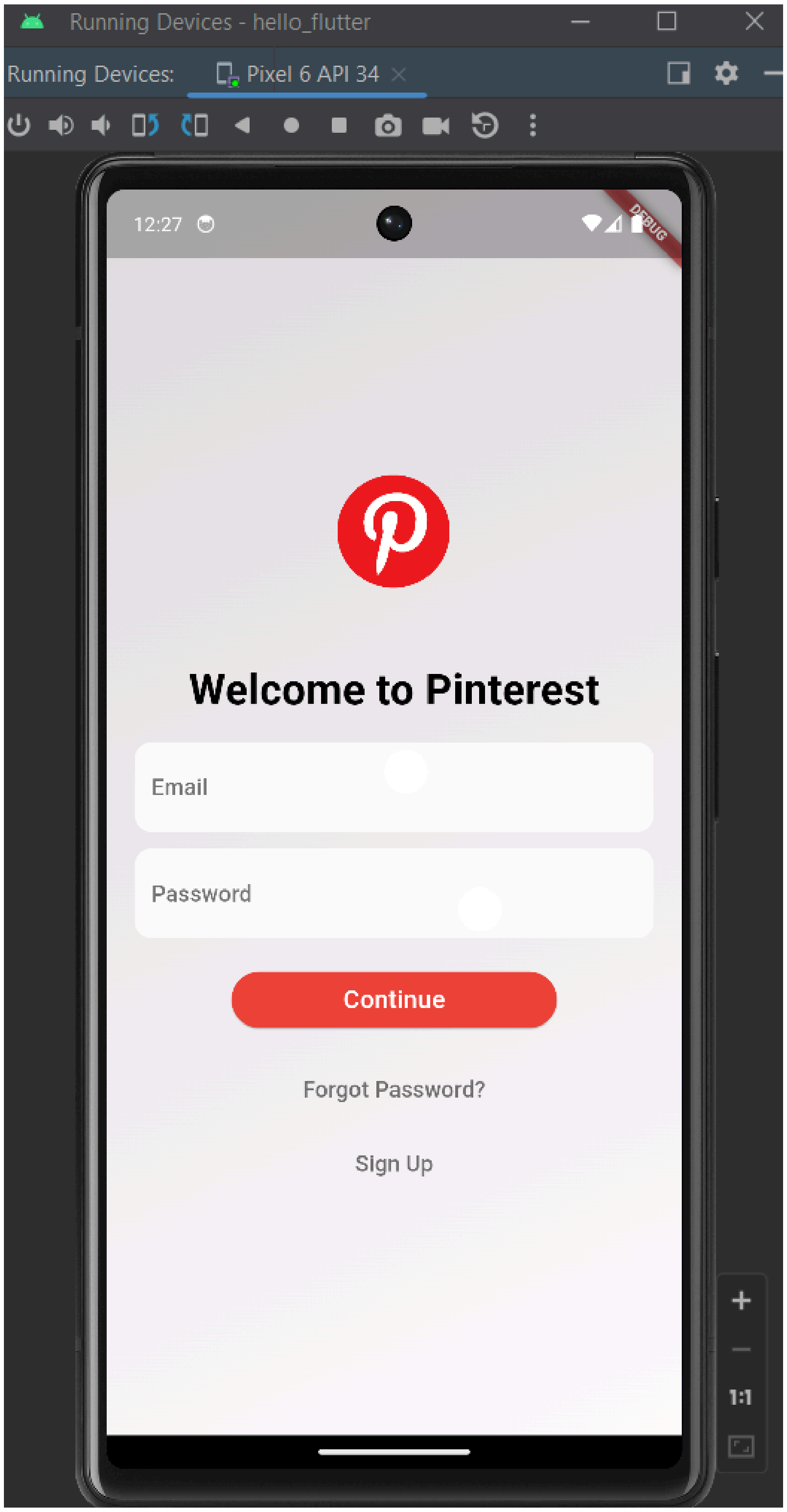
```
import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: LoginPage(),
    );
  }
}

class LoginPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        padding: EdgeInsets.symmetric(horizontal: 20.0),
        decoration: BoxDecoration(
          gradient: LinearGradient(
            begin: Alignment.topLeft,
            end: Alignment.bottomRight,
            colors: [Colors.black12, Colors.white12],
          ),
        ),
        child: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Image.asset(
                'assets/pinterest_logo.png', // Replace with your image file path
                width: 150.0,
                height: 150.0,
              ),
              SizedBox(height: 16.0),
              Text(
                'Welcome to Pinterest',
                style: TextStyle(
                  color: Colors.black,
                  fontSize: 30.0,
                  fontWeight: FontWeight.bold,
                ),
              ),
              SizedBox(height: 16.0),
              TextField(
                decoration: InputDecoration(
                  hintText: 'Email',
                  hintStyle: TextStyle(color: Colors.black54),
                  filled: true,
                  fillColor: Colors.white.withOpacity(0.8),
                  border: OutlineInputBorder(
                    borderRadius: BorderRadius.circular(12.0),
                    borderSide: BorderSide.none,
                  ),
                ),
                style: TextStyle(color: Colors.white),
              ),
              SizedBox(height: 12.0),
              TextField(
                obscureText: true,
                decoration: InputDecoration(
                  hintText: 'Password',
                  hintStyle: TextStyle(color: Colors.black54),
                  filled: true,
                  fillColor: Colors.white.withOpacity(0.8),
                  border: OutlineInputBorder(
                    borderRadius: BorderRadius.circular(12.0),
                    borderSide: BorderSide.none,
                  ),
                ),
                style: TextStyle(color: Colors.white),
              ),
              SizedBox(height: 20.0),
              ElevatedButton(
                onPressed: () {
                  // Add your login logic here
                },
                child: Text('Continue',
                  style: TextStyle(
                    color: Colors.white,
                      fontSize: 18
                  ),
                ),
                style: ElevatedButton.styleFrom(
                  primary: Colors.red,
                  padding: EdgeInsets.symmetric(horizontal: 80.0),
                  shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(20.0),
                  ),
                ),
              ),
              SizedBox(height: 16.0),
              TextButton(
                onPressed: () {
                  // Add your forgot password logic here
                },
                child: Text(
                  'Forgot Password?',
                  style: TextStyle(color: Colors.black54,
                  fontSize: 16 ),
                ),
              ),
              SizedBox(height: 5.0),
              TextButton(
                onPressed: () {
                  // Add your sign-up logic here
                },
                child: Text(
                  'Sign Up',
                  style: TextStyle(color: Colors.black54,
                      fontSize: 16),
                ),
              ),
            ],
          ),
        ),
      ),
     );
     }
    }
```

App UI:



Conclusion: Thus, understood the use of basic common widgets used in Mobile App Development and used some of them to create the login page for the chosen mini project application.