

Experiment 02

Aim : To design Flutter UI by including common widgets.

Theory:

Flutter is Google's UI toolkit for crafting beautiful, natively compiled iOS and Android apps from a single code base. To build any application we start with widgets – The building block of flutter applications. Widgets describe what their view should look like given their current configuration and state. It includes a text widget, row widget, column widget, container widget, and many more.

Widgets: Each element on a screen of the Flutter app is a widget. The view of the screen completely depends upon the choice and sequence of the widgets used to build the apps. And the structure of the code of an apps is a tree of widgets.

Types of Widgets:

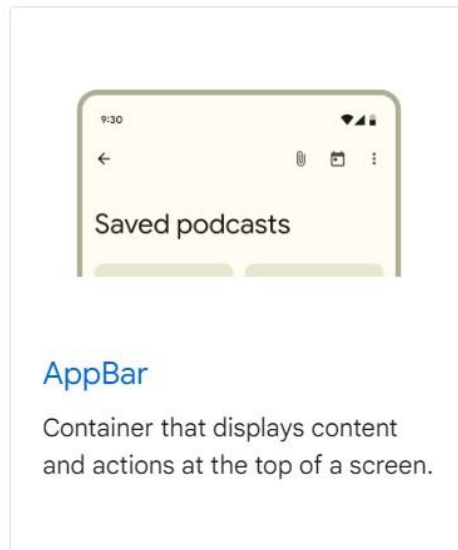
There are broadly two types of widgets in the flutter:

Stateless Widget

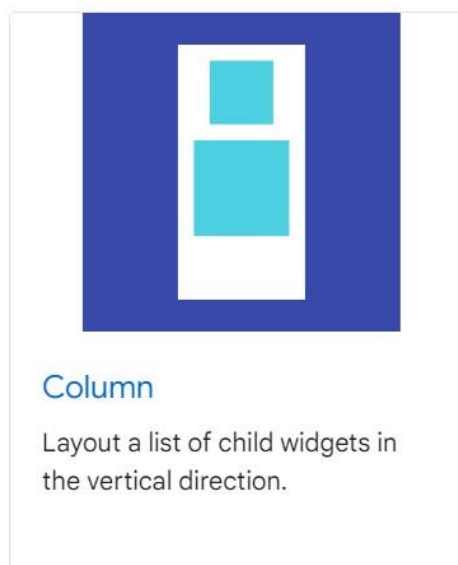
Stateful Widget

Common Widgets in Flutter :

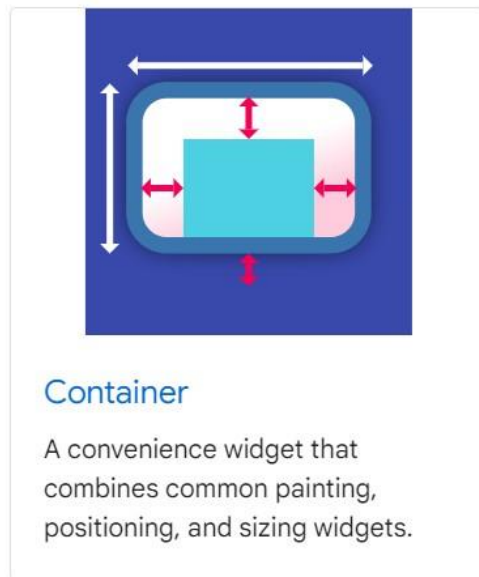
AppBar: AppBar is a material design widget used for representing the top app bar. It typically contains the app's title, leading and trailing widgets, and actions.



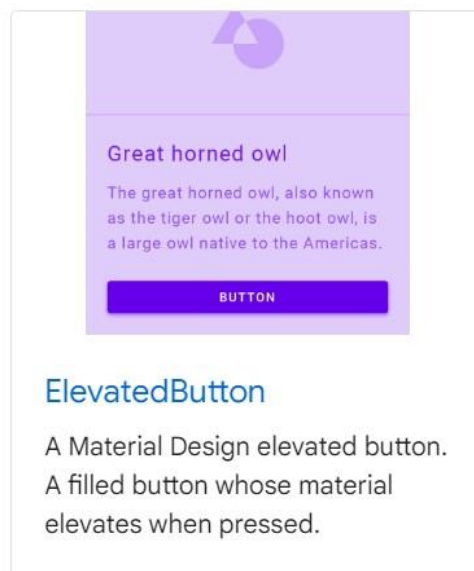
Column: Column is a layout widget that arranges its children vertically in a single column. It is commonly used to create vertical layouts.



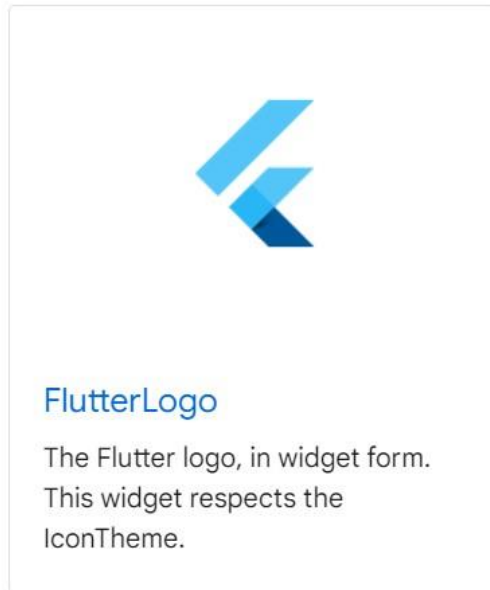
Container: Container is a versatile widget used to contain other widgets and apply various styling properties like padding, margin, background color, etc.



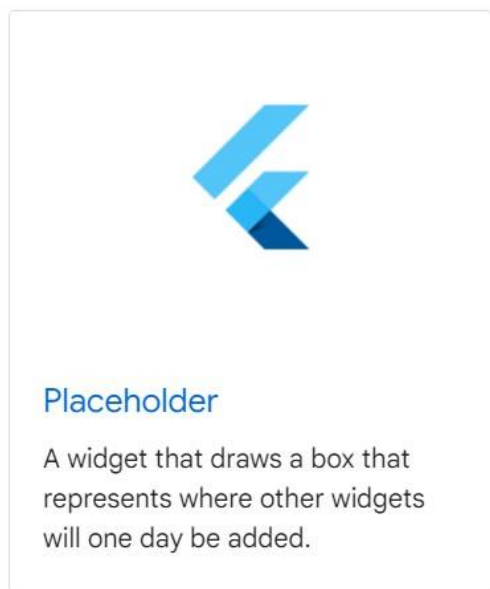
ElevatedButton: `ElevatedButton` is a button widget that displays a material design button with elevation. It's typically used for actions like submitting forms or initiating important actions.



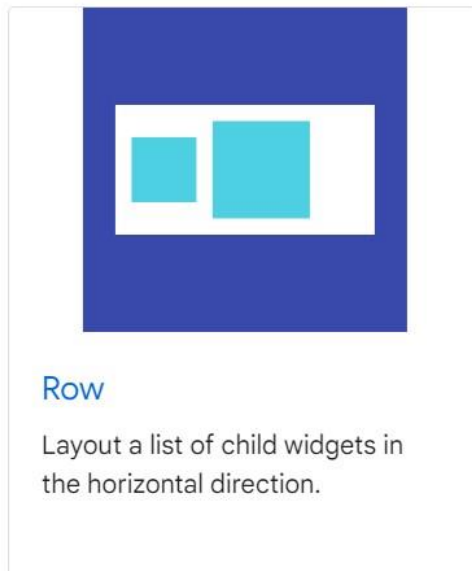
FlutterLogo: `FlutterLogo` is a widget that displays the Flutter logo. It's commonly used as a placeholder for apps during development or as a decorative element.



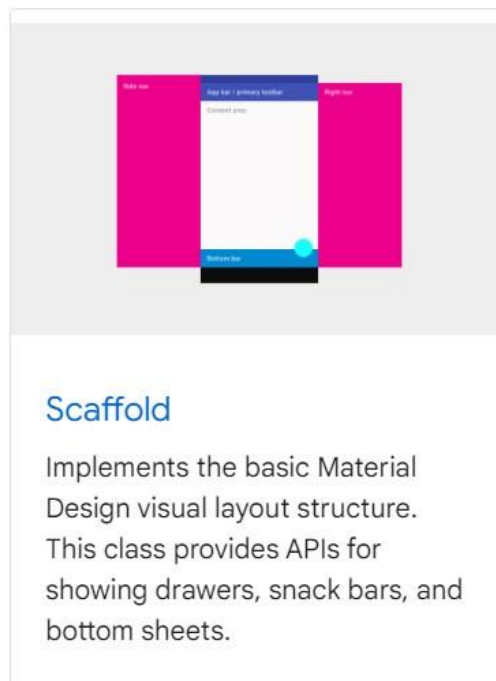
Placeholder: Placeholder is a widget used as a temporary visual placeholder for widgets that are under development or will be replaced later.



Row: Row is a layout widget that arranges its children horizontally in a single row. It's often used to create horizontal layouts.



Scaffold: Scaffold is a layout widget that provides a basic structure for material design apps. It typically contains an app bar, a body, floating action buttons, and other standard app elements.



Code:

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

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

```
}
```

```
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'Login Page',  
      theme: ThemeData(  
        primarySwatch: Colors.blue,  
      ),  
      home: LoginScreen(),  
    );  
  }  
}
```

```
class LoginScreen extends StatelessWidget {  
  final TextEditingController usernameController = TextEditingController();  
  final TextEditingController passwordController = TextEditingController();  
  
  void _handleLogin() {  
    String username = usernameController.text;  
    String password = passwordController.text;  
  
    if (username.isNotEmpty && password.isNotEmpty) {  
  
      print('Login successful');  
    } else {  
  
      print('Login failed. Please enter both username and password.');    }  
  }  
}
```

```
@override  
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(  
      title: Text('Login Page'),  
    ),  
    body: Padding(  
      padding: EdgeInsets.all(16.0),  
      child: Column(  
        mainAxisAlignment: MainAxisAlignment.center,  
        children: [  

```

```
        TextField(
          controller: usernameController,
          decoration: InputDecoration(
            labelText: 'Username',
          ),
        ),
        SizedBox(height: 16.0),
        TextField(
          controller: passwordController,
          obscureText: true,
          decoration: InputDecoration(
            labelText: 'Password',
          ),
        ),
        SizedBox(height: 32.0),
        ElevatedButton(
          onPressed: _handleLogin,
          child: Text('Login'),
        ),
      ],
    ),
  );
}
```

Explanation:

Scaffold and AppBar:

Description: Scaffold is the main container with an AppBar displaying the title "Login".

Padding:

Description: Used for adding spacing around the content.

Container Styling:

Description: Containers are employed to stylize the login form elements, providing flexibility for layout adjustments.

TextFormField:

Description: Utilized for creating text input fields for user email and password.

Details: The password field has the `obscureText: true` property, enhancing security by hiding entered text.

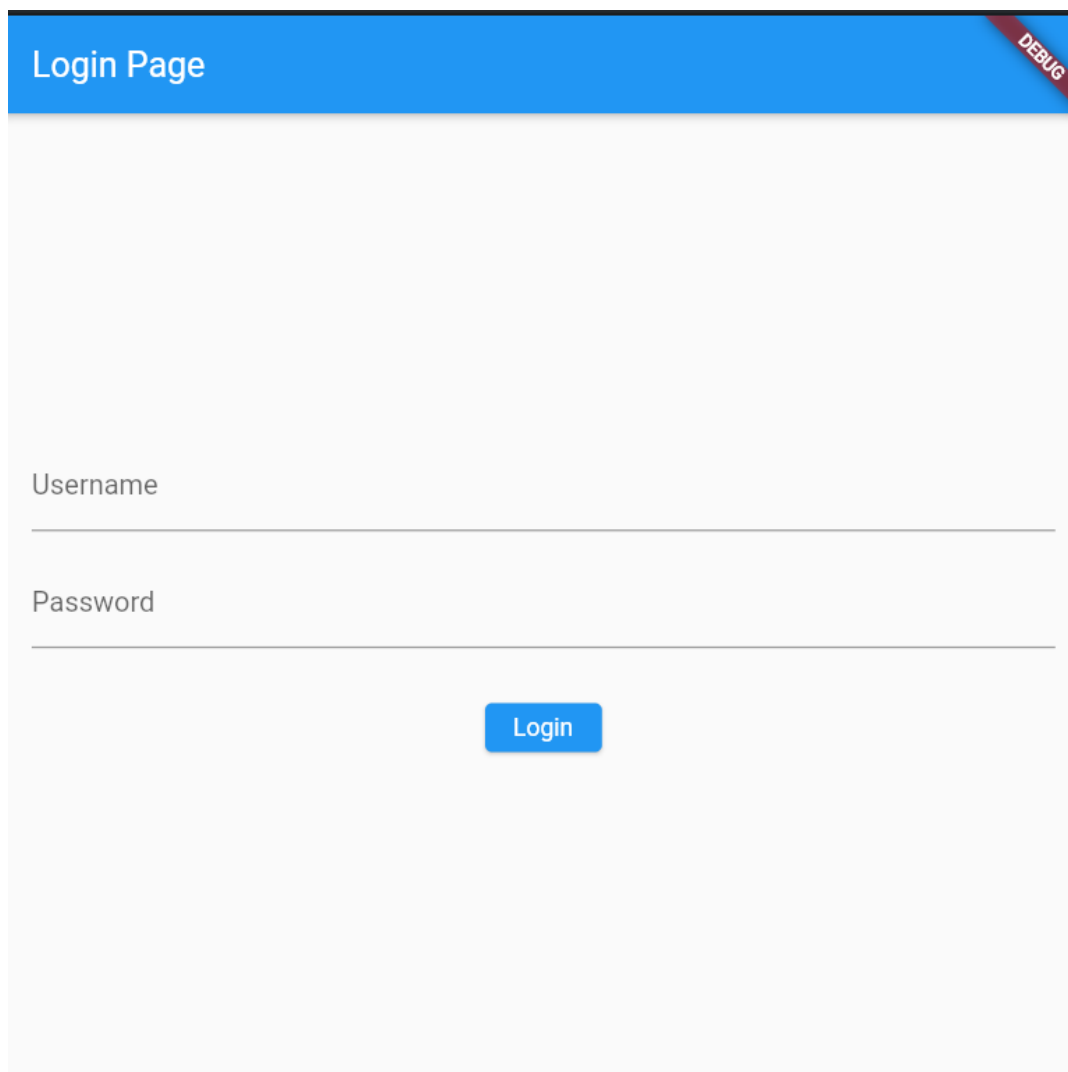
`ElevatedButton`:

Description: Implements the login action with an `ElevatedButton`.

Details: The button has the text "Login" for a clear call to action.

`TextEditingController`:

Description: Manages user input, allowing for the retrieval and modification of entered text.



Conclusion:

We have created a simple Flutter UI. For that we used some common widgets like Scaffold, padding, container, TextFormField, ElevatedButton, Row ,etc.