# Experiment 2

**Aim**: To design flutter ui by including common widgets

**Theory**: Flutter provides a rich set of built-in widgets to design responsive, beautiful, and user-friendly UIs. The UI in Flutter is built using a widget tree, where each UI component is a widget.

Common Flutter widgets include:

- **Text**: Displays text on the screen.
- **Container**: Adds padding, margin, decoration, etc., to a child widget.
- **Row / Column**: Aligns widgets horizontally / vertically.
- **Card**: Displays content inside a stylized box.
- **ListView**: Displays a scrollable list of items.
- **TextField**: Allows user input.
- **ElevatedButton**: A button with elevation and styling options.

In the context of a **Meal Delivery** app, these widgets help create screens like:

- Home screen with a list of food items.
- Search and filter features.

**Code**:

1. StatefulWidget

```
class LogIn extends StatefulWidget {
  const LogIn({super.key});
  @override
  State<LogIn> createState() => _LogInState();
}
```

- StatefulWidget is used when the UI can change dynamically. For example, in our case, we update the UI based on user input like email and password.

- LogIn is the name of our screen class.

2. Scaffold

```
return Scaffold(
  backgroundColor: Color(0xff14141d),
  body: SingleChildScrollView(
    child: Column(
```

- Scaffold provides the basic layout structure for the app screen (like app bar, body, background color, etc.).

3. SingleChildScrollView & Column

```
body: SingleChildScrollView(
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
```

- SingleChildScrollView allows the whole content to be scrollable (important for small screens).

- Column arranges widgets vertically.

4. Image.asset

```
Image.asset("images/signin.png", height: 200, fit: BoxFit.cover),
```

- This displays a static image at the top of the screen.

5. Padding

```
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 30.0, vertical: 30.0),
  child: Column(
```

- Adds space around widgets to make UI more user-friendly and clean.

6. Text

```
Text(
  "Log In",
  style: TextStyle(
    color: Colors.white,
    fontSize: 32.0,
    fontWeight: FontWeight.bold,
  ),
),
```

- Text widget displays static labels like “Log In”, “Email”, etc.

7. TextField

```
TextField(
  controller: mailcontroller,
  decoration: InputDecoration(
    hintText: "Enter Email",
    hintStyle: TextStyle(color: Colors.white54),
    suffixIcon: Icon(Icons.email, color: Colors.white),
```

```
  ),
  style: TextStyle(color: Colors.white),
  cursorColor: Colors.white,
),
```

- TextField allows the user to enter input.
- We use controller to retrieve the user input.
- Similar widget is used for Password.

8. GestureDetector and Container

```
GestureDetector(
  onTap: () {
    if (mailcontroller.text != "" && passwordcontroller.text != "") {
      setState(() {
        email = mailcontroller.text;
        password = passwordcontroller.text;
      });
      userLogin();
    }
  },
  child: Container(
    width: 160,
    padding: EdgeInsets.all(12),
    decoration: BoxDecoration(
      color: Color(0xff6b63ff),
      borderRadius: BorderRadius.circular(30),
    ),
    child: Center(
      child: Text(
        "Log In",
        style: TextStyle(
          color: Colors.white,
```

```
          fontSize: 20.0,
          fontWeight: FontWeight.bold,
        ),
      ),
    ),
  ),
),
```

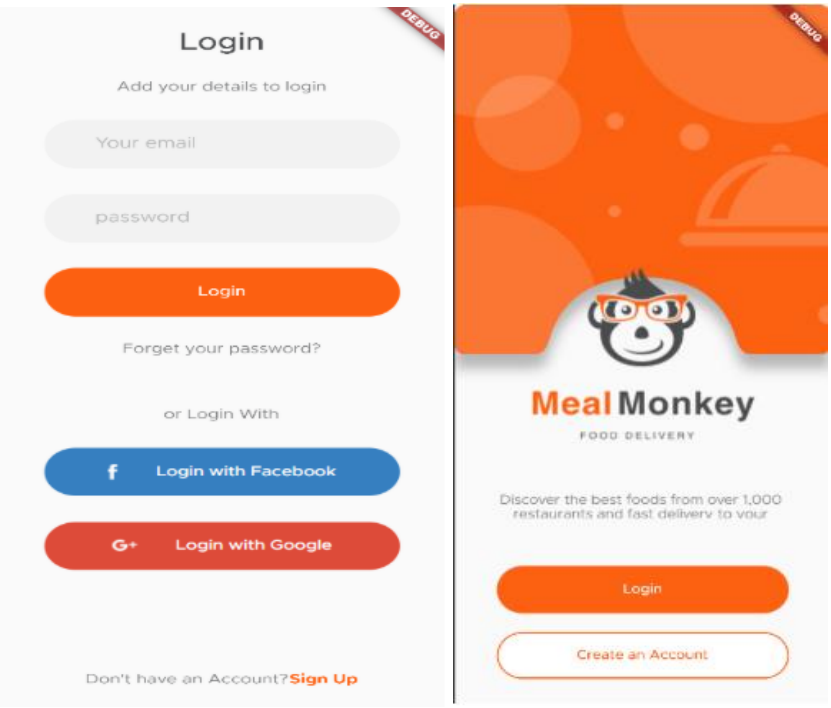
- GestureDetector detects taps.
- Container is styled to look like a button.
- On tap, the userLogin() function is called.

9. Navigator

```
Navigator.push(
  context,
  MaterialPageRoute(builder: (context) => const SignUp()),
),
```

- Navigates to the Signup screen when "New user? Sign Up" is tapped.

**Screenshots:**




**Conclusion:** In this lab, we successfully included and configured images, custom fonts, and Cupertino icons in a Flutter app. Through the Meal Monkey homepage, we saw how to declare and use these assets to enhance the UI. Using pubspec.yaml correctly and applying these features helped make the app visually appealing and platform-friendly.