Name: Prathamesh Shetty
Class: D15B
Roll No: 53

MPL Experiment 2

In **Flutter**, the entire user interface (UI) is built using **widgets**. A widget is essentially a **descriptive representation** of part of the user interface, whether it's a simple button, an image, a list, or even the layout structure of the screen itself. Unlike some other frameworks where UI elements are pre-built components, in Flutter, **everything is a widget**.

Widgets in Flutter are **immutable** (cannot be changed directly), and they are often used to **build and describe the UI** in terms of properties, layout, and interaction. A widget is **just a blueprint**, but when combined with a **StatefulWidget**, it allows the UI to **react to state changes** like user input or asynchronous operations.

1 Stateless vs Stateful Widgets

🔹 **Stateless Widgets**

- These widgets **do not change** once built.
- Used for static UI elements like text, icons, and images.

🔹 **Stateful Widgets**

- These widgets **can change dynamically** during runtime.
- Used for interactive UI components like buttons, text fields, and animations.

2 Visible (Structural) Widgets

These widgets define the **visual layout** of the UI.

🔹 **Container Widgets** – Used for layout styling, including padding, margin, and background color.
🔹 **Text Widgets** – Displays text in different styles.
🔹 **Image Widgets** – Displays images from assets, network, or memory.
🔹 **Button Widgets** – Includes `ElevatedButton`, `TextButton`, and `IconButton` for user interaction.
🔹 **List & Grid Widgets** – Used for displaying scrollable lists (`ListView`) or grids (`GridView`).

3 Layout (Structural) Widgets

These widgets help in arranging other widgets on the screen.

- **Row & Column** – Arranges widgets horizontally or vertically.
- **Stack** – Places widgets on top of each other.
- **Expanded & Flexible** – Adjusts widget size dynamically based on available space.
- **SizedBox** – Adds spacing between widgets.

4 Interactive Widgets

Widgets that handle user interactions.

- **GestureDetector** – Detects gestures like taps, swipes, and long presses.
- **TextField** – Used for user input.
- **Checkbox, Radio, and Switch** – Used for selections and toggles.
- **Slider** – Allows selecting values within a range.

5 Animation & Effects Widgets

These widgets help in adding animations and special effects.

- **AnimatedContainer** – Allows smooth transitions when properties change.
- **Hero** – Provides shared element transitions between screens.
- **Opacity** – Controls widget transparency.
- **Transform** – Allows rotation, scaling, and translation of widgets.

6 Navigation & Routing Widgets

These widgets help in navigating between different screens.

- **Navigator** – Manages the navigation stack.
- **PageRoute** – Defines different types of screen transitions.
- **Drawer** – A slide-out navigation menu.
- **BottomNavigationBar** – A tabbed navigation bar at the bottom.

7 State Management Widgets

These widgets help in managing the state of an application.

- **InheritedWidget** – The base class for propagating state down the widget tree.
- **Provider** – A widely used state management solution.
- **Bloc & Riverpod** – Advanced state management approaches for complex apps.

Code:

main.dart

```
import 'package:flutter/material.dart';
import './screens/home_page.dart';  // Import the HomePage

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

class MyApp extends StatelessWidget {
 const MyApp({Key? key}) : super(key: key);

 @override
 Widget build(BuildContext context) {
   return MaterialApp(
     title: 'ShareIt',
     debugShowCheckedModeBanner: false,  // Removes the debug banner
     theme: ThemeData(
       primarySwatch: Colors.lightBlue,
       scaffoldBackgroundColor: Colors.white,
     ),
     home: const HomePage(),  // Use the HomePage as the home screen
   );
 }
}
```

home_page.dart

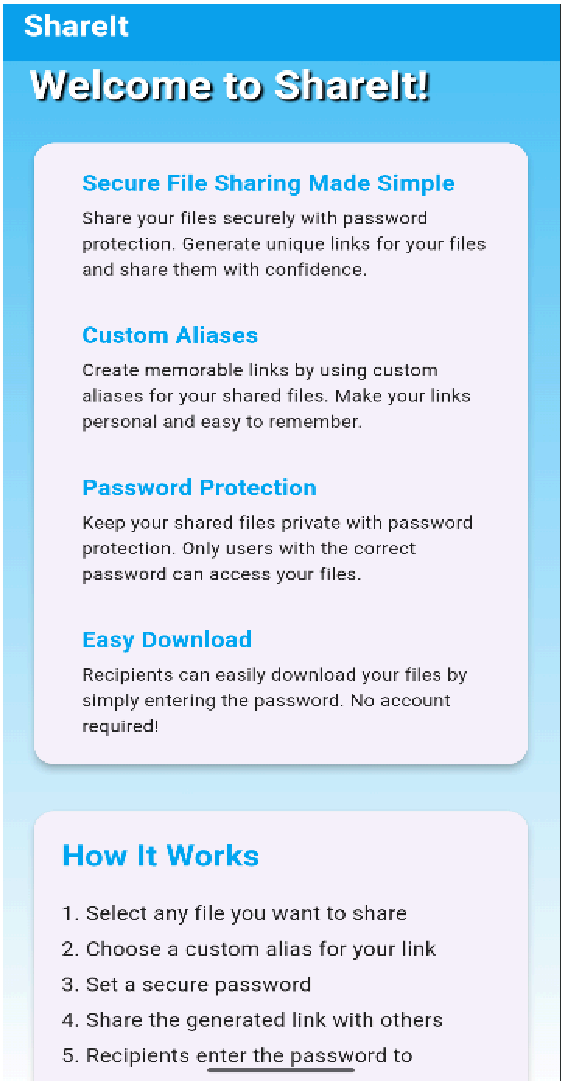
```
import 'package:flutter/material.dart';

class HomePage extends StatelessWidget {
 const HomePage({Key? key}) : super(key: key);

 @override
 Widget build(BuildContext context) {
   return Scaffold(
     appBar: AppBar(
       title: const Text(
         'ShareIt',
         style: TextStyle(
           fontSize: 24,
           fontWeight: FontWeight.bold,
           color: Colors.white,
         ),
       ),
       backgroundColor: Colors.lightBlue,
       elevation: 0,
     ),
     body: Container(
       decoration: BoxDecoration(
         gradient: LinearGradient(
           begin: Alignment.topCenter,
           end: Alignment.bottomCenter,
           colors: [
             Colors.lightBlue.shade300,
             Colors.white,
           ],
         ),
       ),
       child: SingleChildScrollView(
         child: Padding(
           padding: const EdgeInsets.all(20.0),
           child: Column(
             crossAxisAlignment: CrossAxisAlignment.start,
             children: [
               const Text(
                 'Welcome to ShareIt!',
                 style: TextStyle(
                   fontSize: 32,
                   fontWeight: FontWeight.bold,
                   color: Colors.white,
                   shadows: [
                     Shadow(
                       offset: Offset(2, 2),
                       blurRadius: 3.0,
                       color: Color.fromARGB(255, 0, 0, 0),),),],),),),
               const SizedBox(height: 20),
               Card(
                 elevation: 5,
                 shape: RoundedRectangleBorder(
                   borderRadius: BorderRadius.circular(15),
                 ),
                 child: Padding(
                   padding: const EdgeInsets.all(20.0),
                   child: Column(
                     crossAxisAlignment: CrossAxisAlignment.start,
                     children: [
                       _buildFeatureSection(
                         'Secure File Sharing Made Simple',
                         'Share your files securely with password protection. Generate
unique links for your files and share them with confidence.',
                       ),
                       const SizedBox(height: 30),
                       _buildFeatureSection(
                         'Custom Aliases',
                         'Create memorable links by using custom aliases for your shared
files. Make your links personal and easy to remember.',
                       ),
                       const SizedBox(height: 30),
                       _buildFeatureSection(
                         'Password Protection',
                         'Keep your shared files private with password protection. Only
users with the correct password can access your files.',
                       ),
                       const SizedBox(height: 30),
                       _buildFeatureSection(
                         'Easy Download',
                         'Recipients can easily download your files by simply entering the password. No account required!',
                       ),],),),),
               const SizedBox(height: 30),
               Card(
                 elevation: 5,
                 shape: RoundedRectangleBorder(
                   borderRadius: BorderRadius.circular(15),
                 ),
                 child: Padding(
                   padding: const EdgeInsets.all(20.0),
                   child: Column(
                     crossAxisAlignment: CrossAxisAlignment.start,
                     children: const [
                       Text(
                         'How It Works',
                         style: TextStyle(
                           fontSize: 24,
                           fontWeight: FontWeight.bold,
                           color: Colors.lightBlue,
                         ),
                       ),
                       SizedBox(height: 15),
                       Text(
                         '1. Select any file you want to share\n'
                             '2. Choose a custom alias for your link\n'
                             '3. Set a secure password\n'
                             '4. Share the generated link with others\n'
                             '5. Recipients enter the password to download',
                         style: TextStyle(
                           fontSize: 16,
                           height: 1.8,
                           color: Colors.black87,
                         ),),],),),),
               const SizedBox(height: 40),
             ],),),),),),);}

 Widget _buildFeatureSection(String title, String description,) {
   return Row(
     crossAxisAlignment: CrossAxisAlignment.start,
     children: [
       const SizedBox(width: 15),
       Expanded(
         child: Column(
           crossAxisAlignment: CrossAxisAlignment.start,
           children: [
             Text(
               title,
               style: const TextStyle(
                 fontSize: 18,
                 fontWeight: FontWeight.bold,
                 color: Colors.lightBlue,
               ),
             ),
             const SizedBox(height: 5),
             Text(
               description,
               style: const TextStyle(
                 fontSize: 14,
                 color: Colors.black87,
                 height: 1.5,
               ),),],),),],);}}
```

Output:



ShareIt

Welcome to ShareIt!

Secure File Sharing Made Simple

Share your files securely with password protection. Generate unique links for your files and share them with confidence.

Custom Aliases

Create memorable links by using custom aliases for your shared files. Make your links personal and easy to remember.

Password Protection

Keep your shared files private with password protection. Only users with the correct password can access your files.

Easy Download

Recipients can easily download your files by simply entering the password. No account required!

How It Works

1. Select any file you want to share
2. Choose a custom alias for your link
3. Set a secure password
4. Share the generated link with others
5. Recipients enter the password to

Conclusion:

In conclusion, the experiment was successfully completed by creating a simple homepage in Flutter, which allowed for hands-on exploration of various widgets. The process involved utilizing basic widgets like Text, Container, Column to design a functional UI. This exercise provided valuable insights into how widgets are structured and combined to build layouts in Flutter. By experimenting with different widgets, the understanding of Flutter's declarative UI framework was enhanced, laying the foundation for more complex app development in the future.