MAD PWA Lab

EXPERIMENT NO.2

Name : Pranathi Narsupalli
Div : D15B Roll no. : 45

AIM : To design Flutter UI by including common widgets.

THEORY :

Flutter widgets are the building blocks of Flutter applications, allowing developers to create user interfaces by assembling and combining various widgets. Widgets in Flutter are objects that represent parts of the user interface, such as buttons, text, images, layout structures, and more. They are the primary elements used to compose the visual and interactive components of a Flutter app. Flutter widgets empower developers to craft dynamic and expressive user interfaces effortlessly. From foundational elements like Container and Text to complex structures like Row, Column, and Stack, Flutter's widget-based architecture offers a rich toolkit for building aesthetically pleasing and responsive mobile applications.

Flutter Widgets:

1. StatelessWidget:

- Represents widgets that are immutable and don't change over time.
- They don't store or manage any mutable state.

2. StatefulWidget:

- Represents widgets that can change dynamically during the lifetime of the application.
- They have mutable state, and changes in state trigger a rebuild of the widget tree.

3. Container:

- A box model that can contain other widgets and provides features like padding, margin, and decoration.
- Often used to group and style other widgets.

4. Row and Column:

- Used to arrange child widgets horizontally (Row) or vertically (Column).
- Flexibility in distributing space among child widgets.

5. Stack:

- Allows widgets to be overlaid on top of each other.
- Widgets are positioned relative to the edges or the center of the stack.

6. ListView:

- A scrollable list of widgets.
- Can display a large number of children, either in a vertical or horizontal direction.

7. Text:

- Displays a styled text string.
- Supports rich formatting and styling options.

8. Image:

- Displays images from various sources, such as assets, the network, or memory.
- Supports caching and different fit options.

9. AppBar:

- A material design app bar that typically contains the app's title and various actions.
- Positioned at the top of the screen.

10. Scaffold:

- Represents the basic material design visual structure of a Flutter app.
- Provides a framework for implementing the basic layout structure.

INPUT (Code) :

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('Common Widgets Example'),
        ),
```

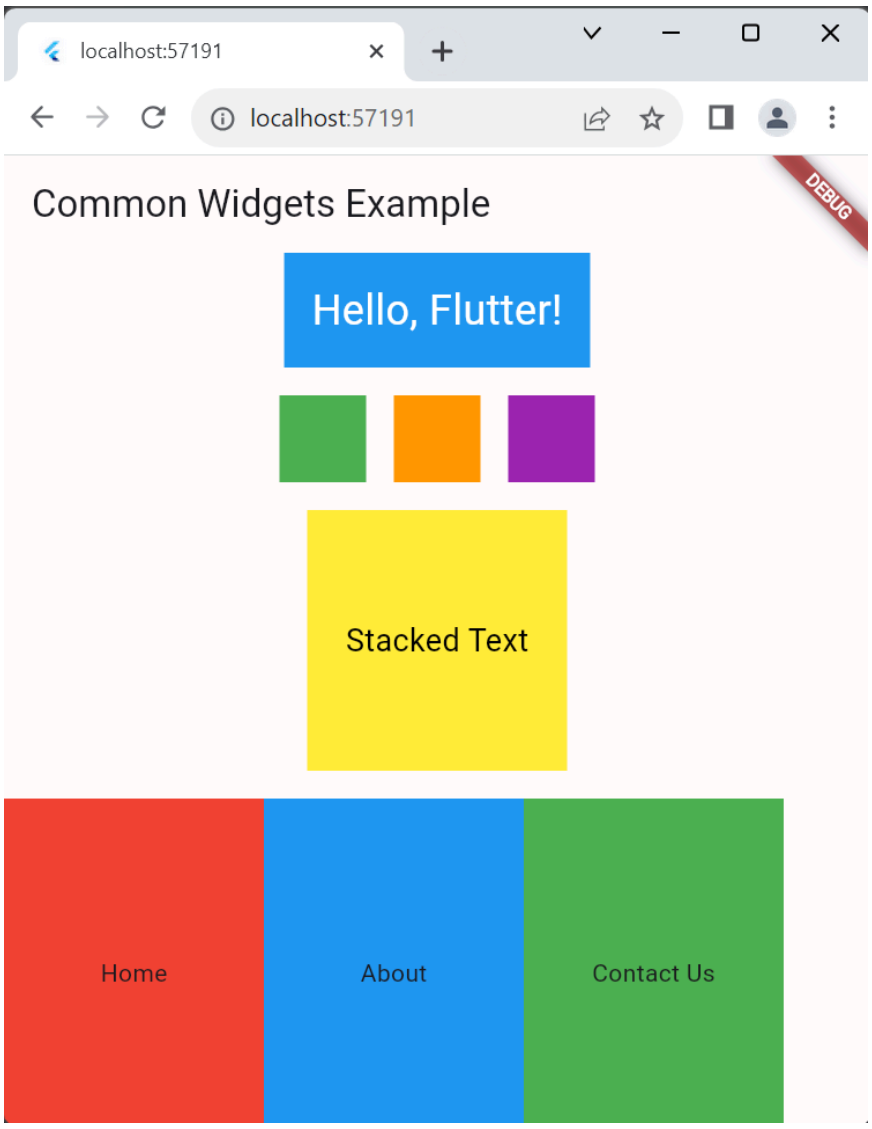
```
      body: MyWidgetContent(),
    ),
   );
  }
}

class MyWidgetContent extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
   return SingleChildScrollView(
    child: Column(
     crossAxisAlignment: CrossAxisAlignment.center,
     children: [
      Container(
       padding: EdgeInsets.all(16.0),
       color: Colors.blue,
       child: Text(
        'Hello, Flutter!',
        style: TextStyle(
         fontSize: 24.0,
         color: Colors.white,
        ),
       ),
      ),
      SizedBox(height: 16.0),
      Row(
       mainAxisAlignment: MainAxisAlignment.center,
       children: [
        Container(
         color: Colors.green,
         width: 50.0,
         height: 50.0,
        ),
```

```
    SizedBox(width: 16.0),
    Container(
      color: Colors.orange,
      width: 50.0,
      height: 50.0,
    ),
    SizedBox(width: 16.0),
    Container(
      color: Colors.purple,
      width: 50.0,
      height: 50.0,
    ),
  ],
),
SizedBox(height: 16.0),
Stack(
  alignment: Alignment.center,
  children: [
    Container(
      width: 150.0,
      height: 150.0,
      color: Colors.yellow,
    ),
    Text(
      'Stacked Text',
      style: TextStyle(fontSize: 18.0, color: Colors.black),
    ),
  ],
),
SizedBox(height: 16.0),
Container(
  height: 200.0,
  child: ListView(
```

```
            scrollDirection: Axis.horizontal,
            children: [
              Container(
                width: 150.0,
                color: Colors.red,
                child: Center(
                  child: Text('Item 1'),
                ),
              ),
              Container(
                width: 150.0,
                color: Colors.blue,
                child: Center(
                  child: Text('Item 2'),
                ),
              ),
              Container(
                width: 150.0,
                color: Colors.green,
                child: Center(
                  child: Text('Item 3'),
                ),
              ),
            ],
          ),
        ),
      ],
    ),
  );
 }
}
```

OUTPUT :



CONCLUSION : In this experiment, we successfully understood, designed and implemented a responsive Flutter UI by incorporating fundamental widgets like Column, Row, Stack, Container, Text, and ListView. This versatile set of widgets allowed us to create a visually engaging and interactive user interface, showcasing the flexibility and power of Flutter for crafting diverse and dynamic app layouts using Android Studio.