EXPERIMENT NO.2

Experiment No 1 **2: To design Flutter UI by including common widgets.**	
ROLL NO	**35**
NAME	**Gaurang Milind Mapuskar**
CLASS	**D15-B**
SUBJECT	**MAD & PWA Lab**
LO-MAPPE D	

Theory:

Flutter is an open-source UI toolkit developed by Google for building natively compiled applications across multiple platforms, including mobile, web, and desktop, from a single codebase. It utilizes the Dart programming language and offers a rich set of customizable widgets for creating beautiful, fast, and responsive user interfaces. Widgets in Flutter are building blocks of the UI, ranging from basic elements like buttons and text to complex layouts like lists and grids. These widgets are highly flexible, enabling developers to create immersive and dynamic user experiences with ease, making Flutter a powerful choice for modern app development.

Basic Widgets:

- **Appbar**: A fundamental Flutter widget representing a material design app bar at the top of the screen. It typically contains a title, leading and trailing icons, and actions. It's commonly used for navigation and displaying key information.

- **Scaffold**: A foundational widget providing layout structure for apps, offering features like app bars, drawers, bottom navigation, and floating action buttons, simplifying app development.

- **Column**: A layout widget arranging its children vertically from top to bottom, enabling flexible alignment and spacing along the vertical axis for building UIs.

- **Text**: A widget displaying text content with customizable properties like font size, color, alignment, and decoration, facilitating text rendering in Flutter apps.

- **Container**: A versatile widget serving as a visual element with customizable properties such as size, color, padding, margin, alignment, decoration, and child widget composition, offering flexibility in UI design and layout.

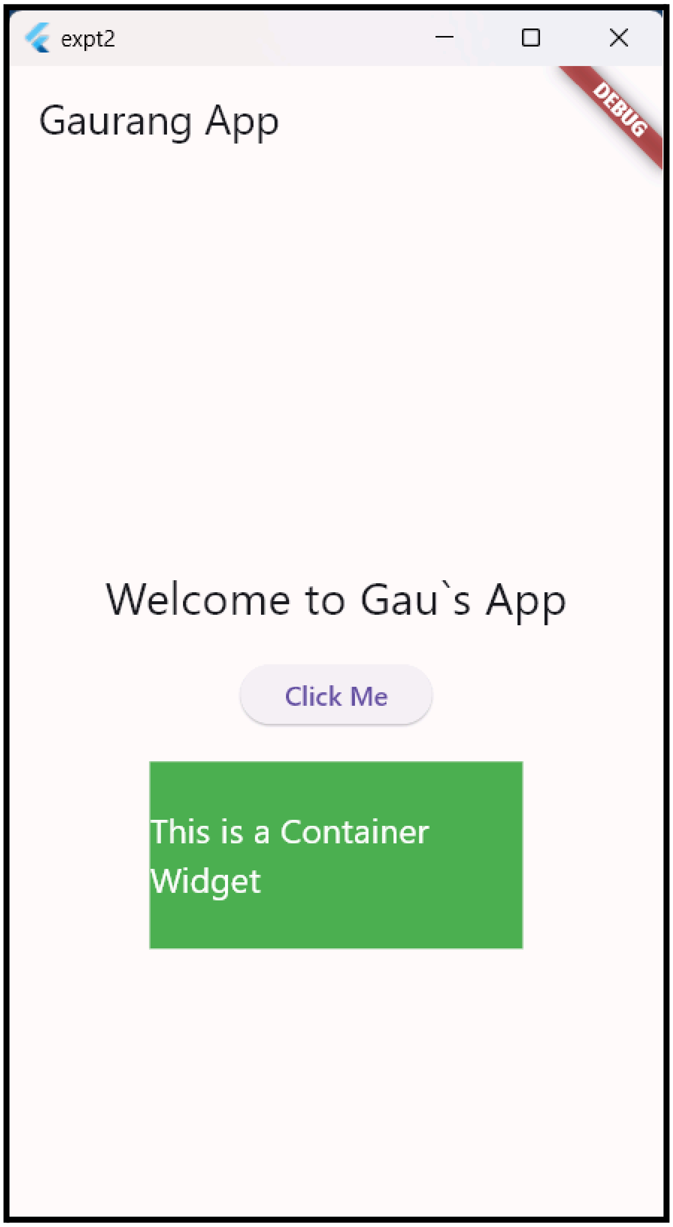
- **Row**: A layout widget arranging its children horizontally from left to right within the parent widget's constraints. It enables flexible alignment and spacing along the horizontal axis, allowing developers to create UIs with multiple widgets laid out side by side.

Code:

```
import 'dart:math';

import 'package:flutter/material.dart';

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

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

```
class MyHomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Gaurang App'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            const Text(
              'Welcome to Gau`s App',
              style: TextStyle(fontSize: 24.0),
            ),
            const SizedBox(height: 20.0),
            ElevatedButton(
              onPressed: () {
                print("Button was Clicked");
              },
              child: const Text('Click Me'),
            ),
            const SizedBox(height: 20.0), // Add some spacing
            Container(
              // Adding Container widget
              width: 200.0,
              height: 100.0,
              color: Colors.green,
              child: const Center(
                child: Text(
                  'This is a Container Widget',
                  style: TextStyle(fontSize: 18.0, color: Colors.white),
                ),
              ),
            ),
          ],
        ),
      ),
    );
  }
}
```

Output:



Gaurang App

Welcome to Gau`s App

This is a Container Widget