EXPERIMENT 2

AIM : To design flutter UI by including common widgets

Theory :
Widgets are the fundamental components of the user interface in Flutter. Everything in Flutter is a widget, and the application's layout and visual components are made with widgets. You can design your user interface (UI) using a variety of pre-built widgets from Flutter, or you can make your own unique widgets.

There are two primary categories of widgets in Flutter:

● Stateless Widgets: These widgets have immutable properties, which means that once they are set, they cannot change. For areas of the user interface that don't alter dynamically, stateless widgets are utilised
.
● Stateful Widgets: These widgets have the ability to dynamically alter in response to user input, data modifications, and other variables. Stateful widgets are employed in UI components that require

Because Flutter widgets are modular, you can nest and combine them to create intricate user interfaces. The framework offers a wide range of widgets for typical user interface components including layouts, text, photos, buttons, and more.

The expressiveness and flexibility of Flutter apps are enhanced by the composability and responsiveness of Flutter widgets.

Widgets in Flutter are the fundamental building blocks for managing layout, interactivity, animation, and other features in addition to their aesthetic value. The following are some of the main characteristics of Flutter widgets:

Widget Tree:
● Flutter applications are built using a widget tree. The widget tree is a hierarchical structure of widgets, where each widget can contain other widgets as its children.

● The root of the widget tree is typically the MaterialApp or CupertinoApp widget, depending on the design language you are using (Material Design for Android, Cupertino for iOS).

Composable and Nestable:
● Flutter widgets are composable, meaning you can combine and nest them to create complex UIs. This composability allows for the easy construction of sophisticated user interfaces.

Layout Widgets:
● Flutter provides a variety of layout widgets to structure and position other widgets on the screen. Examples include Container, Row, Column, Stack, Expanded, and more.

Gesture Handling:
● Flutter has widgets for handling gestures like taps, swipes, and drags. For example, GestureDetector is a versatile widget that can recognize various gestures and trigger corresponding callbacks.

Animation:
● Flutter makes it easy to create animations using widgets. The Animation and Tween classes, along with widgets like AnimatedContainer and Hero, allow you to add motion and transitions to your UI.

Material Design and Cupertino Widgets:
● Flutter provides widgets that follow the Material Design guidelines (for Android) and Cupertino design (for iOS). For example, AppBar, BottomNavigationBar, TextField, and others adhere to the respective platform's design principles.

Code:

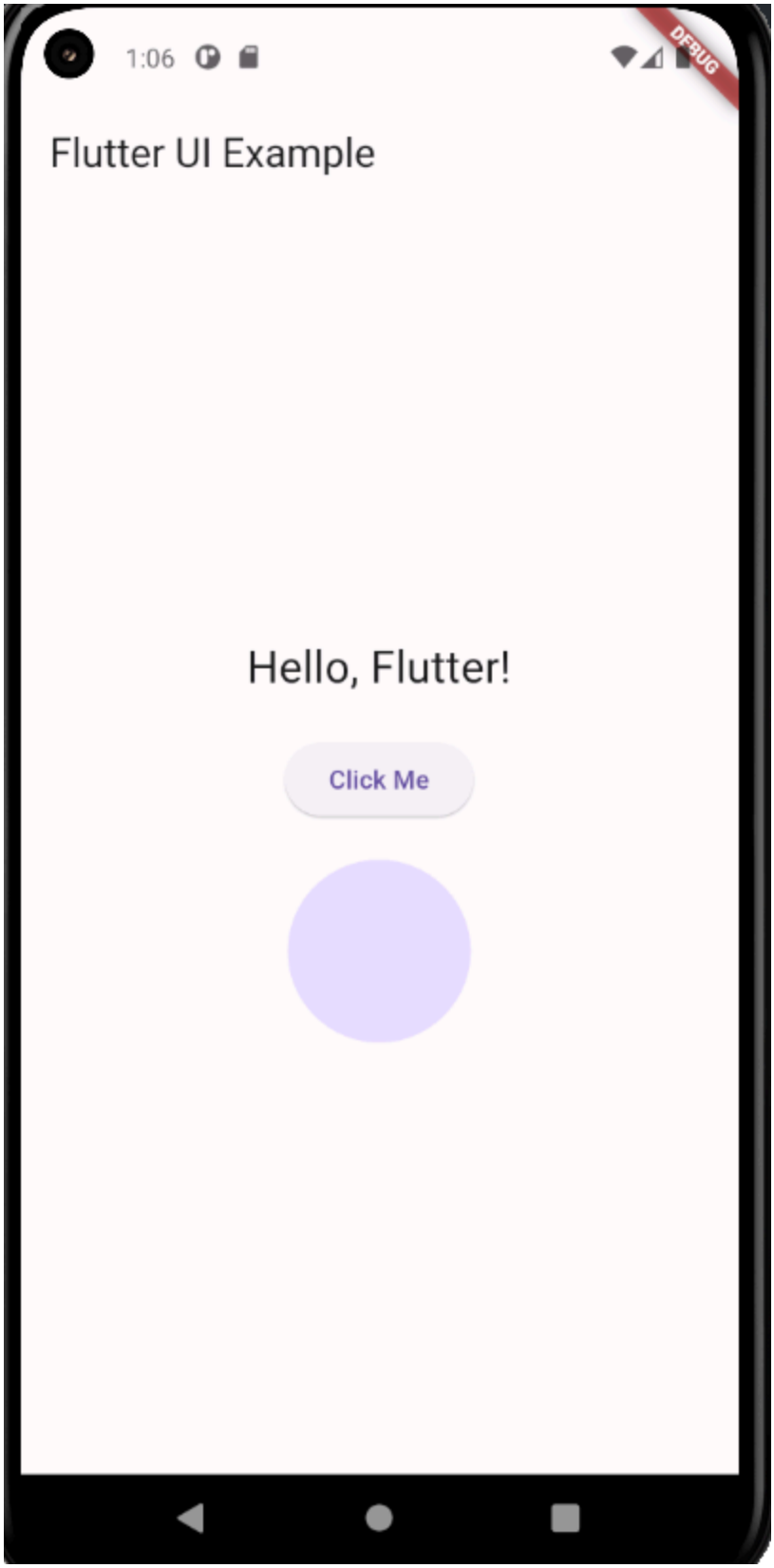
main.dart

lib > main.dart > ...

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

Run | Debug | Profile
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('Flutter UI Example'),
        ), // AppBar
        body: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Text(
                'Hello, Flutter!',
                style: TextStyle(fontSize: 24),
              ), // Text
              SizedBox(height: 20),
              ElevatedButton(
                onPressed: () {
                  // Add your button functionality here
                },
                child: Text('Click Me'),
              ), // ElevatedButton
              SizedBox(height: 20),
              CircleAvatar(
                radius: 50,
                backgroundImage: NetworkImage(
                    'https://example.com/your_image_url.jpg'), // NetworkImage
              ), // CircleAvatar
            ],
          ), // Column
        ), // Center
      ), // Scaffold
    ); // MaterialApp
  }
}
```


1:06
DEBUG
Flutter UI Example
Hello, Flutter!
Click Me

Conclusion : We Have Successfully designed Flutter UI by including common widgets.