Name : Pooja Panjwani **Exp : 3** D15B_51

Aim: To include icons, images, and fonts in the Flutter app

Theory:

- *Icons in Flutter:*

Overview: Icons in Flutter are visual symbols used to represent actions, objects, or concepts in the user interface. Flutter provides access to a wide range of icons from the Material Design and Cupertino icon sets.

Common Widgets and Classes:

1. `Icon` Widget:
 - The `Icon` widget is used to display icons in Flutter.
 - Example: Icon(Icons.star, color: Colors.yellow, size: 60.0);

2. `IconButton` Widget:
 - The `IconButton` widget is a button that contains an icon.
 - Example:

```
 IconButton(
  icon: Icon(Icons.add),
     onPressed: ()
        {
     // Add your action here
         },
           )
```

- *Images in Flutter:*

Overview:

Images are used to display visual content in a Flutter app. Flutter supports various image formats, including JPEG, PNG, GIF, and WebP. You can display images from the network, assets, or the device.

Common Widgets and Classes:

1. Image Widget:
- The `Image` widget is used to display images in Flutter.
 - Example: Image.network('https://example.com/image.jpg');

2. . Image.network` Widget:
- The `Image.network` widget loads and displays images from the network.
- Example: Image.network('https://example.com/image.jpg');

- *Fonts in Flutter:*

Overview: Fonts in Flutter are used to define the style and appearance of text. Flutter supports custom fonts, allowing you to use a variety of typefaces and styles in your app.

Common Widgets and Classes:

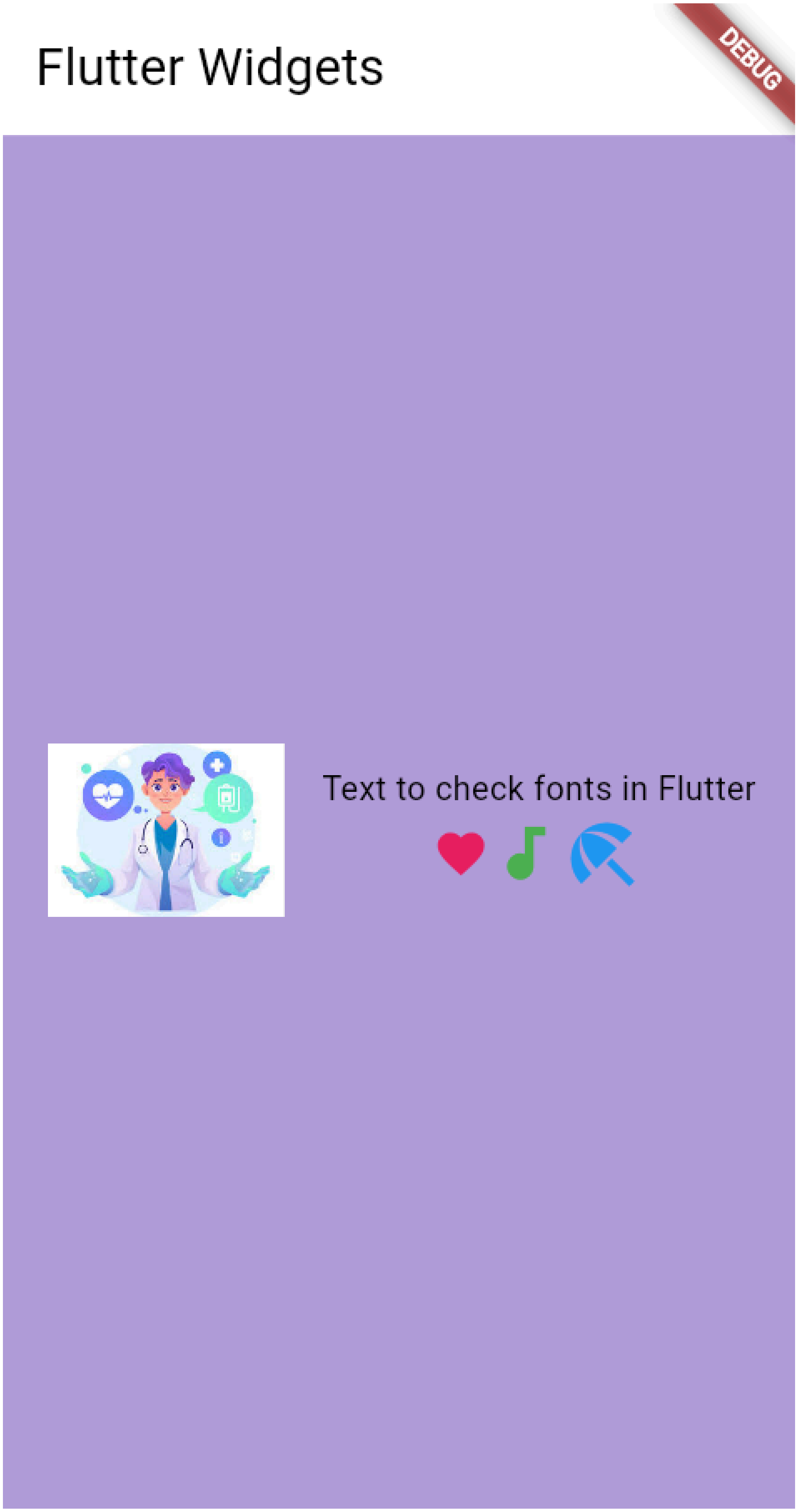
1. TextStyle` Class:
- The `TextStyle` class is used to define the style of text, including font family, size, color, etc.
- Example: TextStyle(fontFamily: 'Roboto', fontSize: 16.0, color: Colors.black,);

2. Google Fonts Package:
- The `google_fonts` package provides a convenient way to use Google Fonts in your Flutter app.
- Example: GoogleFonts.openSans(textStyle: TextStyle(fontSize: 16.0, color: Colors.black),

Name : Pooja Panjwani **Exp : 3** D15B_51

Code :

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

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSwatch(primarySwatch: Colors.deepPurple),
        // Removed useMaterial3 as it's not a valid property in ThemeData
      ),
      home: const MyHomePage(title: 'Flutter Widgets'),
    );
  }
}

class MyHomePage extends StatefulWidget {
  final String title;

  const MyHomePage({Key? key, required this.title}) : super(key: key);

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.title),
      ),
      body: Center(
child: Column(
  mainAxisAlignment: MainAxisAlignment.center,
  children: <Widget>[
    Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        // ignore: sized_box_for_whitespace
        Container(
          width: 100,
          height: 100,
          child: Image.asset('assets/images/icLogin.png'),
        ),
        const SizedBox(width: 16), // Add spacing between image and text
        // ignore: prefer_const_constructors
        Column(
          mainAxisAlignment: MainAxisAlignment.center,
          // ignore: prefer_const_literals_to_create_immutables
          children: <Widget>[
            const Text(
              'Text to check fonts in Flutter',
              style: TextStyle(fontFamily: 'ProtestRiot'),
            ),
            const Row(
              mainAxisAlignment: MainAxisAlignment.spaceAround,
              children: <Widget>[
                Icon(
                  Icons.favorite,
                  color: Colors.pink,
                  size: 24.0,
                  semanticLabel: 'Text to announce in accessibility modes',
                ),
                Icon(
                  Icons.audiotrack,
                  color: Colors.green,
                  size: 30.0,
                ),
                Icon(
                  Icons.beach_access,
                  color: Colors.blue,
                  size: 36.0,
                        ),
                      ],
                    ),
                  ],
                ),
              ],
            ),
          ],
        ),
      ),
    );
  }
}
```



Flutter Widgets

Text to check fonts in Flutter

♥ ♪ ☂

Conclusion : In summary, this experiment taught us how to use icons, images, and custom fonts in Flutter apps. We learned how to make our apps look better by adding visual elements like icons and images, and how to give them a unique style with custom fonts. These additions not only make the app more visually appealing but also give us more options for creativity. With this knowledge, we can create more attractive and personalized apps in Flutter for our projects.