Experiment 3
Name:Khushi Jeswani
Div: D15A
Roll no: 26

Aim: To include icons, images, fonts in Flutter app Theory:

Text Widget:

- The Text widget is used to display textual content within a Flutter application.
- It allows you to customize the appearance of text, including font family, size, weight, style, color, alignment, and more.
- Text widgets support both single-line and multi-line text.
- You can use Text widgets within various Flutter layout widgets such as Column, Row, ListView, etc., to display text in different parts of the screen.
- Text widgets can also be styled dynamically using theming or state management techniques.

Button Widget:

- Flutter provides several types of buttons, including ElevatedButton, TextButton, OutlinedButton, and IconButton.
- Buttons are interactive elements that users can tap or click to trigger actions or events in the application.
- Each type of button has its own style and appearance, but they all support customization of properties such as text, color, padding, shape, and onPressed callback.
- Buttons can be placed within Flutter layout widgets like Row, Column, Container, etc., to create interactive user interfaces.
- Flutter buttons can also be disabled or enabled based on certain conditions, and their appearance can be adjusted accordingly.

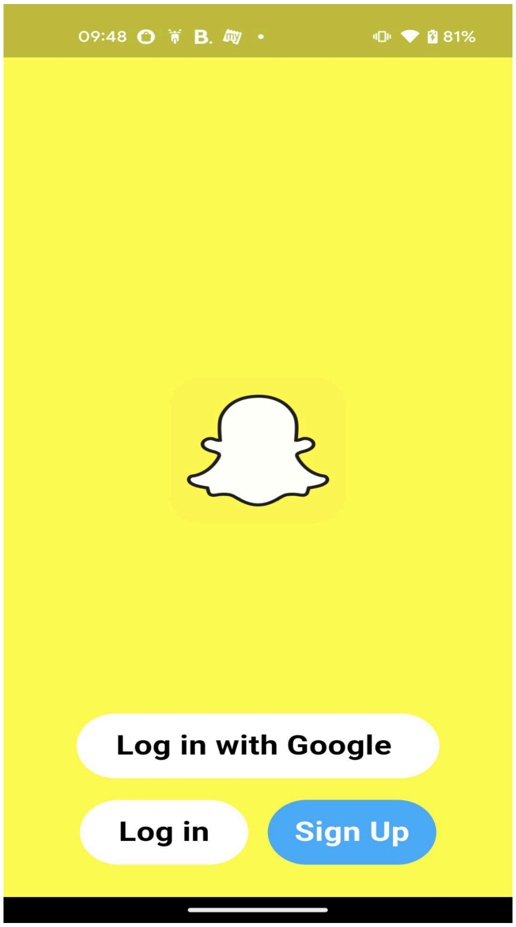
Image Widget:

- The Image widget is used to display images within a Flutter application.
- It supports various image formats such as JPEG, PNG, GIF, WebP, and SVG (using the flutter_svg package).
- Images can be loaded from different sources including local assets, network URLs, memory, and file paths.
- The Image widget provides properties to control the image's size, alignment, fit, repeat mode, color filters, and more.
- Flutter also provides advanced features for image caching, resizing, and processing to optimize performance and memory usage.
- Images are often used to enhance the visual appeal of an application and to convey information to the user through graphics and icons.

Code:

```
import 'package:flutter/material.dart';
import 'package:snapchatfinal/page/chat_screen.dart';
import 'reels_page.dart';
import 'stories_page.dart';
import 'caemra_page.dart';
import 'chat_page.dart';
import 'initial_page.dart';
import 'location_page.dart';

class HomePage extends StatefulWidget {
  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  int _selectedIndex = 0;
  static const List<Widget> _widgetOptions = <Widget>[
    LocationPage(),
    ChatPage(),
    CameraPage(),
    StoriesPage(),
    ReelPage(),
  ];

  void _onItemTapped(int index) {
    setState(() {
      _selectedIndex = index;
      if(index == 1){ // Check if the chat icon is tapped
        Navigator.push(
          context,
          MaterialPageRoute(builder: (context) => ChatPage2()), // Navigate to ChatScreen
        );
      }
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      body: SafeArea(child: _widgetOptions[_selectedIndex]),
      bottomNavigationBar: BottomNavigationBar(
        items: <BottomNavigationBarItem>[
BottomNavigationBarItem(
  backgroundColor: Colors.black,
  icon: Icon(
    Icons.location_on_outlined,
    size: 25.0,
    color: Colors.white,
  ),
  label: "",
),
BottomNavigationBarItem(
  backgroundColor: Colors.black,
  icon: Icon(
    Icons.chat_bubble_outline_rounded,
    size: 25.0,
    color: Colors.white,
  ),
  label: "",
),
BottomNavigationBarItem(
  backgroundColor: Colors.black,
  icon: Icon(
    Icons.camera_alt_outlined,
    size: 25.0,
    color: Colors.white,
  ),
  label: "",
),
BottomNavigationBarItem(
  backgroundColor: Colors.black,
  icon: Icon(
    Icons.group_outlined,
    size: 25.0,
    color: Color(0XFF10ACFF),
  ),
  label: "",
),
BottomNavigationBarItem(
  backgroundColor: Colors.black,
  icon: Icon(
    Icons.play_arrow_outlined,
    size: 25.0,
    color: Colors.white,
  ),
  label: "",
          ),
        ],
        type: BottomNavigationBarType.fixed,
        currentIndex: _selectedIndex,
        selectedItemColor: Color(0XFF10ACFF),
        backgroundColor: Colors.black,
        onTap: _onItemTapped,
        unselectedItemColor: Colors.white,
      ),
    );
  }
}
```

App UI:



Widgets used: Image, Text, Icons,

Conclusion: Thus, understood the use of Icons, images and font widgets in Flutter. Implemented Icons, Images and fonts in my Flutter application.