Experiment 3

Name: Sonali Makhijani
Div: D15A
Roll no: 34

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

Theory:

1. 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.

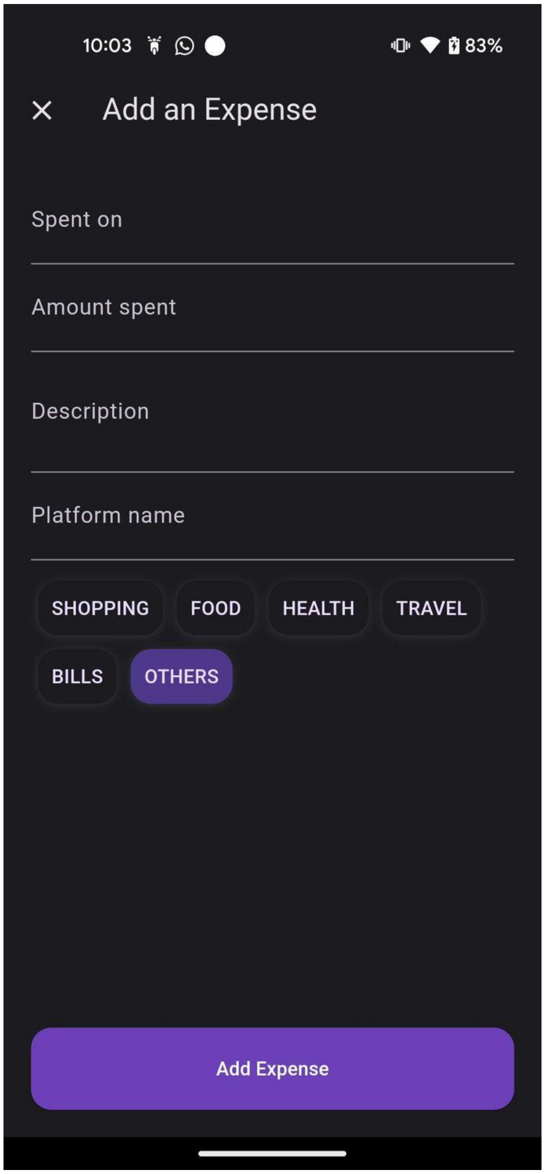
2. Button Widget:
    - Flutter provides several types of buttons, including ElevatedButton, TextButton, OutlinedButton, and IconButton.
    - Buttons are interactive elements that users can tap or click to trigger actions or events in the application.
    - Each type of button has its own style and 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.

3. 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:flutter_bloc/flutter_bloc.dart';

import '/core/constants/styles.dart';
import '/domain/bloc/expenses_bloc/expenses_bloc.dart';
import '../add_or_edit_expense/add_or_edit_expense.dart';
import '/presentation/screens/search/search_page.dart';
import '/data/repositories/common_interfaces/expenses_repo_interface.dart';
import '/presentation/screens/homescreen/components/expense_tile.dart';
import '/presentation/screens/homescreen/components/category_tile.dart';
import '/presentation/screens/homescreen/components/app_drawer.dart';
import '/presentation/widgets/error_widget.dart';
import '/presentation/widgets/loader.dart';
import '../view_expenses/view_expenses_list.dart';

class HomeScreen extends StatefulWidget {
  const HomeScreen({super.key});

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  @override
  void didChangeDependencies() {
    context.read<ExpensesBloc>().add(ExpensesLoadEvent());
    super.didChangeDependencies();
  }

  final GlobalKey<ScaffoldState> _key = GlobalKey();
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      key: _key,
      appBar: AppBar(
        title: const Text(
          'expensio',
          style: TextStyle(
            fontWeight: FontWeight.w500,
          ),
        ),
        centerTitle: true,
        leading: IconButton(
          icon: const Icon(Icons.menu),
          onPressed: () => _key.currentState!.openDrawer(),
        ),
        actions: [
          IconButton(
            onPressed: () {
              Navigator.push(context,
                  MaterialPageRoute(builder: (context) => const SearchPage()));
            },
            icon: const Icon(Icons.search_rounded),
          )
        ],
      ),
drawer: const AppDrawer(),
floatingActionButton: FloatingActionButton(
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (_) => const AddOrEditExpensePage(),
        fullscreenDialog: true,
        barrierDismissible: true,
      ),
    );
  },
  child: const Icon(Icons.add),
),
body: BlocBuilder<ExpensesBloc, ExpensesState>(
  builder: (context, state) {
    if (state is ExpensesLoadingState) {
      return const Loader();
    } else if (state is ExpensesErrorState) {
      return CustomErrorWidget(
        exceptionCaught: state.exception,
        onPressed: () {
          context.read<ExpensesBloc>().add(ExpensesLoadEvent());
        },
      );
    } else {
      return ListView(
        padding: const EdgeInsets.all(paddingDefault),
        physics: const BouncingScrollPhysics(),
        children: [
          Container(
            padding: const EdgeInsets.all(paddingDefault),
            margin: const EdgeInsets.all(5),
            decoration: BoxDecoration(
              color: Theme.of(context)
                  .colorScheme
                  .surfaceVariant
                  .withOpacity(0.4),
              borderRadius: borderRadiusDefault,
            ),
            child: Column(
              children: [
                const Text('Total Monthly Expenses'),
                Text(
                  "₹${context.read<ExpensesBloc>().allExpensesSum.toString()}",
                  style: const TextStyle(
                      fontSize: 22, fontWeight: FontWeight.w600),
                )
              ],
            ),
          ),
          ConstrainedBox(
            constraints:
                const BoxConstraints(maxHeight: 500, maxWidth: 300),
            child: GridView.count(
              physics: const NeverScrollableScrollPhysics(),
              shrinkWrap: true,
                    crossAxisCount: 2,
                    childAspectRatio: 4 / 2.4,
                    children: List.generate(
                      ExpenseCategory.values.length,
                      (index) => ConstrainedBox(
                        constraints:
                            const BoxConstraints(maxHeight: 150, maxWidth: 300),
                        child: CategoryTile(index),
                      ),
                    ),
                  ),
                ),
                Row(
                  mainAxisAlignment: MainAxisAlignment.spaceBetween,
                  children: [
                    const Text(
                      "Recent Expenses",
                      style:
                          TextStyle(fontWeight: FontWeight.w600, fontSize: 18),
                    ),
                    TextButton(
                      onPressed: () {
                        Navigator.push(
                          context,
                          MaterialPageRoute(
                            builder: (context) => const ViewExpenses(),
                          ),
                        );
                      },
                      child: const Text('more'),
                    )
                  ],
                ),
                BlocBuilder<ExpensesBloc, ExpensesState>(
                    builder: (context, state) {
                  return ListView.builder(
                    physics: const NeverScrollableScrollPhysics(),
                    shrinkWrap: true,
                    itemCount:
                        context.watch<ExpensesBloc>().listOfExpenses.length,
                    itemBuilder: (context, index) {
                      final expense =
                          context.watch<ExpensesBloc>().listOfExpenses[index];
                      return ExpenseTile(expense: expense);
                    },
                  );
                }),
              ],
            );
          }
        },
      ),
    );
  }
}
```

App UI:



Widgets used: Text, Icons,

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