Name:Sumeet Kumar Singh Roll no:62 Div:D15B

Aim: To design Flutter ui by including common widgets

Theory:

Flutter UI widgets are building blocks for creating visually appealing and interactive user interfaces in Flutter, a UI toolkit developed by Google. These widgets are pre-designed components that can be customized and arranged to create rich and dynamic user experiences. Flutter offers a wide range of widgets for various purposes, including structural layout (such as rows, columns, and grids), input handling (buttons, text fields), navigation (tabs, drawers), and visual styling (containers, images). With Flutter's widget-based architecture, developers can efficiently create cross-platform applications with consistent behavior and appearance across different devices and platforms.

In the provided Flutter code, there are several types of widgets being used:

StatelessWidget:

MyApp, MyAppBar, MyBody, and ProjectForm are all stateless widgets.
These widgets do not maintain any state internally and their UI representation depends solely on their configuration.
StatefulWidget:

MyHomePage and _ProjectFormState are stateful widgets.
These widgets have associated mutable state that can change over time, influencing the widget's appearance or behavior.
Material Components:

MaterialApp, Scaffold, AppBar, Text, TextField, ElevatedButton, InkWell, and ListView are Material Design widgets provided by the Flutter framework.
These widgets implement the Material Design language for consistent and visually appealing UIs on Android and iOS platforms.
Layout Widgets:

Column, Row, and Expanded are layout widgets used to organize other widgets in a vertical or horizontal arrangement.
SizedBox is used to introduce spacing between widgets.
Text Widgets:

Text widget is used to display text on the screen.
Input Widgets:

TextField widget allows users to input text.
ElevatedButton is a button widget with a Material Design elevation effect.

InkWell provides a touch ripple effect when tapped and is used for making a clickable link.
Other Widgets:

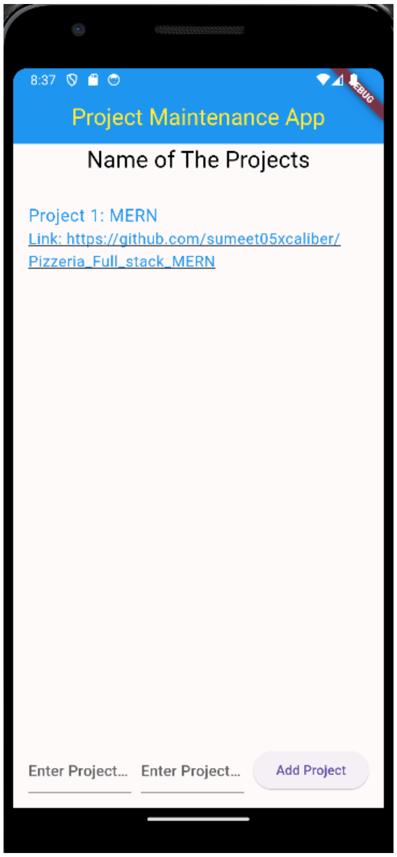
ListTile represents a single fixed-height row that typically contains some text as well as an optional leading or trailing icon.
PreferredSizeWidget is an interface implemented by MyAppBar to provide custom sizing information for the app bar.
Function is a typedef for a function.

Code:

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Project Maintenance App',
      home: MyHomePage(),
    );
  }
}

class MyHomePage extends StatefulWidget {
  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  List<ProjectData> projects = [];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: MyAppBar(),
      body: MyBody(projects: projects, onAddProject: _addProject),
    );
  }

  void _addProject(ProjectData project) {
    setState(() {
      projects.add(project);
    });
  }
}

class ProjectData {
  final String name;
  final String link;

  ProjectData({required this.name, required this.link});
}

class MyAppBar extends StatelessWidget implements PreferredSizeWidget {
  @override
  Widget build(BuildContext context) {
    return AppBar(
      title: Text(
        'Project Maintenance App',
        style: TextStyle(fontSize: 24, color: Colors.yellow),
      ),
      backgroundColor: Colors.blue,
      centerTitle: true,
    );
  }

  @override
  Size get preferredSize => Size.fromHeight(kToolbarHeight);
}

class MyBody extends StatelessWidget {
  final List<ProjectData> projects;
  final Function(ProjectData) onAddProject;

  MyBody({required this.projects, required this.onAddProject});

  @override
  Widget build(BuildContext context) {
    return Column(
      children: [
        Text(
          'Name of The Projects',
          style: TextStyle(fontSize: 24, color: Colors.black),
        ),
        SizedBox(height: 20),
        Expanded(
          child: ListView.builder(
            itemCount: projects.length,
            itemBuilder: (context, index) {
              return ListTile(
                title: Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Text(
                      'Project ${index + 1}: ${projects[index].name}',
                      style: TextStyle(fontSize: 18, color: Colors.blue),
                    ),
                    if (projects[index].link.isNotEmpty)
                      InkWell(
                        onTap: () {
                          _launchURL(projects[index].link);
                        },
                        child: Text(
                          'Link: ${projects[index].link}',
                          style: TextStyle(
                            fontSize: 16,
                            color: Colors.blue,
                            decoration: TextDecoration.underline,
                          ),
                        ),
                      ),
                  ],
                ),
              );
            },
          ),
        ),
        ProjectForm(onAddProject: onAddProject),
      ],
    );
  }

  void _launchURL(String url) async {
    print('Launching URL: $url');
    // You can open the link in the default web browser here.
  }
}
```

```
class ProjectForm extends StatefulWidget {
  final Function(ProjectData) onAddProject;

  ProjectForm({required this.onAddProject});

  @override
  _ProjectFormState createState() => _ProjectFormState();
}

class _ProjectFormState extends State<ProjectForm> {
  final TextEditingController _projectController = TextEditingController();
  final TextEditingController _linkController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        children: [
          Row(
            children: [
              Expanded(
                child: TextField(
                  controller: _projectController,
                  decoration: InputDecoration(
                    hintText: 'Enter Project Name',
                  ),
                ),
              ),
              SizedBox(width: 10),
              Expanded(
                child: TextField(
                  controller: _linkController,
                  decoration: InputDecoration(
                    hintText: 'Enter Project Link (Optional)',
                  ),
                ),
              ),
              SizedBox(width: 10),
              ElevatedButton(
                onPressed: () {
                  String projectName = _projectController.text.trim();
                  String projectLink = _linkController.text.trim();
                  if (projectName.isNotEmpty) {
```

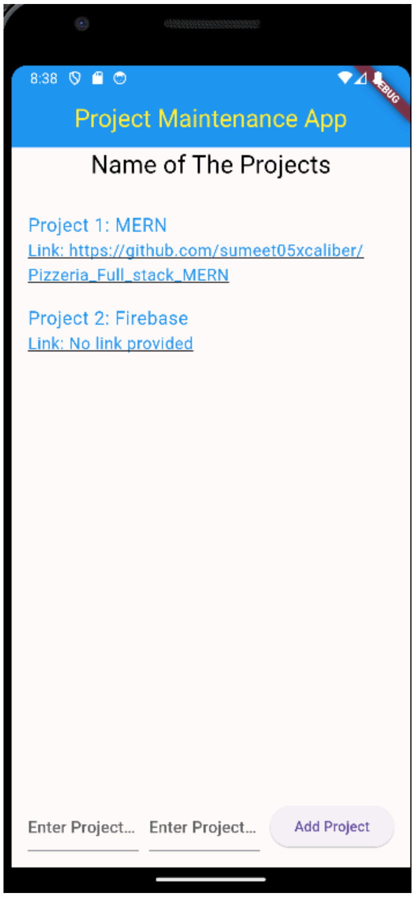
```
                widget.onAddProject(ProjectData(
                  name: projectName,
                  link: projectLink,
                ));
                _projectController.clear();
                _linkController.clear();
              }
            },
            child: Text('Add Project'),
          ),
        ],
      ),
    ],
  ),
 );
 }
}
```



Project Maintenance App

Name of The Projects

Project 1: MERN
Link: https://github.com/sumeet05xcaliber/Pizzeria_Full_stack_MERN



Project Maintenance App

Name of The Projects

Project 1: MERN
Link: https://github.com/sumeet05xcaliber/Pizzeria_Full_stack_MERN

Project 2: Firebase
Link: No link provided

Conclusion: From the above experiment i have successfully understood the working of widgets and its implementation to build different types of ui components in flutter.