Arjun Sudan **Batch :- C**

D15A_56

Experiment no. :- 05

Aim :- : To apply navigation,routing and gestures in Flutter.

Theory :-

In Flutter, navigation, routing, and gestures are essential concepts for creating interactive and navigable user interfaces.

Navigation: Navigation refers to the process of moving between different screens or pages within a Flutter app. Flutter provides the Navigator widget for managing navigation and routing.

Routing: Routing is the mechanism used to define the paths or routes between different screens in your app. Each route typically corresponds to a different widget or screen in your app.

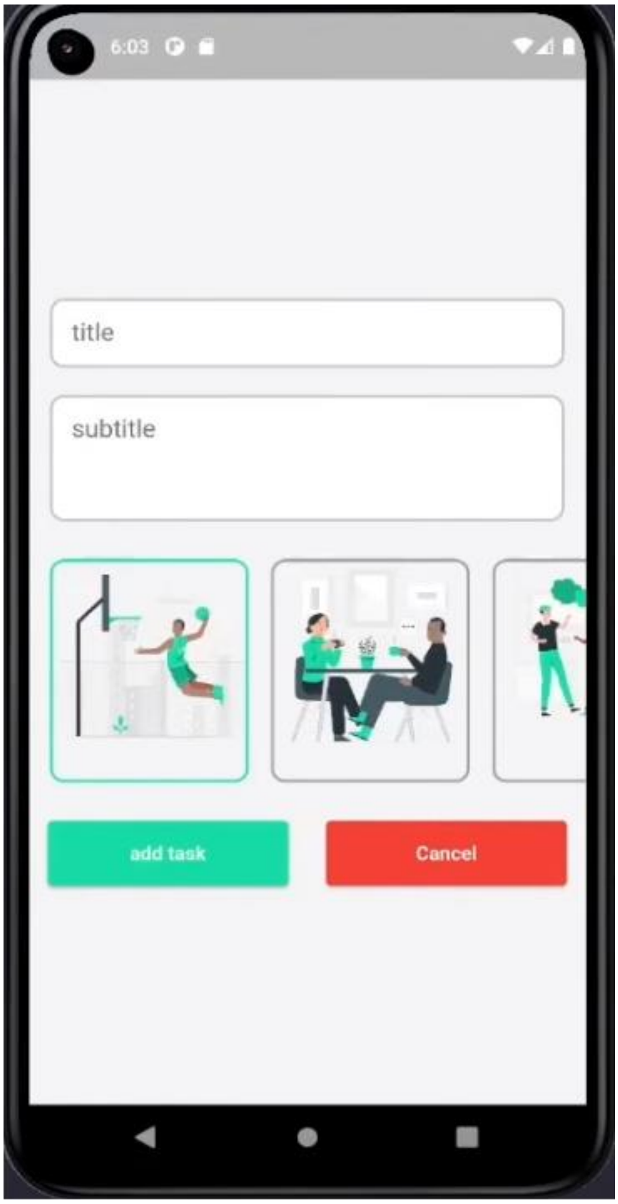
Gesture Detection: Gestures allow users to interact with the app by tapping, dragging, swiping, or performing other touch-based actions. Flutter provides various gesture detection widgets to handle user input.

```
GestureDetector(
onTap: () {
print('Container tapped');
 },
child: Container(
width: 200,
height: 200,
color: Colors.blue,
child: Center(
 child: Text('Tap Me'),
  ),
 ),
)
```

Code :-

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:flutter_to_do_list/const/colors.dart';
import 'package:flutter_to_do_list/data/firestor.dart';

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

  @override
  State<Add_creen> createState() => _Add_creenState();
}

class _Add_creenState extends State<Add_creen> {
  final title = TextEditingController();
  final subtitle = TextEditingController();

  FocusNode _focusNode1 = FocusNode();
  FocusNode _focusNode2 = FocusNode();
  int indexx = 0;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: backgroundColors,
      body: SafeArea(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            title_widgets(),
            SizedBox(height: 20),
            subtite_wedgite(),
            SizedBox(height: 20),
            imagess(),
            SizedBox(height: 20),
            button()
          ],
        ),
      ),
    );
  }

  Widget button() {
    return Row(
      mainAxisAlignment: MainAxisAlignment.spaceAround,
      children: [
        ElevatedButton(
          style: ElevatedButton.styleFrom(
            primary: custom_green,
            minimumSize: Size(170, 48),
          ),
          onPressed: () {
            Firestore_Datasource().AddNote(subtitle.text, title.text, indexx);
            Navigator.pop(context);
          },
          child: Text('add task'),
        ),
        ElevatedButton(
          style: ElevatedButton.styleFrom(
            primary: Colors.red,
            minimumSize: Size(170, 48),
          ),
          onPressed: () {
            Navigator.pop(context);
          },
          child: Text('Cancel'),
        ),
      ],
    );
  }

  Container imagess() {
    return Container(
      height: 180,
      child: ListView.builder(
        itemCount: 4,
        scrollDirection: Axis.horizontal,
        itemBuilder: (context, index) {
          return GestureDetector(
            onTap: () {
              setState(() {
                indexx = index;
              });
            },
            child: Padding(
              padding: EdgeInsets.only(left: index == 0 ? 7 : 0),
              child: Container(
                decoration: BoxDecoration(
                  borderRadius: BorderRadius.circular(10),
                  border: Border.all(
                    width: 2,
                    color: indexx == index ? custom_green : Colors.grey,
                  ),
                ),
                width: 140,
                margin: EdgeInsets.all(8),
                child: Column(
                  children: [
                    Image.asset('images/${index}.png'),
                  ],
                ),
              ),
            ),
          );
        },
      ),
    );
  }

  Widget title_widgets() {
    return Padding(
      padding: const EdgeInsets.symmetric(horizontal: 15),
      child: Container(
        decoration: BoxDecoration(
          color: Colors.white,
          borderRadius: BorderRadius.circular(15),
        ),
        child: TextField(
          controller: title,
          focusNode: _focusNode1,
          style: TextStyle(fontSize: 18, color: Colors.black),
          decoration: InputDecoration(
              contentPadding:
                  EdgeInsets.symmetric(horizontal: 15, vertical: 15),
              hintText: 'title',
              enabledBorder: OutlineInputBorder(
                borderRadius: BorderRadius.circular(10),
                borderSide: BorderSide(
                  color: Color(0xffc5c5c5),
                  width: 2.0,
                ),
              ),
              focusedBorder: OutlineInputBorder(
                borderRadius: BorderRadius.circular(10),
                borderSide: BorderSide(
                  color: custom_green,
                  width: 2.0,
                ),
              )),
        ),
      ),
    );
  }

  Padding subtite_wedgite() {
    return Padding(
      padding: const EdgeInsets.symmetric(horizontal: 15),
      child: Container(
        decoration: BoxDecoration(
          color: Colors.white,
          borderRadius: BorderRadius.circular(15),
        ),
        child: TextField(
          maxLines: 3,
          controller: subtitle,
          focusNode: _focusNode2,
          style: TextStyle(fontSize: 18, color: Colors.black),
          decoration: InputDecoration(
            contentPadding: EdgeInsets.symmetric(horizontal: 15, vertical: 15),
            hintText: 'subtitle',
            enabledBorder: OutlineInputBorder(
              borderRadius: BorderRadius.circular(10),
              borderSide: BorderSide(
                color: Color(0xffc5c5c5),
                width: 2.0,
              ),
            ),
            focusedBorder: OutlineInputBorder(
              borderRadius: BorderRadius.circular(10),
              borderSide: BorderSide(
                color: custom_green,
                width: 2.0,
              ),
            ),
          ),
        ),
      ),
    );
  }
}
```

Output :-



-