Experiment No:05

Aim: To apply navigation, routing and gestures in Flutter App

Theory:

Navigation and Routing in Flutter

Navigation and routing are some of the core concepts of all mobile applications, which allows the user to move between different pages. We know that every mobile application contains several screens for displaying different types of information. For example, an app can have a screen that contains various products. When the user taps on that product, immediately it will display detailed information about that product.

Navigation Routing:

1. Using Navigator Widget:

- Flutter provides the Navigator widget for managing routes and navigating between different screens in an app.
- You can push a new route onto the navigator's stack to navigate to a new screen and pop a route from the stack to go back to the previous screen.

2. Named Routes:

- Named routes are routes identified by a unique string identifier.
- Define named routes in the app's main MaterialApp widget.

3. Passing Data:

- You can pass data to a new screen when navigating.

Gesture Handling

Gesture handling refers to the process of detecting and responding to user interactions, or gestures, on a touch-based interface, such as a smartphone or tablet. In the context of software development, especially in mobile app development, gesture handling involves recognizing various touch events, such as taps, swipes, pinches, and rotations, and translating them into meaningful actions within the application.

1. Gesture Detector:
- Use the GestureDetector widget to detect various gestures like tap, double tap, long press, etc.
- Wrap the widget with the GestureDetector and specify the gesture callbacks.

2. Draggable and DragTarget:
- Use Draggable to make a widget draggable and DragTarget to specify a target area for dropping.

Code:

```
import 'dart:ui';

import 'package:flutter/material.dart';
import 'package:flutter/src/widgets/framework.dart';
import 'package:flutter/src/widgets/placeholder.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';
import 'dart:async';

import 'package:rive/rive.dart';

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

  @override
  State<CollabPage> createState() => _CollabPageState();
}

class _CollabPageState extends State<CollabPage> {
  //Defining an boolean to specify if the user is willing to edit the text or not
  bool f = false;
  var txt = TextEditingController();
  @override
  void initState() {
    super.initState();
    //Calling to update my text in the textformfield every 1 second
```

```
  //Only if the below conditions are satisfied
  Timer.periodic(
    const Duration(milliseconds: 500),
    (timer) async {
      //Condition - If the user has tapped below button so that he can edit the text
      if (f) {
        var res = await http.get(
          Uri.parse("https://collab-me-backend.vercel.app/latest/1"),
        );
        var data1 = await json.decode(res.body);
        var sample = txt.text;
        if ((data1["latest msg"]["text"].length != 0) &&
            (txt.text != data1["latest msg"]["text"])) {
          setState(
            () {
              txt.text = data1["latest msg"]["text"];
            },
          );
        }
        print("Text Updated");
        print(data1["latest msg"]["text"]);
      }
    },
  );
}

Widget build(BuildContext context) {
  return Scaffold(
    // appBar: AppBar(
    //   title: const Text("Collab Me"),
    //   elevation: 0,
    //   flexibleSpace: Container(
    //     decoration: const BoxDecoration(
    //       gradient: LinearGradient(
    //         colors: [Colors.purple, Colors.blue],
    //       ),
    //     ),
    //   ),
```

```
    // ),
    body: Stack(
      children: [
        const RiveAnimation.asset("assets/RiveAssets/shapes.riv"),
        Positioned.fill(
          child: BackdropFilter(
            filter: ImageFilter.blur(
              sigmaX: 20,
              sigmaY: 10,
            ),
            child: SizedBox(),
          ),
        ),
        SafeArea(
          child: Column(
            children: [
              Row(
                children: [
                  const SizedBox(
                    height: 150,
                  ),
                  const Expanded(
                    child: Padding(
                      padding: EdgeInsets.only(
                          left: 15, bottom: 20, right: 20, top: 10),
                      child: Text(
                        "Do you want to turn on collaborative mode and update your text
with others",
                        style: TextStyle(color: Colors.black, fontSize: 25),
                      ),
                    ),
                  ),
                  // ElevatedButton(
                  //   onPressed: () async {
                  //     var res = await http.get(Uri.parse(
                  //         "https://collab-me-backend.vercel.app/latest/1"));
                  //     var data = await json.decode(res.body);
                  //     print(data["latest msg"]["text"]);
```

```
    //    setState(() {
    //      f = !f;
    //    });
    //  },
    //  child: Text(f.toString()),
    // ),
    Switch(
      value: f,
      focusColor: Colors.red,
      onChanged: (bool value) async {
        var res = await http.get(Uri.parse(
            "https://collab-me-backend.vercel.app/latest/1"));
        var data = await json.decode(res.body);
        print(data["latest msg"]["text"]);
        setState(() {
          f = value;
        });
      },
    )
  ],
),
const SizedBox(
  height: 40,
),

TextField(
  controller: txt,
  maxLines: null,
  style: TextStyle(fontSize: 25),
  minLines: 10,
  decoration: const InputDecoration(
      border: InputBorder.none,
      fillColor: Colors.white70,
      filled: true),
  onChanged: (value) async {
    if (txt.text.length != 0) {
      var data = {"text": txt.text};
      final response = await http.post(
```

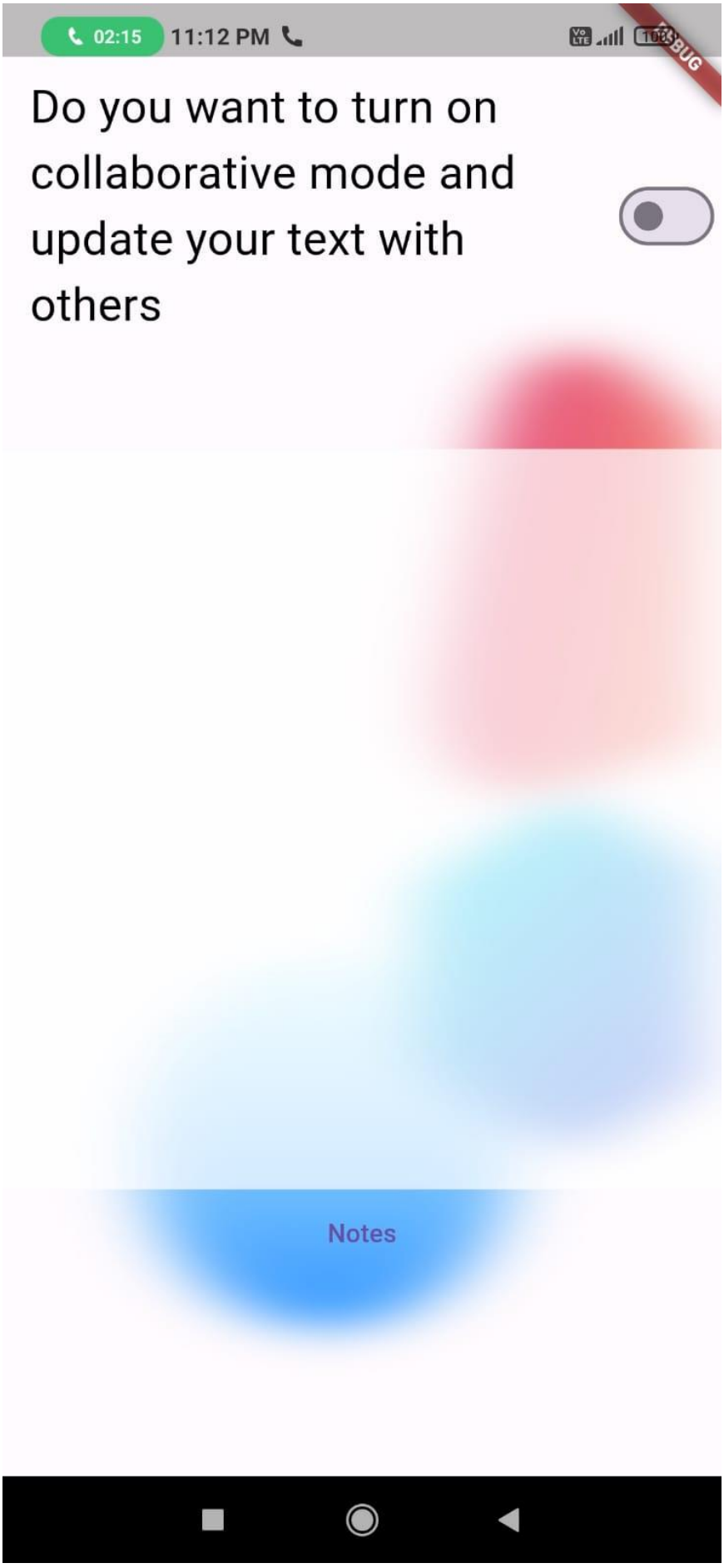
```
                Uri.parse(
                  "https://collab-me-backend.vercel.app/update",
                ),
                body: json.encode(data),
                headers: {"Content-Type": "application/json"},
              );
              print(response.body);
              print("data uploaded");
            }
            //Used to bring my editing cursor on the end of the text
            //Used to fix an annoying bug :)
            txt.selection =
                TextSelection.collapsed(offset: txt.text.length);
          },
        ),
        // Text(txt.text),
      ],
    ),
  ),
 ],
),
);
}
}
```

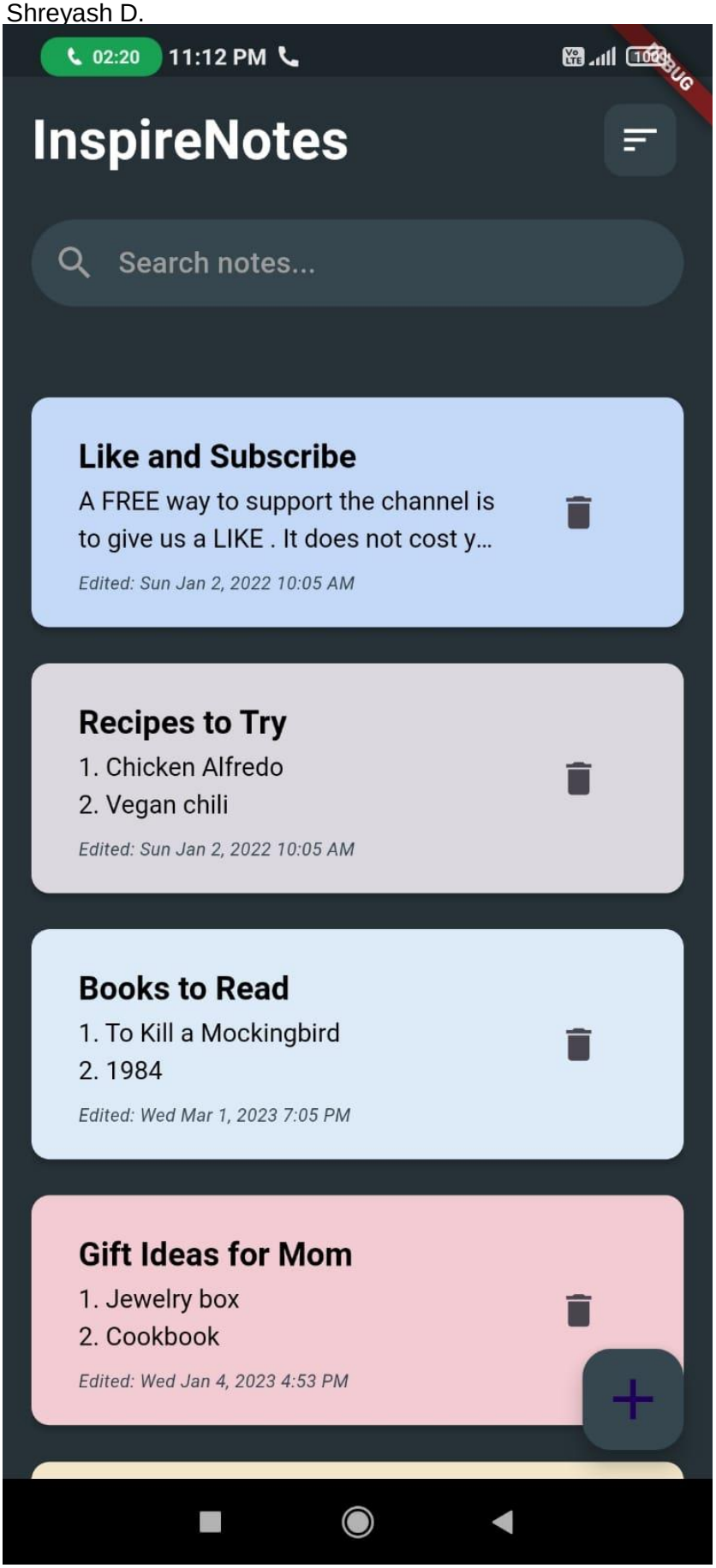
Edit.dart

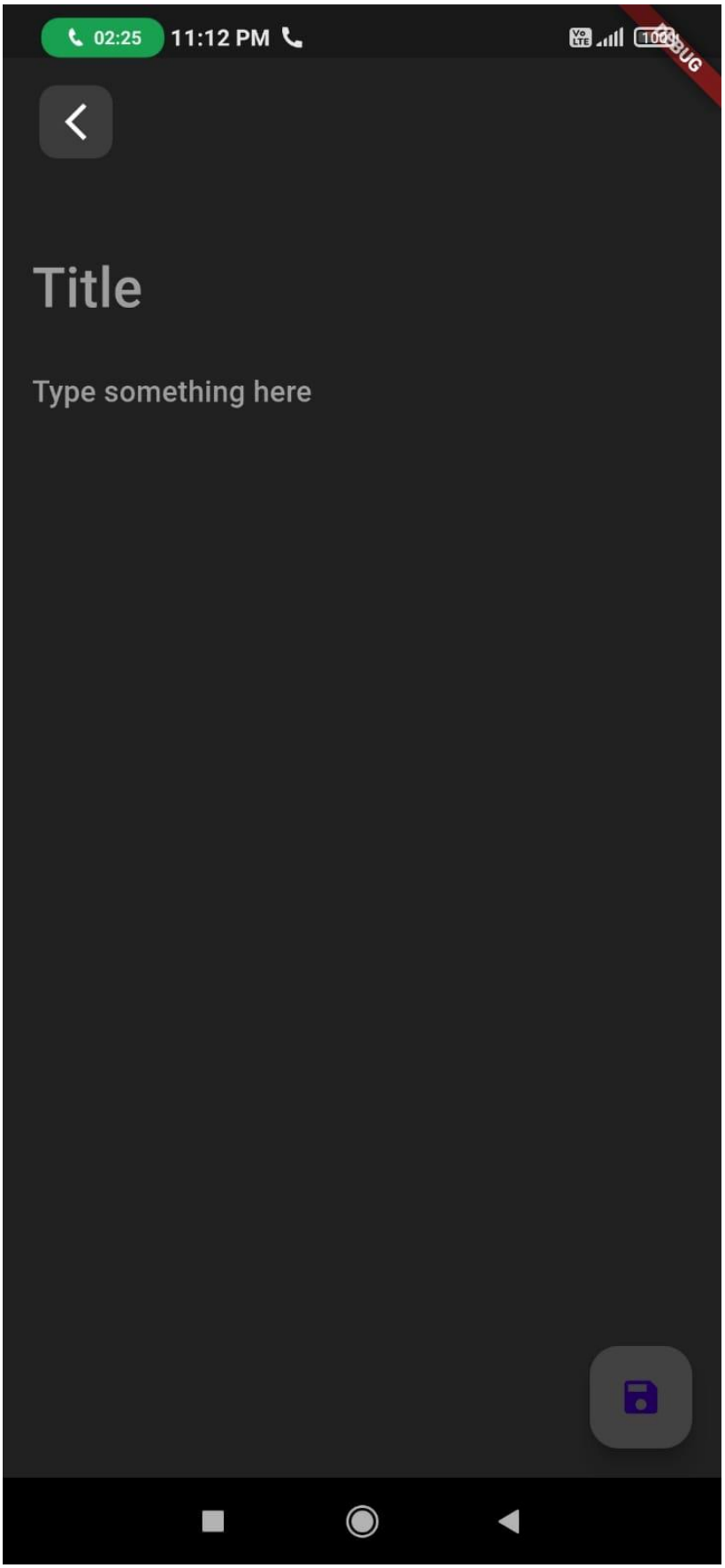
```
import 'package:flutter/material.dart';

import '../models/note.dart';

class EditScreen extends StatefulWidget {
 final Note? note;
 const EditScreen({super.key, this.note});

 @override
 State<EditScreen> createState() => _EditScreenState();
}

class _EditScreenState extends State<EditScreen> {
 TextEditingController _titleController = TextEditingController();
 TextEditingController _contentController = TextEditingController();
```

```
@override
void initState() {
  // TODO: implement initState
  if (widget.note != null) {
    _titleController = TextEditingController(text: widget.note!.title);
    _contentController = TextEditingController(text: widget.note!.content);
  }

  super.initState();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.grey.shade900,
    body: Padding(
      padding: const EdgeInsets.fromLTRB(16, 40, 16, 0),
      child: Column(children: [
        Row(
          mainAxisAlignment: MainAxisAlignment.spaceBetween,
          children: [
            IconButton(
                onPressed: () {
                  Navigator.pop(context);
                },
                padding: const EdgeInsets.all(0),
                icon: Container(
                  width: 40,
                  height: 40,
                  decoration: BoxDecoration(
                      color: Colors.grey.shade800.withOpacity(.8),
                      borderRadius: BorderRadius.circular(10)),
                  child: const Icon(
                    Icons.arrow_back_ios_new,
                    color: Colors.white,
                  ),
                ))
          ],
        ),
        Expanded(
            child: ListView(
          children: [
            TextField(
              controller: _titleController,
              style: const TextStyle(color: Colors.white, fontSize: 30),
              decoration: const InputDecoration(
                  border: InputBorder.none,
                  hintText: 'Title',
```

```
                hintStyle: TextStyle(color: Colors.grey, fontSize: 30)),
            ),
            TextField(
              controller: _contentController,
              style: const TextStyle(
                color: Colors.white,
              ),
              maxLines: null,
              decoration: const InputDecoration(
                  border: InputBorder.none,
                  hintText: 'Type something here',
                  hintStyle: TextStyle(
                    color: Colors.grey,
                  )),
            ),
          ],
        ))
      ]),
    ),
    floatingActionButton: FloatingActionButton(
      onPressed: () {
        Navigator.pop(
            context, [_titleController.text, _contentController.text]);
      },
      elevation: 10,
      backgroundColor: Colors.grey.shade800,
      child: const Icon(Icons.save),
    ),
  );
 }
}
```


02:15
11:12 PM
DEBUG
Do you want to turn on collaborative mode and update your text with others
Notes


02:20
11:12 PM
DEBUG
InspireNotes
Search notes...
Like and Subscribe
A FREE way to support the channel is to give us a LIKE . It does not cost y...
Edited: Sun Jan 2, 2022 10:05 AM
Recipes to Try
1. Chicken Alfredo
2. Vegan chili
Edited: Sun Jan 2, 2022 10:05 AM
Books to Read
1. To Kill a Mockingbird
2. 1984
Edited: Wed Mar 1, 2023 7:05 PM
Gift Ideas for Mom
1. Jewelry box
2. Cookbook
Edited: Wed Jan 4, 2023 4:53 PM


02:25
11:12 PM
DEBUG
Title
Type something here

Conclusion:

We understood the concepts of navigation , routing and gestures in Flutter. We implemented navigation and routing for the above shown pages. We implemented gestures in a basic Flutter application.