Name: Mitesh A. Dalvi Roll no.: 11
Class: D15B

MAD Experiment 5

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

Theory :

Flutter Navigation and Routing :
Navigation and routing are some of the core concepts of all mobile application, 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.
In Flutter, the screens and pages are known as routes, and these routes are just a widget. In Android, a route is similar to an Activity, whereas, in iOS, it is equivalent to a ViewController. In any mobile app, navigating to different pages defines the workflow of the application, and the way to handle the navigation is known as routing. Flutter provides a basic routing class MaterialPageRoute and two methods Navigator.push() and Navigator.pop() that shows how to navigate between two routes. The following steps are required to start navigation in your application.
Step 1: First, you need to create two routes.
Step 2: Then, navigate to one route from another route by using the Navigator.push() method.
Step 3: Finally, navigate to the first route by using the Navigator.pop() method.

Gestures :
Gestures are used to interact with an application. It is generally used in touch-based devices to physically interact with the application. It can be as simple as a single tap on the screen to a more complex physical interaction like swiping in a specific direction to scrolling down an application. It is heavily used in gaming and more or less every application requires it to function as devices turn more touch-based than ever. In this article, we will discuss them in detail.
Some widely used gestures are mentioned here :

- **Tap:** Touching the surface of the device with the fingertip for a small duration of time period and finally releasing the fingertip.
- **Double Tap:** Tapping twice in a short time.
- **Drag:** Touching the surface of the device with the fingertip and then moving the fingertip in a steadily and finally releasing the fingertip.
- **Flick:** Similar to dragging, but doing it in a speedier way.
- **Pinch:** Pinching the surface of the device using two fingers.
- **Zoom:** Opposite of pinching.
- **Panning:** Touching the device surface with the fingertip and moving it in the desired direction without releasing the fingertip.

The GestureDetector widget in flutter is used to detect physical interaction with the application on the UI. If a widget is supposed to experience a gesture, it is kept inside the GestureDetector widget. The same widget catches the gesture and returns the appropriate action or response.

Below is the list of gestures and their corresponding events :

Tap

- onTapDown
- onTapUp
- onTap
- onTapCancel

Double tap

- onDoubleTap

Long press

- onLongPress

Vertical drag

- onVerticalDragStart
- onVerticalDragUpdate
- onVerticalDragEnd

Horizontal drag

- onHorizontalDragStart
- onHorizontalDragUpdate
- onHorizontalDragEnd

Pan

- onPanStart
- onPanUpdate
- onPanEnd

Execution :

home.dart file:

```
import 'dart:convert';
import 'dart:developer';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:recipe_app/RecipeView.dart';
import 'package:recipe_app/feedback.dart';
import 'package:recipe_app/mealplanner.dart';
import 'package:recipe_app/model.dart';
import 'package:recipe_app/search.dart';
import 'package:http/http.dart';

class Home extends StatefulWidget {
  @override
  _HomeState createState() => _HomeState();
}

class _HomeState extends State<Home> {
  final user = FirebaseAuth.instance.currentUser;

  signout() async {
    await FirebaseAuth.instance.signOut();
  }

  bool isLoading = true;
  List<RecipeModel> recipeList = <RecipeModel>[];
  TextEditingController searchController = new TextEditingController();
  List reciptCatList = [
    {
```

```
    "imgUrl": "https://images.unsplash.com/photo-1593560704563-f176a2eb61db",
    "heading": "Chilli Food"
   },
   {
    "imgUrl": "https://images.unsplash.com/photo-1593560704563-f176a2eb61db",
    "heading": "Chilli Food"
   },
   {
    "imgUrl": "https://images.unsplash.com/photo-1593560704563-f176a2eb61db",
    "heading": "Chilli Food"
   },
   {
    "imgUrl": "https://images.unsplash.com/photo-1593560704563-f176a2eb61db",
    "heading": "Chilli Food"
   }
  ];

  getRecipes(String query) async {
   String url =

"https://api.edamam.com/search?q=$query&app_id=2159af20&app_key=58924e51bb9295724d0a13dc230be5d
4";
   Response response = await get(Uri.parse(url));
   Map data = jsonDecode(response.body);

   setState(() {
    data["hits"].forEach((element) {
     RecipeModel recipeModel = new RecipeModel();
     recipeModel = RecipeModel.fromMap(element["recipe"]);
     recipeList.add(recipeModel);
     setState(() {
      isLoading = false;
     });
     log(recipeList.toString());
    });
   });

   recipeList.forEach((Recipe) {
    print(Recipe.applabel);
    print(Recipe.appcalories);
   });
  }

  @override
  void initState() {
   super.initState();
   getRecipes("Ladoo");
  }

  @override
  Widget build(BuildContext context) {
   return Scaffold(
    appBar: AppBar(
     title: Text('Home'),
    ),
    drawer: Drawer(
  child: ListView(
   padding: EdgeInsets.zero,
   children: <Widget>[
    DrawerHeader(
```

```
        decoration: BoxDecoration(
          color: Colors.blue,
        ),
        child: Text(
          'Menu',
          style: TextStyle(
            color: Colors.white,
            fontSize: 24,
          ),
        ),
      ),
      ListTile(
        title: Text('Meal Planner'),
        onTap: () {
          Navigator.push(
            context,
            MaterialPageRoute(builder: (context) => MealPlanner()),
          );
        },
      ),
      ListTile(
        title: Text('Feedback Form'),
        onTap: () {
          Navigator.push(
            context,
            MaterialPageRoute(builder: (context) => FeedbackForm()),
          );
        },
      ),
    ],
  ),
),

      body: Stack(
        children: [
          Container(
            width: MediaQuery.of(context).size.width,
            height: MediaQuery.of(context).size.height,
            decoration: BoxDecoration(
              gradient: LinearGradient(
                  colors: [Color(0xff213A50), Color(0xff071938)]),
            ),
          ),
          SingleChildScrollView(
            child: Column(
              children: [
                SafeArea(
                  child: Container(
                    padding: EdgeInsets.symmetric(horizontal: 8),
                    margin: EdgeInsets.symmetric(horizontal: 24, vertical: 20),
                    decoration: BoxDecoration(
                        color: Colors.white,
                        borderRadius: BorderRadius.circular(24)),
                    child: Row(
                      children: [
                        GestureDetector(
                          onTap: () {
                            if ((searchController.text).replaceAll(" ", "") ==
                                "") {
                              print("Blank search");
```

```
          } else {
            Navigator.push(context, MaterialPageRoute(
                builder: (context) => Search(searchController.text)));
          }
        },
        child: Container(
          child: Icon(
            Icons.search,
            color: Colors.blueAccent,
          ),
          margin: EdgeInsets.fromLTRB(3, 0, 7, 0),
        ),
      ),
      Expanded(
        child: TextField(
          controller: searchController,
          decoration: InputDecoration(
              border: InputBorder.none,
              hintText: "Let's Cook Something!"),
        ),
      )
    ],
  ),
 ),
),
Container(
 padding: EdgeInsets.symmetric(horizontal: 20),
 child: Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: [
   Text(
    "WHAT DO YOU WANT TO COOK TODAY?",
    style: TextStyle(fontSize: 33, color: Colors.white),
   ),
   SizedBox(
    height: 10,
   ),
   Text(
    "Let's Cook Something New!",
    style: TextStyle(fontSize: 20, color: Colors.white),
   )
  ],
 ),
),
Container(
  child: isLoading ? CircularProgressIndicator() : ListView.builder(
    physics: NeverScrollableScrollPhysics(),
    shrinkWrap: true,
    itemCount: recipeList.length,
    itemBuilder: (context, index) {
     return InkWell(
      onTap: () {
       Navigator.push(context, MaterialPageRoute(
           builder: (context) => RecipeView(recipeList[index].appurl)));
      },
      child: Card(
       margin: EdgeInsets.all(20),
       shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(10),
       ),
```

```
            elevation: 0.0,
            child: Stack(
              children: [
                ClipRRect(
                    borderRadius: BorderRadius.circular(10.0),
                    child: Image.network(
                      recipeList[index].appimgUrl,
                      fit: BoxFit.cover,
                      width: double.infinity,
                      height: 200,
                    )),
                Positioned(
                    left: 0,
                    right: 0,
                    bottom: 0,
                    child: Container(
                        padding: EdgeInsets.symmetric(
                            vertical: 5, horizontal: 10),
                        decoration: BoxDecoration(
                            color: Colors.black26),
                        child: Text(
                          recipeList[index].applabel,
                          style: TextStyle(
                              color: Colors.white,
                              fontSize: 20),
                        ))),
                Positioned(
                  right: 0,
                  height: 40,
                  width: 80,
                  child: Container(
                      decoration: BoxDecoration(
                          color: Colors.white,
                          borderRadius: BorderRadius.only(
                              topRight: Radius.circular(10),
                              bottomLeft: Radius.circular(10)
                          )
                      ),
                      child: Center(
                        child: Row(
                          mainAxisAlignment: MainAxisAlignment.center,
                          children: [
                            Icon(Icons.local_fire_department, size: 15,),
                            Text(recipeList[index].appcalories.toString().substring(0, 6)),
                          ],
                        ),
                      )),
                )
              ],
            ),
          ),
        );
      })),

Container(
  height: 100,
  child: ListView.builder( itemCount: reciptCatList.length, shrinkWrap: true,
    scrollDirection: Axis.horizontal,
    itemBuilder: (context, index){
```

```
                return Container(
                    child: InkWell(
                      onTap: () {
                        Navigator.push(context, MaterialPageRoute(
                            builder: (context) => Search(reciptCatList[index]["heading"])));
                      },
                      child: Card(
                        margin: EdgeInsets.all(20),
                        shape: RoundedRectangleBorder(
                          borderRadius: BorderRadius.circular(18),
                        ),
                        elevation: 0.0,
                        child:Stack(
                          children: [
                            ClipRRect(
                              borderRadius: BorderRadius.circular(18.0),
                              child: Image.network(reciptCatList[index]["imgUrl"], fit: BoxFit.cover,
                                width: 200,
                                height: 250,)
                            ),
                            Positioned(
                              left: 0,
                              right: 0,
                              bottom: 0,
                              top: 0,
                              child: Container(
                                padding: EdgeInsets.symmetric(
                                    vertical: 5, horizontal: 10),
                                decoration: BoxDecoration(
                                    color: Colors.black26),
                                child: Column(
                                  mainAxisAlignment: MainAxisAlignment.center,
                                  children: [
                                    Text(
                                      reciptCatList[index]["heading"],
                                      style: TextStyle(
                                          color: Colors.white,
                                          fontSize: 28),
                                    ),
                                  ],
                                ))),
                          ],
                        )
                      ),
                    )
                );
                }),
            )
          ],
        ),
      ),
    ],
  ),
  floatingActionButton: FloatingActionButton(
    onPressed: (() => signout()),
    child: Icon(Icons.login_rounded),
  ),
 );
 }
}
```

mealplanner.dart file:

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

void main() {
  runApp(MaterialApp(
    home: MealPlanner(),
  ));
}

class MealPlanner extends StatefulWidget {
  @override
  _MealPlannerState createState() => _MealPlannerState();
}

class _MealPlannerState extends State<MealPlanner> {
  Map<String, String> _meals = {};


  void _addMeal(String day, String meal) {
    setState(() {
      _meals[day] = meal;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Meal Planner'),
      ),
      body: ListView.builder(
        itemCount: 7,
        itemBuilder: (context, index) {
          String day = _getDayOfWeek(index + 1);
          return ListTile(
            title: Text(day),
            subtitle: _meals.containsKey(day) ? Text(_meals[day]!) : null,
            trailing: IconButton(
              icon: Icon(Icons.add),
              onPressed: () {
                showDialog(
                  context: context,
                  builder: (context) => _buildAddMealDialog(day),
                );
              },
            ),
          );
        },
      ),
    );
  }

  Widget _buildAddMealDialog(String day) {
    TextEditingController _textController = TextEditingController();
    return AlertDialog(
      title: Text('Add Meal for $day'),
      content: TextField(
        controller: _textController,
        decoration: InputDecoration(hintText: 'Enter meal name'),
```

```
      ),
      actions: [
        TextButton(
          onPressed: () {
            _addMeal(day, _textController.text);
            Navigator.pop(context);
          },
          child: Text('Add'),
        ),
        TextButton(
          onPressed: () {
            Navigator.pop(context);
          },
          child: Text('Cancel'),
        ),
      ],
    );
  }

  String _getDayOfWeek(int day) {
    switch (day) {
      case 1:
        return 'Monday';
      case 2:
        return 'Tuesday';
      case 3:
        return 'Wednesday';
      case 4:
        return 'Thursday';
      case 5:
        return 'Friday';
      case 6:
        return 'Saturday';
      case 7:
        return 'Sunday';
      default:
        return '';
    }
  }
}
```

feedback.dart file:

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

class FeedbackForm extends StatefulWidget {
  @override
  _FeedbackFormState createState() => _FeedbackFormState();
}

class _FeedbackFormState extends State<FeedbackForm> {
  final _formKey = GlobalKey<FormState>();

  String _name = '';
  String _email = '';
  String _feedbackType = 'General';
  String _comments = '';

  @override
  Widget build(BuildContext context) {
```

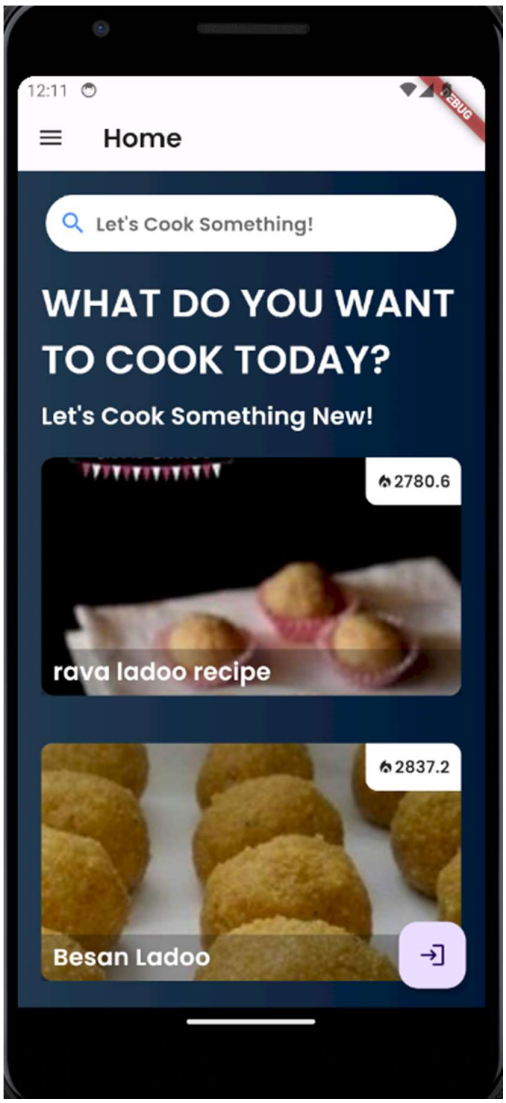
```
return Scaffold(
  appBar: AppBar(
    title: Text('Feedback Form'),
  ),
  body: Padding(
    padding: const EdgeInsets.all(16.0),
    child: Form(
      key: _formKey,
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: <Widget>[
          TextFormField(
            decoration: InputDecoration(labelText: 'Name'),
            validator: (value) {
              if (value == null || value.isEmpty) {
                return 'Please enter your name';
              }
              return null;
            },
            onSaved: (value) {
              _name = value!;
            },
          ),
          TextFormField(
            decoration: InputDecoration(labelText: 'Email'),
            validator: (value) {
              if (value == null || value.isEmpty) {
                return 'Please enter your email address';
              }
              return null;
            },
            onSaved: (value) {
              _email = value!;
            },
          ),
          DropdownButtonFormField(
            value: _feedbackType,
            decoration: InputDecoration(labelText: 'Feedback Type'),
            items: <String>['General', 'Bug', 'Feature Request']
                .map<DropdownMenuItem<String>>((String value) {
              return DropdownMenuItem<String>(
                value: value,
                child: Text(value),
              );
            }).toList(),
            onChanged: (String? value) {
              setState(() {
                _feedbackType = value!;
              });
            },
          ),
          TextFormField(
            decoration: InputDecoration(labelText: 'Comments'),
            maxLines: 3,
            validator: (value) {
              if (value == null || value.isEmpty) {
                return 'Please enter your comments';
              }
              return null;
            },
```

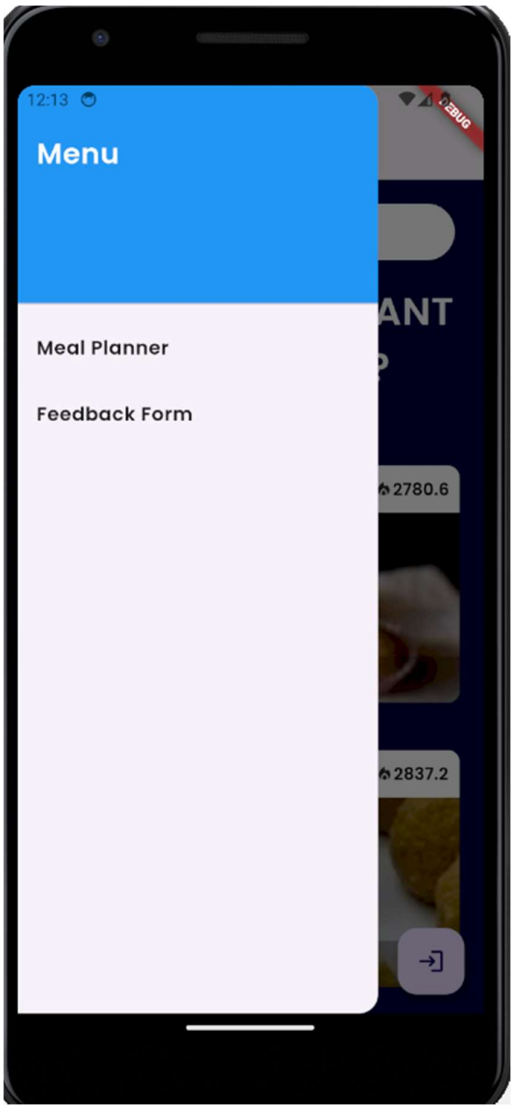
```
            onSaved: (value) {
              _comments = value!;
            },
          ),
          SizedBox(height: 16.0),
          ElevatedButton(
            onPressed: () {
              if (_formKey.currentState!.validate()) {
                _formKey.currentState!.save();
                // Send feedback to your server or process it as needed
                // For now, just print the feedback data
                print('Name: $_name');
                print('Email: $_email');
                print('Feedback Type: $_feedbackType');
                print('Comments: $_comments');
              }
            },
            child: Text('Submit'),
          ),
        ],
      ),
    ),
  ),
 );
 }
}

void main() {
 runApp(MaterialApp(
   title: 'Feedback Form Example',
   home: FeedbackForm(),
 ));
}
```

Output :

We have routed and navigated to meal planner page and feedback form page using Navigate.push() function.

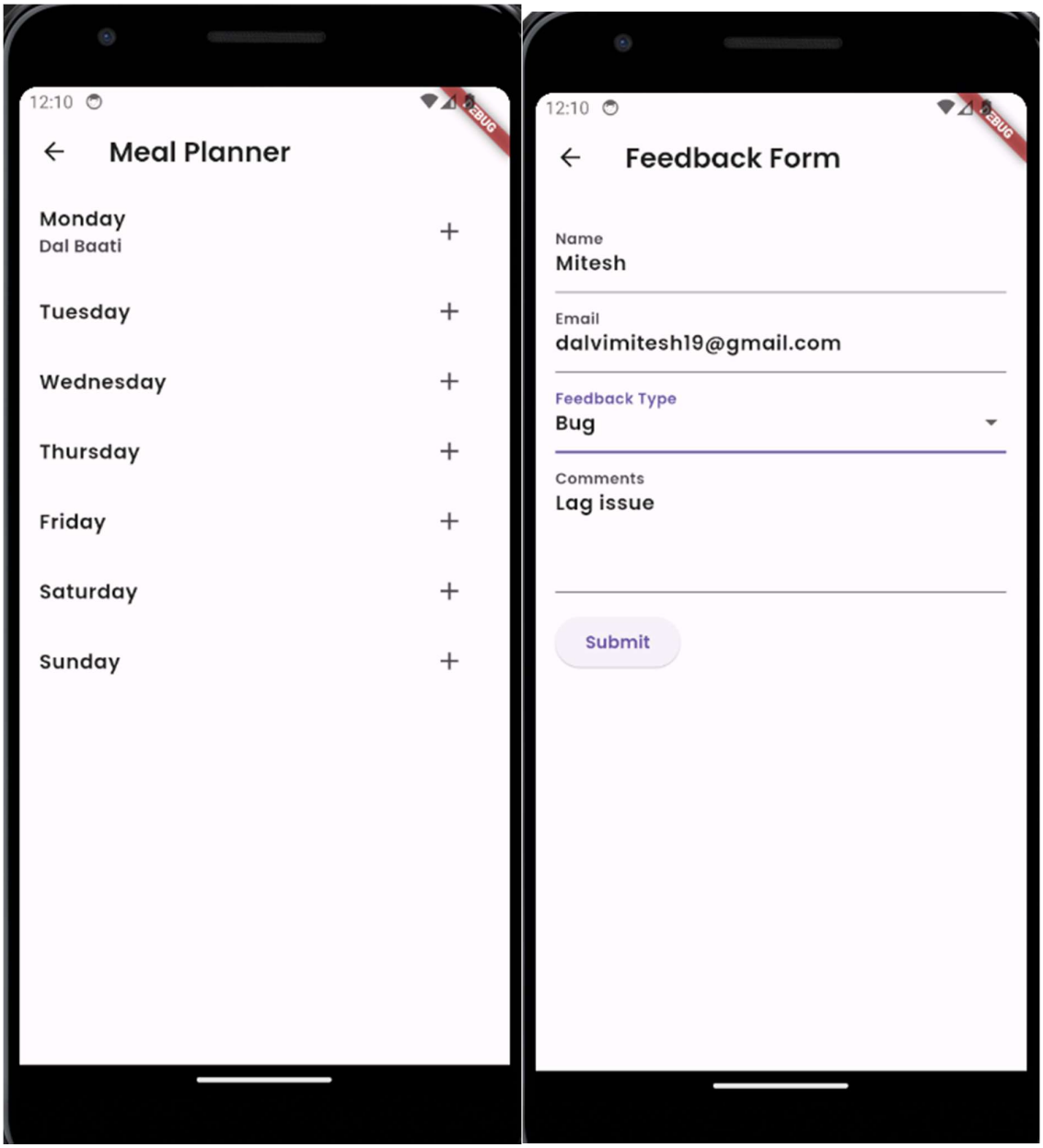




This is the home page of our recipe app. Here the search bar at the top is the usage of GestureDetector method.

There is side drawer menu for the two routes in which navigation takes place:

1. Meal planner
2. Feedback Form



Conclusion : We have understood the concept of gestures, their use and implemented it in our flutter app as a search bar. Also, we created two pages and route it in our app and navigate using Navigate.push method. The flow is smooth for our app.