EXPERIMENT NO.5

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

Theory:

Navigation:

- Navigation refers to the process of moving between different screens or "routes" within the app.
- Flutter provides the Navigator class, which manages a stack of routes and facilitates navigation between them.
- You can push new routes onto the stack using Navigator.push() and remove routes using Navigator.pop().
- Named routes can be pre-defined in the app's route table, making it easier to navigate to specific screens by providing their names.
- Nested navigation allows for hierarchical navigation structures, such as tab-based navigation or modal dialogs.

Routing:

- Routing involves defining and managing the routes or paths that users can take through the app.
- Routes are logical representations of screens or pages within the app and are associated with unique identifiers (route names or route keys).
- Route management includes defining routes, specifying transitions between routes, passing data between routes, and handling route navigation events.

Gestures:

- Gestures enable users to interact with the app's UI elements through touch-based interactions.
- Flutter provides a wide range of gesture recognizers, such as GestureDetector, InkWell, Draggable, LongPressGestureDetector, etc., to detect and respond to user gestures.
- Gesture recognizers can detect taps, swipes, drags, pinches, and other touch-based actions, allowing for rich and intuitive user interactions.
- You can customize gesture behaviors, such as sensitivity, velocity, and directionality, to meet specific app requirements.

By implementing navigation, routing, and gestures effectively in your Flutter app, you can create a smooth and engaging user experience, enabling users to navigate between screens, interact with UI elements, and perform actions with ease

Code:

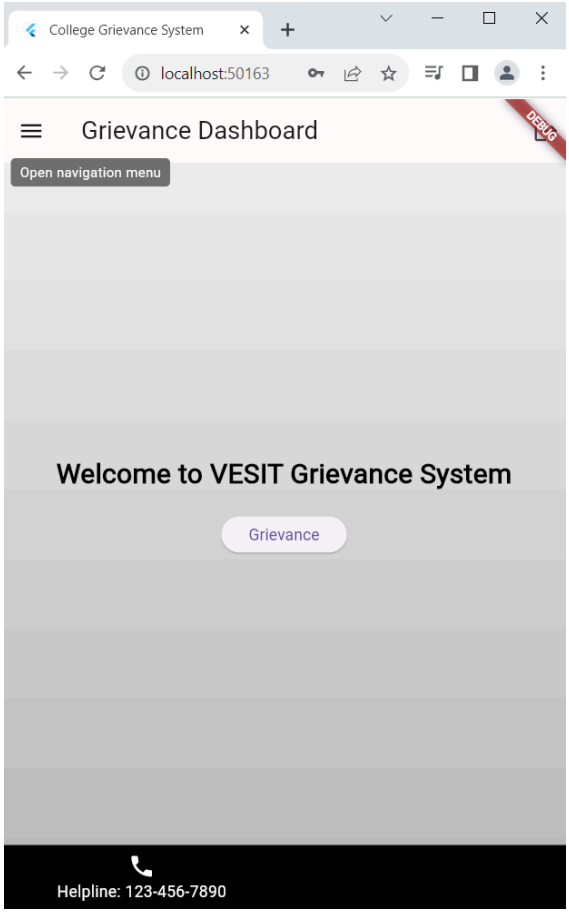
```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:loginagain/firebase_options.dart';
import 'package:loginagain/pages/about.dart';
import 'package:loginagain/pages/contact_info_page.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:loginagain/pages/profilepage.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(GrievanceDashboard());
}

class GrievanceDashboard extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'College Grievance System',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: DashboardScreen(),
    );
  }
}

class DashboardScreen extends StatelessWidget {
  void signUserOut() {
    FirebaseAuth.instance.signOut();
  }
```

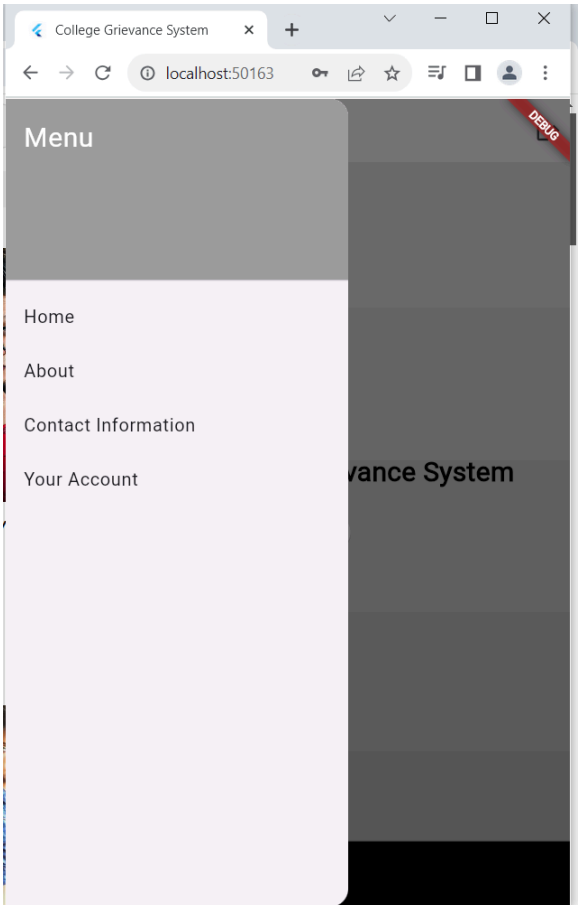
```
  void _showAboutDialog(BuildContext context) {
    showDialog(
      context: context,
      builder: (BuildContext context) {
        return AlertDialog(
          title: Text("About College"),
          content: Text(
              "This is where you can display information about the
college."),
          actions: <Widget>[
            TextButton(
              onPressed: () {
                Navigator.of(context).pop();
              },
              child: Text("Close"),
            ),
          ],
        );
      },
    );
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        actions: [
          IconButton(onPressed: signUserOut, icon: Icon(Icons.logout)),
        ],
        title: Text('Grievance Dashboard'),
      ),
      drawer: Drawer(
        child: ListView(
          padding: EdgeInsets.zero,
          children: <Widget>[
            DrawerHeader(
              decoration: BoxDecoration(
                color: Colors.grey,
              ),
```

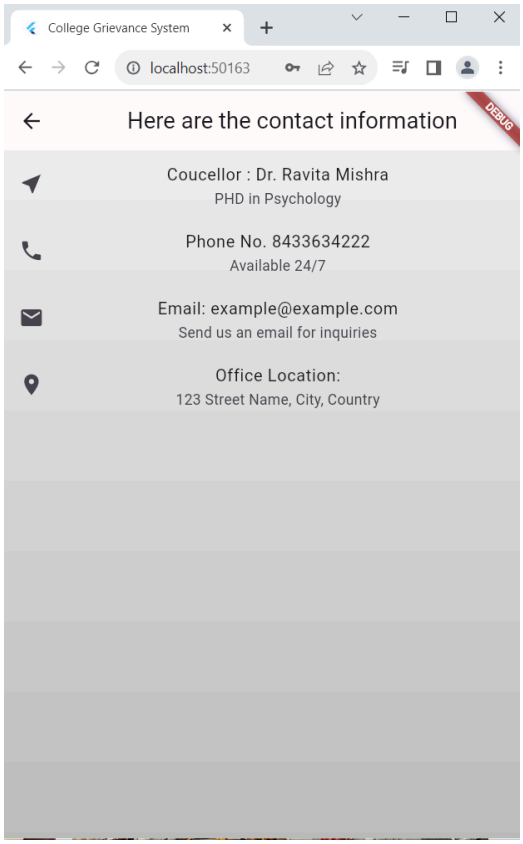
```
              child: Text(
                'Menu',
                style: TextStyle(
                  color: Colors.white,
                  fontSize: 24,
                ),
              ),
            ),
            ListTile(
              title: Text('Home'),
              onTap: () {
                Navigator.pop(context); // Close the drawer
                // Navigate back to the dashboard
                Navigator.pushReplacement(
                  context,
                  MaterialPageRoute(builder: (context) =>
GrievanceDashboard()),
                );
              },
            ),
            ListTile(
              title: Text('About'),
              onTap: () {
                // Navigate to AboutPage screen
                Navigator.push(
                  context,
                  MaterialPageRoute(builder: (context) => AboutPage()),
                );
              },
            ),
  bottomNavigationBar: BottomNavigationBar(
        backgroundColor: Colors.black,
        selectedItemColor: Colors.white,
        items: [
          BottomNavigationBarItem(
            icon: Icon(Icons.phone),
            label: 'Helpline: 123-456-7890',
          ),
          BottomNavigationBarItem(
            icon: Icon(Icons.email),
```

```
            label: 'Email: example@example.com',
          ),
        ],
      ),
    );
  }
}
```

Output :







Conclusion: Thus we have implemented navigation and routing in our app.