Experiment - 5

Name: Preeti Khamkar **Division:** D15A **Roll no. :** 28

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

Theory:

Navigation and routing are fundamental concepts in Flutter for managing the flow of screens or "routes" within an application. They allow users to move between different parts of the app, such as navigating from a home screen to a settings screen, or from a product list to a product detail page.

1. Navigation: Navigation refers to the action of moving between different screens or routes in an application. This can be triggered by user interactions like tapping on a button, or programmatically in response to certain events.
2. Routes: In Flutter, a route represents a distinct screen or page within the application. Each route has its own widget tree and can be pushed onto a navigation stack or popped off it.
3. Navigator: The Navigator widget manages a stack of routes. It allows you to push new routes onto the stack, pop existing routes off the stack, or replace routes with new ones. The Navigator typically resides at the root of the widget tree and facilitates navigation within the app.
4. Routing: Routing refers to the process of defining and managing the mapping between route names (or route objects) and their corresponding widgets in the application. Flutter provides mechanisms for both named routing and dynamic routing.

Syntax:

```
 Navigator.push(
             context,
             MaterialPageRoute(builder: (context) => const SecondRoute()),
           );
```

Widget and Properties:

1. Drawer: The Drawer widget provides a panel that slides in from the left side of the screen and contains navigation options and user information. It includes user

details like profile image, username, phone number, and options like creating a new group, accessing contacts, settings, etc.

2. StreamBuilder: It's used to listen to changes in Firestore collections and update the UI accordingly. In this code, it's used to fetch user data and display a list of users.
3. ListView: Displays a list of users fetched from Firestore.
4. Navigation: Navigation between screens is handled using Navigator.push() and MaterialPageRoute.

Code and Output:

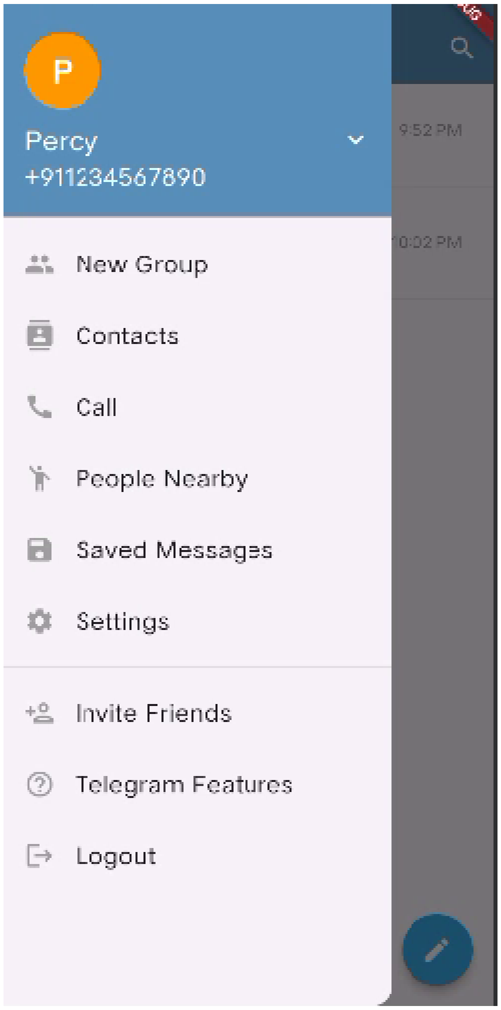
```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:telegram/screen/chatPage.dart';
import '../contacts.dart';
import '../search.dart';
import 'drawer/settings.dart';
import 'package:intl/intl.dart';

class HomePage extends StatefulWidget {
 const HomePage({Key? key});

 @override
 State<HomePage> createState() => _HomePageState();

 // Make the method public
 static String constructChatRoomId(String phoneNumber1, String phoneNumber2) {
   List<String> phoneNumbers = [phoneNumber1, phoneNumber2];
   phoneNumbers.sort();
   return phoneNumbers.join("");
 }
}

class _HomePageState extends State<HomePage> {
 final FirebaseAuth _auth = FirebaseAuth.instance;

 void signOut() async {
   await _auth.signOut();
   // Navigate to login screen
   Navigator.pushReplacementNamed(context, '/login');
 }
```

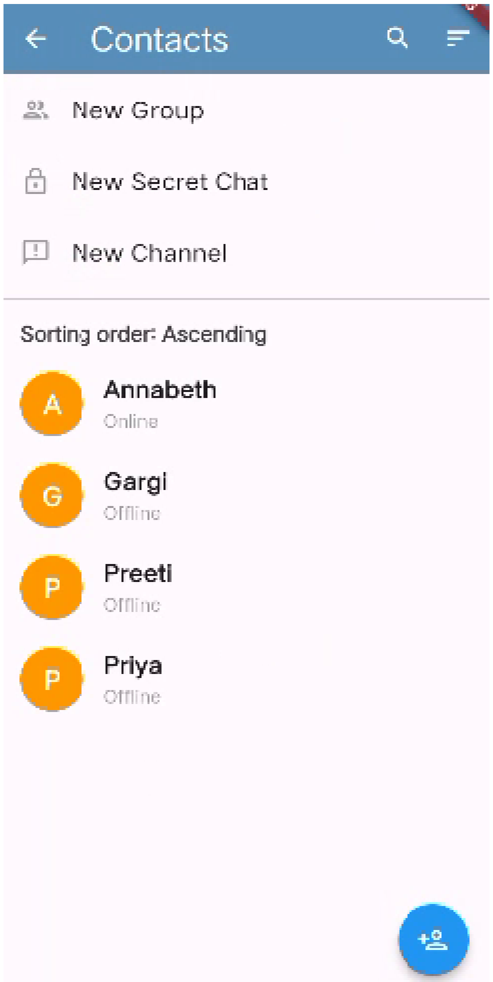
```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Telegram', style: TextStyle(color: Colors.white)),
      backgroundColor: Color(0xFF588FBA),
      actions: [
        IconButton(
          icon: Icon(Icons.search),
          onPressed: () {
            // Navigate to search screen
            Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) => Search(),
              ),
            );
          },
        ),
      ],
      iconTheme: IconThemeData(color: Colors.white), // Set the color of drawer icon to white
    ),

    drawer: Drawer(
      child: StreamBuilder<DocumentSnapshot>(
        stream: FirebaseFirestore.instance
            .collection('users')
            .doc(_auth.currentUser!.uid)
            .snapshots(),
        builder: (context, snapshot) {
          if (snapshot.hasError) {
            return Text('Error: ${snapshot.error}');
          }
          if (snapshot.connectionState == ConnectionState.waiting) {
            return Center(child: CircularProgressIndicator());
          }

          var userData = snapshot.data!.data() as Map<String, dynamic>;
          var profileImageUrl = userData['profileImageUrl'] ?? '';
          var username = userData['username'] ?? '';
          var phoneNumber = userData['phone'] ?? '';

          return ListView(
            padding: EdgeInsets.zero,
children: [
  DrawerHeader(
    decoration: BoxDecoration(color: Color(0xFF588FBA)), // Changed to #588FBA
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        CircleAvatar(
          backgroundImage: profileImageUrl.isNotEmpty
              ? NetworkImage(profileImageUrl)
              : null,
          backgroundColor:
          Colors.orange, // Default background color
          radius: 30, // Adjust as needed
          child: profileImageUrl.isEmpty
              ? Text(
            username[0].toUpperCase(), // Display the first letter of the username
            style: TextStyle(
              color: Colors.white,
              fontSize: 24, // Adjust the font size as needed
              fontWeight: FontWeight.bold, // Make the font bold
            ),
          )
              : null,
        ),
        SizedBox(height: 8),
        Row(
          children: [
            Expanded(
              child: Text(
                username,
                style:
                TextStyle(color: Colors.white, fontSize: 18),
              ),
            ),
            Icon(Icons.keyboard_arrow_down,
                color: Colors.white), // Arrow color set to white
          ],
        ),
        Text(
          phoneNumber,
          style: TextStyle(color: Colors.white),
        ),
      ],
    ),
),
ListTile(
  leading:
  Icon(Icons.group, color: Colors.grey), // Changed icon color
  title: Text('New Group'),
  onTap: () {
    // Navigate to new group screen
  },
),
ListTile(
  leading: Icon(Icons.contacts,
      color: Colors.grey), // Changed icon color
  title: Text('Contacts'),
  onTap: () {
    // Navigate to Contacts screen
  },
),
ListTile(
  leading: Icon(Icons.call,
      color: Colors.grey), // Changed icon color
  title: Text('Call'),
  onTap: () {
    // Navigate to Contacts screen
  },
),
ListTile(
  leading: Icon(Icons.emoji_people_outlined,
      color: Colors.grey), // Changed icon color
  title: Text('People Nearby'),
  onTap: () {
    // Navigate to People Nearby screen
  },
),
ListTile(
  leading: Icon(Icons.save,
      color: Colors.grey), // Changed icon color
  title: Text('Saved Messages'),
  onTap: () {
    // Navigate to Saved Messages screen
  },
),
ListTile(
  leading: Icon(Icons.settings,
      color: Colors.grey), // Changed icon color
```

```
            title: Text('Settings'),
            onTap: () {
              // Navigate to settings screen
              Navigator.push(
                  context,
                  MaterialPageRoute(
                      builder: (context) => ProfilePage(
                          userId: _auth.currentUser!.uid)));
            },
          ),
          Divider(
            color: Colors.grey[300], // Added a light grey line separator
            thickness: 1,
          ),
          ListTile(
            leading: Icon(Icons.person_add_outlined,
                color: Colors.grey), // Changed icon color
            title: Text('Invite Friends'),
            onTap: () {
              // Invite Friends
            },
          ),
          ListTile(
            leading: Icon(Icons.help_outline,
                color: Colors.grey), // Changed icon color
            title: Text('Telegram Features'),
            onTap: () {
              signOut();
            },
          ),
          ListTile(
            leading: Icon(Icons.logout,
                color: Colors.grey), // Changed icon color
            title: Text('Logout'),
            onTap: () {
              signOut();
            },
          ),
        ],
      );
    },
  ),
),
body: _buildUserList(),
  floatingActionButton: FloatingActionButton(
   onPressed: () {
    // Navigate to ContactsPage
    Navigator.push(
     context,
     MaterialPageRoute(
      builder: (context) => ContactsPage(), // Replace ContactsPage with the actual page
     ),
    );
   },
   child: Icon(Icons.edit, color: Colors.white,),
   backgroundColor: Color(0xFF3BACEB),
   shape: CircleBorder(),// Adjust the color to match the theme
  ),
  floatingActionButtonLocation: FloatingActionButtonLocation.endFloat,
 );
}


// build list of users except the current user
Widget _buildUserList() {
 return StreamBuilder<QuerySnapshot>(
  stream: FirebaseFirestore.instance.collection('users').snapshots(),
  builder: (context, snapshot) {
   if (snapshot.hasError) {
    return Text('Error: ${snapshot.error}');
   }
   if (snapshot.connectionState == ConnectionState.waiting) {
    return Center(child: CircularProgressIndicator());
   }

   return ListView(
    children: snapshot.data!.docs
      .map<Widget>((doc) => _buildUserListItem(doc))
      .toList(),
   );
  },
 );
}

// build individual user list items
Widget _buildUserListItem(DocumentSnapshot document) {
 Map<String, dynamic>? data = document.data() as Map<String, dynamic>?;
```

```
      if (_auth.currentUser != null &&
          _auth.currentUser!.phoneNumber != null &&
          data != null) {
        if (data['phone'] != null &&
            _auth.currentUser!.phoneNumber != data['phone']) {
          String displayName = data['username'] ?? data['phone'];

          return StreamBuilder<QuerySnapshot>(
            stream: FirebaseFirestore.instance
                .collection('chat_rooms')
                .doc(HomePage.constructChatRoomId(_auth.currentUser!.phoneNumber.toString(),
data['phone']))
                .collection('messages')
                .orderBy('timestamp', descending: true)
                .limit(1)
                .snapshots(),
            builder: (context, snapshot) {
              if (snapshot.hasData &&
                  snapshot.data!.docs.isNotEmpty) {
                var lastMessage = snapshot.data!.docs.first;
                var timestamp = lastMessage['timestamp'];
                var messageTimestamp = DateTime.fromMillisecondsSinceEpoch(timestamp.seconds * 1000);
                var formattedTimestamp = DateFormat.jm().format(messageTimestamp);

                return Column(
                  children: [
                    ListTile(
                      title: Text(
                        displayName,
                        style: TextStyle(fontWeight: FontWeight.bold),
                      ), // Make the username bold
                      subtitle: Text(
                        lastMessage['message'],
                        maxLines: 1,
                        overflow: TextOverflow.ellipsis,
                        style: TextStyle(color: Colors.grey), // Set text color to grey
                      ),
                      trailing: Text(
                        formattedTimestamp,
                        style: TextStyle(color: Colors.grey), // Set text color to grey
                      ),
                      leading: CircleAvatar(
                        backgroundImage: data['profileImageUrl'] != null
                            ? NetworkImage(data['profileImageUrl'])
```

```
                         : null,
                     backgroundColor: Colors.orange, // Default background color
                     radius: 40, // Increased the radius to make the profile icon bigger
                     child: data['profileImageUrl'] == null
                         ? Text(
                       displayName[0].toUpperCase(), // Display the first letter of the username
                       style: TextStyle(
                         color: Colors.white,
                         fontSize: 24, // Adjust the font size as needed
                         fontWeight: FontWeight.bold, // Make the font bold
                       ),
                     )
                         : null, // Show the first letter only when no profile image is available
                   ),
                   onTap: () {
                     Navigator.push(
                       context,
                       MaterialPageRoute(
                         builder: (context) => ChatPage(
                           receivedUserPhoneNumber: data['phone'] ?? '',
                         ),
                       ),
                     );
                   },
                 ),
                 Divider(
                   color: Colors.grey[300], // Added a light grey line separator
                   thickness: 1,
                 ),
               ],
             );
           } else {
             return SizedBox.shrink(); // Return an empty SizedBox if there are no messages
           }
         },
       );
     }
    }
   return SizedBox.shrink(); // Return an empty SizedBox if user data is invalid
 }
}
```


P
Percy
+911234567890
9:52 PM
10:02 PM
New Group
Contacts
Call
People Nearby
Saved Messages
Settings
Invite Friends
Telegram Features
Logout


Contacts
New Group
New Secret Chat
New Channel
Sorting order: Ascending
A
Annabeth
Online
G
Gargi
Offline
P
Preeti
Offline
P
Priya
Offline