Prachit Paralikar
D15A
43

MPL EXPT 5

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

THEORY:
Navigation:
Movement between different screens or pages in a Flutter app. Achieved using widgets like Navigator or MaterialApp. Triggered by user actions like tapping buttons or selecting items.

Gesture:
User's physical interaction with the touchscreen, like tapping or swiping.
Detected using widgets like GestureDetector.
Used to capture user input and trigger actions or events.

Routing:
Defines and manages navigation paths within the app. Done using Navigator widget and routes parameter of MaterialApp. Maps routes to screens or widgets, facilitating structured navigation.

SYNTAX:
Navigation:
```
Navigator.push(
  context,
  MaterialPageRoute(builder: (context) => NextScreen()),
);
```

Routing:
```
MaterialApp(
  routes: {
    '/next': (context) => NextScreen(),
  },
  // Main app content
);
```

Gestures:
```
GestureDetector(
  onTap: () {
    // Handle tap gesture
  },
  child: Container(
   // Widget content
  ),
);
```

WIDGET AND PROPERTIES:
In this experiment we have focussed on the navigation, gesture and routing widgets and we are using various properties like onpressed, pushnamed, etc.

CODE:

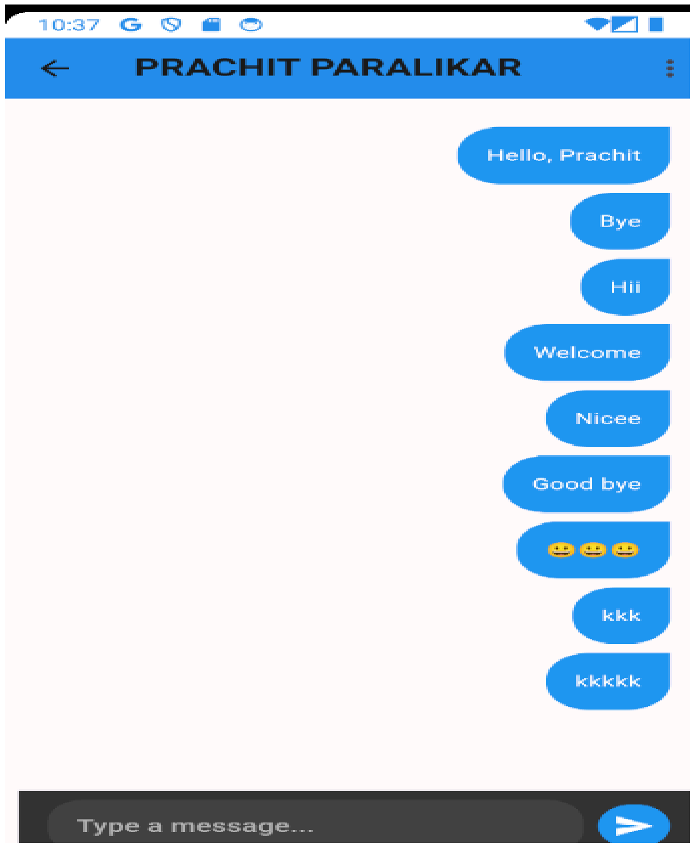
```
import 'package:flutter/material.dart';

import 'package:messenger_clone/helperfunctions/sharedpref_helper.dart';

import 'package:messenger_clone/services/auth.dart';

import 'package:messenger_clone/services/database.dart';

import 'package:messenger_clone/views/chatscreen.dart';

import 'package:messenger_clone/views/signin.dart';

import 'package:messenger_clone/views/stories_screen.dart'; // Import the Stories
screen

import 'package:cloud_firestore/cloud_firestore.dart';



class Home extends StatefulWidget {

  @override

  _HomeState createState() => _HomeState();

}



class _HomeState extends State<Home> {

  bool isSearching = false;
```

```
String? myName, myProfilePic, myUserName, myEmail;

Stream? usersStream, chatRoomsStream;


TextEditingController searchUsernameEditingController =

    TextEditingController();


getMyInfoFromSharedPreference() async {

  myName = await SharedPreferenceHelper().getDisplayName();

  myProfilePic = await SharedPreferenceHelper().getUserProfileUrl();

  myUserName = await SharedPreferenceHelper().getUserName();

  myEmail = await SharedPreferenceHelper().getUserEmail();

  setState(() {});

}


getChatRoomIdByUsernames(String a, String b) {

  if (a.substring(0, 1).codeUnitAt(0) > b.substring(0, 1).codeUnitAt(0)) {

    return "$b\_$a";

  } else {

    return "$a\_$b";

  }

}
```

```
onSearchBtnClick() async {
  isSearching = true;
  setState(() {});
  usersStream = await DatabaseMethods()
      .getUserByUserName(searchUsernameEditingController.text);

  setState(() {});
}

Widget chatRoomsList() {
  return StreamBuilder(
    stream: chatRoomsStream,
    builder: (context, snapshot) {
      return snapshot.hasData
          ? ListView.builder(
              itemCount: snapshot.data!.docs.length,
              shrinkWrap: true,
              itemBuilder: (context, index) {
                DocumentSnapshot ds = snapshot.data!.docs[index];
                return ChatRoomListTile(
                    ds["lastMessage"], ds.id, myUserName!);
              })
          : Center(child: CircularProgressIndicator());
    },
  );
}

Widget searchListUserTile(
    {required String profileUrl,
    required String name,
    required String username,
    required String email}) {
  return GestureDetector(
    onTap: () {
      var chatRoomId = getChatRoomIdByUsernames(myUserName!, username);
      Map<String, dynamic> chatRoomInfoMap = {
        "users": [myUserName, username]
      };
      DatabaseMethods().createChatRoom(chatRoomId, chatRoomInfoMap);
      Navigator.push(
          context,
          MaterialPageRoute(
              builder: (context) => ChatScreen(username, name)));
    },
    child: Container(
      margin: EdgeInsets.symmetric(vertical: 8),
      child: Row(
        children: [
          CircleAvatar(
            radius: 20,
            backgroundImage: NetworkImage(profileUrl),
          ),
          SizedBox(width: 12),
          Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [Text(name), Text(email)])
        ],
      ),
    ),
  );
}

Widget searchUsersList() {
  return StreamBuilder(
    stream: usersStream,
    builder: (context, snapshot) {
      return snapshot.hasData
          ? ListView.builder(
              itemCount: snapshot.data!.docs.length,
              shrinkWrap: true,
              itemBuilder: (context, index) {
                DocumentSnapshot ds = snapshot.data!.docs[index];
                return searchListUserTile(
                    profileUrl: ds["imgUrl"],
                    name: ds["name"],
                    email: ds["email"],
                    username: ds["username"]);
              },
            )
          : Center(
              child: CircularProgressIndicator(),
            );
    },
  );
}

getChatRooms() async {
  chatRoomsStream = await DatabaseMethods().getChatRooms();
  setState(() {});
}

onScreenLoaded() async {
  await getMyInfoFromSharedPreference();
  getChatRooms();
}

@override
void initState() {
  onScreenLoaded();
  super.initState();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text("Chats"),
      actions: [
        InkWell(
          onTap: () {
```

```
        AuthMethods().signOut(context).then((s) {
          Navigator.pushReplacement(
              context, MaterialPageRoute(builder: (context) => SignIn()));
        });
      },
      child: Padding(
        padding: const EdgeInsets.all(8.0),
        child: Icon(Icons.exit_to_app),
      ),
    )
  ],
),
body: Container(
  margin: EdgeInsets.symmetric(horizontal: 20),
  child: Column(
    children: [
      Row(
        children: [
          isSearching
              ? GestureDetector(
                  onTap: () {
                    isSearching = false;
        searchUsernameEditingController.text = "";
        setState(() {});
      },
      child: Padding(
          padding: EdgeInsets.only(right: 12),
          child: Icon(Icons.arrow_back)),
    )
  : Container(),
Expanded(
  child: Container(
    margin: EdgeInsets.symmetric(vertical: 16),
    padding: EdgeInsets.symmetric(horizontal: 16),
    decoration: BoxDecoration(
      border: Border.all(
        color: Colors.grey,
        width: 1,
      ),
      borderRadius: BorderRadius.circular(24),
    ),
    child: Row(
      children: [
        Expanded(
              child: TextField(
                controller: searchUsernameEditingController,
                decoration: InputDecoration(
                  border: InputBorder.none,
                  hintText: "Search",
                ),
              ),
            ),
            GestureDetector(
              onTap: () {
                if (searchUsernameEditingController.text != "") {
                  onSearchBtnClick();
                }
              },
              child: Icon(Icons.search),
            ),
          ],
        ),
      ),
    ),
  ],
),
      isSearching ? searchUsersList() : chatRoomsList(),
    ],
  ),
),
bottomNavigationBar: BottomAppBar(
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceAround,
    children: [
      IconButton(
        icon: Icon(Icons.chat),
        onPressed: () {
          // Add functionality for chat
        },
      ),
      IconButton(
        icon: Icon(Icons.view_headline),
        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(builder: (context) => StoriesScreen()),
          );
        },
          ),
        ],
      ),
    ),
  );
 }
}

class ChatRoomListTile extends StatefulWidget {
 final String lastMessage, chatRoomId, myUsername;
 ChatRoomListTile(this.lastMessage, this.chatRoomId, this.myUsername);

 @override
 _ChatRoomListTileState createState() => _ChatRoomListTileState();
}

class _ChatRoomListTileState extends State<ChatRoomListTile> {
 String profilePicUrl = "", name = "", username = "";

 getThisUserInfo() async {
  username =
    widget.chatRoomId.replaceAll(widget.myUsername, "").replaceAll("_", "");
    QuerySnapshot querySnapshot = await DatabaseMethods().getUserInfo(username);

    print(

        "something bla bla ${querySnapshot.docs[0].id} ${querySnapshot.docs[0]["name"]}
${querySnapshot.docs[0]["imgUrl"]}");

    name = "${querySnapshot.docs[0]["name"]}";

    profilePicUrl = "${querySnapshot.docs[0]["imgUrl"]}";

    setState(() {});

  }



  @override

  void initState() {

    getThisUserInfo();

    super.initState();

  }



  @override

  Widget build(BuildContext context) {

    return GestureDetector(

      onTap: () {

        Navigator.push(

            context,

            MaterialPageRoute(
            builder: (context) => ChatScreen(username, name)));
      },
      child: Container(
        margin: EdgeInsets.symmetric(vertical: 8),
        child: Row(
          children: [
            CircleAvatar(
              radius: 20,
              backgroundImage: NetworkImage(profilePicUrl),
            ),
            SizedBox(width: 12),
            Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Text(
                  name,
                  style: TextStyle(fontSize: 16),
                ),
                SizedBox(height: 3),
                Text(widget.lastMessage)
              ],
            )
          ],
        ),
      ),
    );
  }
}
```

OUTPUT:



Hello, Prachit

Bye

Hii

Welcome

Nicee

Good bye

😀😀😀

kkk

kkkkk

CONCLUSION:

Learned about navigation, routing and gestures.