Aim- To apply navigation, routing and gestures in Flutter

Theory -

- Gestures are an interesting feature in Flutter that allows us to interact with the mobile app (or any touch-based device). Generally, gestures define any physical action or movement of a user in the intention of specific control of the mobile device. Some of the examples of gestures are:

- When the moble screen is locked, you slide your finger across the screen to unlock it.
- Tapping a button on your mobile screen and.
- Tapping and holding on app icon on a touch-based device to drag it across screens.

We use all these gestures in everyday life to interact with your phone or touch-based device. Flutter divides the gesture system into two different layers, which are given below:

1. Pointers
2. Gestures

- Navigation in Flutter is theprocess of moving between different pages or screen in an app. This is achieved by using the 'Navigator' widget.
  Flutter provides several built-in widgets for implementing navigation in your app, including 'Bottom NavigationBar', 'TabBar', 'Drawer', 'AppBar' and 'materialPage Route'.
  To navigate between different pages, you can use the 'Navigator.push()' method to push a new route onto the stack and the 'Navigator.pop()' method to remove the current route from the stack.

- ~~Flutter~~ Flutter uses a routing system called Navigator to navigate between pages. It maintains a stack of pages.
  You can also use named routes to navigate to specific pages and pass data between pages using the Navigator's arguements.

**Page 1 - Login Page**
**Code -**
**login_tab_page.dart**

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:users_app/splashScreen/my_splash_screen.dart';

import '../global/global.dart';

import '../widgets/custom_text_field.dart';
import '../widgets/loading_dialog.dart';

class LoginTabPage extends StatefulWidget {
  @override
  State<LoginTabPage> createState() => _LoginTabPageState();
}

class _LoginTabPageState extends State<LoginTabPage> {
  TextEditingController emailTextEditingController = TextEditingController();
  TextEditingController passwordTextEditingController = TextEditingController();
  GlobalKey<FormState> formKey = GlobalKey<FormState>();

  validateForm() {
    if (emailTextEditingController.text.isNotEmpty &&
        passwordTextEditingController.text.isNotEmpty) {
      // allow user to login
      loginNow();
    } else {
      Fluttertoast.showToast(msg: 'Please provide email and password');
    }
  }

  loginNow() async {
```

```
  showDialog(
      context: context,
      builder: (c) {
        return LoadingDialogWidget(message: 'Checking credentials');
      });

  //authenticate the user
  User? currentUser;
  await FirebaseAuth.instance
      .signInWithEmailAndPassword(
          email: emailTextEditingController.text.trim(),
          password: passwordTextEditingController.text.trim())
      .then((auth) {
    currentUser = auth.user;
  }).catchError((errorMessage) {
    Navigator.pop(context);
    Fluttertoast.showToast(msg: 'Error Occurred: \n $errorMessage');
  });

  if (currentUser != null) {
    checkIfUserRecordExists(currentUser!);
  }
}

checkIfUserRecordExists(User currentUser) async {
  await FirebaseFirestore.instance
      .collection('users')
      .doc(currentUser.uid)
      .get()
      .then((record) async {
    // record exists
    if (record.exists) {
      // status approved
      if (record.data()!['status'] == 'approved') {
        // mozemo pribavljati podatke preko authenticationa ili direktno iz storaga
```

```
     await sharedPreferences!.setString('uid', currentUser.uid);
     await sharedPreferences!.setString('email', record.data()!['email']);
     await sharedPreferences!.setString('name', record.data()!['name']);
     await sharedPreferences!
       .setString('photoUrl', record.data()!['photoUrl']);

     List<String> userCartList = record.data()!['userCart'].cast<String>();
     await sharedPreferences!.setStringList('userCart', userCartList);
     // send user to homeScreen
     Navigator.push(
       context, MaterialPageRoute(builder: (c) => MySplashScreen()));
    } else {
     // status not approved
     FirebaseAuth.instance.signOut();
     Navigator.pop(context);
     Fluttertoast.showToast(
       msg:
         'You have been BLOCKED by admin.\nContact admin:
admin1@gmail.com');
    }
   } else {
    // record not exists
    FirebaseAuth.instance.signOut();
    Navigator.pop(context);
    Fluttertoast.showToast(msg: 'This users record do not exist.');
   }
  });
 }

 @override
 Widget build(BuildContext context) {
  return SingleChildScrollView(
   reverse: true,
   child: Column(
    children: [
Padding(
  padding: const EdgeInsets.all(8.0),
  child: Image.asset(
    'images/login.png',
    height: MediaQuery.of(context).size.height * 0.40,
  ),
),
Form(
  key: formKey,
  child: Column(
    children: [
      CustomTextField(
        textEditingController: emailTextEditingController,
        iconData: Icons.email,
        hintText: 'Email',
        isObsecure: false,
        enabled: true,
      ),
      CustomTextField(
        textEditingController: passwordTextEditingController,
        iconData: Icons.lock,
        hintText: 'Password',
        isObsecure: true,
        enabled: true,
      ),
      const SizedBox(height: 10.0),
    ],
  ),
),
ElevatedButton(
  style: ElevatedButton.styleFrom(
    backgroundColor: Colors.pinkAccent,
    padding: const EdgeInsets.symmetric(horizontal: 50, vertical: 12),
  ),
  onPressed: () {
```

```
              validateForm();
            },
            child: const Text(
              'Login',
              style:
                  TextStyle(color: Colors.white, fontWeight: FontWeight.bold),
            ),
          ),
          const SizedBox(height: 30),
        ],
      ),
    );
  }
}
```

### Page 2 - Home Page

**Code -**

**home_screen.dart**

```
import 'package:carousel_slider/carousel_slider.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:flutter_staggered_grid_view/flutter_staggered_grid_view.dart';
import 'package:users_app/functions/functions.dart';
import 'package:users_app/global/global.dart';
import 'package:users_app/models/sellers.dart';
import 'package:users_app/pushNotifications/push_notifications_system.dart';
import 'package:users_app/sellersScreens/sellers_ui_design_widget.dart';

import 'package:users_app/widgets/my_drawer.dart';

import '../splashScreen/my_splash_screen.dart';

class HomeScreen extends StatefulWidget {
  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  //metoda za onemogucavanje usera koji je blokiran da pristupi
  restrictBlockedUsersFromUsingUsersApp() async {
    await FirebaseFirestore.instance
        .collection('users')
        .doc(sharedPreferences!.getString('uid'))
        .get()
        .then((snapshot) {
      if (snapshot.data()!['status'] != 'approved') {
        showReusableSnackBar(context, 'You are blocked by admin.');
        showReusableSnackBar(context, 'Contact admin: admin1@gmail.com');

        FirebaseAuth.instance.signOut();
        Navigator.push(
            context, MaterialPageRoute(builder: (c) => MySplashScreen()));
      } else {
        cartMethods.clearCart(context);
      }
    });
  }

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

    PushNotificationSystem pushNotificationSystem = PushNotificationSystem();
    pushNotificationSystem.whenNotificationRecived(context);
    pushNotificationSystem.generateDeviceRecognitionToken();

    restrictBlockedUsersFromUsingUsersApp();
  }
```

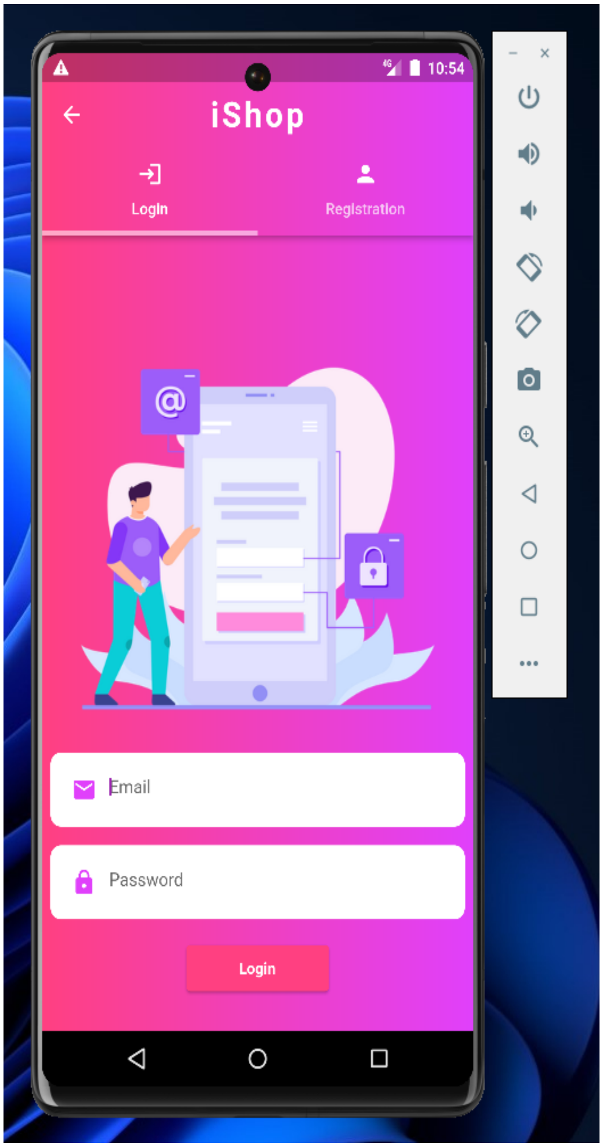
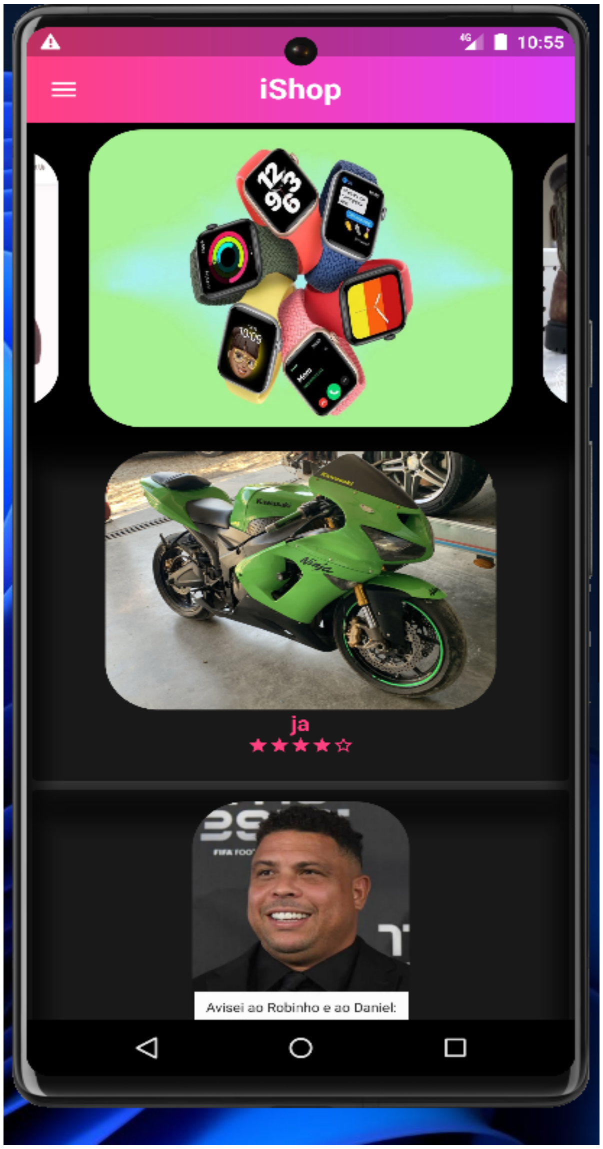
```
@override
Widget build(BuildContext context) {
 return Scaffold(
  backgroundColor: Colors.black,
  drawer: MyDrawer(),
  appBar: AppBar(
   flexibleSpace: Container(
    decoration: const BoxDecoration(
      gradient: LinearGradient(
     colors: [
      Colors.pinkAccent,
      Colors.purpleAccent,
     ],
     begin: FractionalOffset(0.0, 0.0),
     end: FractionalOffset(1.0, 0.0),
     stops: [0.0, 1.0],
     tileMode: TileMode.clamp,
    )),
   ),
   title: const Text(
    "iShop",
    style: TextStyle(
     fontSize: 24,
     fontWeight: FontWeight.bold,
    ),
   ),
   centerTitle: true,
  ),
  body: CustomScrollView(
   slivers: [
    //image slider
    SliverToBoxAdapter(
     child: Padding(
      padding: const EdgeInsets.all(6.0),
      child: SizedBox(
  height: MediaQuery.of(context).size.height * .3,
  width: MediaQuery.of(context).size.width,
  child: CarouselSlider(
    options: CarouselOptions(
      height: MediaQuery.of(context).size.height * .9,
      aspectRatio: 16 / 9,
      viewportFraction: 0.8,
      initialPage: 0,
      enableInfiniteScroll: true,
      reverse: false,
      autoPlay: true,
      autoPlayInterval: const Duration(seconds: 2),
      autoPlayAnimationDuration:
          const Duration(milliseconds: 800),
      autoPlayCurve: Curves.fastOutSlowIn,
      enlargeCenterPage: true,
      scrollDirection: Axis.horizontal,
    ),
    items: itemsImagesList.map((index) {
      return Builder(builder: (BuildContext c) {
        return Container(
          width: MediaQuery.of(context).size.width,
          margin: const EdgeInsets.symmetric(horizontal: 1.0),
          child: ClipRRect(
            borderRadius: BorderRadius.circular(40),
            child: Image.asset(
              index,
              fit: BoxFit.fill,
            ),
          ),
        );
      });
    }).toList(),
  ),
),
  ),
),

//query
//model
//design widget

StreamBuilder(
  stream:
      FirebaseFirestore.instance.collection("sellers").snapshots(),
  builder: (context, AsyncSnapshot dataSnapshot) {
    if (dataSnapshot.hasData) {
      return SliverStaggeredGrid.countBuilder(
        crossAxisCount: 1,
        staggeredTileBuilder: (c) => const StaggeredTile.fit(1),
        itemBuilder: (context, index) {
          Sellers model = Sellers.fromJson(
              dataSnapshot.data.docs[index].data()
                  as Map<String, dynamic>);

          return SellersUIDesignWidget(
            model: model,
          );
        },
        itemCount: dataSnapshot.data.docs.length,
      );
    } else {
      return const SliverToBoxAdapter(
        child: Center(
          child: Text(
            "No Sellers Data exists.",
          ),
        ),
      );
    }
```

```
            },
          ),
        ],
      ),
    );
  }
}
```

**pubspec.yaml**

```
name: users_app
description: A new Flutter project.

publish_to: "none"

version: 1.0.0+1

environment:
  sdk: ">=2.18.2 <3.0.0"

dependencies:
  carousel_slider: ^4.1.1
  cart_stepper: ^3.0.2
  cloud_firestore: ^4.0.4
  cupertino_icons: ^1.0.2
  firebase_auth: ^4.1.1
  firebase_core: ^2.1.1
  firebase_messaging: ^14.0.4
  firebase_storage: ^11.0.4
  flutter:
    sdk: flutter
  flutter_staggered_grid_view: ^0.4.1
  fluttertoast: ^8.1.1
  http: ^0.13.5
  image_picker: ^0.8.6
  intl: null
  provider: ^6.0.4
  shared_preferences: ^2.0.15
  smooth_star_rating_nsafe: ^1.0.0+1
dev_dependencies:
  flutter_lints: ^2.0.0
  flutter_test:
    sdk: flutter


flutter:
  uses-material-design: true


  assets:
    - images/
    - slider/
```

## Output Screen -

<table>
<tr><th colspan="2">Login Page ➡️Home Screen</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>