

EXPERIMENT NO: - 05

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

Theory: -

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.

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.

Gestures enable the app to respond to user interactions, making the application more dynamic and responsive.

➤ Navigation and Routing in Flutter

Navigation is the process of moving between different screens or pages in an app. Flutter provides a simple and effective way to handle this through the use of the `Navigator` widget and routes.

1. Using Navigator Widget

The `Navigator` widget manages a stack of routes, allowing for pushing and popping routes on the stack.

- Pushing a Route: To navigate to a new screen, use `Navigator.push()`.
- Popping a Route: To go back to the previous screen, use `Navigator.pop()`.

```
ElevatedButton(  
  onPressed: () {
```

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

2. Named Routes

Flutter also allows the use of named routes to navigate, which can make the routing process cleaner, especially in larger applications. `MaterialApp`

```
initialRoute: '/',  
routes: {  
  
  '/': (context) => HomeScreen(),  
  '/second': (context) => SecondScreen(),  
},  
);
```

Navigate to the route using `Navigator.pushNamed()`

```
Navigator.pushNamed(context, '/second');
```

Handling Gestures in Flutter

Gestures refer to user interactions with the app, such as taps, swipes, pinches, and drags. Flutter provides several widgets and gesture detectors to handle these interactions.

Tap Gestures

The most common gesture is the tap, which can be handled using the `GestureDetector` widget or specific buttons like `InkWell` or `ElevatedButton`.

Long Press Gesture

For long press gestures, Flutter provides the `onLongPress` callback in `GestureDetector` or `InkWell`.

Swipe and Drag Gestures

Flutter also provides swipe and drag gesture handling. The `onHorizontalDragUpdate` and `onVerticalDragUpdate` callbacks are used for dragging gestures.

Code:-

main.dart

```
import 'package:filmy_fun/pages/detail_page.dart';
import 'package:filmy_fun/service/constant.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart'; import
'package:filmy_fun/pages/login.dart'; import
'package:filmy_fun/pages/signup.dart'; import
'package:filmy_fun/pages/home.dart'; import
'package:filmy_fun/pages/bottomnav.dart'; import
'package:filmy_fun/pages/booking.dart';

import 'package:flutter_stripe/flutter_stripe.dart';

void main()async {

  WidgetsFlutterBinding.ensureInitialized();
  Stripe.publishableKey= publishedKey; await
  Firebase.initializeApp();

  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'FilmyFun',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        // This is the theme of your application.
        //
        // TRY THIS: Try running your application with "flutter run". You'll see
        // the application has a purple toolbar. Then, without quitting the app,
        // try changing the seedColor in the colorScheme below to Colors.green
```

```
    // and then invoke "hot reload" (save your changes or press the "hot
// reload" button in a Flutter-supported IDE, or press "r" if you used    //
the command line to start the app).
```

```
    //
```

```
    // Notice that the counter didn't reset back to zero; the application
// state is not lost during the reload. To reset the state, use hot    //
restart instead.
```

```
    //
```

```
    // This works for code too, not just values: Most code changes can be
// tested with just a hot reload.
```

```
    colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
    useMaterial3: true,
```

```
  ),
```

```
  home: Signup(),
```

```
);
```

```
}
```

```
}
```

```
class MyHomePage extends StatefulWidget {
```

```
  const MyHomePage({super.key, required this.title});
```

```
  // This widget is the home page of your application. It is stateful, meaning
// that it has a State object (defined below) that contains fields that affect //
how it looks.
```

```
  // This class is the configuration for the state. It holds the values (in this
```

```
  // case the title) provided by the parent (in this case the App widget) and
// used by the build method of the State. Fields in a Widget subclass are //
always marked "final".
```

```
  final String title;
```

```
  @override
```

```
  State<MyHomePage> createState() => _MyHomePageState();
```

```
}
```

```

class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;

  void _incrementCounter() {
    setState(() {
      // This call to setState tells the Flutter framework that something has
      // changed in this State, which causes it to rerun the build method below
      // so that the display can reflect the updated values. If we changed
      // _counter without calling setState(), then the build method would not be
      // called again, and so nothing would appear to happen.
      _counter++;
    });
  }

  @override
  Widget build(BuildContext context) {
    // This method is rerun every time setState is called, for instance as done
    // by the _incrementCounter method above.
    //
    // The Flutter framework has been optimized to make rerunning build methods
    // fast, so that you can just rebuild anything that needs updating rather
    // than having to individually change instances of widgets.
    return Scaffold(
      appBar: AppBar(
        // TRY THIS: Try changing the color here to a specific color (to
        // Colors.amber, perhaps?) and trigger a hot reload to see the AppBar
        // change color while the other colors stay the same.
        backgroundColor: Theme.of(context).colorScheme.inversePrimary, //
        // Here we take the value from the MyHomePage object that was created by
        // the App.build method, and use it to set our appBar title.
        title: Text(widget.title),
      ),
      body: Center(
        // Center is a layout widget. It takes a single child and positions it
        // in the middle of the parent. child: Column(

```

```

    // Column is also a layout widget. It takes a list of children and
    // arranges them vertically. By default, it sizes itself to fit its    //
    children horizontally, and tries to be as tall as its parent.

    //

    // Column has various properties to control how it sizes itself and
    // how it positions its children. Here we use mainAxisAlignment to
    // center the children vertically; the main axis here is the vertical
    // axis because Columns are vertical (the cross axis would be
    // horizontal).

    //

    // TRY THIS: Invoke "debug painting" (choose the "Toggle Debug Paint"
    // action in the IDE, or press "p" in the console), to see the    //
    wireframe for each widget.

```

```

    mainAxisAlignment: MainAxisAlignment.center,
    children: <Widget>[
      const Text(
        'You have pushed the button this many times:',
      ),
      Text(
        '$_counter',
        style: Theme.of(context).textTheme.headlineMedium,
      ),
    ],
  ),
),
),

floatingActionButton: FloatingActionButton(
  onPressed: _incrementCounter,    tooltip:
  'Increment',    child: const Icon(Icons.add),

  ), // This trailing comma makes auto-formatting nicer for build methods.

);

}}

```

Signup.dart

```

import 'package:filmy_fun/service/database.dart';
import 'package:firebase_auth/firebase_auth.dart';

```

```

import 'package:flutter/material.dart'; import
'package:filmy_fun/pages/login.dart'; import
'package:filmy_fun/service/shared_pref.dart'; import
'package:random_string/random_string.dart'; import
'bottomnav.dart';

class Signup extends StatefulWidget {
  const Signup({super.key});

  @override
  State<Signup> createState() => _SignupState();
}

class _SignupState extends State<Signup> {
  String email = "", password = "", name = "";

  TextEditingController namecontroller = TextEditingController();
  TextEditingController passwordcontroller = TextEditingController();
  TextEditingController mailcontroller = TextEditingController();

  registration() async {
    if (passwordcontroller.text.isNotEmpty &&
        namecontroller.text.isNotEmpty &&
        mailcontroller.text.isNotEmpty) {
      try {
        UserCredential userCredential = await FirebaseAuth.instance
          .createUserWithEmailAndPassword(
            email: mailcontroller.text, password: passwordcontroller.text);
        String id = randomAlphaNumeric(10);
        Map<String, dynamic> userInfoMap = {
          "Name": namecontroller.text,
          "Email": mailcontroller.text,
          "Id": id,
          "Image": ""
        };

        await SharedPreferencesHelper().saveUserDislayName(namecontroller.text);
        await SharedPreferencesHelper().saveUserEmail(mailcontroller.text);      await

```

```

SharedPreferenceHelper().saveUserID(id);    await
SharedPreferenceHelper().saveUserImage("");    await
DatabaseMethods().addUserDetails(userInfoMap, id);
ScaffoldMessenger.of(context).showSnackBar(SnackBar(

    backgroundColor: Colors.green,
content: Text(

    "Registered Successfully!",

    style: TextStyle(        color:
Colors.white,        fontWeight:
FontWeight.bold,

    fontSize: 20.0),

    ));

Navigator.pushReplacement(
    context, MaterialPageRoute(builder: (context) => Bottomnav()));

} on FirebaseAuthException catch (e) {
if (e.code == 'weak-password') {

    ScaffoldMessenger.of(context).showSnackBar(SnackBar(

        backgroundColor: Colors.orangeAccent,
content: Text(

            "Password Provided is too Weak",

            style: TextStyle(fontSize: 18.0),

            ));

    } else if (e.code == "email-already-in-use") {

        ScaffoldMessenger.of(context).showSnackBar(SnackBar(

            backgroundColor: Colors.orangeAccent,
content: Text(

                "Account Already exists",

                style: TextStyle(fontSize: 18.0),

                ));

        }

    }

}

}

@override

```



```

Widget build(BuildContext context) {
return Scaffold(  backgroundColor:
Colors.black,  body:
SingleChildScrollView(

  child: Padding(

    padding: const EdgeInsets.symmetric(horizontal: 20.0),
  child: Column(

    crossAxisAlignment: CrossAxisAlignment.start,
    children: [

      const SizedBox(height: 50.0),
      Image.asset("images/signin.png"),      const
      SizedBox(height: 20.0),

      const Text(
"Welcome!",

        style: TextStyle(

          color: Color.fromARGB(157, 255, 255, 255),
fontSize: 34.0,

          fontWeight: FontWeight.w500,

        ),

      ),

      const Text(
"SignUp",      style:
TextStyle(      color:
Colors.white,

        fontSize: 45.0,

        fontWeight: FontWeight.bold,

      ),

    ),

    const SizedBox(height: 30.0),

    buildInputField("Name", "Enter Name", Icons.person, namecontroller),
buildInputField("Email", "Enter Email", Icons.email, mailcontroller),
buildInputField("Password", "Enter Password", Icons.lock, passwordcontroller, obscureText:
true),

    const SizedBox(height: 30.0),
Center(      child: Column(
children: [

```

```
GestureDetector(
onTap: registration,
child: GestureDetector(
    onTap: () {
        if (namecontroller.text.isNotEmpty &&
            mailcontroller.text.isNotEmpty &&
                passwordcontroller.text.isNotEmpty) {
            setState(() {
                name = namecontroller.text;
                email = mailcontroller.text;
                password = passwordcontroller.text;
            });
            registration();
        }
    },
    child: Container(
        width: 170,
        padding: const EdgeInsets.all(12),
        decoration: BoxDecoration(
            color: Colors.white,
            borderRadius: BorderRadius.circular(30),
        ),
        child: const Text(
            "SignUp", style:
            TextStyle(
                color:
                Color(0xff6b63ff),
                fontSize: 25.0,
                fontWeight: FontWeight.bold,
            ),
            textAlign: TextAlign.center,
        ),
    ),
),
),
```

```

        const SizedBox(height: 20.0),
      Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          const Text(
            "Already have an account?",
            style: TextStyle(
              color: Color.fromARGB(175, 255, 255, 255),
              fontSize: 18.0,
              fontWeight: FontWeight.w500,
            ),
          ),
          GestureDetector(
            onTap: () {
              Navigator.push(
                context,
                MaterialPageRoute(
                  builder: (context) => const Login()),
                ),
            child: const Text(
              " LogIn",
              style:
                TextStyle(
                  color:
                    Colors.white,
                  fontSize: 18.0,
                  fontWeight: FontWeight.bold,
                ),
            ),
          ],
        ),
        const SizedBox(height: 30.0),
      ],
    ),
  ),
),

```

```

    ],
  ),
),
),
);
}

```

```

Widget buildInputField(String label, String hint, IconData icon,
  TextEditingController controller,
  {bool obscureText = false}) {
  return Padding(
    padding: const EdgeInsets.only(bottom: 20.0),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Text(
          label,
          style: const TextStyle(
            color: Colors.white,
            fontSize: 20.0,
            fontWeight: FontWeight.w500,
          ),
        ),
        TextField(
          controller:
            controller,
          obscureText:
            obscureText,
          cursorColor: Colors.white, // Cursor color set to white
          style:
            const TextStyle(color: Colors.white), // Text color set to white
          decoration: InputDecoration(
            hintText: hint,
            hintStyle: const TextStyle(color: Colors.white), // Hint text color set to white
            suffixIcon: Icon(icon, color: Colors.white),
            enabledBorder: const
              UnderlineInputBorder(
                borderSide: BorderSide(color: Colors.white), // Underline color
              ),

```

```

        focusedBorder: const UnderlineInputBorder(
borderSide:
BorderSide(color: Colors.white), // Underline when focused
        ),
    ),
    ),
],
),
);
}}

```

Login.dart

```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:filmy_fun/pages/bottomnav.dart';
import 'package:filmy_fun/pages/home.dart'; import
'package:filmy_fun/service/database.dart'; import
'package:filmy_fun/service/shared_pref.dart'; import
'package:firebase_auth/firebase_auth.dart'; import
'package:flutter/material.dart';

import 'package:filmy_fun/pages/signup.dart';

class Login extends StatefulWidget {
  const Login({super.key});

  @override
  State<Login> createState() => _LoginState();
}

class _LoginState extends State<Login> {
  String email = "", password = "", myname = "", myid = "", myimage = "";
  TextEditingController passwordcontroller = TextEditingController();
  TextEditingController mailcontroller = TextEditingController();

  userLogin() async {
    try {
      await FirebaseAuth.instance.signInWithEmailAndPassword(
email: email,

```

```

        password: password,
    );

    QuerySnapshot querySnapshot = await
    DatabaseMethods().getUserbyemail(email);
    myname = "${querySnapshot.docs[0]["Name"]}";
    myid = "${querySnapshot.docs[0]["Id"]}";

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

    await SharedPreferencesHelper().saveUserImage(myimage);
    await SharedPreferencesHelper().saveUserEmail(email);    await
    SharedPreferencesHelper().saveUserDislayName(myname);

    await SharedPreferencesHelper().saveUserID(myid);

    Navigator.push(context, MaterialPageRoute(builder: (context) => Bottomnav()));

    } on FirebaseAuthException catch (e) {
    String errorMessage = "";    if (e.code
    == 'user-not-found') {

        ScaffoldMessenger.of(context).showSnackBar(SnackBar(
        content: Text("No user found for that email.",

            style: TextStyle(fontSize: 18.0, color: Colors.black),

            ));

    } else if (e.code == 'wrong-password') {

        ScaffoldMessenger.of(context).showSnackBar(SnackBar(

            backgroundColor: Colors.white,
        content: Text("Wrong password provided."

            ));

    }

    }

    }

    @override

    Widget build(BuildContext context) {

    return Scaffold(

        backgroundColor: Colors.black,

        body: Padding(

```

```

padding: const EdgeInsets.symmetric(horizontal: 20.0),
child: Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: [
    Image.asset("images/signin.png"),
    const SizedBox(height: 20.0),
    const Text(
      "Welcome!",
      style: TextStyle(
        color: Color.fromARGB(157, 255, 255, 255),
        fontSize: 34.0,
        fontWeight: FontWeight.w500,
      ),
    ),
    GestureDetector(
      onTap: () {
        if (mailcontroller.text.isNotEmpty &&
            passwordcontroller.text.isNotEmpty) {
          setState(() {
            email =
            mailcontroller.text;
            password =
            passwordcontroller.text;
            userLogin();
          });
        }
      },
      child: const Text(
        "Login",
        style:
        TextStyle(
          color:
          Colors.white,
          fontSize: 45.0,
          fontWeight: FontWeight.bold,
        ),
      ),
    ),
  ],
),

```

```

        const SizedBox(height: 50.0),

        const Text(
"Email",          style:
TextStyle(        color:
Colors.white,

        fontSize: 20.0,

        fontWeight: FontWeight.w500,

        ),

        ),

        TextField(      controller:
mailcontroller,      cursorColor: Colors.black,
style: const TextStyle(color: Colors.white),
decoration: const InputDecoration(
hintText: "Enter Email",

        hintStyle: TextStyle(color: Colors.grey),

        ),

        ),

        const SizedBox(height: 50.0),

        const Text(
"Password",       style:
TextStyle(        color:
Colors.white,

        fontSize: 20.0,

        fontWeight: FontWeight.w500,

        ),

        ),

        TextField(

        controller: passwordcontroller,
cursorColor: Colors.black,      style: const
TextStyle(color: Colors.white),
obscureText: true,      decoration: const
InputDecoration(      hintText: "Enter
Password",      hintStyle: TextStyle(color:
Colors.grey),

        ),

        ),

```



```

        const SizedBox(height: 20.0),
      Row(
        mainAxisAlignment: MainAxisAlignment.end,
        children: [
          Text(
            "Forgot Password?",
            style: TextStyle(
              color: Colors.white,
              fontSize: 18.0,
              fontWeight: FontWeight.w500,
            ),
          ),
        ],
      ),
      const SizedBox(height: 50.0),
      Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
          GestureDetector(
            onTap: () {
              Navigator.push(context,
                MaterialPageRoute(builder: (context) => Signup()));
            },
            child: Container(
              width: 170,
              padding: EdgeInsets.all(10),
              decoration: BoxDecoration(
                color: Colors.white,
                borderRadius: BorderRadius.circular(30)),
              child: Text(
                "SignUp",
                style: TextStyle(
                  color: Color(0xff6b63ff),
                  fontSize: 25.0,
                  fontWeight:
                    FontWeight.bold),
                textAlign: TextAlign.center,
              ),
            ),
          ),
        ],
      ),
    ],
  ),
)

```

```

GestureDetector(
onTap: () {
Navigator.push(
context,

MaterialPageRoute(

builder: (context) => const Bottomnav()));

},

child: Container(
width: 170,
padding: EdgeInsets.all(10),
decoration:
BoxDecoration(
color: Color(0xff6b63ff),
borderRadius: BorderRadius.circular(30)),
child: Text(
"LogIn",
style:
TextStyle(
color: Colors.white,

fontSize: 25.0,
fontWeight: FontWeight.bold),

textAlign: TextAlign.center,

),

),

),

],

)

],

),

),

);

```

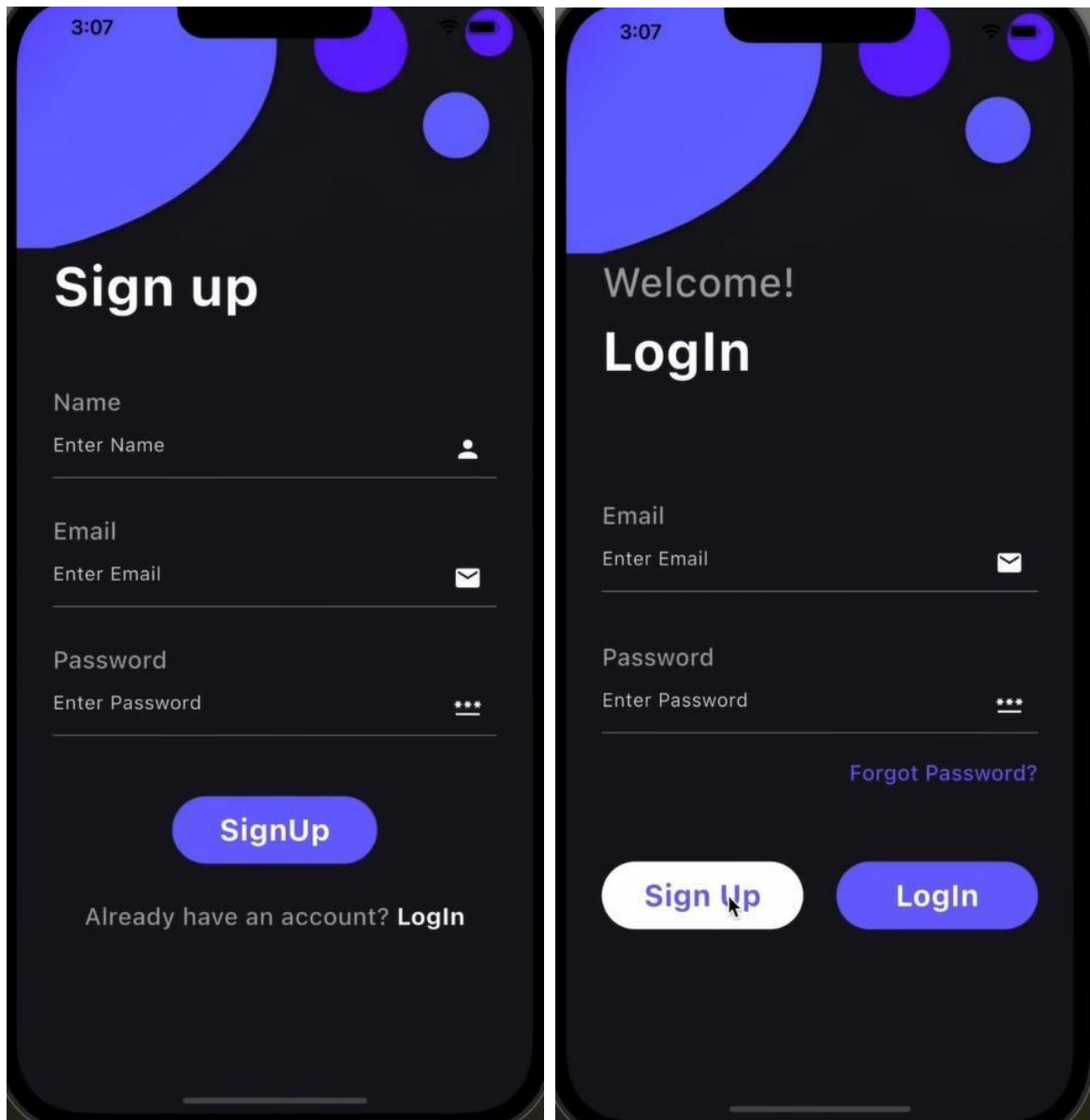
}}

OU

TPU

T:

After clicking on Already have an account? It navigates to login



On Homepage after clicking on Movie image it navigates to Details page

