

EXPERIMENT NO: - 05

Name:-Pranav Pol

Class:- D15A

Roll:No: - 41

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

Theory: -

Navigation, Routing, and Gesture Handling in Flutter

In Flutter, screens or pages are referred to as **routes**, and each route is essentially a widget. This concept is similar to **Activities** in Android. Navigating between pages defines an app's workflow, and the mechanism for handling this is known as **routing**.

Flutter provides a built-in routing system using **MaterialPageRoute**, along with the `Navigator.push()` and `Navigator.pop()` methods to move between routes.

Additionally, **gestures** allow apps to respond to user interactions like taps, swipes, and drags, making applications more dynamic and user-friendly.

Navigation and Routing in Flutter

1. Using the Navigator Widget

Flutter's Navigator widget manages a **stack** of routes, enabling seamless navigation between screens.

- **Pushing a Route:** Moves to a new screen using `Navigator.push()`.
- **Popping a Route:** Returns to the previous screen using `Navigator.pop()`.

Example:

```
ElevatedButton(  
  onPressed: () {  
    Navigator.push(  
      context,  
      MaterialPageRoute(builder: (context) => SecondScreen()),  
    );  
  },  
  child: Text('Go to Second Screen'),  
);
```

2. Using Named Routes

For larger applications, **named routes** provide a cleaner and more structured way to manage navigation.

Step 1: Define Routes in MaterialApp

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

Step 2: Navigate Using Navigator.pushNamed()

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

Handling Gestures in Flutter

Gestures enable user interaction through **taps, swipes, pinches, and drags**. Flutter provides various widgets and gesture detectors to manage these interactions effectively.

1. Tap Gestures

Taps are one of the most common interactions and can be handled using:

- GestureDetector
- InkWell
- ElevatedButton

Example (Tap Gesture using GestureDetector):

```
GestureDetector(  
  onTap: () {  
    print("Tapped!");  
  },  
  child: Container(  
    padding: EdgeInsets.all(20),  
    color: Colors.blue,  
    child: Text('Tap Me'),  
  ),  
);
```

2. Long Press Gestures

Long-press interactions can be captured using the `onLongPress` callback in `GestureDetector` or `InkWell`.

```
InkWell(  
  onLongPress: () {  
    print("Long Pressed!");  
  },  
  child: Container(  
    padding: EdgeInsets.all(20),  
    color: Colors.red,  
    child: Text('Long Press Me'),  
  ),  
);
```

3. Swipe and Drag Gestures

Flutter provides built-in methods like `onHorizontalDragUpdate` and `onVerticalDragUpdate` to detect swipe and drag actions.

Example (Swipe Detection):

```
GestureDetector(  
  onHorizontalDragUpdate: (details) {  
    if (details.primaryDelta! > 0) {  
      print("Swiped Right!");  
    } else {  
      print("Swiped Left!");  
    }  
  },  
  child: Container(  
    padding: EdgeInsets.all(20),  
    color: Colors.green,  
    child: Text('Swipe Me'),  
  ),  
);
```

sign_in_page.dart

```
class SignInPage extends StatefulWidget {
  @override
  _SignInPageState createState() =>
  _SignInPageState();
}
```

```
class _SignInPageState extends
State<SignInPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        padding: EdgeInsets.symmetric(horizontal:
15, vertical: 15),
        child: Column(
          children: [
            _headerWidget(),
            SizedBox(
              height: 10,
            ),
            _formWidget(),
          ],
        ),
      ),
    );
  }
}
```

```
child: Image.asset('assets/logo.png'),
)
```

$$\left\{ \begin{array}{l}] \\) \\ \end{array} \right.$$

```
Widget _formWidget() {
  return Expanded(
    child: Column(
      mainAxisAlignment:
MainAxisAlignment.center,
      children: [
        Container(
          padding:
EdgeInsets.symmetric(horizontal: 8),
          decoration: BoxDecoration(
            color: Colors.grey[800],
            borderRadius: BorderRadius.all(
              Radius.circular(5),
            ),
          ),
        ),
        child: TextFormField(
          decoration: InputDecoration(
            labelStyle: TextStyle(
              fontSize: 14,
              color: Colors.white,
            ),
            border: InputBorder.none,
            labelText: "Email or phone
number"),
          ),
        ),
        SizedBox(
          height: 10,
        ),
        Container(
          padding:
EdgeInsets.symmetric(horizontal: 8),
          decoration: BoxDecoration(
            color: Colors.grey[800],
            borderRadius: BorderRadius.all(
              Radius.circular(5),
            ),
          ),
        ),
        child: TextFormField(
          obscureText: true,
          decoration: InputDecoration(
```

```

);
labelStyle: TextStyle(
    fontSize: 14,
),
color: Colors.white,
border: InputBorder.none,
labelText: "Password"),
),
),
SizedBox(
    height: 15,
),
Container(
    alignment: Alignment.center,
    padding: EdgeInsets.symmetric(vertical:
15),
    width: double.maxFinite,
    decoration: BoxDecoration(
        color: Colors.transparent,
        border: Border.all(color:
Colors.grey[600] ?? Colors.grey, width: 2),
    ),
    child: Text("Sign In"),
),
SizedBox(
    height: 15,
),
Text("Need Help?"),
SizedBox(
    height: 15,
),
Text(
    "New to Netflix? Sign up now.",
    style: TextStyle(
        fontWeight: FontWeight.bold,
    ),
),
SizedBox(
    height: 20,
),
Text(
    "Sign-in is protected by Google
reCAPTCHA to ensure you're not a bot. Learn
more.",
    style: TextStyle(fontSize: 12,),
    textAlign: TextAlign.center,
),
],
),

```

homescreen.dart

```
import
'package:cached_network_image/cached_network_image.dart';
import 'package:flutter/material.dart';
import 'package:movflix/common/utils.dart';
import 'package:movflix/models/movie_model.dart';
import
'package:movflix/models/tv_series_model.dart';
import 'package:movflix/screens/search_screen.dart';
import 'package:movflix/services/api_services.dart';
import
'package:movflix/widgets/upcoming_movie_card_widget.dart';
```

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

```
  @override
  State<HomeScreen> createState() =>
  _HomeScreenState();
}
```

```
class _HomeScreenState extends
State<HomeScreen> {
  ApiService apiServices = ApiService();

  late Future<MovieModel> upcomingFuture;
  late Future<MovieModel> nowPlaying;
  late Future<TvSeriesModel> topRatedShows;
```

```
  @override
  void initState() {
    upcomingFuture =
apiServices.getUpcomingMovies();
    nowPlaying =
apiServices.getNowPlayingMovies();
    topRatedShows =
apiServices.getTopRatedSeries();
    super.initState();
  }
```

```
  @override
  Widget build(BuildContext context) {
    var size = MediaQuery.of(context).size;
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.black,
        title: Image.asset(
          'assets/logo.png',
          height: 50,
          width: 120,
        ),
```

```
actions: [
  Padding(
    padding: const EdgeInsets.only(right: 20.0),
    child: InkWell(
      onTap: () {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => const
SearchScreen(),
          ),
        );
      },
      child: const Icon(
        Icons.search,
        size: 30,
        color: Colors.white,
      ),
    ),
  ),
  ClipRRect(
    borderRadius: BorderRadius.circular(6),
    child: InkWell(
      onTap: () {},
      child: Container(
        color: Colors.blue,
        height: 27,
        width: 27,
      ),
    ),
  ),
  const SizedBox(width: 20),
],
),
body: SingleChildScrollView(
  child: Column(
    children: [
      FutureBuilder<TvSeriesModel>(
        future: topRatedShows,
        builder: (context, snapshot) {
          if (snapshot.connectionState ==
ConnectionState.waiting) {
            return const Center(child:
CircularProgressIndicator());
          }
          if (snapshot.hasData &&
snapshot.data!.results.isNotEmpty) {
            return SizedBox(
              // width: size.width,
              height: (size.height * 0.5 < 300) ? 300 :
size.height * 0.5,
              child: PageView.builder(
                itemCount:
snapshot.data!.results.length
controller: PageController(viewportFraction:
```

```

snapshot.data!.results.length == 1 ? 1.0 : 0.9),
  itemBuilder: (context, index) {
    var show =
snapshot.data!.results[index];
    return ClipRRect(
      borderRadius:
BorderRadius.circular(10),
      child: Stack(
        children: [
          CachedNetworkImage(
            imageUrl:
"$imageUrl${show.backdropPath}",
            fit: BoxFit.cover,
            width: double.infinity,
            height: double.infinity,
          ),
          Container(
            decoration: BoxDecoration(
              gradient: LinearGradient(
                colors:
[Colors.black.withOpacity(0.6), Colors.transparent],
                begin:
Alignment.bottomCenter,
                end: Alignment.topCenter,
              ),
            ),
          Positioned(
            bottom: 50,
            left: 20,
            child: Column(
              crossAxisAlignment:
CrossAxisAlignment.start,
              children: [
                Text(
                  show.name.toString(),
                  style: const TextStyle(
                    fontSize: 24,
                    fontWeight:
FontWeight.bold,
                    color: Colors.white,
                  ),
                ),
                const SizedBox(height: 10),
                Row(
                  children: [
                    ElevatedButton.icon(
                      onPressed: () {}, icon: const
Icon(Icons.play_arrow),
                      label: const Text('Play'),
                      style:
ElevatedButton.styleFrom(
                        backgroundColor:
Colors.white,
                        foregroundColor:

```

```

Colors.black,
                      ),
                    ),
                    const SizedBox(width: 10),
                    ElevatedButton.icon(
                      onPressed: () {},
                      icon: const
Icon(Icons.info_outline),
                      label: const Text('Info'),
                      style:
ElevatedButton.styleFrom(
                        backgroundColor:
Colors.grey[800],
                        foregroundColor:
Colors.white,
                      ),
                    ),
                  ],
                ),
              ],
            ),
          );
        },
      ),
    );
  }
  return const SizedBox();
},
const SizedBox(height: 20),
SizedBox(
  height: 220,
  child: UpcomingMovieCard(
    future: nowPlaying,
    headlineText: 'Now Playing',
  ),
),
const SizedBox(height: 20),
SizedBox(
  height: 220,
  child: UpcomingMovieCard(
    future: upcomingFuture
headlineText: 'Upcoming Movies',
  ),
),
],
),
);
}
}

```

movie_detailed_screen.dart

```
import
'package:cached_network_image/cached_network_image.dart';
import 'package:flutter/material.dart';
import 'package:movflix/common/utils.dart';
import
'package:movflix/models/movie_detail_model.dart';
import
'package:movflix/models/movie_recommendation_model.dart';
import
'package:movflix/services/api_services.dart';

class MovieDetailScreen extends
StatefulWidget {
  final int movieId;
  const MovieDetailScreen({super.key, required
this.movieId});

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

class MovieDetailScreenState extends
State<MovieDetailScreen> {
  ApiService apiServices = ApiService();

  late Future<MovieDetailModel> movieDetail;
  late Future<MovieRecommendationsModel>
movieRecommendationModel;

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

  fetchInitialData() {
    movieDetail =
apiServices.getMovieDetail(widget.movieId);
    movieRecommendationModel =
apiServices.getMovieRecommendations(widget.
movieId);
    setState(() {});
  }
}
```

```
}

@override
Widget build(BuildContext context) {
  var size = MediaQuery.of(context).size;
  print(widget.movieId);
  return Scaffold(
    body: SingleChildScrollView(
      child: FutureBuilder(
        future: movieDetail,
        builder: (context, snapshot) {
          if (snapshot.hasData) {
            final movie = snapshot.data;

            String genresText =
movie!.genres.map((genre) =>
genre.name).join(', ');

            return Column(
              children: [
                Stack(
                  children: [
                    Container(
                      height: size.height * 0.4,
                      decoration: BoxDecoration(
                        image: DecorationImage(
                          image: NetworkImage(
"$imageUrl${movie.posterPath}"),
                          fit: BoxFit.cover),
                        child: SafeArea(
                          child: Row(
                            mainAxisAlignment:
MainAxisAlignment.spaceBetween,
                            crossAxisAlignment:
CrossAxisAlignment.start,
                            children: [
                              IconButton(
                                icon: const
Icon(Icons.arrow_back_ios,
                                  color: Colors.white),
                                onPressed: () {
                                  Navigator.pop(context);
                                },
                              ),
                              ),
                            ),
                          ),
                        ),
                      ],
                    ),
                  ],
                ),
              ],
            ),
          ),
        ),
      ],
    ),
  );
}
```



```

),
    Padding(
      padding:
        const EdgeInsets.only(top: 25, left:
10, right: 10),
      child: Column(
        crossAxisAlignment:
CrossAxisAlignment.start,
        children: [
          Text(
            movie.title,
            style: const TextStyle(
              fontSize: 22,
              fontWeight: FontWeight.bold,
            ),
          ),
          const SizedBox(height: 15),
          Row(
            children: [
              Text(
movie.releaseDate.year.toString(),
                style: const TextStyle(
                  color: Colors.grey,
                ),
              ),
              const SizedBox(
                width: 30,
              ),
              Text(
                genresText,
                style: const TextStyle(
                  color: Colors.grey,
                  fontSize: 17,
                ),
              ),
            ],
          ),
          const SizedBox(
            height: 30,
          ),
          Text(
            movie.overview,
            maxLines: 6,
            overflow:
TextOverflow.ellipsis,
            style: const TextStyle(
              color: Colors.white, fontSize:
16),
          ),
        ],
      ),
    ),
  ),

```

```

],
    ),
  ),
  const SizedBox(
    height: 30,
  ),
  FutureBuilder(
    future:
movieRecommendationModel,
    builder: (context, snapshot) {
      if (snapshot.hasData) {
        final movie = snapshot.data;

        return movie!.results.isEmpty
          ? const SizedBox()
          : Column(
              crossAxisAlignment:
CrossAxisAlignment.start,
              children: [
                const Text(
                  "More like this",
                  maxLines: 6,
                  overflow:
TextOverflow.ellipsis,
                  style: TextStyle(
                    color: Colors.white,
                    fontWeight:
FontWeight.bold,
                  ),
                ),
                const SizedBox(height: 20),
                GridView.builder(
                  physics:
const
NeverScrollableScrollPhysics(),
                  shrinkWrap: true,
                  padding: EdgeInsets.zero,
                  scrollDirection: Axis.vertical,
                  itemCount:
movie.results.length,
                  gridDelegate:
const
SliverGridDelegateWithFixedCrossAxisCount(
                    crossAxisCount: 3,
                    mainAxisSpacing: 15,
                    childAspectRatio: 1.5 / 2,
                  ),
                  itemBuilder: (context, index)
{
                    return InkWell(

```

```

onTap: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) =>
          MovieDetailScreen(
            movieId: movie

.results[index].id),
      ),
    );
  },
  child:
CachedNetworkImage(
  imageUrl:

"$imageUrl${movie.results[index].posterPath}"
,
  ),
  ),
  },
  ),
  ],
  );
}
return const Text("Something
Went wrong");
},
),
],
);
}
return const SizedBox();
},
),
),
);
}
}

```





