MAD and PWA Lab

Name: Sarvadnya Awaghad Class: D15A Roll no:04

Experiment - 5

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

Theory:

Navigation:
Navigation is a fundamental aspect of mobile app development that involves transitioning between different screens or pages. In Flutter, the Navigator class plays a central role in managing the navigation stack, allowing developers to push and pop routes as users move through the app.

Navigator Class:
The Navigator class handles the navigation stack, maintaining a history of routes. The push method adds a new route to the stack, typically triggered by user actions. The pop method removes the current route from the stack, enabling backward navigation.
Routing:
Routing in Flutter involves defining and organizing the paths or routes within the application. Routes represent different screens or pages, and their effective use is crucial for structuring the app's architecture.

MaterialPageRoute and CupertinoPageRoute:

MaterialPageRoute is employed in apps following Material Design principles, providing a standard Android-style transition between screens.
CupertinoPageRoute is used for iOS-style designs, ensuring a consistent and native user experience.
Named Routes:

Named routes offer a more organized and readable approach to navigation.
By assigning names to routes, such as '/details', developers can easily navigate to

specific screens using Navigator.pushNamed.

Gestures in Flutter:
Gestures enhance user interaction by allowing the app to respond to touch or mouse inputs. In Flutter, the GestureDetector widget is instrumental in recognizing and handling various gestures.

GestureDetector Widget:

The GestureDetector widget is used to detect a variety of gestures, including taps, drags, and long presses.
By wrapping UI components with GestureDetector, developers can specify callback functions to execute when specific gestures are detected.
Gesture Recognizers:

Flutter provides gesture recognizers for more complex gestures, such as panning, pinching, and swiping.
These recognizers, when combined with a GestureDetector, enable the app to respond to a broader range of user inputs.
InkWell and InkWell:

The InkWell and InkResponse widgets bring a material ripple effect to touchable UI components.
Integrating these widgets enhances the visual feedback during user interactions, contributing to a more polished and intuitive user experience.

Code:
profile_screen.dart

```
import 'package:flutter/material.dart';
import 'package:flutter_custom_clippers/flutter_custom_clippers.dart';
import 'package:flutter_svg/flutter_svg.dart';
import 'package:tinder/authenticationScreen/chat_screen.dart';
import 'package:tinder/authenticationScreen/explore_screen.dart';
import 'package:tinder/authenticationScreen/main_page.dart';
import 'package:tinder/data/account_json.dart';
import 'package:tinder/authenticationScreen/colors.dart';

class AccountPage extends StatefulWidget {
 @override
 _AccountPageState createState() => _AccountPageState();
}
```

```
class _AccountPageState extends State<AccountPage> {
 @override
 Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.white, // Set background color to white
    body: getBody(),
    bottomNavigationBar: getBottomNavigationBar(),
  );
 }

 Widget getBody() {
  var size = MediaQuery.of(context).size;
  return Column(
    children: [
     ClipPath(
       clipper: OvalBottomBorderClipper(),
       child: Container(
        width: size.width,
        height: size.height * 0.6,
        decoration: BoxDecoration(color: white, boxShadow: [
         BoxShadow(
          color: Colors.grey,
          spreadRadius: 10,
          blurRadius: 10,
         ),
        ]),
        child: Padding(
         padding: const EdgeInsets.only(left: 30, right: 30, bottom: 40),
         child: Column(
          mainAxisAlignment: MainAxisAlignment.end,
          children: [
           Row(
            children: [
             Expanded(
              child:
                  Container(), // This container takes the left space
             ),
             GestureDetector(
              onTap: () {
               // Add your logic for the person SVG tap
              },
              child: SvgPicture.asset(
               'images/person.svg',
               width: 16,
              ),
             ),
             SizedBox(
                 width:
                     20), // Add spacing between person and tinder icons
             Expanded(
              child:
                  Container(), // This container takes the remaining space
             ),
             GestureDetector(
              onTap: () {
               // Add your logic for the tinder Image tap
              },
```

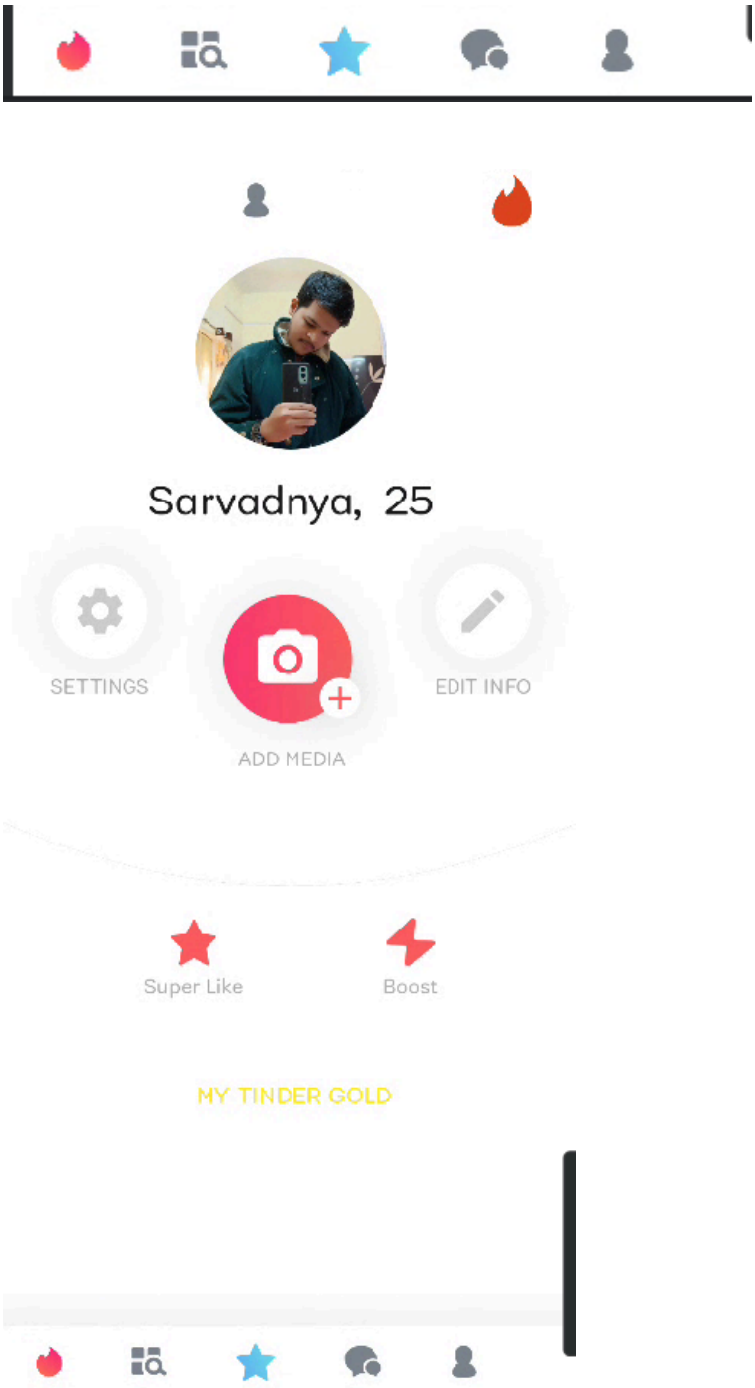
```
      child: Image.asset(
        'images/tinder_logo2.png', // Replace with the actual path to your tinder Image file
        width: 24,
      ),
    ),
  ],
),
SizedBox(height: 20),
Container(
  width: 120,
  height: 120,
  decoration: BoxDecoration(
      shape: BoxShape.circle,
      image: DecorationImage(
          image: AssetImage(account_json[0]['img']),
          fit: BoxFit.cover)),
),
SizedBox(
  height: 15,
),
Text(
  account_json[0]['name'] + ", " + account_json[0]['age'],
  style: TextStyle(fontSize: 25, fontWeight: FontWeight.w600),
),
SizedBox(
  height: 20,
),
Row(
  crossAxisAlignment: CrossAxisAlignment.start,
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    Column(
      children: [
        Container(
          width: 60,
          height: 60,
          decoration: BoxDecoration(
              shape: BoxShape.circle,
              color: white,
              boxShadow: [
                BoxShadow(
                  color: grey.withOpacity(0.1),
                  spreadRadius: 10,
                  blurRadius: 15,
                ),
              ]),
          child: Icon(
            Icons.settings,
            size: 35,
            color: grey.withOpacity(0.5),
          ),
        ),
        SizedBox(
          height: 10,
        ),
        Text(
          "SETTINGS",
```

```
                    style: TextStyle(
                        fontSize: 12,
                        fontWeight: FontWeight.w600,
                        color: grey.withOpacity(0.8)),
                  )
                ],
              ),

Widget getBottomNavigationBar() {
 return Container(
   padding: EdgeInsets.symmetric(horizontal: 20, vertical: 10),
   decoration: BoxDecoration(
    color: white,
    boxShadow: [
     BoxShadow(
      color: grey.withOpacity(0.1),
      spreadRadius: 10,
      blurRadius: 10,
     ),
    ],
   ),
   child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
     GestureDetector(
      onTap: () {
       Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => MainPage()),
       );
      },
      child: Image.asset(
       'images/logo.png',
       height: 20,
      ),
     ),
     GestureDetector(
      onTap: () {
       Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => ExploreScreen()),
       );
      },
      child: SvgPicture.asset(
       'images/search.svg',
       width: 22,
      ),
     ),
     GestureDetector(
      onTap: () {
       Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => ChatPage()),
       );
      },
      child: SvgPicture.asset(
       'images/star.svg',
```

```
            width: 24,
          ),
        ),
        GestureDetector(
          onTap: () {
            Navigator.push(
              context,
              MaterialPageRoute(builder: (context) => ChatPage()),
            );
          },
          child: SvgPicture.asset(
            'images/chat.svg',
            width: 24,
          ),
        ),
        Row(
          mainAxisAlignment: MainAxisAlignment.end,
          children: [
            GestureDetector(
              onTap: () {
                // Add your logic for the person SVG tap
              },
              child: SvgPicture.asset(
                'images/person.svg',
                width: 16,
              ),
            ),
            SizedBox(
                width: 20), // Add spacing between person and tinder icons
            GestureDetector(
              onTap: () {
                // Add your logic for the tinder SVG tap
              },
              child: SvgPicture.asset(
                'path/to/tinder.svg', // Replace with the actual path to your tinder SVG file
                width: 24,
              ),
            ),
          ],
        ),
      ],
    ),
  );
 }
}
```

Output:



Conclusion:

In this experiment, we have successfully created routing in the bottom navigation bar and connected all pages successfully using Navigator class and implemented it successfully.