Experiment 5

Shashwat Tripathi
D15A   Batch C
Roll No: 64

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

THEORY:
Understanding the Flow:

Flutter apps often involve navigating between different screens or views.
Navigation and routing provide the mechanism to manage this flow of users through your app.
Gestures allow users to interact with the UI and trigger navigation actions.
Navigation Concepts:

Navigator: A widget responsible for managing the navigation stack and transitions.
Routing: The process of defining routes and their corresponding destinations.
Routes: Screens or views that users can navigate to within your app.
Navigation Stack: A stack of routes, where the top element represents the currently active screen.
Transitions: Animations used to create visual continuity between screens during navigation.
Routing Strategies:

Declarative Routing: Routes are defined explicitly using the Navigator widget and its associated classes.
Package-based Routing: Third-party packages like go_router or routemaster offer additional features and flexibility.
Dynamic Routing: Routes can be generated based on runtime conditions or user input.
Gesture Recognition:

Flutter uses a gesture recognizer system to detect various user interactions like taps, swipes, and drags.
Common gesture recognizers include TapGestureRecognizer, SwipeGestureRecognizer, and PanGestureRecognizer.
You can define custom gesture recognizers for specific needs.
Interactions and Navigation:

Gestures can be used to trigger navigation actions.
For example, a tap on a button might use Navigator.push to push a new route onto the stack.
Swiping left or right might use Navigator.pop to go back to the previous screen.
Gestures can also be used to manipulate UI elements within a screen.
Best Practices:

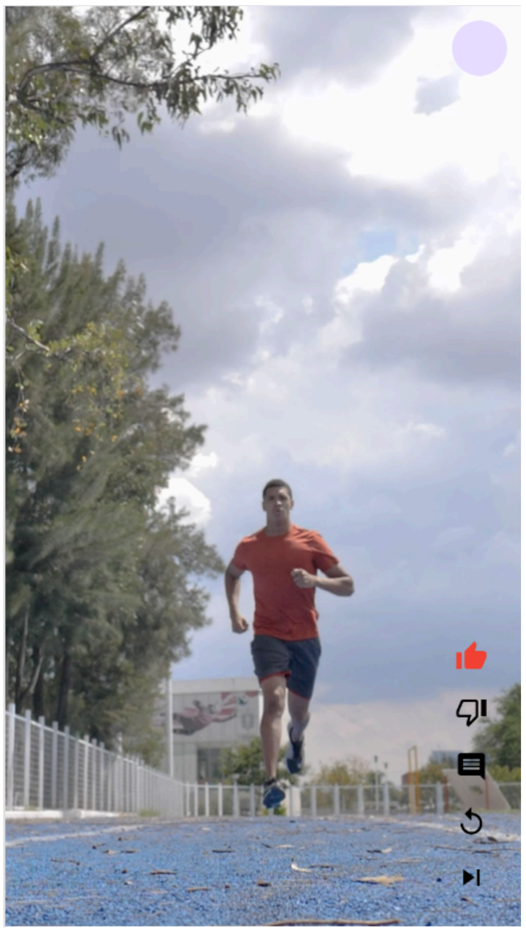
Use clear and consistent navigation patterns throughout your app.

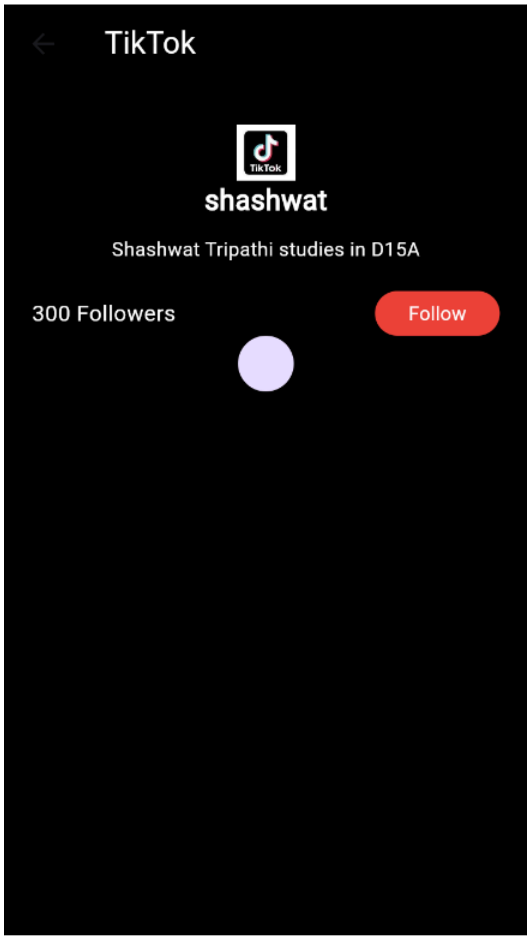
Provide visual cues to indicate interactive elements.
Design gestures that feel natural and intuitive for users.
Consider accessibility guidelines when designing gestures.
Test your navigation flow on different devices and screen sizes.
Additional Considerations:

Explore advanced navigation features like nested navigators and deep linking.
Use animation and transitions to enhance the user experience.
Consider integrating gesture recognizers with other UI elements like images and text.

Code:
home_page.dart
```
import 'package:flutter/material.dart';
import
'package:flutter_tiktok_shashwat/features/user_auth/presentation/page
s/profile.dart';
import
'package:flutter_tiktok_shashwat/features/user_auth/presentation/widg
ets/bottom_navbar.dart';
import 'package:video_player/video_player.dart';

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

  String get profileImageUrl => "null";

  // Navigator.pushNamed(context, '/homepage', arguments: {'user':
user, 'currentIndex': 0});

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  late VideoPlayerController controller;
  bool isLiked = false;
  bool isDisliked = false;
  bool isCommentOpen = false;
  List<String> videoUrls = [
    'assets/running.mp4',
    'assets/flowers.mp4',
    'assets/lights.mp4',
    'assets/rowing.mp4',
    'assets/traffic.mp4',
  ];
  int currentVideoIndex = 0;

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

  loadVideoPlayer() async {
    controller =
VideoPlayerController.asset(videoUrls[currentVideoIndex]);
    await controller.initialize();
    controller.addListener(() {
      setState(() {});
    });
    controller.play(); // Start playing the video automatically
    setState(() {});
  }

  @override
  void dispose() {
    controller.removeListener(() {});
    controller.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    const String profileImageUrl = "";
    return Scaffold(
      backgroundColor: Colors.black,
      body: Stack(
        children: [
          // Video Player
          Expanded(
            child: controller.value.isInitialized
                ? AspectRatio(
                    aspectRatio: controller.value.aspectRatio,
                    child: VideoPlayer(controller),
                  )
                : Center(child: CircularProgressIndicator()),
          ),
          Positioned(
            top: 10.0,
            right: 10.0,
            child: GestureDetector(
              onTap: () {
                Navigator.of(context).push(
                  MaterialPageRoute(builder: (context) =>
ProfilePage()),
                );
              },
              child: CircleAvatar(
                backgroundImage:
NetworkImage(widget.profileImageUrl),
                radius: 20.0,
              ),
            ),
          ),

          // Controls
          Positioned(
            bottom: 0,
            right: 0,
            child: Padding(
              padding: const EdgeInsets.all(16.0),
              child: Column(
                mainAxisAlignment: MainAxisAlignment.end,
                children: [
                  IconButton(
                    icon: Icon(
                      isLiked ? Icons.thumb_up :
Icons.thumb_up_outlined,
                      color: isLiked ? Colors.red : Colors.black,
                    ),
                    onPressed: () {
                      setState(() {
                        isLiked = !isLiked;
                        isDisliked = false;
                      });
                    },
                  ),
                  IconButton(
                    icon: Icon(
                      isDisliked ? Icons.thumb_down :
Icons.thumb_down_outlined,
                      color: isDisliked ? Colors.blue : Colors.black,
                    ),
                    onPressed: () {
                      setState(() {
                        isDisliked = !isDisliked;
                        isLiked = false;
                      });
                    },
                  ),
                  IconButton(
                    icon: Icon(Icons.comment, color: Colors.black),
                    onPressed: () {
                      // ... (Show comments as before)
                    },
                  ),
                  IconButton(
                    icon: Icon(Icons.replay, color: Colors.black),
                    onPressed: () {
                      controller.seekTo(Duration.zero);
                      setState(() {});
                    },
                  ),
                  IconButton(
                    icon: Icon(Icons.skip_next, color: Colors.black),
                    onPressed: () {
                      setState(() {
                        currentVideoIndex =
                            (currentVideoIndex + 1) %
videoUrls.length;
                        loadVideoPlayer();
                      });
                    },
                  ),
                ],
              ),
            ),
          ),
        ],
      ),
    );
  }
}
```

OUTPUT :





shashwat

Shashwat Tripathi studies in D15A

300 Followers

CONCLUSION: Thus, we have used navigation, routing and gestures in Flutter App.