

MAD and PWA Lab Mini Project

Name : Shubham Nakashe

Class:D15A

Roll no: 38

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

Theory:

In Flutter, navigation gestures and routing are used to navigate between different screens or pages .

- **Navigation:** Flutter provides a `Navigator` widget that manages a stack of routes (screens/pages) in your app. You can push a new route onto the stack to navigate to a new screen and pop a route to go back to the previous screen.
- **Gestures:** Flutter supports various gestures for navigation, such as tapping, swiping, and dragging. These gestures can be used to trigger navigation actions, like going to the next or previous screen.
- **Routing:** Flutter uses a named routing system, where each route/screen has a unique name. You can define routes in your app's main `MaterialApp` widget using the `routes` property, or you can use the `Navigator` directly to navigate to a specific route.
- **Navigation Stack:** The `Navigator` manages a stack of routes, which represents the navigation history of your app. When you push a new route, it gets added to the top of the stack, and when you pop a route, it gets removed from the stack.

CODE:

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:twitch_clone/screens/browse.dart';
import 'package:twitch_clone/screens/feed_screen.dart';
import 'package:twitch_clone/screens/go_live_screen.dart';
import 'package:twitch_clone/screens/login_screen.dart';
import 'package:twitch_clone/screens/onboarding_screen.dart';
import 'package:twitch_clone/screens/profile.dart';
import 'package:twitch_clone/utils/colors.dart';
import 'package:twitch_clone/widgets/custom_button.dart';

class HomeScreen extends StatefulWidget {
  static String routeName = '/home';
```

```

const HomeScreen({super.key});

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

class _HomeScreenState extends State<HomeScreen> {

  int _page = 0;
  List<Widget> pages = [];

  @override
  void initState() {
    super.initState();
    pages = [
      browse(),
      const GoLiveScreen(),
      const FeedScreen(),
      ProfileScreen(),

    ];
  }

  onPageChanged(int page) {
    setState(() {
      _page = page;
    });
  }

  @override
  Widget build(BuildContext context) {
    // final userProvider = Provider.of<UserProvider>(context);
    return Scaffold(
      bottomNavigationBar: BottomNavigationBar(
        selectedItemColor: buttonColor,
        unselectedItemColor: primaryColor,
        backgroundColor: backgroundColor,
        onTap: onPageChanged,
        currentIndex: _page,
        unselectedFontSize: 12,
        items: const [
          BottomNavigationBarItem(
            icon: Icon(Icons.favorite),
            label: 'Browse',
          ),
          BottomNavigationBarItem(

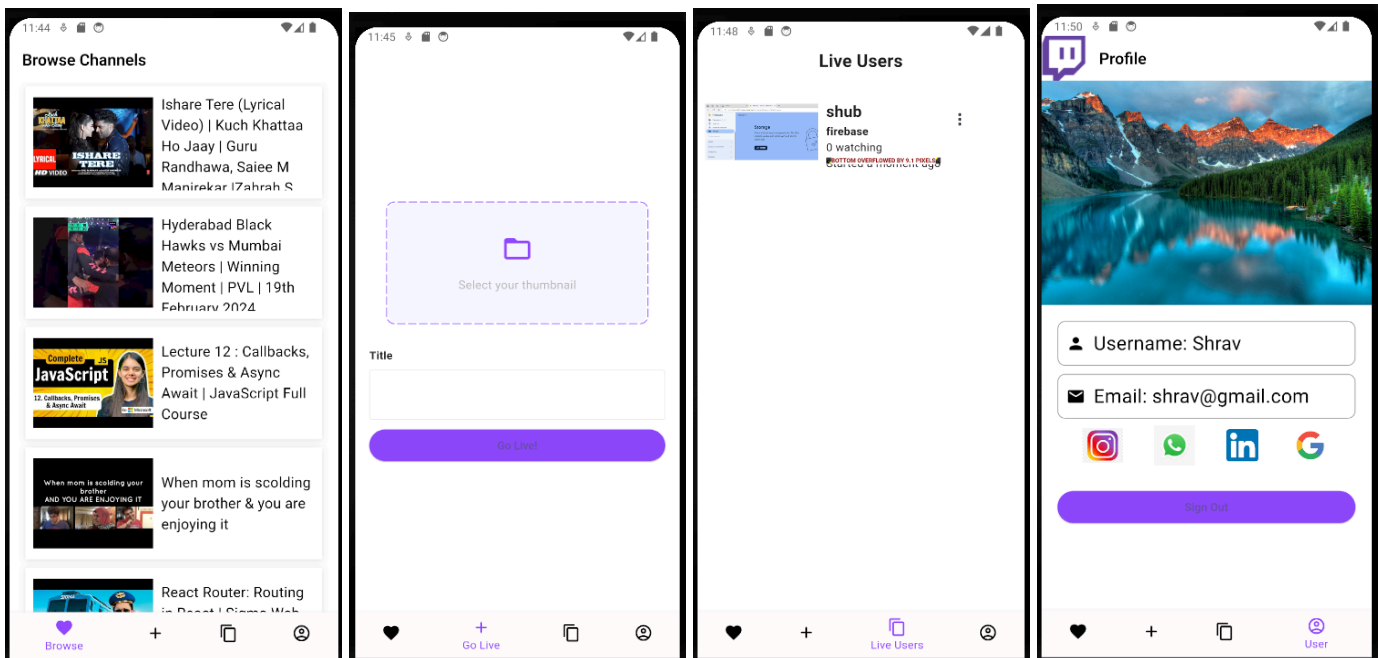
```

```

        icon: Icon(Icons.add_rounded),
        label: 'Go Live',
      ),
      BottomNavigationBarItem(
        icon: Icon(Icons.copy),
        label: 'Live Users',
      ),
      BottomNavigationBarItem(
        icon: Icon(Icons.account_circle_outlined),
        label: 'User',
      ),
    ],
  ),
  body: pages[_page],
);
}
}

```

OUTPUT:



Navigation from Browse channels → Live Stream → Current live stream → Profile Page