Atharva Mulam Batch B
D15A Roll No: 37

Exp No: 5

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

Theory:
In Flutter, navigation, routing, and gestures play crucial roles in creating engaging and intuitive user interfaces. Here's a brief overview of each concept:

Navigation:
Navigation in Flutter refers to the process of moving between different screens or pages within an app. Flutter provides a Navigator class that manages a stack of Route objects, allowing you to push and pop routes onto and off of the navigation stack.
There are several ways to navigate between screens in Flutter:
Pushing a new route: You can push a new route onto the navigation stack using the Navigator.push() method.
Popping a route: You can pop the current route off the stack using the Navigator.pop() method.
Named routes: You can define named routes for your app's screens and use them to navigate using the Navigator.pushNamed() method.
Modal routes: You can show a modal route that covers the screen using the showDialog() or showModalBottomSheet() methods.

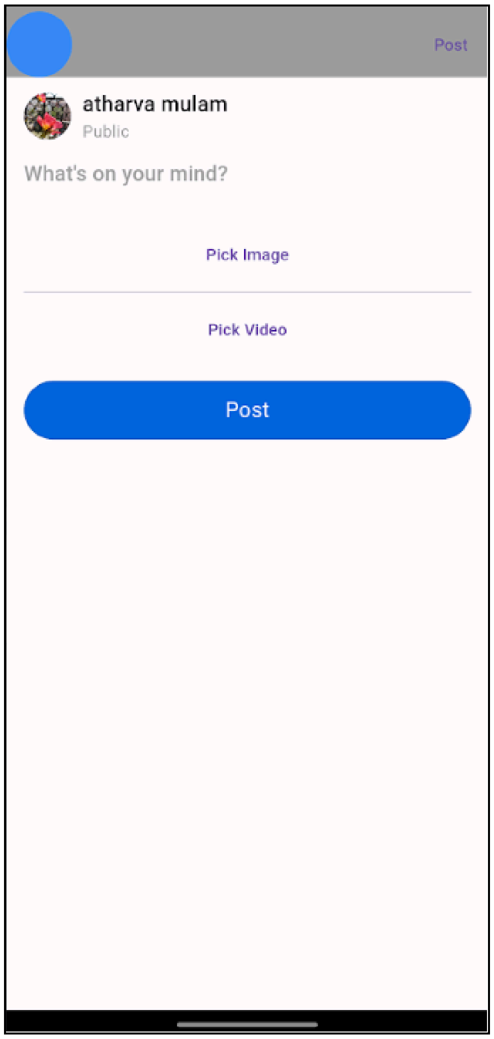
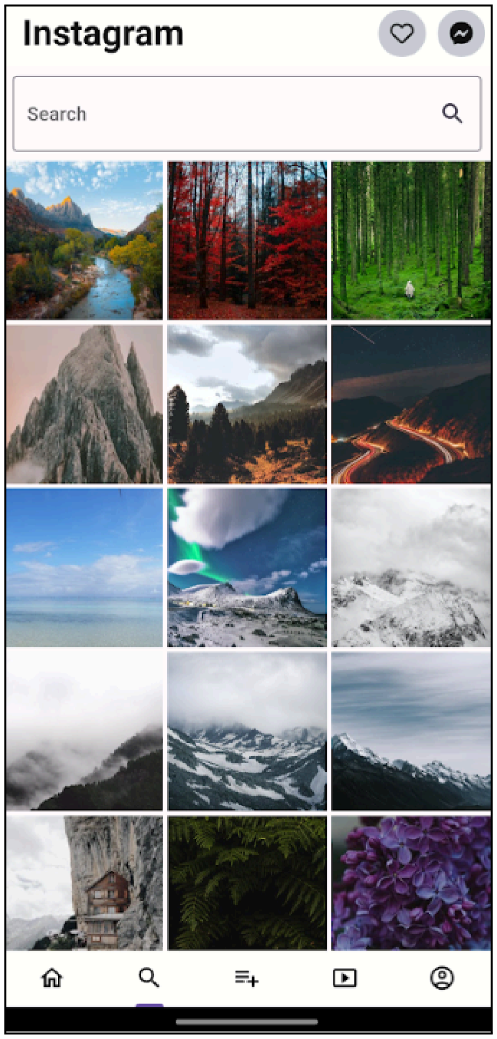
Routing:
Routing in Flutter refers to the process of defining how the app's screens or pages are structured and organized. Flutter uses a hierarchical routing system, where each route corresponds to a widget subtree that can be pushed and popped onto and off of the navigation stack. Here are some key concepts related to routing in Flutter:
MaterialPageRoute: This is the most commonly used route in Flutter apps, which represents a full-screen page that transitions in and out using a material design-style animation.
PageRouteBuilder: This class allows you to create custom page transition animations by specifying a builder function that returns the widget subtree for the route.
Nested navigation: You can nest Navigator widgets within your app's widget tree to create nested navigation hierarchies, allowing for more complex navigation flows.
Gestures:
Gestures in Flutter refer to user interactions such as tapping, dragging, swiping, pinching, etc. Flutter provides a rich set of gesture recognizer classes that make it easy

to handle these interactions. Here are some commonly used gesture recognizers in Flutter:
GestureDetector: This widget allows you to detect various gestures such as taps, drags, and long-presses on its child widget and respond to them with custom callback functions.
InkWell: This widget provides a material design-style ink splash effect in response to taps, and it's commonly used for creating clickable elements in Flutter apps.
Draggable: This widget allows you to make its child widget draggable, enabling users to drag it around the screen using touch gestures.

Code:
```
import 'package:flutter/material.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:ig/core/constants/app_colors.dart';
import 'package:ig/core/constants/constants.dart';
import 'package:ig/core/widgets/round_icon_button.dart';

class HomeScreen extends StatefulWidget {
  const HomeScreen({Key? key}) : super(key: key);

  static const routeName = '/home';

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

class _HomeScreenState extends State<HomeScreen> with TickerProviderStateMixin {
  late final TabController _tabController;

  @override
  void initState() {
    _tabController = TabController(length: 5, vsync: this);
    super.initState();
  }

  @override
  void dispose() {
    _tabController.dispose();
    super.dispose();
  }
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: AppColors.greyColor,
    appBar: AppBar(
      backgroundColor: AppColors.whiteColor,
      elevation: 0,
      title: _buildFacebookText(),
      actions: [
        _buildSearchWidget(),
        _buildMessengerWidget(),
      ],
    ),
    body: TabBarView(
      controller: _tabController,
      children: Constants.screens,
    ),
    bottomNavigationBar: Material(
      color: AppColors.whiteColor,
      child: TabBar(
        tabs: Constants.getHomeScreenTabs(_tabController.index),
        controller: _tabController,
        onTap: (index) {
          setState(() {});
        },
      ),
    ),
  );
}

Widget _buildFacebookText() => const Text(
      'Instagram',
      style: TextStyle(
        color: AppColors.blackColor,
        fontSize: 30,
        fontWeight: FontWeight.bold,
      ),
    );

Widget _buildSearchWidget() => const RoundIconButton(
      icon: FontAwesomeIcons.heart,
    );

  Widget _buildMessengerWidget() => InkWell(
      onTap: () {},
      child: const RoundIconButton(
        icon: FontAwesomeIcons.facebookMessenger,
      ),
    );
}
```
Routing:
```
import 'package:ig/core/screens/error_screen.dart';
import 'package:flutter/cupertino.dart';
import 'package:ig/core/screens/home_screen.dart';
import 'package:ig/core/screens/profile_screen.dart';
import 'package:ig/features/auth/presentation/screens/create_account_screen.dart';
import 'package:ig/features/posts/presentation/screens/comments_screen.dart';
import 'package:ig/features/posts/presentation/screens/create_post_screen.dart';

class Routes {
  static Route onGenerateRoute(RouteSettings settings) {
    switch (settings.name) {
      case CreateAccountScreen.routeName:
      return _cupertinoRoute(const CreateAccountScreen(),);
      case HomeScreen.routeName:
      return _cupertinoRoute(const HomeScreen(),);
      case CreatePostScreen.routeName:
      return _cupertinoRoute(const CreatePostScreen(),);
      case CommentsScreen.routeName:
        final postId = settings.arguments as String;
        return _cupertinoRoute(
          CommentsScreen(postId: postId),
        );
       case ProfileScreen.routeName:
        final userId = settings.arguments as String;
        return _cupertinoRoute(
          ProfileScreen(
            userId: userId,
          ),
        );
      default:
        return _cupertinoRoute(
          ErrorScreen(
            error: 'Wrong Route provided ${settings.name}',
          ),
        );
    }
  }

  static Route _cupertinoRoute(Widget view) => CupertinoPageRoute(
        builder: (_) => view,
      );

  Routes._();
}
```

Output:


Post
atharva mulam
Public
What's on your mind?
Pick Image
Pick Video
Post

Instagram
Search