Shilpa Pandey
42 D15A

Experiment 5

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

Theory:

Navigate between screens: Use Navigator methods like push() and pop() to navigate forward and backward through the route stack.

Map route names to their corresponding widgets/screens in the MaterialApp widget using the routes property or use Navigator to push/pop routes dynamically.

Gestures:
Gesture detection: Wrap widgets with gesture detectors such as GestureDetector to detect user gestures.
Gesture handling: Implement callback functions like onTap, onPanUpdate, etc., to respond to specific gestures.

Step 1: First, you need to create two routes.

Step 2: Then, navigate to one route from another route by using the Navigator.push() method.

Step 3: Finally, navigate to the first route by using the Navigator.pop() method.

Code:

Screen layout

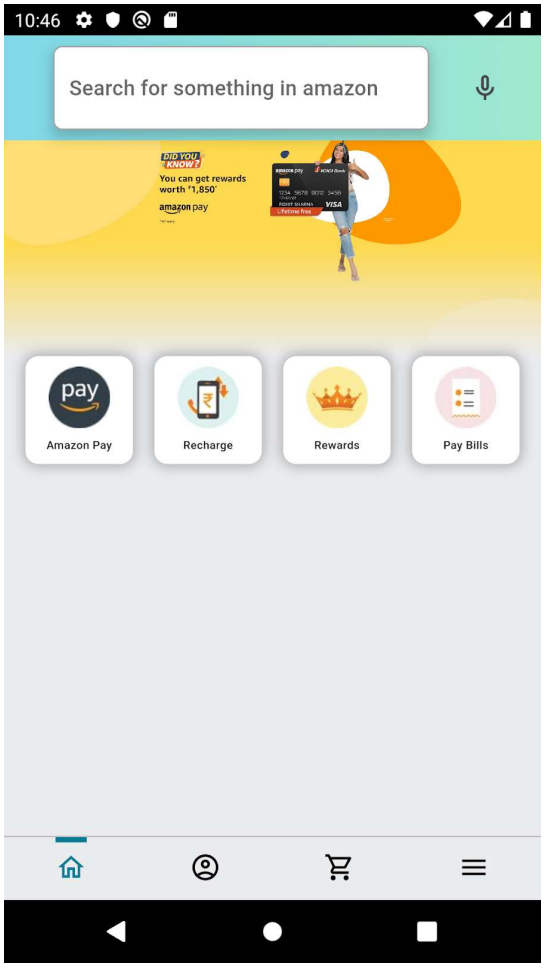
```
 import 'package:amazon_clone/utils/color_themes.dart';
import 'package:amazon_clone/utils/constants.dart';
import 'package:flutter/material.dart';

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

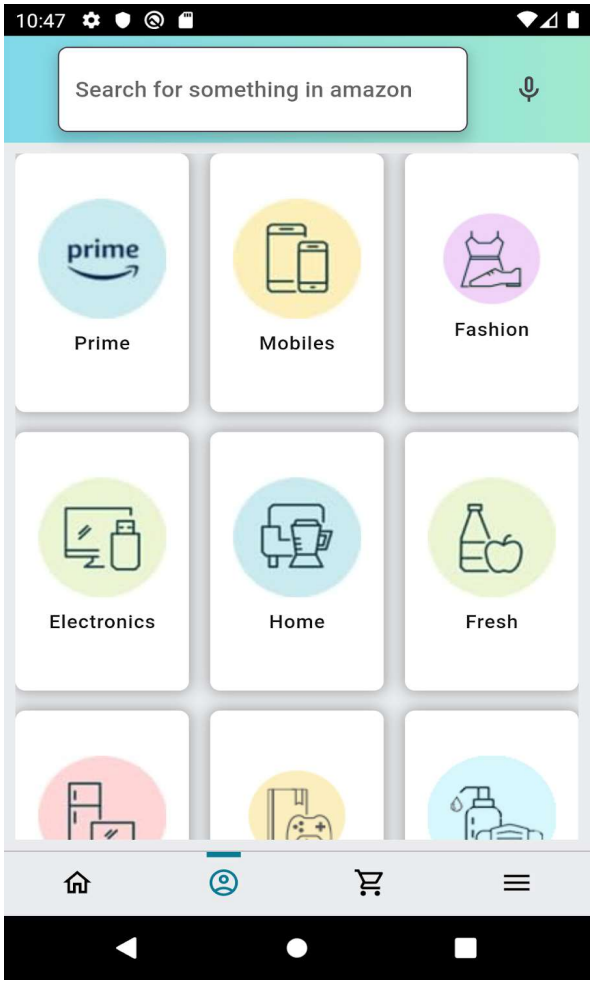
```
  @override
  State<ScreenLayout> createState() => _ScreenLayoutState();
}


class _ScreenLayoutState extends State<ScreenLayout> {
  PageController pageController = PageController();
  int currentPage = 0;


  @override
  void dispose() {
    super.dispose();
    pageController.dispose();
  }


  changePage(int page) {
    pageController.jumpToPage(page);
    setState(() {
      currentPage = page;
    });
  }


  @override
  void initState() {
    super.initState();
    //CloudFirestoreClass().getNameAndAddress();
  }


  @override
  Widget build(BuildContext context) {
   // Provider.of<UserDetailsProvider>(context).getData();
    return DefaultTabController(
      length: 4,
      child: SafeArea(
        child: Scaffold(
```

```
body: PageView(
  physics: NeverScrollableScrollPhysics(),
  controller: pageController,
  children: screens,
),
bottomNavigationBar: Container(
  decoration: BoxDecoration(
    border: Border(
      top: BorderSide(color: Colors.grey[400]!, width: 1),
    ),
  ),
  child: TabBar(
    indicator: const BoxDecoration(
      border: Border(
        top: BorderSide(color: activeCyanColor, width: 4),
      ),
    ),
    onTap: changePage,
    indicatorSize: TabBarIndicatorSize.label,
    tabs: [
      Tab(
        child: Icon(
          Icons.home_outlined,
          color: currentPage == 0 ? activeCyanColor : Colors.black,
        ),
      ),
      Tab(
        child: Icon(
          Icons.account_circle_outlined,
          color: currentPage == 1 ? activeCyanColor : Colors.black,
        ),
      ),
      Tab(
        child: Icon(
          Icons.shopping_cart_outlined,
          color: currentPage == 2 ? activeCyanColor : Colors.black,
        ),
      ),
      Tab(
        child: Icon(
```

```
                Icons.menu,
                color: currentPage == 3 ? activeCyanColor : Colors.black,
              ),
            ),
          ],
        ),
      ),
    ),
  ),
 );
 }
}
```

Output:





Search

```
import 'package:amazon_clone/widgets/search_bar_widget.dart';
import 'package:flutter/material.dart';


class SearchScreen extends StatelessWidget {
  const SearchScreen({Key? key}) : super(key: key);


  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: SearchBarWidget(isReadOnly: false, hasBackButton: true),
```

```
    );
  }
}
```

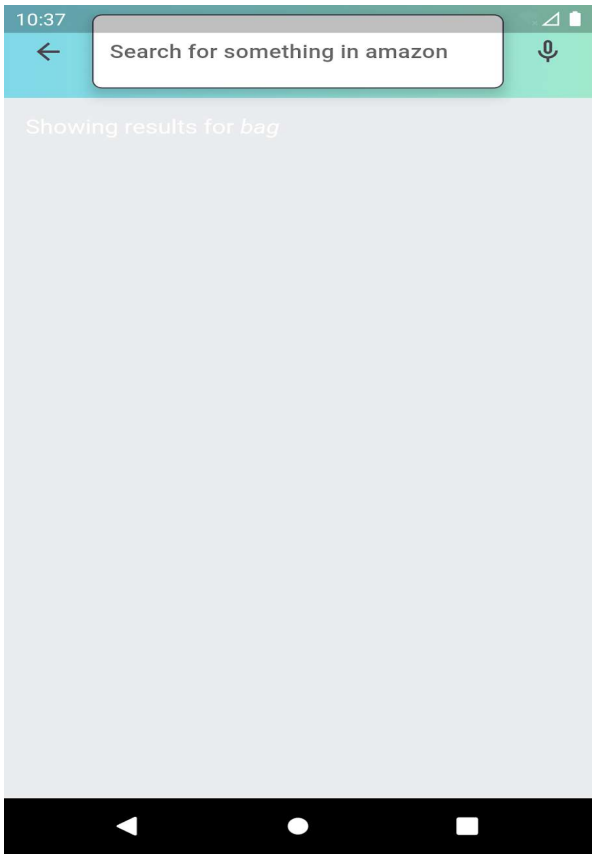
Screen widget

```
import 'package:amazon_clone/screens/results_screen.dart';
import 'package:amazon_clone/screens/search_screen.dart';
import 'package:amazon_clone/utils/color_themes.dart';
import 'package:amazon_clone/utils/constants.dart';
import 'package:amazon_clone/utils/utils.dart';
import 'package:flutter/material.dart';


class SearchBarWidget extends StatefulWidget implements  PreferredSizeWidget {
  final bool isReadOnly;
  final bool hasBackButton;
  SearchBarWidget({
    Key? key,
    required this.isReadOnly,
    required this.hasBackButton,
  })  : preferredSize = const Size.fromHeight(kAppBarHeight),
        super(key: key);


  @override
  final Size preferredSize;


  @override
  State<SearchBarWidget> createState() => _SearchBarWidgetState();
}


class _SearchBarWidgetState extends State<SearchBarWidget> {
```

```
OutlineInputBorder border = OutlineInputBorder(
  borderRadius: BorderRadius.circular(7),
  borderSide: const BorderSide(
    color: Colors.grey,
    width: 1,
  ),
);


@override
Widget build(BuildContext context) {
  Size screenSize = Utils().getScreenSize();
  return Container(
    height: kAppBarHeight,
    decoration: const BoxDecoration(
      gradient: LinearGradient(
        colors: backgroundGradient,
        begin: Alignment.centerLeft,
        end: Alignment.centerRight,
      ),
    ),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      children: [
        widget.hasBackButton
            ? IconButton(
                onPressed: () {
                  Navigator.pop(context);
                },
                icon: const Icon(Icons.arrow_back))
            : Container(),
        SizedBox(
          width: screenSize.width * 0.7,
          child: Container(
            decoration: BoxDecoration(
              boxShadow: [
                BoxShadow(
```

```
          color: Colors.black.withOpacity(0.2),
          blurRadius: 8,
          spreadRadius: 1,
          offset: const Offset(0, 5),
        ),
      ],
    ),
    child: TextField(
      onSubmitted: (String query) {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => ResultsScreen(query: query),
          ),
        );
      },
      readOnly: widget.isReadOnly,
      onTap: () {
        if (widget.isReadOnly) {
          Navigator.push(
              context,
              MaterialPageRoute(
                  builder: (context) => const SearchScreen()));
        }
      },
      decoration: InputDecoration(
        hintText: "Search for something in amazon",
        fillColor: Colors.white,
        filled: true,
        border: border,
        focusedBorder: border,
      ),
    ),
  ),
),
IconButton(
  onPressed: () {},
```

```
            icon: const Icon(Icons.mic_none_outlined),
          )
        ],
      ),
    );
  }
}
```



Conclusion:

Understand the theoretical concept of navigation, routing and gestures and also how to use them in flutter App.