MAD Lab Exp 5

Name: Abhishek Fatate Roll No. 15

Div: D15A Batch: A

Aim:

The aim of this experiment is to create an interactive form using Flutter's form widget. The form will include various form fields such as text fields, dropdown menus, and checkboxes, along

with validation logic. Additionally, a button will be added to validate and submit the form data.

Theory:

Gesture:

In Flutter, gestures are user interactions with the application, such as taps, drags, and scrolls. The GestureDetector widget is used to detect various gestures and provides callbacks to handle these interactions. Here's a brief overview:

GestureDetector Widget: The GestureDetector widget wraps its child and detects various gestures applied to it. It provides properties like onTap, onDoubleTap, onLongPress, onPanUpdate, etc., to handle different types of gestures.

Gesture Detection Process: When a user interacts with the screen, the GestureDetector widget detects the gesture based on the user's input and invokes the appropriate callback function,

allowing developers to respond to the gesture accordingly. Navigation:

Navigation in Flutter refers to moving between different screens or routes within the application. Flutter's navigation system is built around the Navigator class, which manages a stack of routes. Here's how it works:

Routes: In Flutter, each screen or page is called a route. Routes are pushed onto and popped off the Navigator's stack to navigate between screens.

Navigator.push(): To navigate from one route to another, you use the Navigator.push() method. This method adds a new route to the stack, displaying the new screen on top of the current one.

Navigator.pop(): To return to the previous route, you use the Navigator.pop() method. This removes the top route from the stack, returning to the previous screen.

Steps to Implement Navigation and Gestures:

Create Two Routes: Define two separate screens or widgets to represent the two routes in your application.

Navigate to Second Route: Use Navigator.push() to navigate from the first route to the second route when a specific gesture or action is detected.

Return to First Route: Implement a mechanism, such as a button press or gesture detection, to trigger Navigator.pop() and return from the second route to the first route.

By combining gestures and navigation, you can create dynamic and interactive Flutter

applications that provide a seamless user experience for navigating between different screens and responding to user input through gestures.

# Code in main.dart:

import 'package:abhishek\_application/firebase\_options.dart';

import 'package:firebase\_core/firebase\_core.dart';

import 'package:flutter/material.dart';

import 'LoginScreen.dart';

import 'splash\_screen.dart';

void main() async {

WidgetsFlutterBinding.ensureInitialized();

await Firebase.initializeApp(

options: DefaultFirebaseOptions.currentPlatform,

);

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Oyo',

theme: ThemeData(

scaffoldBackgroundColor: Colors.black,

appBarTheme: AppBarTheme(

backgroundColor: Colors.black,

titleTextStyle: const TextStyle(

color: Colors.red,

fontSize: 34.0,

fontWeight: FontWeight.bold,

),

),

),

home: const SplashScreen(), // Display SplashScreen initially

);

}

}

# Code in Home\_screen.dart

import 'package:flutter/material.dart';

import 'HotelDetailsPage.dart';

import 'SearchPage.dart'; // Import the SearchPage.dart file

import 'BookingsPage.dart'; // Import the BookingsPage.dart

import 'NeedHelpPage.dart'; // Import the NeedHelpPage.dart

class HomePage extends StatelessWidget {

@override

Widget build(BuildContext context) {

return Scaffold(

body: CustomScrollView(

slivers: <Widget>[

SliverAppBar(

title: Text('Home'),

centerTitle: true,

floating: true,

snap: true,

elevation: 0,

flexibleSpace: LayoutBuilder(

builder: (BuildContext context, BoxConstraints constraints) {

return constraints.biggest.height > 150

? Image.asset(

'assets/Oyo\_logo.jpeg',

fit: BoxFit.cover,

)

: SizedBox.shrink();

},

),

),

SliverToBoxAdapter(

child: SingleChildScrollView(

scrollDirection: Axis.horizontal,

child: Row(

children: [

CityButton(

cityName: 'Mumbai',

onPressed: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => CityPage(cityName: 'Mumbai'),

),

);

},

),

CityButton(

cityName: 'Pune',

onPressed: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => CityPage(cityName: 'Pune'),

),

);

},

),

CityButton(

cityName: 'Bangalore',

onPressed: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) =>

CityPage(cityName: 'Bangalore'),

),

);

},

),

CityButton(

cityName: 'Delhi',

onPressed: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => CityPage(cityName: 'Delhi'),

),

);

},

),

CityButton(

cityName: 'Chennai',

onPressed: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => CityPage(cityName: 'Chennai'),

),

);

},

),

CityButton(

cityName: 'Hyderabad',

onPressed: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) =>

CityPage(cityName: 'Hyderabad'),

),

);

},

),

],

),

),

),

SliverList(

delegate: SliverChildBuilderDelegate(

(BuildContext context, int index) {

return HotelInfo(

hotelName: 'Hotel ${index + 1}',

distance: '${(index + 1) \* 2} km from current location',

hasRestaurant: index.isEven,

imagePath: 'assets/hotel${index % 3 + 1}.jpg',

hotelIndex: index,

);

},

childCount: 9, // Add more if needed

),

),

],

),

bottomNavigationBar: BottomNavigationBar(

selectedItemColor: Colors.black,

unselectedItemColor: Colors.black,

items: [

BottomNavigationBarItem(

icon: Icon(Icons.home),

label: 'Home',

),

BottomNavigationBarItem(

icon: Icon(Icons.search),

label: 'Search',

),

BottomNavigationBarItem(

icon: Icon(Icons.bookmark),

label: 'Bookings',

),

BottomNavigationBarItem(

icon: Icon(Icons.help),

label: 'Need Help',

),

],

onTap: (int index) {

if (index == 1) {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => SearchPage(),

),

);

} else if (index == 2) {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => BookingsPage()

),

);

} else if (index == 3) {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => NeedHelpPage(),

),

);

}

},

),

);

}

}

class CityButton extends StatelessWidget {

final String cityName;

final VoidCallback onPressed;

const CityButton({

required this.cityName,

required this.onPressed,

});

@override

Widget build(BuildContext context) {

return Padding(

padding: const EdgeInsets.all(8.0),

child: ElevatedButton(

onPressed: onPressed,

child: Text(cityName),

),

);

}

}

class HotelInfo extends StatelessWidget {

final String hotelName;

final String distance;

final bool hasRestaurant;

final String imagePath;

final int hotelIndex;

const HotelInfo({

required this.hotelName,

required this.distance,

required this.hasRestaurant,

required this.imagePath,

required this.hotelIndex,

});

@override

Widget build(BuildContext context) {

double screenWidth = MediaQuery.of(context).size.width;

return Padding(

padding: const EdgeInsets.all(8.0),

child: GestureDetector(

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => HotelDetailsPage(

hotelName: hotelName,

distance: distance,

hasRestaurant: hasRestaurant,

imagePath: imagePath,

hotelIndex: hotelIndex,

),

),

);

},

child: Column(

crossAxisAlignment: CrossAxisAlignment.center,

children: [

ClipRRect(

borderRadius: BorderRadius.circular(10),

child: Image.asset(

imagePath,

width: screenWidth,

height: screenWidth \* 0.75, // Adjust height as needed

fit: BoxFit.cover,

),

),

SizedBox(height: 10),

Text(

'Hotel ${hotelIndex + 1}', // Display hotel number

style: TextStyle(

fontSize: 16,

fontWeight: FontWeight.bold,

color: Colors.red,

),

),

SizedBox(height: 5),

Text(

distance,

style: TextStyle(

color: Colors.red, // Set the color to red

),

),

SizedBox(height: 5),

Text(

hasRestaurant ? 'Restaurant available' : 'No restaurant',

style: TextStyle(

color: hasRestaurant ? Colors.green : Colors.red,

fontWeight: FontWeight.bold,

),

),

],

),

),

);

}

}

// Create separate pages for each city

class CityPage extends StatelessWidget {

final String cityName;

CityPage({required this.cityName});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text(cityName),

centerTitle: true,

),

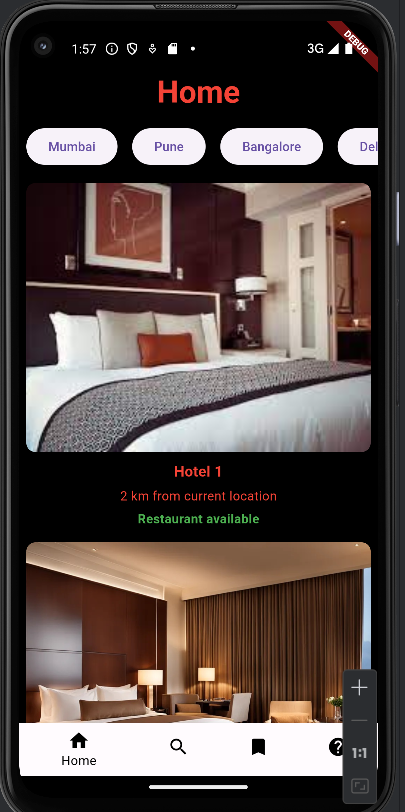
body: HomePage(),

);

}

}}

Home screen:



Switched to search screen:

