EXPERIMENT NO: - 05

Name:- Prajyot Shinde **Class:-** D15A **Roll:No: -** 55

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

Theory: -

In Flutter, the screens and pages are known as routes, and these routes are just a widget. In Android, a route is similar to an Activity.

In any mobile app, navigating to different pages defines the workflow of the application, and the way to handle the navigation is known as routing. Flutter provides a basic routing class MaterialPageRoute and two methods Navigator.push() and Navigator.pop() that shows how to navigate between two routes. The following steps are required to start navigation in your application.

Gestures enable the app to respond to user interactions, making the application more dynamic and responsive.

➢ Navigation and Routing in Flutter

Navigation is the process of moving between different screens or pages in an app. Flutter provides a simple and effective way to handle this through the use of the Navigator widget and routes.

1. Using Navigator Widget

The Navigator widget manages a stack of routes, allowing for pushing and popping routes on the stack.

- **Pushing a Route**: To navigate to a new screen, use Navigator.push().
- **Popping a Route**: To go back to the previous screen, use Navigator.pop().

```
ElevatedButton(
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(builder: (context) => SecondScreen()),
    );},
);
```

2. **Named Routes**

Flutter also allows the use of named routes to navigate, which can make the routing process cleaner, especially in larger applications.

```
MaterialApp(
  initialRoute: '/',
  routes: {
    '/': (context) => HomeScreen(),
    '/second': (context) => SecondScreen(),
  },
);
```

Navigate to the route using Navigator.pushNamed()

```
Navigator.pushNamed(context, '/second');
```

Handling Gestures in Flutter

Gestures refer to user interactions with the app, such as taps, swipes, pinches, and drags. Flutter provides several widgets and gesture detectors to handle these interactions.

Tap Gestures

The most common gesture is the tap, which can be handled using the GestureDetector widget or specific buttons like InkWell or ElevatedButton.

Long Press Gesture

For long press gestures, Flutter provides the onLongPress callback in GestureDetector or InkWell.

Swipe and Drag Gestures

Flutter also provides swipe and drag gesture handling. The onHorizontalDragUpdate and onVerticalDragUpdate callbacks are used for dragging gestures.

After clicking on Don’t have an account? it navigates to the registration page.

Code: -

Main.dart

```
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import '/screens/home_screen.dart';
import '/screens/auth/login_screen.dart';
import '/screens/bookings/bookings_screen.dart';
import '/screens/profile/profile_screen.dart';
import '/screens/service_details_screen.dart';
import '/screens/help/help_screen.dart';
import '/screens/auth/otp_verify.dart';
import '/screens/auth/signup_screen.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Urban Company',
      theme: ThemeData(
        primarySwatch: Colors.blue,
        scaffoldBackgroundColor: Colors.white,
        fontFamily: 'Poppins',
        appBarTheme: const AppBarTheme(
          backgroundColor: Colors.white,
          elevation: 0,
          iconTheme: IconThemeData(color: Colors.black),
          titleTextStyle: TextStyle(
            color: Colors.black,
            fontSize: 18,
            fontWeight: FontWeight.w600,
            fontFamily: 'Poppins',
          ),
        ),
      ),
      initialRoute: '/',
      routes: {
        '/': (context) => const LoginScreen(),
        '/signup': (context) => const SignupScreen(),
       '/otp': (context) {
  final args = ModalRoute.of(context)?.settings.arguments;

  if (args is Map<String, String>) {
    return OtpVerificationScreen(
      phoneNumber: args['phoneNumber'] ?? '',
      verificationId: args['verificationId'] ?? '',
    );
  } else {
    return const Scaffold(
      body: Center(child: Text('Invalid arguments for OTP screen')),
    );
  }
},
        '/home': (context) => const HomeScreen(),
        '/bookings': (context) => const BookingsScreen(),
        '/help': (context) => const HelpScreen(),
        '/profile': (context) => const ProfileScreen(),
        '/service-details': (context) => ServiceDetailsScreen(),
      },
    );
  }
}
```

Handyman_booking_screen.dart:

```
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';

class HandymanBookingScreen extends StatefulWidget {
  final String serviceTitle;
  final String price;

  const HandymanBookingScreen({
    Key? key,
    required this.serviceTitle,
    required this.price,
  }) : super(key: key);

  @override
  State<HandymanBookingScreen> createState() => _HandymanBookingScreenState();
}

class _HandymanBookingScreenState extends State<HandymanBookingScreen> {
  DateTime? _selectedDate;
  TimeOfDay? _selectedTime;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text(
          'Book Handyman',
        ),
        centerTitle: true,
        elevation: 0,
      ),
      body: SingleChildScrollView(
        padding: const EdgeInsets.symmetric(horizontal: 16.0, vertical: 24.0), // Adjusted padding
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            // Service Info Card
            _buildServiceInfoCard(),
            const SizedBox(height: 32),
            // Date & Time Picker
            _buildDateTimePicker(context),
            const SizedBox(height: 32),
            // Confirm Booking Button
            _buildConfirmButton(context),
          ],
        ),
      ),
    );
  }

  Widget _buildServiceInfoCard() {
    return Card(
      elevation: 8,
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(16),
      ),
      child: Container(
        padding: const EdgeInsets.all(24.0),
        decoration: BoxDecoration(
          gradient: const LinearGradient(
            colors: [Colors.deepPurple,
Colors.purpleAccent],
            begin: Alignment.topLeft,
            end: Alignment.bottomRight,
          ),
          borderRadius: BorderRadius.circular(16),
        ),
        child: Column(
          crossAxisAlignment:
CrossAxisAlignment.start,
          children: [
            Text(
              widget.serviceTitle,
              style: const TextStyle(
                fontSize: 28,
                fontWeight: FontWeight.bold,
                color: Colors.white,
              ),
            ),
            const SizedBox(height: 8),
            Text(
              widget.price,
              style: const TextStyle(
                fontSize: 22,
                color: Colors.white70,
                fontWeight: FontWeight.w600,
              ),
            ),
          ],
        ),
      ),
    );
  }

  Widget _buildDateTimePicker(BuildContext
context) {
    return Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        const Text(
          'Select Date & Time',
          style: TextStyle(
            fontSize: 22,
            fontWeight: FontWeight.bold,
            color: Colors.deepPurple,
          ),
        ),
        const SizedBox(height: 16),
        Row(
          children: [
            Expanded(
              child: _buildPickerField(
                label: _selectedDate == null
                    ? 'Select Date'
                    : '${_selectedDate!.day}/${_selectedDate!.month}/${_selectedDate!.year}',
                icon: Icons.calendar_today,
                onTap: () async {
                  final date = await showDatePicker(
                    context: context,
                    initialDate: DateTime.now(),
                    firstDate: DateTime.now(),
                    lastDate: DateTime.now().add(const Duration(days: 365)),
                  );
                  if (date != null) {
                    setState(() => _selectedDate = date);
                  }
                },
              ),
            ),
            const SizedBox(width: 16),
            Expanded(
              child: _buildPickerField(
                label: _selectedTime == null
                    ? 'Select Time'
                    : _selectedTime!.format(context),
                icon: Icons.access_time,
                onTap: () async {
                  final time = await showTimePicker(
                    context: context,
                    initialTime: TimeOfDay.now(),
                  );
                  if (time != null) {
                    setState(() => _selectedTime = time);
                  }
                },
              ),
            ),
          ],
        ),
      ],
    );
  }

  Widget _buildPickerField({
    required String label,
    required IconData icon,
    required VoidCallback onTap,
  }) {
    return GestureDetector(
      onTap: onTap,
      child: Container(
        padding: const EdgeInsets.symmetric(vertical: 18, horizontal: 16),
        decoration: BoxDecoration(
          color: Colors.white,
          borderRadius: BorderRadius.circular(12),
          boxShadow: [
            BoxShadow(
       color: Colors.black.withOpacity(0.1),
       blurRadius: 10,
       spreadRadius: 2,
       offset: const Offset(0, 5),
      ),
     ],
    ),
    child: Row(
     children: [
      Icon(icon, color: Colors.deepPurple),
      const SizedBox(width: 12),
      Expanded( // Added Expanded to prevent overflow
       child: Text(
        label,
        style: const TextStyle(
         fontSize: 16,
         fontWeight: FontWeight.w500,
        ),
       ),
      ),
     ],
    ),
   ),
  );
 }

 Widget _buildConfirmButton(BuildContext context) {
  return SizedBox(
   width: double.infinity,
   child: ElevatedButton(
    onPressed: () => _confirmBooking(context),
    style: ElevatedButton.styleFrom(
     backgroundColor: Colors.deepPurple,
     padding: const EdgeInsets.symmetric(vertical: 20),
     shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(12),
     ),
     elevation: 4,
    ),
    child: const Text(
     'Confirm Booking',
     style: TextStyle(
      fontSize: 20,
      fontWeight: FontWeight.bold,
      color: Colors.white,
     ),
    ),
   ),
  );
 }

 Future<void> _confirmBooking(BuildContext context) async {
  if (_selectedDate == null || _selectedTime == null) {
   ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(content: Text('Please select date and time.')),
   );
   return;
  }
```

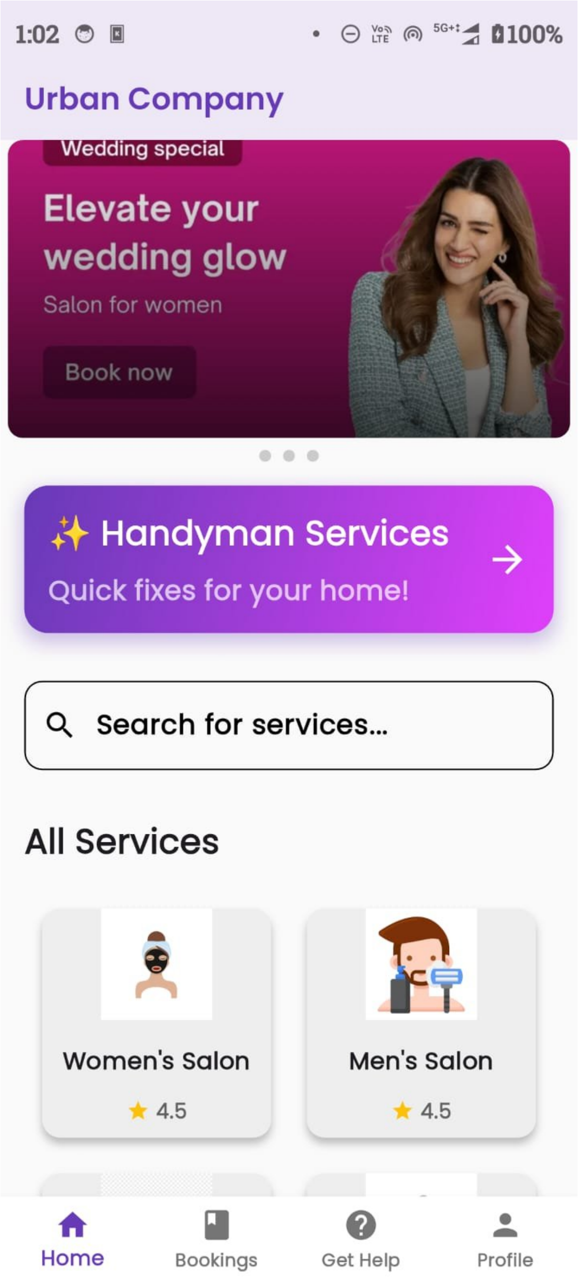
```
    final user = FirebaseAuth.instance.currentUser;
    if (user == null) return;

    final timestamp = DateTime(
      _selectedDate!.year,
      _selectedDate!.month,
      _selectedDate!.day,
      _selectedTime!.hour,
      _selectedTime!.minute,
    );

    await FirebaseFirestore.instance.collection('bookings').add({
      'date': timestamp,
      'price': widget.price,
      'serviceTitle': widget.serviceTitle,
      'status': 'Upcoming',
      'userId': user.uid,
    });

    showDialog(
      context: context,
      builder: (context) => AlertDialog(
        title: const Text('Booking Confirmed'),
        content: Text('Your booking for "${widget.serviceTitle}" has been confirmed.'),
        actions: [
          TextButton(
            onPressed: () => Navigator.popUntil(context, (route) => route.isFirst),
            child: const Text('OK'),
          ),
        ],
      ),
    );
  }
}
```

Booking_success.dart:

```
import 'package:flutter/material.dart';

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Booking Successful'),
        centerTitle: true,
        elevation: 0,
        // backgroundColor: Colors.white24,
        // iconTheme: const IconThemeData(color: Colors.black),
      ),
      body: Center(
        child: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
children: [
  // Success Icon
  const Icon(
    Icons.check_circle,
    color: Colors.green,
    size: 100,
  ),
  const SizedBox(height: 24),

  // Success Message
  const Text(
    'Thank you for your booking!',
    style: TextStyle(
      fontSize: 24,
      fontWeight: FontWeight.bold,
    ),
    textAlign: TextAlign.center,
  ),
  const SizedBox(height: 16),

  // Payment Details
  const Text(
    'Your payment will be collected at the time of service.',
    style: TextStyle(
      fontSize: 16,
      color: Colors.grey,
    ),
    textAlign: TextAlign.center,
  ),
  const SizedBox(height: 24),
  // Payment Method
  Card(
    elevation: 4,
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(12),
    ),
    child: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        children: [
          const Text(
            'Payment Method',
            style: TextStyle(
              fontSize: 18,
              fontWeight: FontWeight.bold,
            ),
          ),
          const SizedBox(height: 16),
          Row(
            mainAxisAlignment: MainAxisAlignment.center,
            children: const [
              Icon(Icons.money, color: Colors.green),
              SizedBox(width: 8),
              Text(
                'Cash on Delivery (COD)',
                style: TextStyle(
                  fontSize: 16,
                  fontWeight: FontWeight.w500,
                ),
              ),
                        ],
                      ),
                      const SizedBox(height: 16),
                      const Text(
                        'UPI/Payment Options Coming Soon!',
                        style: TextStyle(
                          fontSize: 14,
                          color: Colors.grey,
                          fontStyle: FontStyle.italic,
                        ),
                        textAlign: TextAlign.center,
                      ),
                    ],
                  ),
                ),
              ),
              const SizedBox(height: 32),

              // Go to Home Button
              SizedBox(
                width: double.infinity,
                child: ElevatedButton(
                  onPressed: () {
                    Navigator.pushNamedAndRemoveUntil(
                      context,
                      '/home',
                      (route) => false, // Clear the stack
                    );
                  },
                  style: ElevatedButton.styleFrom(
                    backgroundColor: Colors.deepPurple,
                    padding: const EdgeInsets.symmetric(vertical: 16),
                    shape: RoundedRectangleBorder(
                      borderRadius: BorderRadius.circular(8),
                    ),
                  ),
                  child: const Text(
                    'Go to Home',
                    style: TextStyle(
                      fontSize: 18,
                      fontWeight: FontWeight.bold,
                      color: Colors.white,
                    ),
                  ),
                ),
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

Output:

Prajyot Shinde – D15A / 55


1:02 100%

Urban Company

Wedding special

Elevate your wedding glow

Salon for women

Book now

✨ Handyman Services

Quick fixes for your home!

Search for services...

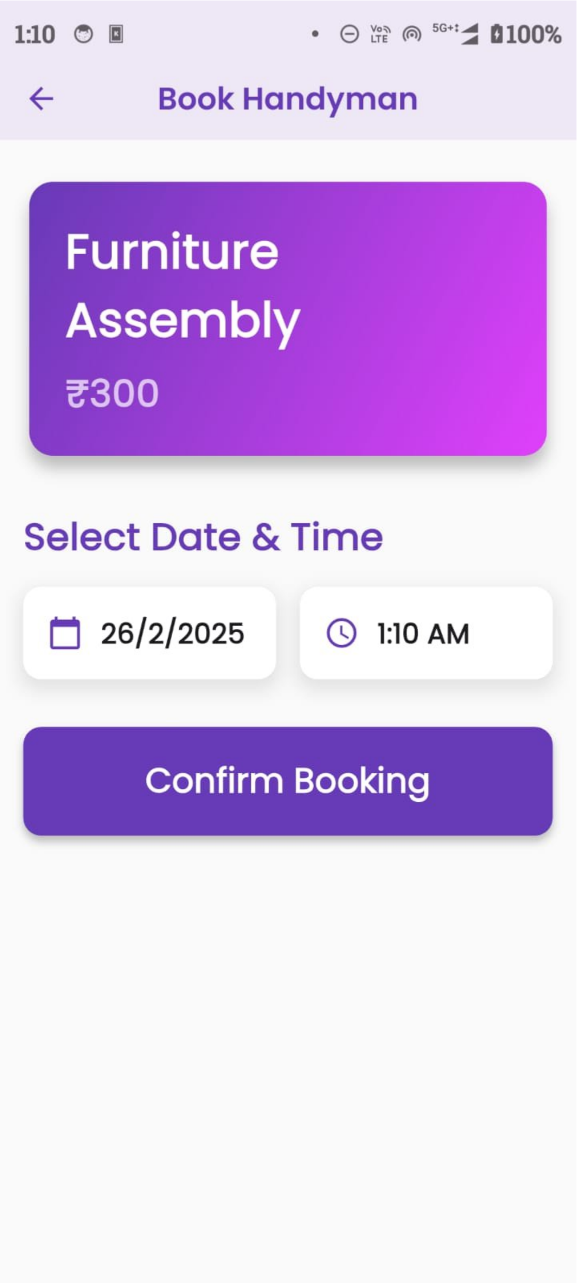
All Services

Women's Salon

★ 4.5

Men's Salon

★ 4.5


1:10 100%

Book Handyman

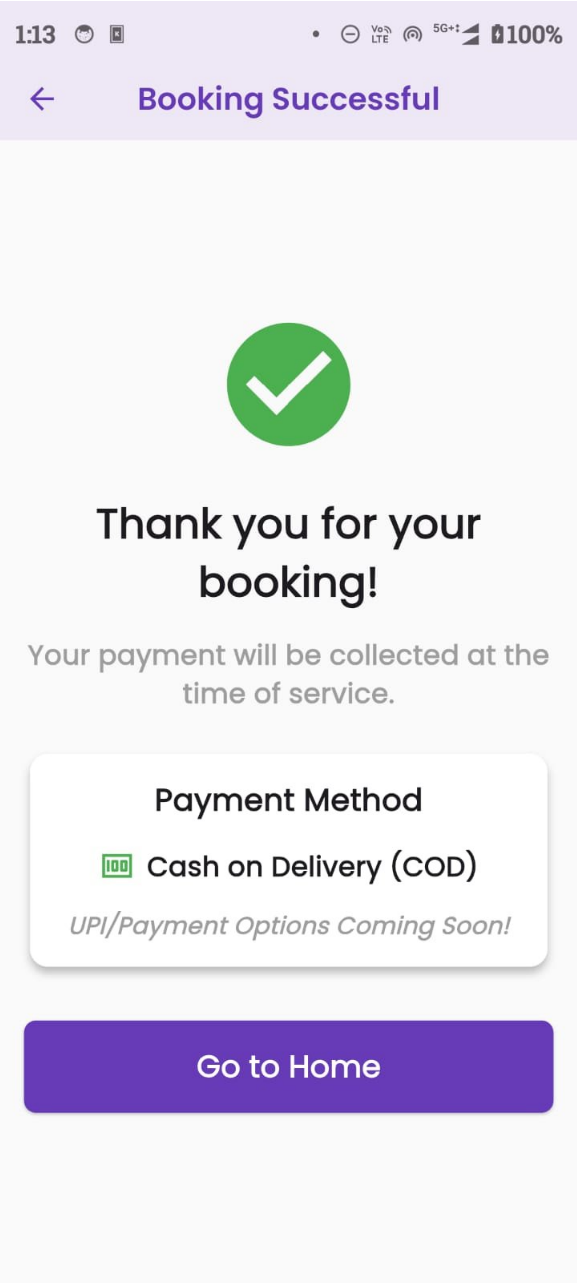
Furniture Assembly

₹300

Select Date & Time

26/2/2025

1:10 AM



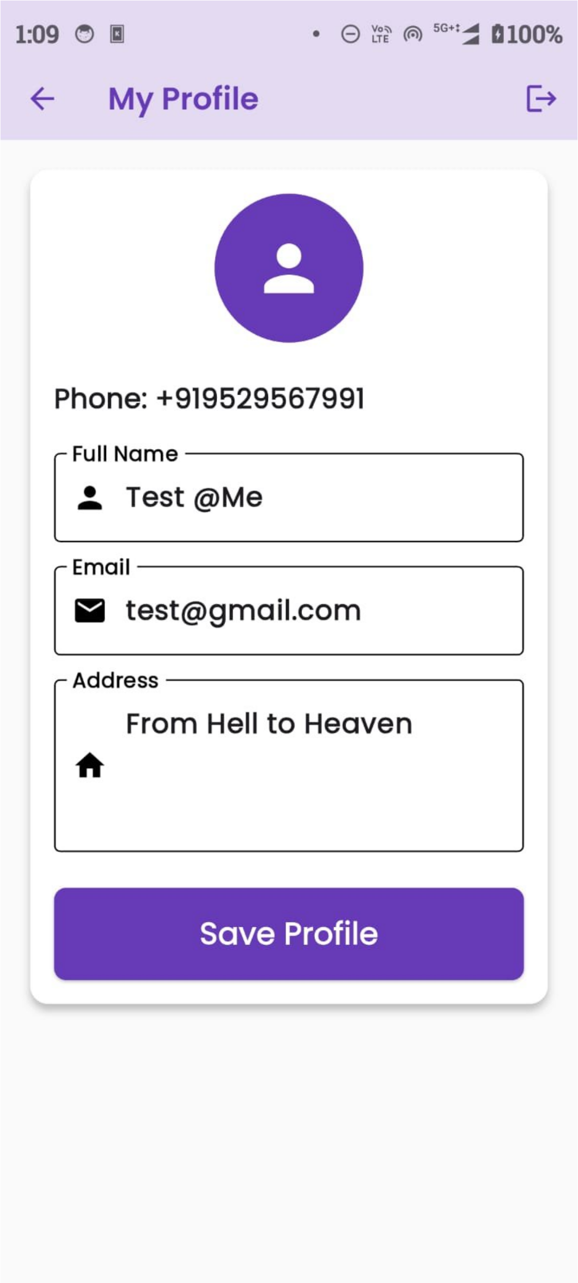
Thank you for your booking!

Your payment will be collected at the time of service.

Payment Method

Cash on Delivery (COD)

UPI/Payment Options Coming Soon!



Phone: +919529567991

Full Name: Test @Me

Email: test@gmail.com

Address: From Hell to Heaven