Experiment 5

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

Theory :

In Flutter, navigation, routing, and gestures play vital roles in creating engaging and intuitive user experiences.

Navigation and Routing:

Navigation refers to moving between different screens or pages within an app. Routing is the mechanism that defines the paths and transitions between these screens. In Flutter, navigation and routing are managed using a Navigator widget.

- Navigator Widget: Each Flutter app has a Navigator widget that manages a stack of Route objects. These Route objects represent individual screens or pages in the app.

- Navigation Stack: The Navigator maintains a stack of routes. Pushing a new route onto the stack navigates to a new screen, and popping a route removes the topmost route from the stack, navigating back to the previous screen.

- Routes: Routes are represented by instances of PageRoute or its subclasses like MaterialPageRoute. They define the content and behavior of each screen.

- Routing Methods: Flutter provides methods like Navigator.push, Navigator.pop, Navigator.pushNamed, and Navigator.popUntil to navigate between routes.

Gestures:

Gestures enable users to interact with the app using touch inputs like tapping, dragging, swiping, etc. Flutter provides a rich set of gesture recognizers that make it easy to handle various touch interactions.

- Gesture Detector: The GestureDetector widget is the primary way to add gesture recognition to widgets in Flutter. It listens for various touch events and provides callbacks for handling those events.

- Handling Gestures: When a gesture is detected, the corresponding callback (e.g., onTap, onLongPress, onPanUpdate, etc.) is invoked, allowing you to respond to user interactions accordingly.

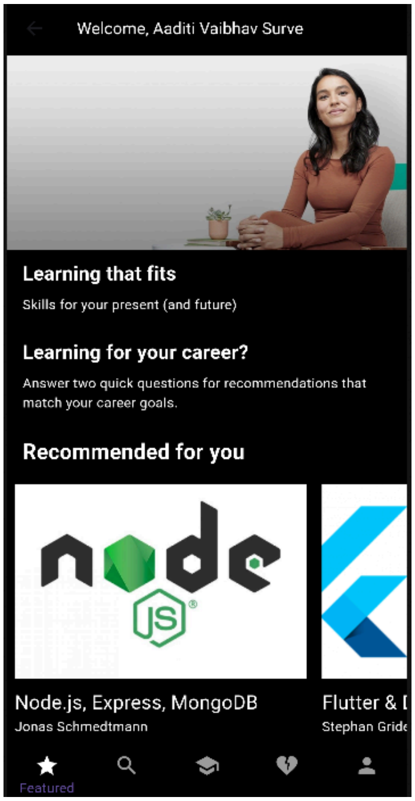
- Gesture Widgets: Some widgets like ListView, GestureDetector, InkWell, Draggable, Dismissible, etc., come with built-in gesture handling capabilities.

Code:

```
//home_page
import 'package:flutter/material.dart';
import 'package:flutter_firebase/features/user_auth/presentation/pages/profile.dart';
import 'package:flutter_firebase/features/user_auth/presentation/pages/search_page.dart';
import 'package:flutter_firebase/features/user_auth/presentation/pages/course1_details_page.dart';
import 'package:flutter_firebase/features/user_auth/presentation/pages/course2_details_page.dart';
import 'package:flutter_firebase/features/user_auth/presentation/pages/wishlist.dart';
class HomePage extends StatefulWidget {
 @override
 _HomePageState createState() => _HomePageState();
}
class _HomePageState extends State<HomePage> {
 int _selectedIndex = 0;
 List<String> courseImages = [
  'assets/images/course_1.jpg',
  'assets/images/course_2.png',
  'assets/images/course_3.png',
  'assets/images/course_4.png',
  'assets/images/course_5.png',
 ];
 List<String> courseTitles = [
  'Node.js, Express, MongoDB',
  'Flutter & Dart-The Complete Guide',
  'Learn JAVA Programming',
  'React JS - Complete Guide',
  'The Complete Python Developer',
 ];
 List<String> courseDescriptions = [
  'Jonas Schmedtmann',
  'Stephan Grider',
  'EdYoda Digital University',
  'Andrei Neagoie',
  'Yihua Zhang',
 ];
 void _onItemTapped(int index) {
  setState(() {
   _selectedIndex = index;
  });
  if (index == 0) {
   Navigator.push(
    context,
    MaterialPageRoute(builder: (context) => HomePage()),
   );
  } else if (index == 1) {
   Navigator.push(
    context,
    MaterialPageRoute(builder: (context) => SearchPage()),
   );
  } else if (index == 3) { // Index 4 corresponds to the Profile icon
   Navigator.push(
    context,
    MaterialPageRoute(builder: (context) => WishlistPage()),
   );
  }else if (index == 4) { // Index 4 corresponds to the Profile icon
   Navigator.push(
    context,
    MaterialPageRoute(builder: (context) => ProfilePage()),
   );
  }
 }
 void _navigateToCourseDetailsPage(int index) {
  Navigator.push(
   context,
   MaterialPageRoute(
    builder: (context) {
     if (index == 0) {
      return Course1DetailsPage();
     } else if (index == 1) {
      return Course2DetailsPage();
     } else {
      // Handle other indices if needed
      return Container(); // Placeholder, you can return a different page or null
     }
    },
   ),
  );
 }

 @override
 Widget build(BuildContext context) {
  return Scaffold(
   appBar: AppBar(
    title: const Text(
     'Welcome, Aaditi Vaibhav Surve',
     style: TextStyle(color: Colors.white, fontSize: 17.0),
    ),
   backgroundColor: Colors.black,
 ),
 backgroundColor: Colors.black,
 body: SingleChildScrollView(
   child: Column(
     crossAxisAlignment: CrossAxisAlignment.start,
     children: [
       Image.asset(
         'assets/images/home.jpg',
         height: 200,
         width: MediaQuery.of(context).size.width,
         fit: BoxFit.cover,
       ),
       const SizedBox(height: 10),
       const Padding(
         padding: EdgeInsets.only(left: 16.0),
         child: Text(
           'Learning that fits',
           style: TextStyle(
             color: Colors.white,
             fontFamily: 'Montserrat',
             fontSize: 20.0,
             fontWeight: FontWeight.bold,
           ),
         ),
       ),
       const SizedBox(height: 7),
       const Padding(
         padding: EdgeInsets.only(left: 16.0),
         child: Text(
           'Skills for your present (and future)',
           style: TextStyle(color: Colors.white),
         ),
       ),
       const SizedBox(height: 25),
       const Padding(
         padding: EdgeInsets.only(left: 16.0),
         child: Text(
           'Learning for your career?',
           style: TextStyle(
             color: Colors.white,
             fontSize: 20,
             fontWeight: FontWeight.bold,
           ),
   ),
 ),
 const SizedBox(height: 7),
 const Padding(
   padding: EdgeInsets.only(left: 16.0),
   child: Text(
     'Answer two quick questions for recommendations that match your career goals.',
     style: TextStyle(color: Colors.white),
   ),
 ),
 const SizedBox(height: 25),
 const Padding(
   padding: EdgeInsets.only(left: 16.0),
   child: Text(
     'Recommended for you',
     style: TextStyle(color: Colors.white, fontSize: 22.0, fontWeight: FontWeight.bold),
   ),
 ),
 const SizedBox(height: 10),
 SingleChildScrollView(
   scrollDirection: Axis.horizontal,
   child: Row(
     mainAxisSize: MainAxisSize.min,
     children: List.generate(
       courseImages.length,
           (index) => GestureDetector(
         onTap: () {
           _navigateToCourseDetailsPage(index);
         },
         child: Padding(
           padding: const EdgeInsets.all(8.0),
           child: Column(
             crossAxisAlignment: CrossAxisAlignment.start,
             children: [
               Container(
                 width: 300,
                 height: 200,
                 decoration: BoxDecoration(
                   color: Colors.grey,
                   borderRadius: BorderRadius.circular(10.0),
                 ),
                 child: Image.asset(
                   courseImages[index],
                   width: 300,
                         height: 180,
                         fit: BoxFit.cover,
                       ),
                     ),
                     const SizedBox(height: 10),
                     Text(
                       courseTitles[index],
                       style: TextStyle(color: Colors.white, fontSize: 20.0),
                     ),
                     Text(
                       courseDescriptions[index],
                       style: TextStyle(color: Colors.white),
                     ),
                   ],
                 ),
               ),
             ),
           ),
         ),
       ),
       const SizedBox(height: 20),
     ],
   ),
 ),
 bottomNavigationBar: BottomNavigationBar(
   backgroundColor: Colors.black,
   selectedIconTheme: const IconThemeData(color: Colors.white),
   selectedLabelStyle: const TextStyle(color: Colors.white),
   unselectedIconTheme: const IconThemeData(color: Colors.grey),
   unselectedLabelStyle: const TextStyle(color: Colors.grey),
   iconSize: 26.0,
   selectedFontSize: 14.0,
   unselectedFontSize: 12.0,
   currentIndex: _selectedIndex,
   onTap: _onItemTapped,
   type: BottomNavigationBarType.fixed,
   items: const [
     BottomNavigationBarItem(icon: Icon(Icons.star), label: 'Featured'),
     BottomNavigationBarItem(icon: Icon(Icons.search), label: 'Search'),
     BottomNavigationBarItem(
         icon: Icon(Icons.school), label: 'My Learning'),
     BottomNavigationBarItem(
         icon: Icon(Icons.heart_broken), label: 'Wishlist'),
     BottomNavigationBarItem(icon: Icon(Icons.person), label: 'Profile'),
        ],
      ),
    );
  }
}

//course1_details_page
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:video_player/video_player.dart';
import 'package:youtube_player_flutter/youtube_player_flutter.dart';
class Course1DetailsPage extends StatefulWidget {
  @override
  _Course1DetailsPageState createState() => _Course1DetailsPageState();
}
class _Course1DetailsPageState extends State<Course1DetailsPage> {
  final CollectionReference wishlistCollection =
  FirebaseFirestore.instance.collection('Wishlist');
  late VideoPlayerController _controller;
  late YoutubePlayerController _youtubeController;
  @override
  void initState() {
    super.initState();
    _controller = VideoPlayerController.networkUrl(
      Uri.parse('https://youtu.be/yEHCfRWz-EI?si=gpoEN2MAhqrzbZCT'),
    )..initialize().then((_) {
      setState(() {}); // Update the state once the video is initialized
    });
    _youtubeController = YoutubePlayerController(
      initialVideoId: 'yEHCfRWz-EI',
      flags: YoutubePlayerFlags(
        autoPlay: false,
      ),
    );
  }
  @override
  void dispose() {
    _controller.dispose();
    _youtubeController.dispose();
    super.dispose();
  }
  void addToWishlist(BuildContext context) async {
    try {
      // Check if the course already exists in the wishlist
      QuerySnapshot querySnapshot = await wishlistCollection.where('course_name', isEqualTo: 'Node.js,
Express, MongoDB & More: The Complete Bootcamp 2024').get();
      if (querySnapshot.docs.isNotEmpty) {
        // Course already exists in the wishlist
        showErrorMessage(context, 'Course already exists in the wishlist');
        return;
      }
      // If the course does not exist, add it to the wishlist
      await wishlistCollection.add({
        'course_name': 'Node.js, Express, MongoDB & More: The Complete Bootcamp 2024',
        'author': 'Jonas Schmedtmann',
        'price': 19.99,
        'ratings': 4.5,
      });
      // Show success message using ScaffoldMessenger
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Course added to wishlist')),
      );
    } catch (e) {
      // Show error message using ScaffoldMessenger
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Failed to add course to wishlist: $e')),
      );
    }
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Node.js, Express, MongoDB', style: TextStyle(color: Colors.white)),
        backgroundColor: Colors.black,
      ),
      body: Container(
        color: Colors.black,
        child: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              // Image of the course
              Stack(
                alignment: Alignment.center,
                children: [
                  Image.asset(
  'assets/images/course_1.jpg',
  width: MediaQuery.of(context).size.width,
  height: 200,
  fit: BoxFit.cover,
),
GestureDetector(
  onTap: () {
    showDialog(
      context: context,
      builder: (BuildContext context) {
        return AlertDialog(
          content: Container(
            width: double.infinity,
            child: YoutubePlayer(
              controller: _youtubeController,
              showVideoProgressIndicator: true,
              progressIndicatorColor: Colors.red,
              progressColors: ProgressBarColors(
                playedColor: Colors.red,
                handleColor: Colors.redAccent,
              ),
              onReady: () {
                _youtubeController.play();
              },
            ),
          ),
        );
      },
    );
  },
  child: Icon(
    Icons.play_circle_outline,
    size: 50,
    color: Colors.black,
  ),
),
Positioned(
  bottom: 8,
  child: Text(
    'Preview this course',
    style: TextStyle(
      color: Colors.black,
      fontWeight: FontWeight.bold,
      fontSize: 17,
                      ),
                    ),
                  ),
                ],
              ),
              SizedBox(height: 20),
              // Title of the course
              Text(
                'Node.js, Express, MongoDB & More: The Complete Bootcamp 2024',
                style: TextStyle(
                  fontSize: 24,
                  fontWeight: FontWeight.bold,
                  color: Colors.white,
                ),
              ),
              SizedBox(height: 5),
              // Description of the course
              Text(
                'Master Node by building a real-world RESTful API and web app (with authentication, Node.js
security, payments & more)',
                style: TextStyle(fontSize: 18, color: Colors.grey[500]),
              ),
              SizedBox(height: 5),
              Row(
                children: [
                  Text(
                    'Created by ',
                    style: TextStyle(
                      fontSize: 16,
                      color: Colors.white,
                    ),
                  ),
                  Text(
                    'Jonas Schmedtmann',
                    style: TextStyle(
                      fontSize: 16,
                      color: Colors.purpleAccent,
                    ),
                  ),
                ],
              ),
              SizedBox(height: 5),
              // Bestseller box
              Container(
  padding: EdgeInsets.all(8),
  decoration: BoxDecoration(
    color: Color(0xFFFFF9C4), // Customize the color as needed
    borderRadius: BorderRadius.circular(5),
  ),
  child: Text(
    'Bestseller',
    style: TextStyle(
      fontSize: 15,
      fontWeight: FontWeight.bold,
      color: Colors.black,
    ),
  ),
),
SizedBox(height: 5),
// Rating of the course with star icons
Row(
  children: [
    Text(
      '4.5', // Rating value
      style: TextStyle(
        fontSize: 16,
        color: Colors.white,
        fontWeight: FontWeight.bold,
      ),
    ),
    SizedBox(
        width: 5), // Add space between rating number and stars
    Icon(Icons.star, color: Colors.yellow),
    Icon(Icons.star, color: Colors.yellow),
    Icon(Icons.star, color: Colors.yellow),
    Icon(Icons.star, color: Colors.yellow),
    Icon(Icons.star_half, color: Colors.yellow),
  ],
),
SizedBox(height: 5),
Text(
  '(22,383 ratings) 136,141 students',
  style: TextStyle(
    fontSize: 14,
    color: Colors.white,
  ),
),
SizedBox(height: 25),
// Price of the course
Text(
  '\$19.99',
  style: TextStyle(
    fontSize: 24,
    fontWeight: FontWeight.bold,
    color: Colors.white,
  ),
),
SizedBox(height: 10),
SizedBox(
  width: double.infinity,
  height: 50,
  child: ElevatedButton(
    onPressed: () {
      // Implement Buy Now functionality
    },
    child: Text(
      'Buy Now',
      style: TextStyle(
        fontSize: 26,
        fontWeight: FontWeight.bold,
        color: Colors.white,
      ),
    ),
    style: ElevatedButton.styleFrom(
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(2),
      ),
      backgroundColor: Colors.purpleAccent,
    ),
  ),
),
SizedBox(height: 10),
// Add to Cart button
SizedBox(
  width: double.infinity,
  height: 50,
  child: ElevatedButton(
    onPressed: () {
      // Add the course to the cart
      addToWishlist(context);
      // Show success message
      showSuccessMessage(context, 'Course added to wishlist');
```

```
                },
                child: Text(
                  'Add to Wishlist',
                  style: TextStyle(
                    fontSize: 20,
                    color: Colors.white,
                  ),
                ),
                style: ElevatedButton.styleFrom(
                  shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(2),
                    side: BorderSide(
                      color: Colors.white, // Border color
                      width: 2, // Border width
                    ),
                  ),
                  backgroundColor: Colors.black,
                ),
              ),
            ),
          ],
        ),
      ),
    ),
  );
}
void showSuccessMessage(BuildContext context, String message) {
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return AlertDialog(
        title: Text('Success'),
        content: Text(message),
        actions: [
          TextButton(
            onPressed: () {
              Navigator.of(context).pop();
            },
            child: Text('OK'),
          ),
        ],
      );
    },
  );
  }
  void showErrorMessage(BuildContext context, String message) {
    showDialog(
      context: context,
      builder: (BuildContext context) {
        return AlertDialog(
          title: Text('Error'),
          content: Text(message),
          actions: [
            TextButton(
              onPressed: () {
                Navigator.of(context).pop();
              },
              child: Text('OK'),
            ),
          ],
        );
      },
    );
  }
}

//profile
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:flutter_firebase/features/user_auth/presentation/pages/login_page.dart';

class ProfilePage extends StatelessWidget {
  void _signOut(BuildContext context) async {
    try {
      await FirebaseAuth.instance.signOut(); // Sign out the user
      // Navigate to the login page
      Navigator.pushReplacement(context, MaterialPageRoute(builder: (context) => LoginPage()));
    } catch (e) {
      print('Error signing out: $e');
      // Handle sign-out error
      // Show a snackbar, toast, or dialog to inform the user about the error
    }
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(
    'Profile',
    style: TextStyle(color: Colors.white),
  ),
  backgroundColor: Colors.black,
),
backgroundColor: Colors.black,
body: SingleChildScrollView(
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.center,
    children: [
      SizedBox(height: 20),
      CircleAvatar(
        radius: 75,
        backgroundColor: Colors.black,
        child: Icon(
          Icons.account_circle,
          size: 150,
          color: Colors.white,
        ),
      ),
      SizedBox(height: 10),
      Text(
        'Aaditi',
        style: TextStyle(fontSize: 24, color: Colors.white),
      ),
      SizedBox(height: 10),
      Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Icon(
            Icons.account_box,
            size: 20,
            color: Colors.grey,
          ),
          SizedBox(width: 10),
          Text(
            '2021.aaditi.surve@ves.ac.in',
            style: TextStyle(fontSize: 18, color: Colors.white),
          ),
        ],
      ),
      SizedBox(height: 10),
      Text(
        'Become an instructor',
              style: TextStyle(fontSize: 16, color: Colors.purpleAccent),
            ),
            SizedBox(height: 20),
            OptionWidget(text: 'Download options'),
            OptionWidget(text: 'Video playback options'),
            OptionWidget(text: 'Career goal'),
            OptionWidget(text: 'Account Security'),
            OptionWidget(text: 'Email notification preferences'),
            OptionWidget(text: 'Learning reminders'),
            OptionWidget(text: 'About Udemy'),
            OptionWidget(text: 'About Udemy Business'),
            OptionWidget(text: 'Frequently asked questions'),
            OptionWidget(text: 'Share the Udemy App'),
            OptionWidget(text: 'Status'),
            // Add more OptionWidget instances for other options
            SizedBox(height: 15),
            ElevatedButton(
              onPressed: () {
                _signOut(context); // Call the sign-out method
              },
              style: ButtonStyle(
                backgroundColor: MaterialStateProperty.all(Colors.black), // Set button background color to
black),
              child: Text(
                'Sign Out',
                style: TextStyle(color: Colors.purpleAccent, fontSize: 18.0, fontWeight: FontWeight.bold),
              ),
            ),
            SizedBox(height: 20),
          ],
        ),
      ),
    );
  }
}
class OptionWidget extends StatelessWidget {
  final String text;
  const OptionWidget({Key? key, required this.text}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return Container(
      padding: EdgeInsets.symmetric(horizontal: 20, vertical: 10),
      margin: EdgeInsets.symmetric(vertical: 5),
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(20),
        color: Colors.black,
      ),
      child: Row(
        children: [
          Expanded(
            child: Text(
              text,
              style: TextStyle(fontSize: 18, color: Colors.white),
            ),
          ),
          Icon(
            Icons.arrow_forward_ios,
            color: Colors.white,
          ),
        ],
      ),
    );
  }
}
```

Output :



Welcome, Aaditi Vaibhav Surve

Learning that fits

Skills for your present (and future)

Learning for your career?

Answer two quick questions for recommendations that match your career goals.

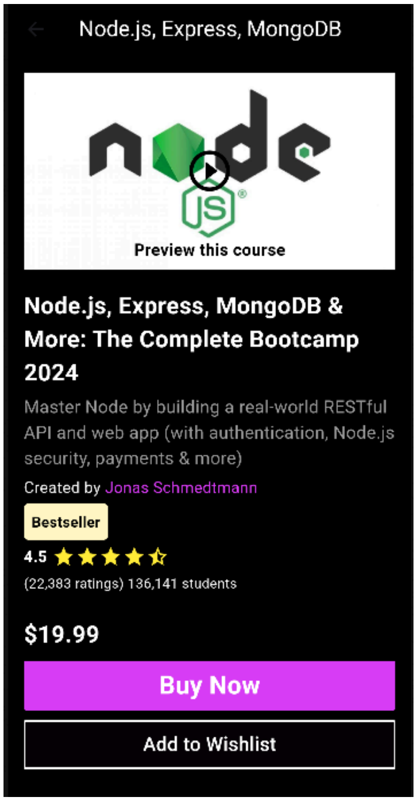
Recommended for you

Node.js, Express, MongoDB

Jonas Schmedtmann

Flutter & I

Stephan Gride



Node.js, Express, MongoDB

Preview this course

Node.js, Express, MongoDB & More: The Complete Bootcamp 2024

Master Node by building a real-world RESTful API and web app (with authentication, Node.js security, payments & more)

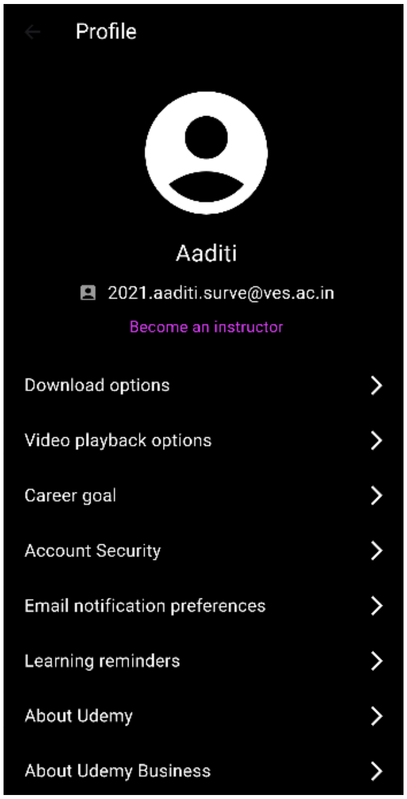
Created by Jonas Schmedtmann

Bestseller

4.5 ★★★★☆

(22,383 ratings) 136,141 students

$19.99



Profile

Aaditi

2021.aaditi.surve@ves.ac.in

Become an instructor