Experiment No - 05

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

Theory:

Flutter provides tools to handle navigation, routing, and gestures, allowing users to move between screens and interact with the app smoothly. These features help create a user-friendly experience in mobile applications.

1. Navigation in Flutter

Navigation is the process of moving between different screens (or pages) in a Flutter app. Flutter uses a stack-based approach for navigation, where new screens are pushed onto the stack and removed when the user navigates back. Types of Navigation:

- Push Navigation: Moves to a new screen and adds it to the stack.
- Pop Navigation: Removes the current screen and returns to the previous one.
- Named Routes: Uses pre-defined route names to navigate.
- Navigation with Data: Allows passing data between screens when navigating.

2. Routing in Flutter

Routing helps in managing different screens efficiently. Instead of manually handling each screen transition, Flutter allows defining routes in a structured way. Types of Routing:

- Direct Routing: Navigates to a specific screen using explicit methods.
- Named Routing: Uses a predefined route name to navigate, making the app more organized.

Routing improves app maintainability, especially in apps with multiple screens.

3. Gestures in Flutter

Gestures enable user interaction in Flutter applications. Flutter provides built-in gesture detection capabilities for touch-based interactions.Common Gestures:

- Tap: A single touch interaction.
- Double Tap: Two quick consecutive taps.
- Long Press: Holding a touch for a longer duration.
- Swipe: Moving a finger across the screen.
- Drag: Moving an object by pressing and holding it.

Gestures are essential for making apps interactive and responsive.

4. Combining Navigation and Gestures

Navigation and gestures can be combined to enhance user experience. For example:

- Tapping on a button can navigate to another screen.
- Swiping a card can delete an item or move to another page.
- Dragging an element can reposition items within the app.

Navigation, routing, and gestures are fundamental to creating an interactive Flutter application. Navigation allows movement between screens, routing helps manage screens efficiently, and gestures enable touch interactions. Mastering these concepts helps in developing dynamic and user-friendly Flutter applications.

Code:

1. **Home_page.dart**

```
import 'package:flutter/material.dart';
import
'package:linkedin/create_post_page.dart';
import
'package:linkedin/network_page.dart';
import
'package:linkedin/notifications_page.dart';
import 'package:linkedin/job_page.dart';
import
'package:linkedin/profile_page.dart';


void main() {
  runApp(HomePage());
}

class HomePage extends StatelessWidget
{
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner:
false,
      home: HomeScreen(),
    );
  }
}

class HomeScreen extends
StatefulWidget {
  @override
  _HomeScreenState createState() =>
_HomeScreenState();
}
```

```
class _HomeScreenState extends
State<HomeScreen> {
  int currentIndex = 0;
  final GlobalKey<ScaffoldState>
_scaffoldKey =
GlobalKey<ScaffoldState>();

  final List<Widget> screens = [
    HomeContent(),
    NetworkPage(),
    CreatePostPage(),
    NotificationPage(),
    JobPage(),
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      key: _scaffoldKey,
      appBar: AppBar(
        title: Row(
          children: [
            GestureDetector(
              onTap: () {
_scaffoldKey.currentState?.openDrawer();
      },

              child: CircleAvatar(
                radius: 18,
                backgroundImage:
AssetImage('assets/sannidhi_illusions.jpg'
),
              ),
            ),
      SizedBox(width: 10),
      Expanded(
       child: TextField(
        decoration: InputDecoration(
         hintText: 'Search',
         prefixIcon: Icon(Icons.search),
         contentPadding:
EdgeInsets.symmetric(vertical: 5),
         border: OutlineInputBorder(
          borderRadius:
BorderRadius.circular(15),
          borderSide: BorderSide.none,
         ),
         filled: true,
         fillColor:
Colors.white.withOpacity(0.1),
        ),
       ),
      ),
     ],
    ),
    backgroundColor: Colors.blue,
    elevation: 0,
    actions: [
     IconButton(
      icon:
Icon(Icons.chat_bubble_outline),
      onPressed: () {},
     ),
    ],
   ),
   drawer: Drawer(
    child: ListView(
     padding: EdgeInsets.zero,
     children: [
      DrawerHeader(
       decoration: BoxDecoration(
        color: Colors.blue,
       ),
       child: Column(
        crossAxisAlignment:
CrossAxisAlignment.start,
        children: [
         CircleAvatar(
          radius: 30,
          backgroundImage:

AssetImage('assets/sannidhi_illusions.jpg'
),
         ),
SizedBox(height: 10),
         Text('Sannidhi Kailaje', style:
TextStyle(color: Colors.white, fontSize: 18,
fontWeight: FontWeight.bold)),
 Text('Software Engineer', style:
TextStyle(color: Colors.white70)),
        ],
       ),
      ),
      ListTile(
       leading: Icon(Icons.person),
       title: Text('Profile'),
       onTap: () {
        Navigator.push(
         context,
         MaterialPageRoute(builder:
(context) => ProfilePage()),
        );
       },
      ),
      ListTile(
       leading: Icon(Icons.settings),
       title: Text('Settings'),
       onTap: () {},
      ),
      ListTile(
       leading: Icon(Icons.logout),
       title: Text('Logout'),
       onTap: () {},
      ),
     ],
    ),
   ),
   body: screens[currentIndex],
   bottomNavigationBar:
BottomNavigationBar(
    type:
BottomNavigationBarType.fixed,
    currentIndex: currentIndex,
    onTap: (index) {
     setState(() {
      currentIndex = index;
     });
    },
    items: [
     BottomNavigationBarItem(icon:
Icon(Icons.home), label: 'Home'),
     BottomNavigationBarItem(icon:
Icon(Icons.group), label: 'Network'),
     BottomNavigationBarItem(icon:
Icon(Icons.add_box), label: 'Post'),
     BottomNavigationBarItem(icon:
Icon(Icons.notifications), label:
'Notifications'),
     BottomNavigationBarItem(icon:
```

```
Icon(Icons.work), label: 'Jobs'),
        ],
      ),
    );
  }
}

// Home Feed Content
class HomeContent extends
StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return ListView(
      children: [
        PostCard(
          name: 'Doug Stevenson',
          profession: 'MERN-STACK
Developer, REST APIs',
          content: "Look at this JavaScript
code I don't believe it, what kind of people
are living in this world haha",
          hashtags: "#javascript
#mernstackdev #development
#developers #html #css #laravel",
          imageUrl:
'https://images.unsplash.com/photo-
1610832958506-
aa56368176cf?w=600&auto=format&fit=cr
op&q=60&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxzZWFyY2
h8M3x8ZnJ1aXRzfGVufDB8fDB8fHww',
          likes: '10',
          comments: '44',
          reposts: '11',
        ),
        Divider(),
        PostCard(
          name: 'Doug Stevenson',
          profession: 'MERN-STACK
Developer, REST APIs',
          content: "Look at this JavaScript
code I don't believe it, what kind of people
are living in this world haha",
          hashtags: "#javascript
#mernstackdev #development
#developers #html #css #laravel",
          imageUrl:
'https://images.unsplash.com/photo-
1610832958506-
aa56368176cf?w=600&auto=format&fit=cr
op&q=60&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxzZWFyY2
h8M3x8ZnJ1aXRzfGVufDB8fDB8fHww',
          likes: '10',
          comments: '44',
          reposts: '11',
        ),
 Divider(),
        PostCard(
          name: 'Doug Stevenson',
          profession: 'MERN-STACK
Developer, REST APIs',
          content: "Look at this JavaScript
code I don't believe it, what kind of people
are living in this world haha",
          hashtags: "#javascript
#mernstackdev #development
#developers #html #css #laravel",
          imageUrl:
'https://images.unsplash.com/photo-
1610832958506-
aa56368176cf?w=600&auto=format&fit=cr
op&q=60&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxzZWFyY2
h8M3x8ZnJ1aXRzfGVufDB8fDB8fHww',
          likes: '10',
          comments: '44',
          reposts: '11',
        ),

      ],
    );
  }
}

// Placeholder Screens
class NetworkScreen extends
StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Center(child: Text("Network
Screen", style: TextStyle(fontSize: 20)));
  }
}

class PostScreen extends
StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Center(child: Text("Create a
Post", style: TextStyle(fontSize: 20)));
  }
}

class NotificationsScreen extends
StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Center(child: Text("Notifications",
style: TextStyle(fontSize: 20)));
  }
}
```

```
class JobsScreen extends
StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Center(child: Text("Jobs
Listings", style: TextStyle(fontSize: 20)));
  }
}

// Post Card Component
class PostCard extends StatelessWidget {
  final String name;
  final String profession;
  final String content;
  final String hashtags;
  final String imageUrl;
  final String likes;
  final String comments;
  final String reposts;

  PostCard({
    required this.name,
    required this.profession,
    required this.content,
    required this.hashtags,
    required this.imageUrl,
    required this.likes,
    required this.comments,
    required this.reposts,
  });

  @override
  Widget build(BuildContext context) {
    return Card(
      margin: EdgeInsets.all(10),
      child: Column(
        crossAxisAlignment:
CrossAxisAlignment.start,
        children: [
          ListTile(
            leading: CircleAvatar(
              backgroundImage:
NetworkImage('https://via.placeholder.com
/50'),
            ),
            title: Text(name, style:
TextStyle(fontWeight: FontWeight.bold)),
            subtitle: Text(profession),
            trailing: Icon(Icons.more_vert),
          ),
          Padding(
            padding: EdgeInsets.all(8.0),
            child: Text(content),
          ),
          Padding(
            padding:
EdgeInsets.symmetric(horizontal: 8.0),
            child: Text(
              hashtags,
              style: TextStyle(color:
Colors.blue),
            ),
          ),
          Image.network(imageUrl, fit:
BoxFit.cover),
          Padding(
            padding: EdgeInsets.all(8.0),
            child: Row(
              mainAxisAlignment:
MainAxisAlignment.spaceBetween,
              children: [
                Row(
                  children: [
                    Icon(Icons.thumb_up_alt,
color: Colors.blue),
                    Icon(Icons.favorite, color:
Colors.red),
                    Icon(Icons.lightbulb, color:
Colors.yellow),
                    SizedBox(width: 5),
                    Text(likes),
                  ],
                ),
                Text('$comments comments -
$reposts reposts'),
              ],
            ),
          ),
          Divider(),
          Padding(
            padding:
EdgeInsets.symmetric(horizontal: 8.0),
            child: Row(
              mainAxisAlignment:
MainAxisAlignment.spaceAround,
              children: [
                IconButton(icon:
Icon(Icons.thumb_up_alt_outlined),
onPressed: () {}),
                IconButton(icon:
Icon(Icons.chat_bubble_outline),
onPressed: () {}),
                IconButton(icon:
Icon(Icons.repeat), onPressed: () {}),
                IconButton(icon:
Icon(Icons.send), onPressed: () {}),
              ],
            ),
          ),
        ],
      ),
    );
  }
}
```

2. CreatePost_page.dart

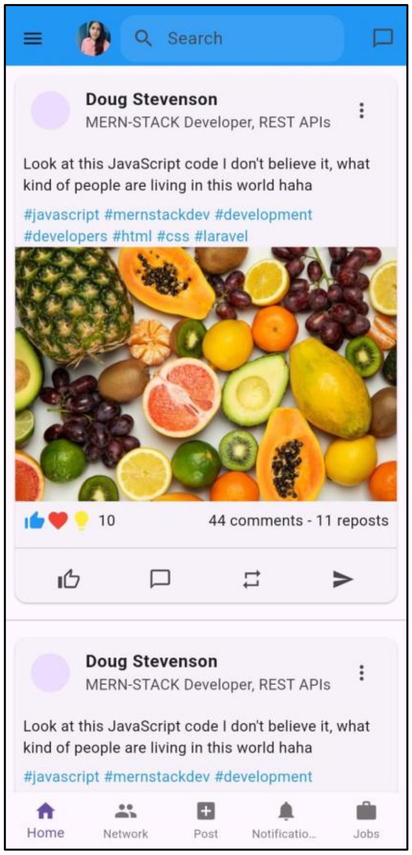
```
import 'dart:io';
import 'package:flutter/material.dart';
import
'package:image_picker/image_picker.
dart';

class CreatePostPage extends
StatefulWidget {
 @override
 _CreatePostPageState createState()
=> _CreatePostPageState();
}

class _CreatePostPageState extends
State<CreatePostPage> {
 final TextEditingController
titleController =
TextEditingController();
 final TextEditingController
descriptionController =
TextEditingController();
 final TextEditingController
hashtagController =
TextEditingController();
 String privacyOption = "Public";
 File? mediaFile;

 final ImagePicker picker =
ImagePicker();

 // Pick Image or Video
 Future<void>
pickMedia(ImageSource source, {bool
isVideo = false}) async {
  final pickedFile = isVideo
    ? await picker.pickVideo(source:
source)
    : await picker.pickImage(source:
source);

  if (pickedFile != null) {
   setState(() {
    mediaFile = File(pickedFile.path);
   });
  }
 }

 // Submit Post (Dummy Function for
Now)
 void submitPost() {
  if (titleController.text.isEmpty ||
descriptionController.text.isEmpty) {

ScaffoldMessenger.of(context).showS
nackBar(
    SnackBar(content: Text('Title and
Description are required!')),
   );
   return;
  }

  // Clear Fields
  setState(() {
   titleController.clear();
   descriptionController.clear();
   hashtagController.clear();
   mediaFile = null;
  });


ScaffoldMessenger.of(context).showS
nackBar(
   SnackBar(content: Text('Post
Created Successfully!')),
  );
 }

 @override
 Widget build(BuildContext context) {
  return Scaffold(
   appBar: AppBar(
    title: Text("Create Post"),
    actions: [
     TextButton(
      onPressed: submitPost,
      child: Text(
       "Post",
       style: TextStyle(color:
Colors.blue, fontSize: 16),
      ),
     )
    ],
   ),
   body: SingleChildScrollView(
    padding: EdgeInsets.all(16),
    child: Column(
     crossAxisAlignment:
CrossAxisAlignment.start,
     children: [
      // Title Input
      TextField(
       controller: titleController,
       decoration: InputDecoration(
        hintText: "Title",
        border: InputBorder.none,
       ),
      ),
 Divider(),

      // Description Input
      TextField(
       controller:
descriptionController,
       maxLines: 3,
       decoration: InputDecoration(
        hintText: "What do you want
to talk about?",
        border: InputBorder.none,
       ),
      ),
      Divider(),

      // Hashtag Input
      TextField(
       controller: hashtagController,
       decoration: InputDecoration(
        hintText: "Add hashtags
(e.g., #Flutter #Firebase)",
        border: InputBorder.none,
       ),
      ),
      SizedBox(height: 10),
      Divider(),
      SizedBox(height: 20),

      // Privacy Selector
      Row(
       children: [
        Text(
         "Privacy",
         style: TextStyle(fontSize:
16, fontWeight: FontWeight.w500),
        ),
        SizedBox(width: 30), //
Spacing between text and dropdown
        Container(
         padding:
EdgeInsets.symmetric(horizontal: 10,
vertical: 0), // Inner padding
         decoration: BoxDecoration(
          color:
Colors.blue.shade100, // Background
color
          borderRadius:
BorderRadius.circular(8), // Rounded
corners
         ),
         child:
DropdownButtonHideUnderline( //
Removes default underline
          child:
DropdownButton<String>(
           value: privacyOption,
           items: ["Public",
"Connections", "Private"].map((option)
{
```

```
 return DropdownMenuItem<String>(
             value: option,
             child: Text(option,
style: TextStyle(fontSize: 14)),
            );
           }).toList(),
           onChanged: (value) {
            setState(() {
             privacyOption =
value!;
            });
           },
           dropdownColor:
Colors.blue.shade50, // Background
color of dropdown menu
          ),
         ),
        ),
       ],
      ),

      SizedBox(height: 40),

      Text(
       "Upload Image or Video",
       style: TextStyle(fontSize: 16,
fontWeight: FontWeight.bold),
      ),
      SizedBox(height: 20),


      // Media Upload Buttons
      Row(
       mainAxisAlignment:
MainAxisAlignment.spaceEvenly,
       children: [
        ElevatedButton.icon(
         icon: Icon(Icons.image),
         label: Text("Image"),
         onPressed: () =>
pickMedia(ImageSource.gallery),
        ),
        ElevatedButton.icon(
         icon: Icon(Icons.video_file),
         label: Text("Video"),
         onPressed: () =>
pickMedia(ImageSource.gallery,
isVideo: true),
        ),
       ],
      ),

      SizedBox(height: 20),

      // Display selected media
      if (mediaFile != null)
       Stack(
        alignment:
```

```
Alignment.topRight,
                children: [
                  // Check if the file is a
video or an image

mediaFile!.path.endsWith(".mp4")
                      ? Container(
                    height: 200,
                    width: double.infinity,
                    decoration:
BoxDecoration(
                      color: Colors.black12,
                      borderRadius:
BorderRadius.circular(10),
                    ),
                    child: Center(
                      child:
Icon(Icons.video_file, size: 50, color:
Colors.black54),
                    ),
                  )
                      : ClipRRect(
                    borderRadius:
BorderRadius.circular(10),
                    child:
Image.file(mediaFile!,
                        height: 200, width:
double.infinity, fit: BoxFit.cover),
                  ),

                  // Circular close button on
the media itself
```

```
                  Positioned(
                    top: 8,
                    right: 8,
                    child: Container(
                      decoration:
BoxDecoration(
                        shape:
BoxShape.circle,
                        color: Colors.red, //
Background color for visibility
                      ),
                      child: IconButton(
                        icon: Icon(Icons.close,
color: Colors.white, size: 20),
                        onPressed: () =>
setState(() => mediaFile = null),
                      ),
                    ),
                  ),
                ],
              ),

            SizedBox(height: 20),
          ],
        ),
      ),
    );
  }
}
```

Output:



Look at this JavaScript code I don't believe it, what kind of people are living in this world haha

#javascript #mernstackdev #development #developers #html #css #laravel

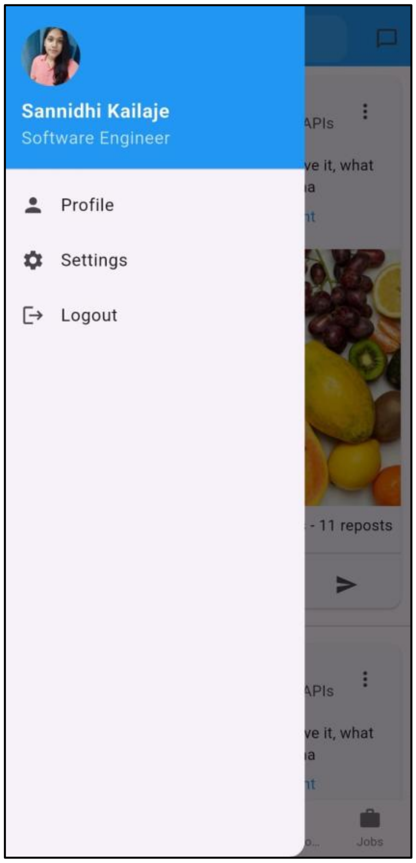
Doug Stevenson

MERN-STACK Developer, REST APIs

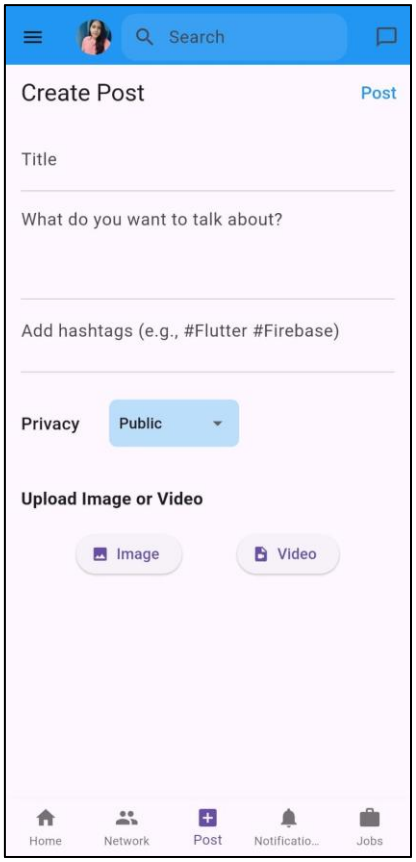
Look at this JavaScript code I don't believe it, what kind of people are living in this world haha

#javascript #mernstackdev #development

Home page



Sidebar



Create Post page