

## **EXPERIMENT NO: - 05**

**Name:-** Arnav Sawant

**Class:-** D15A

**Roll:No: -** 52

**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) => NewScreen())  
    );  
  },  
)
```

```
context,  
    MaterialPageRoute(builder: (context) => SecondScreen()),  
  );},  
);
```

## 2. Using a list for Named routes

Flutter allows the use of a list to manage named routes dynamically, making navigation cleaner and more scalable, especially in larger applications. Instead of writing multiple `Navigator.pushNamed()` calls, routes can be stored in the list and can be accessed by the index

Eg:-

```
final List<String> _routes = ['/search', '/login', '/profile', '/settings'];  
void _onItemTapped(int index) {  
  Navigator.pushReplacementNamed(context, _routes[index]);  
}
```

## 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.

## Code: -

main.dart

```
import 'package:flutter/material.dart';
import 'login/login.dart';
import 'profile/editprofile.dart';
import 'profile/profile.dart';
import 'settings.dart';
import 'search/searchpage.dart';
import 'home.dart'; // Import HomeScreen

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: HomeScreenWrapper(), // Use a
      wrapper for bottom navigation
    );
  }
}

class HomeScreenWrapper extends
StatefulWidget {
  @override
  _HomeScreenWrapperState createState() =>
  _HomeScreenWrapperState();
}

class _HomeScreenWrapperState extends
State<HomeScreenWrapper> {
  int _selectedIndex = 1;

  final List<Widget> _pages = [
    SearchPage(),
    LoginPage(), // Home Page
    ProfileScreen(),
    SettingsPage(),
  ];

  void _onItemTapped(int index) {
    setState() {
      _selectedIndex = index;
    }
  }
}
```

```
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: IndexedStack(
      index: _selectedIndex,
      children: _pages,
    ),
    bottomNavigationBar:
BottomNavigationBar(
  items: const <BottomNavigationBarItem>[
    BottomNavigationBarItem(
      icon: Icon(Icons.search),
      label: 'Search',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.home),
      label: 'Home',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.account_circle),
      label: 'Profile',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.settings),
      label: 'Settings',
    ),
  ],
  currentIndex: _selectedIndex,
  selectedItemColor: Colors.blue,
  unselectedItemColor: Colors.grey,
  onTap: _onItemTapped,
  type: BottomNavigationBarType.fixed,
),
);
}
```

editprofile.dart

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

class EditProfileScreen extends StatefulWidget {
  @override
  _EditProfileScreenState createState() =>
    _EditProfileScreenState();
}

class _EditProfileScreenState extends
    State<EditProfileScreen> {
  TextEditingController nameController =
    TextEditingController(text: "Arnav");
  TextEditingController bioController =
    TextEditingController();

  int nameMaxLength = 20;
  int bioMaxLength = 120;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        leading: IconButton(
          icon: Icon(Icons.arrow_back, color:
Colors.black),
          onPressed: () => Navigator.pop(context),
        ),
        title: Text("Edit Profile", style:
TextStyle(color: Colors.black)),
        backgroundColor: Colors.white,
        elevation: 0,
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment:
CrossAxisAlignment.start,
          children: [
            // Profile Picture
            Center(
              child: CircleAvatar(
                radius: 40,
                backgroundImage:
AssetImage('assets/profile.jpg'), // Replace with
your image
              ),
            ),
          ],
        ),
      ),
    );
  }
}
```

```
        SizedBox(height: 20),

        // Name Field
        _buildLabel("Name"),
        TextField(
          controller: nameController,
          maxLength: nameMaxLength,
          decoration: InputDecoration(
            border: OutlineInputBorder(),
          ),
        ),
        _buildCharacterLimit(nameController.te
xt.length, nameMaxLength),

        SizedBox(height: 16),

        // Bio Field
        _buildLabel("Your Bio"),
        TextField(
          controller: bioController,
          maxLength: bioMaxLength,
          maxLines: 3,
          decoration: InputDecoration(
            border: OutlineInputBorder(),
            hintText: "Enter your bio here",
          ),
        ),
        _buildCharacterLimit(bioController.text.
length, bioMaxLength),

        SizedBox(height: 20),

        // Submit Button
        SizedBox(
          width: double.infinity,
          child: ElevatedButton(
            onPressed: () {},
            style: ElevatedButton.styleFrom(
              backgroundColor: Colors.blue,
              padding:
EdgeInsets.symmetric(vertical: 14),
            ),
            child: Text("Submit", style:
TextStyle(fontSize: 16, color: Colors.white)),
          ),
        ),
      ],
    ),
  ),
}
```

```

    ),
  );
}

// Helper to create labels
Widget _buildLabel(String text) {
  return Padding(
    padding: const EdgeInsets.only(bottom: 8),
    child: Text(text, style: TextStyle(fontSize:
16, fontWeight: FontWeight.bold)),
  );
}

```

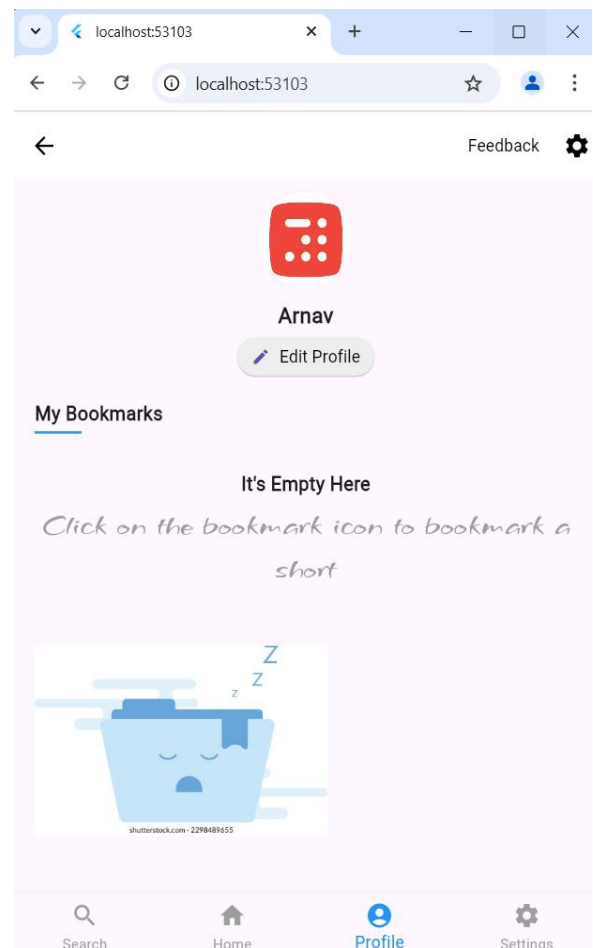
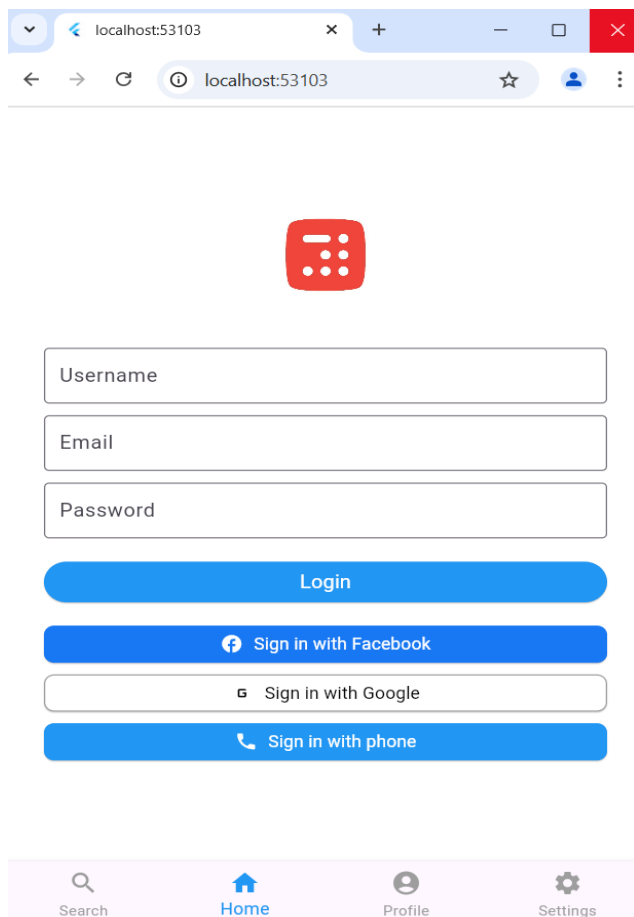
```

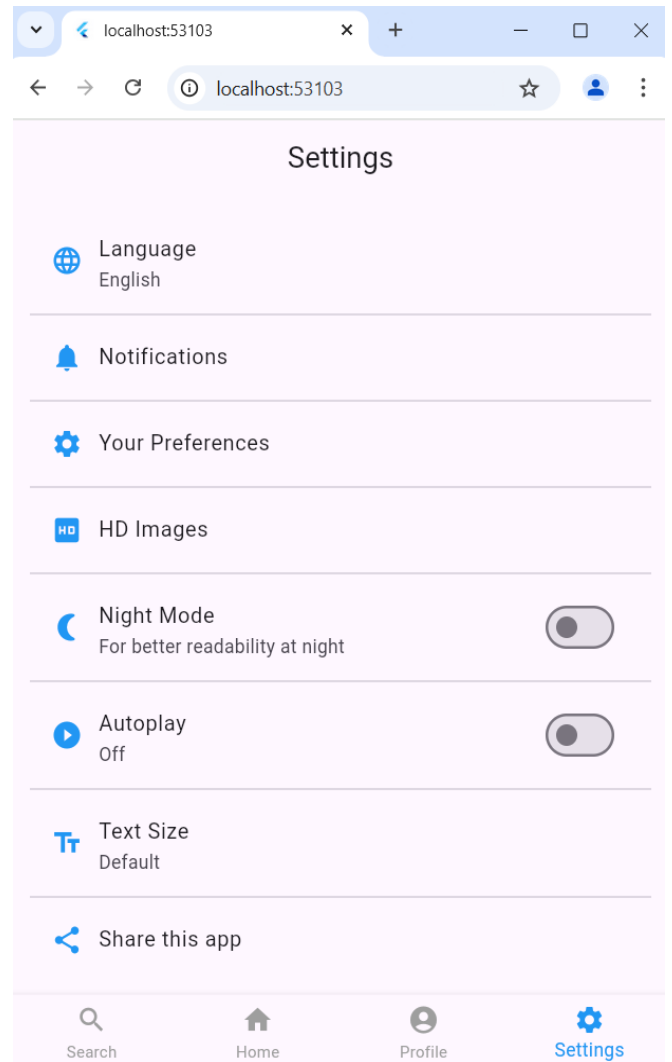
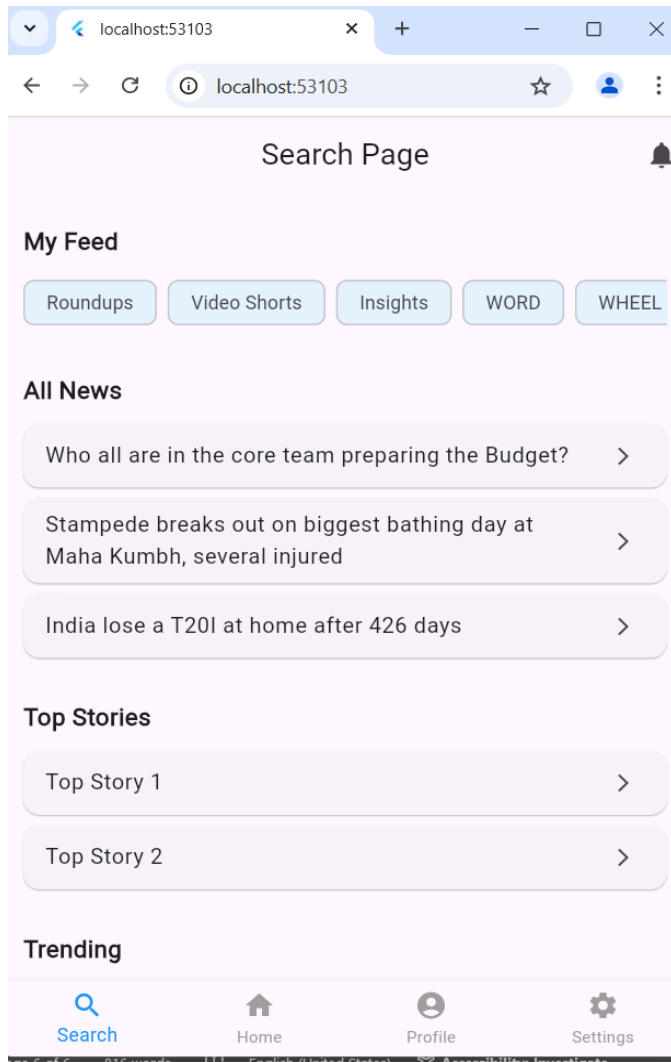
// Character counter
Widget _buildCharacterLimit(int current, int
max) {
  return Align(
    alignment: Alignment.centerRight,
    child: Text("$current/$max", style:
TextStyle(color: Colors.grey, fontSize: 12)),
  );
}
}

```

#### Screenshots:-

1. The first image is the home screen, when I click on profile it would go to the second image.
2. Then when I click on search, it would go to the third image, then when I click on settings it would go on 4<sup>th</sup> image
3. Then when I click on 5<sup>th</sup> image it would go on the 6<sup>th</sup> Image of editprofile.





localhost:53103

localhost:53103

←

Edit Profile

Name

Arnav

5/20

5/20

Your Bio

Enter your bio here

0/120

0/120

Submit