

## Experiment No : 7

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

### Theory :

In Android, a route is equivalent to an Activity. In iOS, a route is equivalent to a ViewController. In Flutter, a route is just a widget.

This recipe uses the Navigator to navigate to a new route.

The next few sections show how to navigate between two routes, using these steps:

Create two routes.

Navigate to the second route using Navigator.push().

Return to the first route using Navigator.pop().

1. First, create two routes to work with. Since this is a basic example, each route contains only a single button. Tapping the button on the first route navigates to the second route. Tapping the button on the second route returns to the first route.
2. To switch to a new route, use the Navigator.push() method. The push() method adds a Route to the stack of routes managed by the Navigator. Where does the Route come from? You can create your own, or use a MaterialPageRoute, which is useful because it transitions to the new route using a platform-specific animation.
3. By using the Navigator.pop() method. The pop() method removes the current Route from the stack of routes managed by the Navigator.

### Code :

Main.dart

```
import 'package:exp_7/second_page.dart';
import 'package:flutter/material.dart';

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

```

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: FirstPage(),
    );
  }
}

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('Exp - 7'),
        ),
        body: Container(
          child: Center(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                Text('First Page Here'),
                OutlinedButton.icon(
                  onPressed: (){
                    Navigator.push(context, MaterialPageRoute(builder:
(context)=> SecondPage()));
                  },
                  label: Text('Next Page'), icon: Icon(Icons.arrow_forward),)
              ],
            ),
          ),
        ),
      ),
    );
  }
}

```

## Second\_page.dart

```

import 'package:exp_7/main.dart';
import 'package:flutter/material.dart';

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('Exp - 7'),
        ),
        body: Container(
          child: Center(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                Text('Second Page Here'),

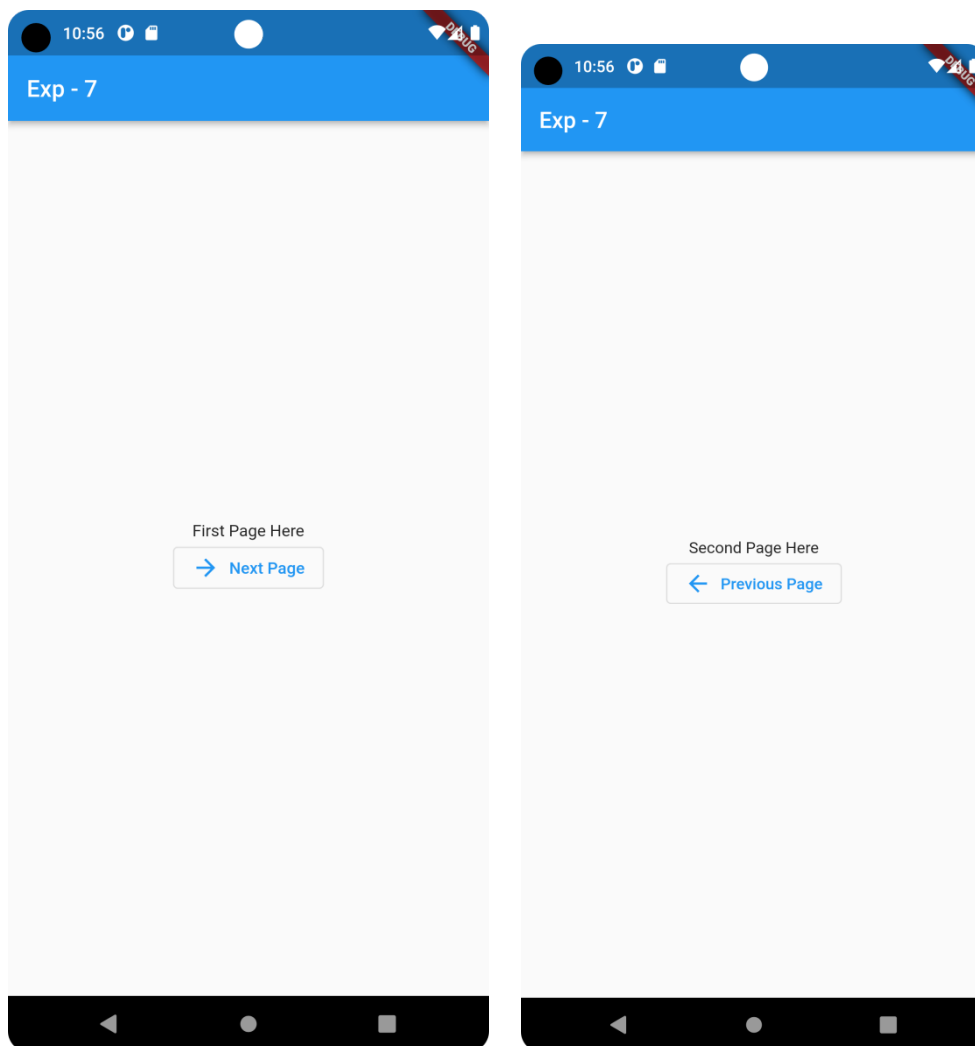
```

```

        OutlinedButton.icon(
          onPressed: () {
            Navigator.push(context, MaterialPageRoute(builder:
(context)=> FirstPage()));
          },
          label: Text('Previous Page'), icon: Icon(Icons.arrow_back),)
        ],
      ),
    ),
  ),
),
);
}
}

```

Output :



**Conclusion :** Therefore we have successfully implemented navigation, routing and gestures in Flutter Application.