Experiment 5: Applying Navigation, Routing, and Gestures in a Flutter App

Aim:

To implement navigation, routing, and gestures in a Flutter application to enhance user experience and interactivity.

Theory:

Navigation and Routing

Navigation in Flutter enables users to move between different screens (routes) within an application. There are two main approaches:

1. Basic Navigation

Navigator.push(): Moves to a new screen.

Navigator.pop(): Returns to the previous screen.

2. Named Routes Navigation

Routes are defined in the MaterialApp widget.

Navigator.pushNamed(): Navigates to a named route.

Navigator.pop(): Returns to the previous screen.

Code:

Basic Navigation

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

void main() {
  runApp(const MaterialApp(
    title: 'Navigation Basics',
    home: FirstRoute(),
  ));
}
```

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('First Route')),
      body: Center(
        child: ElevatedButton(
          child: const Text('Open route'),
          onPressed: () {
            Navigator.push(
              context,
              MaterialPageRoute(builder: (context) => const SecondRoute()),
            );
          },
        ),
      ),
    );
  }
}

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Second Route')),
      body: Center(
        child: ElevatedButton(
          onPressed: () {
            Navigator.pop(context);
          },
          child: const Text('Go back!'),
        ),
      ),
    );
  }
}
```

Named Routes Navigation Example

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

```
void main() {
  runApp(MaterialApp(
    title: 'Named Route Navigation',
    initialRoute: '/',
    routes: {
      '/': (context) => HomeScreen(),
      '/second': (context) => SecondScreen(),
    },
  ));
}

class HomeScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Home Screen')),
      body: Center(
        child: ElevatedButton(
          child: Text('Click Here'),
          onPressed: () {
            Navigator.pushNamed(context, '/second');
          },
        ),
      ),
    );
  }
}

class SecondScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Second Screen")),
      body: Center(
        child: ElevatedButton(
          onPressed: () {
            Navigator.pop(context);
          },
          child: Text('Go back!'),
        ),
      ),
    );
  }
```

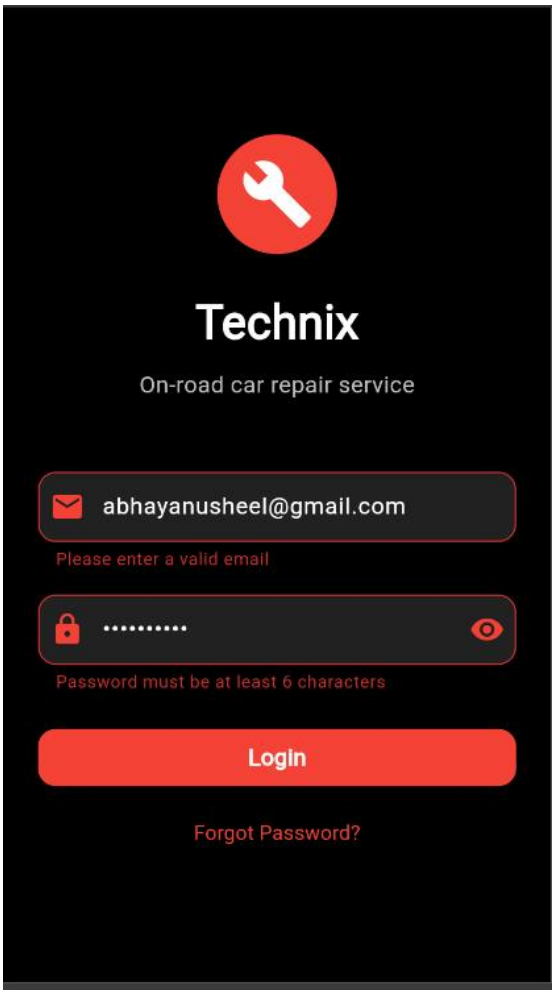
```
}
```

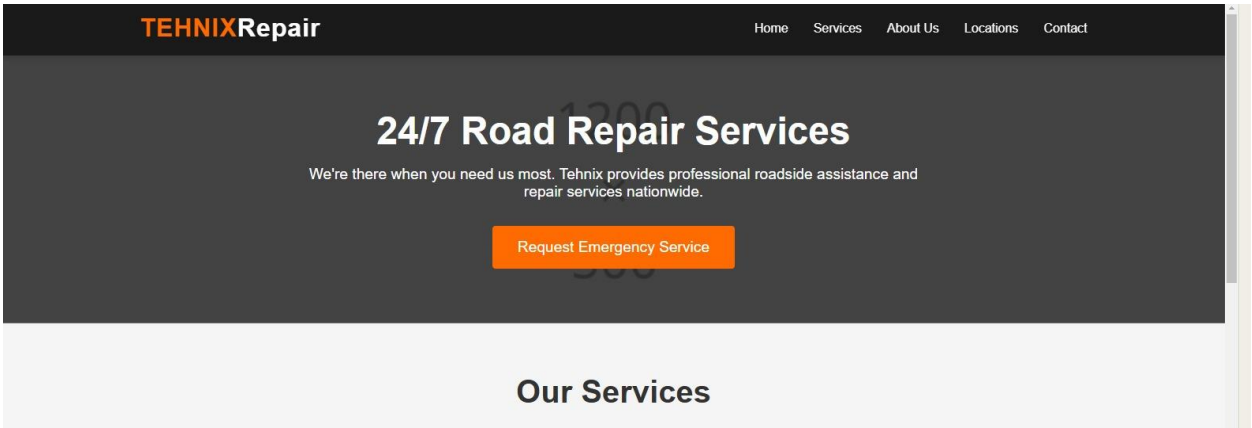
Gesture Handling Example

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

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Gesture Example',
      home: MyHomePage(),
    );
  }
}

class MyHomePage extends StatefulWidget {
  @override
  MyHomePageState createState() => new MyHomePageState();
}

class MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Gestures Example')),
      body: Center(
        child: GestureDetector(
          onTap: () {
            print('Box Clicked');
          },
          child: Container(
            height: 60.0,
            width: 120.0,
            decoration: BoxDecoration(
              color: Colors.blueGrey,
              borderRadius: BorderRadius.circular(15.0),
            ),
            child: Center(child: Text('Click Me')),
          ),
        ),
      ),
```

```
    );
  }
}
```

Output:


Technix
On-road car repair service
abhayanusheel@gmail.com
Please enter a valid email
Password must be at least 6 characters
Login
Forgot Password?



Conclusion:

In this experiment, we successfully implemented navigation, routing, and gestures in a Flutter application. We explored both basic and named navigation techniques and used GestureDetector to detect user interactions. These functionalities enhance app usability by providing seamless screen transitions and interactive elements.