Experiment No 06

Aim: create application that illustrate explain routing, passing data using navigator and navigation rail

Code:

Routing

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

class homePage extends StatelessWidget {
  const homePage();
  @override
  Widget build(BuildContext context) {
    return Scaffold(
        body: Center(
      child: ElevatedButton(
        child: Text("click on the button to see next page"),
        onPressed: () {
          Navigator.of(context).pushNamed(RouteGenerator.randompage);
        },
      ),
    ));
  }
}

void main() {
  runApp(const RandomApp());
}

class RandomApp extends StatelessWidget {
  const RandomApp();
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      onGenerateTitle: (_) => "Random App",
      initialRoute: RouteGenerator.homepage,
      onGenerateRoute: RouteGenerator.generateRoute,
    );
  }
}

class randomPage extends StatelessWidget {
  const randomPage();
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Text("${Random().nextInt(50)}"),
      ),
    );
  }
}

class RouteGenerator {
  static const String homepage = '/';
  static const String randompage = '/random';
  RouteGenerator._() {}

  static Route<dynamic> generateRoute(RouteSettings settings) {
    switch (settings.name) {
      case homepage:
        {
          return MaterialPageRoute(builder: (_) => const homePage());
        }
      case randompage:
        {
          return MaterialPageRoute(builder: (_) => const randomPage());
        }
      default:
        {
          throw FormatException("Route not found");
        }
    }
  }
}

class RouteException implements Exception {
  final String message;
  const RouteException(this.message);
}
```

Passing Data using Navigator

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

void main() {
  runApp(const Myapp());
}

class Details_of_List extends StatelessWidget {
  final Todo item;
  const Details_of_List(this.item);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(title: Text("Aakash Dhotre")),
        body: Center(
          child: Text(
         "${item.title}\n${item.description}",
         textScaleFactor: 4,
        )));
   ;
  }
}

class HomePage extends StatelessWidget {
  HomePage();
  final List<Todo> todolist = [
    const Todo("title 1", "this is the title 1"),
    const Todo("title 2", "this is the title 2"),
    const Todo("title 3", "this is the title 3"),
    const Todo("title 4", "this is the title 4")
  ];
  void _itemPressed(BuildContext context, Todo item) {
    Navigator.of(context)
        .push(MaterialPageRoute(builder: (_) => Details_of_List(item)));
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(title: Text("Aakash Dhotre")),
        body: ListView.builder(
          itemCount: todolist.length,
          itemBuilder: (context, value) {
            return ListTile(
              title: Text(todolist[value].title, textScaleFactor: 2),
              subtitle: Text(todolist[value].description, textScaleFactor: 2),
              onTap: () {
                _itemPressed(context,
                    Todo(todolist[value].title, todolist[value].description));
              },
            );
          },
        ));
  }
}

class Myapp extends StatelessWidget {
  const Myapp();

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      onGenerateTitle: (_) => 'Passing data with Navigator',
      initialRoute: RouteGenerator.homepage,
      onGenerateRoute: RouteGenerator.generateRoute,
    );
  }
}

class Todo {
  final String title;
  final String description;
  const Todo(this.title, this.description);
}

class RouteGenerator {
  static const String homepage = '/';
  static const String details_of_list = '/details_of_list';
  RouteGenerator._() {}
  static Route<dynamic> generateRoute(RouteSettings settings) {
    switch (settings.name) {
      case homepage:
        {
          return MaterialPageRoute(builder: (_) => HomePage());
        }

      default:
        {
          throw FormatException("Route not found");
        }
    }
  }
}

class RouteException implements Exception {
  final String message;
  const RouteException(this.message);
}
```

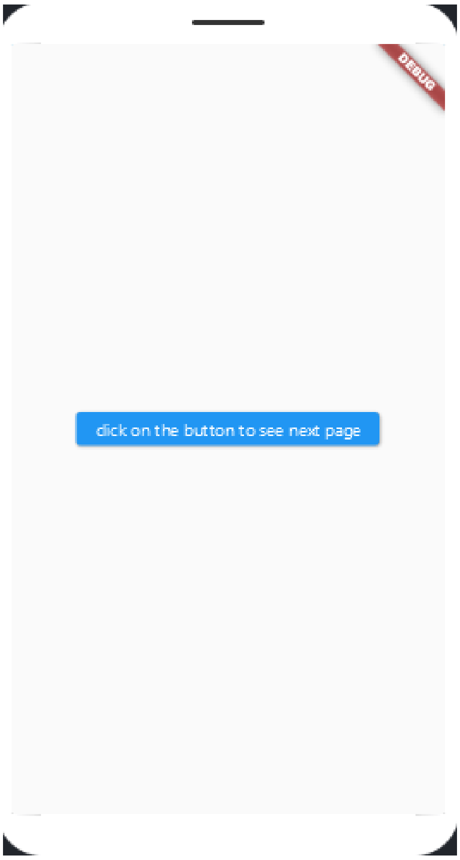
Navigation Rail

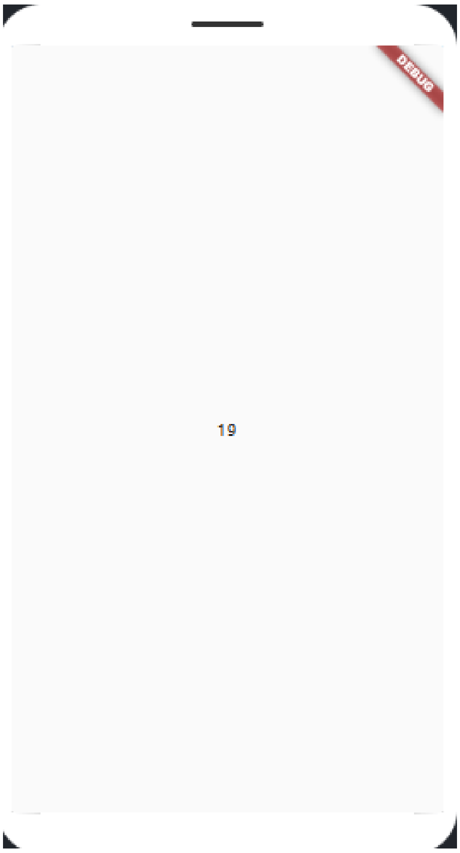
```
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';

void main() {
  runApp(ChangeNotifierProvider<IndexProvider>(
      create: (_) => IndexProvider(), child: MyApp()));
}

class MyApp extends StatelessWidget {
  @override
```

```
 Widget build(BuildContext context) {
  return MaterialApp(
     home: Scaffold(
        appBar: AppBar(title: Text("BottomNavigation Bar")),
        bottomNavigationBar: Buttomnavigation()));
 }
}

class Buttomnavigation extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
  return Consumer<IndexProvider>(builder: (context, IP, _) {
   return BottomNavigationBar(
    currentIndex: IP.index,
    selectedItemColor: Colors.green,
    onTap: (int itemnumber) {
     IP.set_index(itemnumber);
    },
    items: [
     BottomNavigationBarItem(
        icon: Icon(
         Icons.home,
         color: Colors.red,
        ),
        label: "Home"),
     BottomNavigationBarItem(
        icon: Icon(Icons.mail, color: Colors.red), label: "Mail"),
     BottomNavigationBarItem(
        icon: Icon(Icons.settings, color: Colors.red), label: "Setting"),
     BottomNavigationBarItem(
        icon: Icon(Icons.contacts, color: Colors.red),
        label: "Contact Us"),
    ],
   );
  });
 }
}

class IndexProvider with ChangeNotifier {
 int index = 1;
 void set_index(int value) {
  index = value;
  notifyListeners();
  debugPrint("value is updated $value");
 }

 int get_index() {
  return index;
  }
}
```

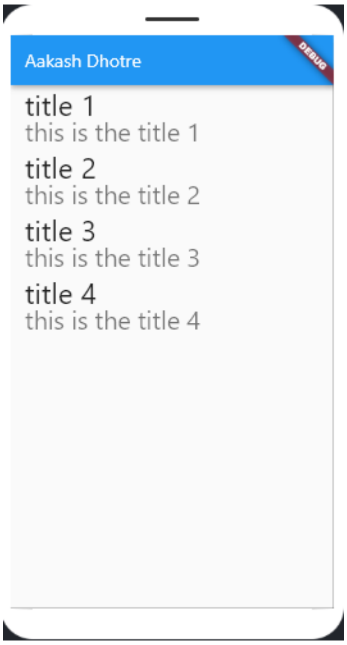
Output:

Routing

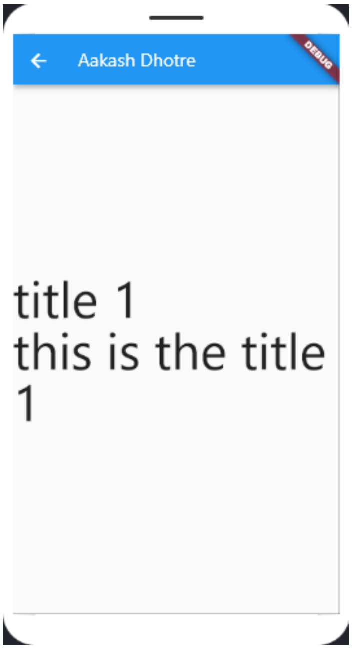




Passing Data using Navigator

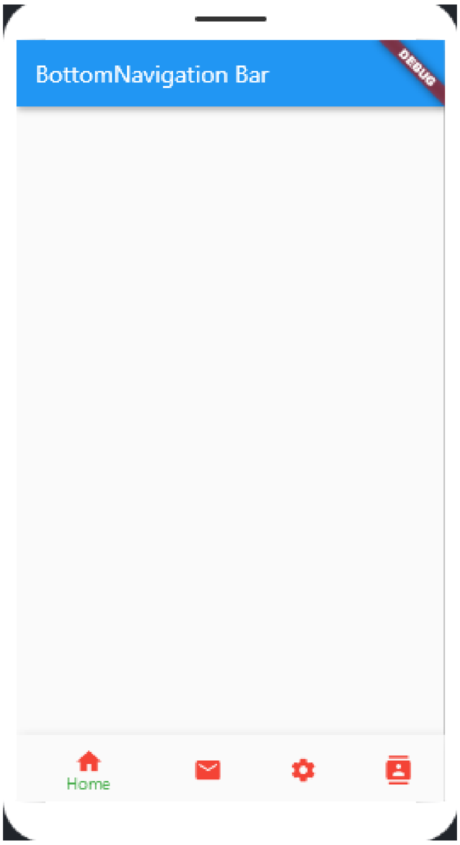


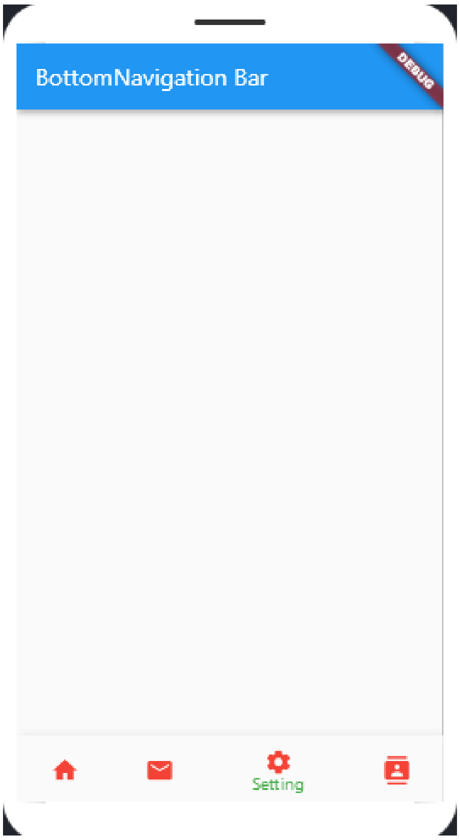
title 1
this is the title 1
title 2
this is the title 2
title 3
this is the title 3
title 4
this is the title 4



title 1
this is the title 1

Navigation Rail


BottomNavigation Bar
DEBUG
Home


BottomNavigation Bar
DEBUG
Setting