EXPERIMENT – 4

AIM:

a) Setup navigation between different screens using navigator.

b) Implement navigation with named routes.

Experiment - 4(a)

AIM: Setup navigation between different screens using navigator.

DESCRIPTION:

In Flutter, Navigator.push() is a method used for navigating to a new screen or "route" within an application. It works by pushing a new Route onto the navigator's stack, effectively placing the new screen on top of the current one. This creates a visual transition to the new screen, and the previous screen remains in the stack, allowing for easy navigation back.

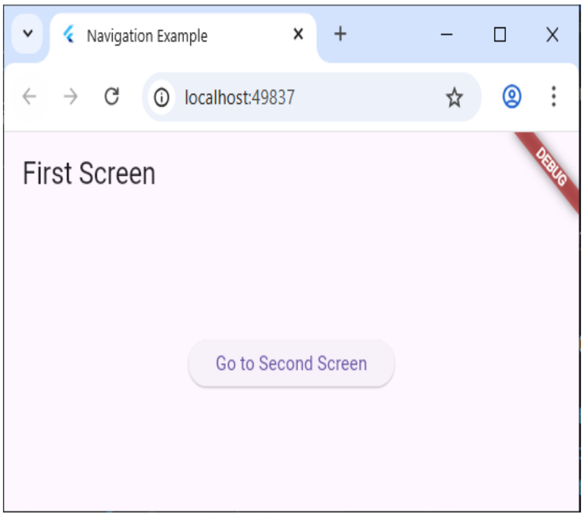
- Navigator.push() is fundamental for implementing navigation flows in Flutter applications.
- It utilizes a stack-based approach to manage the order of screens.
- MaterialPageRoute is frequently used with Navigator.push() to create routes for new screens.
- To return from a pushed screen, Navigator.pop() is used.
- To return to first screen from the current screen

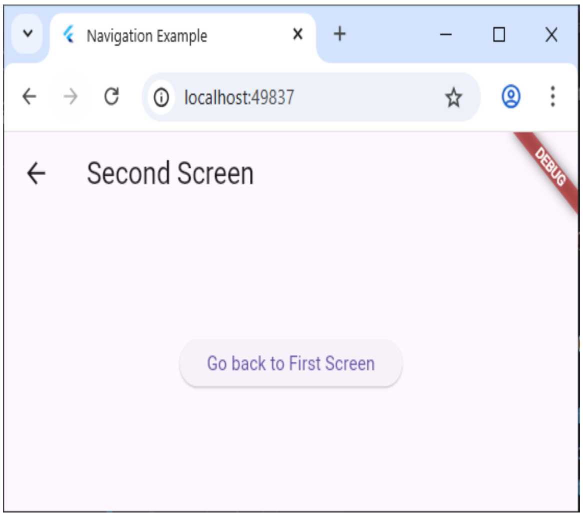
  Navigator.popUntil(context, ModalRoute.withName('/'));

SOLUTION:

```
import 'package:flutter/material.dart';
void main()
{
  runApp(MyApp());
}
class MyApp extends StatelessWidget
{
  @override
  Widget build(BuildContext context)
  {
    return MaterialApp(
      title: 'Navigation Example',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: FirstScreen(),
    );
  }
}
class FirstScreen extends StatelessWidget
{
  @override
  Widget build(BuildContext context)
  {
    return Scaffold(
      appBar: AppBar(
        title: Text('First Screen'),
      ),
      body: Center(
        child: ElevatedButton(
          onPressed: ()
          {
            // Navigate to the second screen
            Navigator.push(
              context,
              MaterialPageRoute(builder: (context) => SecondScreen()),
            );
            },
          child: Text('Go to Second Screen'),
        ),
      ),
    );
  }
}
class SecondScreen extends StatelessWidget
{
  @override
  Widget build(BuildContext context)
  {
    return Scaffold(
    appBar: AppBar(
    title: Text('Second Screen'),
    ),
    body: Center(
      child: ElevatedButton( onPressed: () {
      // Navigate back to the first screen
        Navigator.pop(context);
        },
        child: Text('Go back to First Screen'),
      ),
    ),
  );
  }
}
```

OUTPUT:





Experiment - 4(b)

AIM: Implement navigation with named routes.

DESCRIPTION:

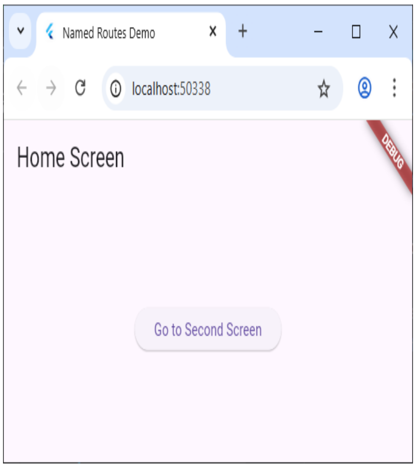
Navigator.pushNamed( ) in Flutter is a method used for navigating to a new screen (route) by its registered name. This approach offers advantages over Navigator.push( ) when dealing with multiple routes, as it promotes better organization and maintainability.

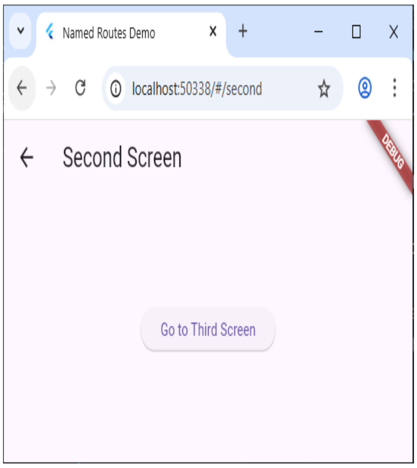
To use pushNamed, the named routes must be declared in the routes property of the MaterialApp widget. This routes property is a map where keys are the route names (strings) and values are builder functions that return the widget associated with that route.

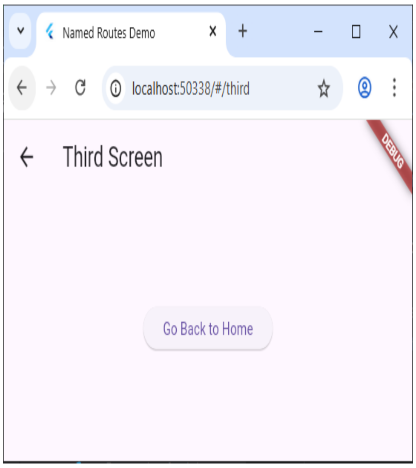
SOLUTION:

```
import 'package:flutter/material.dart';
void main()
{
  runApp(MyApp());
}
class MyApp extends StatelessWidget
{
  @override
  Widget build(BuildContext context)
  {
    return MaterialApp(
      title: 'Named Routes Demo',
      initialRoute: '/',
      routes: {
        '/': (context) => HomeScreen(),
        '/second': (context) => SecondScreen(),
        '/third': (context) => ThirdScreen(),
      },
    );
  }
}
class HomeScreen extends StatelessWidget
{
  @override
  Widget build(BuildContext context)
  {
    return
      Scaffold(
        appBar: AppBar(
          title: Text('Home Screen'),
        ),
        body: Center(
          child: ElevatedButton( onPressed: () {
            Navigator.pushNamed(context, '/second');
            },
            child: Text('Go to Second Screen'),
          ),
        ),
      );
  }
}
class SecondScreen extends StatelessWidget
{
 @override
 Widget build(BuildContext context)
 {
  return Scaffold(
    appBar: AppBar(
      title: Text('Second Screen'),
    ),
    body: Center(
      child: ElevatedButton( onPressed: () {
      Navigator.pushNamed(context, '/third');
    },
      child: Text('Go to Third Screen'),
    ),
   ),
 );
}
}
class ThirdScreen extends StatelessWidget
{
 @override
 Widget build(BuildContext context)
 {
  return Scaffold(
    appBar: AppBar(
    title: Text('Third Screen'),
    ),
    body: Center(
      child: ElevatedButton( onPressed: () {
      Navigator.popUntil(context, ModalRoute.withName('/'));
      },
        child: Text('Go Back to Home'),
      ),
    ),
   );
 }
}
```

OUTPUT:


Named Routes Demo
localhost:50338
Home Screen
DEBUG
Go to Second Screen


Named Routes Demo
localhost:50338/#/second
Second Screen
DEBUG
Go to Third Screen


Named Routes Demo
localhost:50338/#/third
Third Screen
DEBUG
Go Back to Home