

WEEK-4

AIM: a) To set up navigation between different screens using Navigator.

DESCRIPTION:

This is a basic way to move from one screen to another in a Flutter app. You use the Navigator class to push (go to) or pop (go back from) screens.

`Navigator.push()` is used to go to a new screen.

`Navigator.pop()` is used to return to the previous screen.

This method is useful for small apps or when you don't have many screens.

You can also pass data between screens easily using constructors.

Each screen transition is managed like a stack (Last-In-First-Out).

SOURCE CODE:

```
import 'package:flutter/material.dart'; void main() =>  
runApp( MaterialApp( home: Scaffold( appBar:  
AppBar(title: const Text('First Screen')), body:  
Builder(
```

```
builder: (context) {  
    return Center(  
        child: ElevatedButton( child: const  
            Text('Go to Second Screen'),  
            onPressed: () {  
                Navigator.push(  
                    context,  
                    MaterialPageRoute<void>( builder: (context) => Scaffold(  
                        appBar: AppBar(title: const Text('Second Screen'))
```

```
body: Center(  
    child: ElevatedButton(  
        child: const Text('Go Back'),  
        onPressed: () => Navigator.pop(context),  
    ),
```

),

),

);

},

)₂

);

3

),

);

OUTPUT:

First Screen

[Go to Second Screen](#)

DEBUG

← Second Screen

DEBUG

SITY
ge (A))

[Go Back](#)

AIM: b) To implement navigation with named routes.

DESCRIPTION:

This method involves assigning a name (like /home, /profile) to each screen and using those names to navigate.

All screen routes are defined in one place (usually in the main app).

`Navigator.pushNamed()` is used to go to a screen by its name.

`Navigator.pop()` is still used to go back.

Named routes help in keeping the code organized, especially in larger apps.

It's cleaner and more manageable when you have many screens.

You can still pass data with additional setup (e.g., using `onGenerateRoute`).

SOURCE CODE - 1:

```
import 'package:flutter/material.dart';
void main() => runApp(
  MaterialApp(
    initialRoute: '/',
    routes: {
      '/': (context) => Scaffold( appBar: AppBar(title: const
        Text('Home Screen')), body: Center(
          child: Column(
            mainAxisSize: MainAxisSize.min,
            children: [
              ElevatedButton( child: const
                Text('Go to Screen 1'), onPressed:
                  () {
                    Navigator.pushNamed(context, '/screen1');
                  },
                ),
              const SizedBox(height: 20),
              ElevatedButton( child: const
                Text('Go to Screen 2'), onPressed:
                  () {
                    Navigator.pushNamed(context, '/screen2');
                  },
                ),
              ],
            ),
          ),
        ),
      '/screen1': (context) => Scaffold( appBar:
        AppBar(title: const Text('Screen 1')), body:
        Center(
          child: ElevatedButton(
            child: const Text('Go back to home'),
            onPressed: () {
              Navigator.pop(context);
            },
          ),
        ),
      ),
    },
  ),
);
```

```
        ),  
        ),  
        '/screen2': (context) => Scaffold( appBar:  
          AppBar(title: const Text('Screen 2')), body:  
          Center(  
            child: ElevatedButton(  
              child: const Text('Go back to Home'),  
              onPressed: () {  
                Navigator.pop(context);  
              },  
            ),  
            ),  
            ),  
            ),  
            ),  
            ),  
            ),  
            );  
        );
```

OUTPUT -1:

