

Duhok Polytechnic University

Technical College of Zakho

Department of CIS



LAB 3:

In-App Navigation

MOBILE APPLICATION II

Objectives

Imperative Navigation & Screen Stack

Named Routes & Centralized Route Table

Data Passing

Returning Values

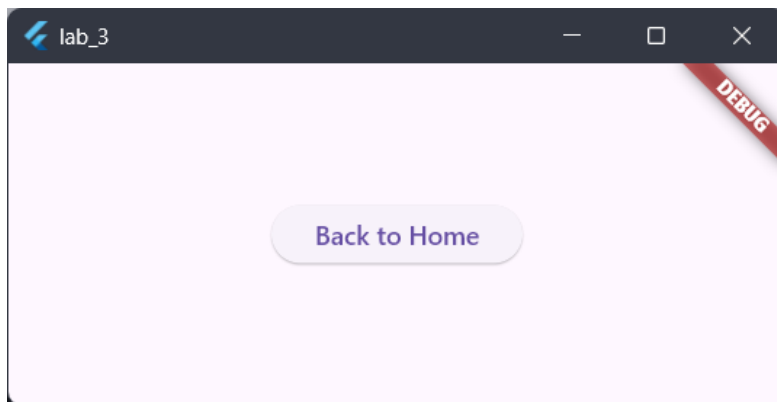
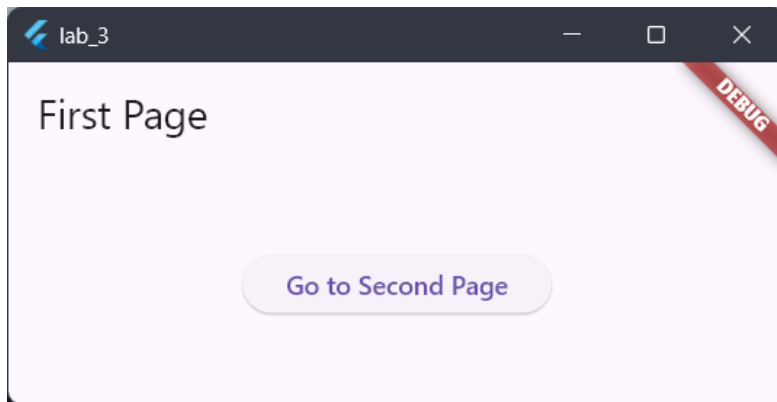
Imperative Navigation & Screen Stack

Mobile applications organize user interfaces into a stack of screens. Flutter's Navigator manages this Last-In, First-Out (LIFO) stack by pushing new screens on top when navigating forward and popping them off when going back.

Key Methods & Classes:

- `Navigator.push()`: Pushes a new route onto the Navigator stack using a `MaterialPageRoute` transition.
- `Navigator.pop()`: Removes the top route from the stack, revealing the previous screen underneath.
- `MaterialPageRoute`: A modal route that replaces the entire screen with a platform-adaptive transition animation (slides up on iOS, fades/slides on Android).
- `BuildContext (context)`: Provides the location anchor in the widget tree so Navigator can locate the nearest `NavigatorState` ancestor.

Imperative Navigation Example



```
1 class FirstPage extends StatelessWidget {
2   @override
3   Widget build(BuildContext context) => Scaffold(
4     appBar: AppBar(title: const Text('First Page')),
5     body: Center(
6       child: ElevatedButton(
7         onPressed: () {
8           Navigator.push(
9             context,
10            MaterialPageRoute(builder: (context) => SecondPage()),
11          );
12        },
13        child: const Text('Go to Second Page'),
14      ),
15    ),
16  );
17 }
18
19 class SecondPage extends StatelessWidget {
20   @override
21   Widget build(BuildContext context) => Scaffold(
22     body: Center(
23       child: ElevatedButton(
24         onPressed: () {
25           Navigator.pop(context);
26         },
27         child: const Text('Back to Home'),
28       ),
29     ),
30   );
31 }
```

Named Routes & Centralized Configuration

For larger applications, defining inline routes becomes hard to maintain. Named routes centralize screen paths as string identifiers inside the MaterialApp widget, decoupling screen creation from route navigation.

Key Attributes & Methods:

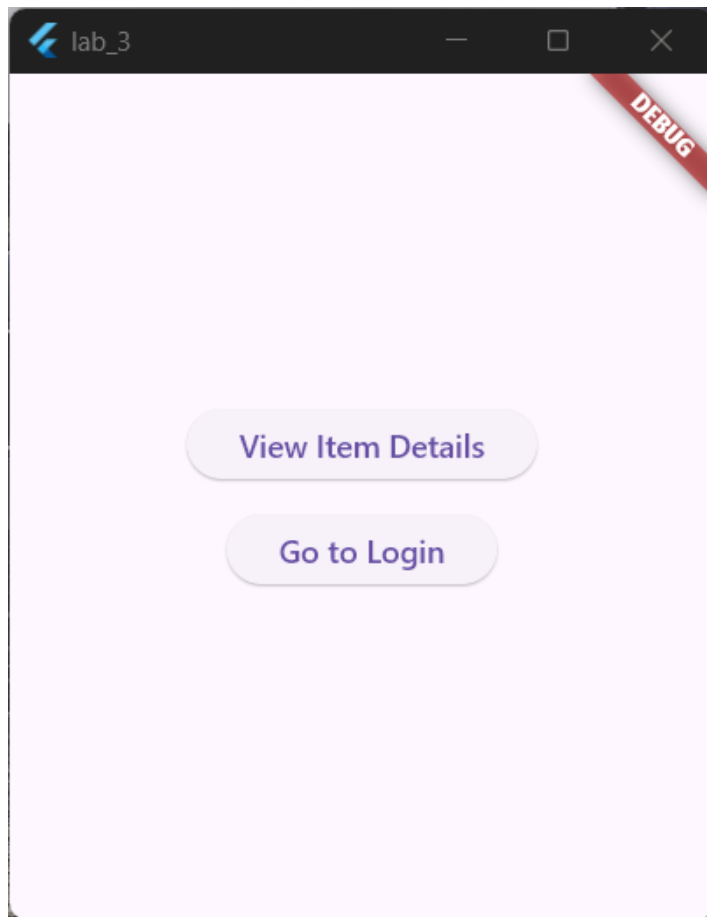
- `initialRoute`: Sets the starting route string (e.g., '/') launched when the application starts up.
- `routes`: A `Map<String, WidgetBuilder>` linking string path names (like '/details') to builder functions returning screen widgets.
- `Navigator.pushNamed()`: Navigates to a specific screen by referencing its string identifier in the route table.
- `Navigator.pushReplacementNamed()`: Pushes a new named route and disposes of the current route so the user cannot navigate back (e.g., Login to Dashboard).

Named Routes Example



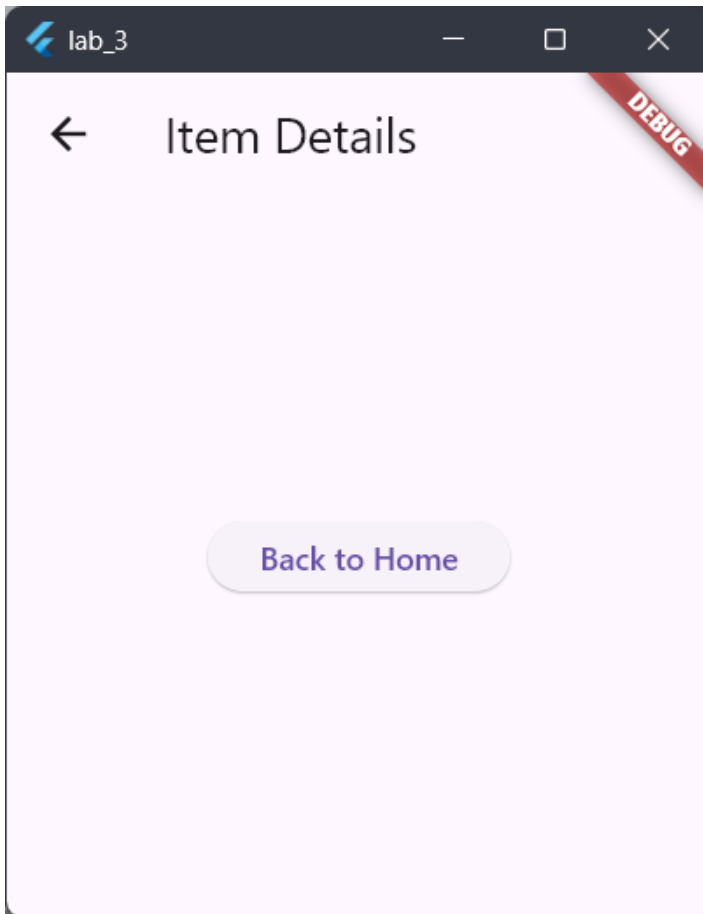
```
1 // main.dart
2 import 'package:flutter/material.dart';
3 import 'pages/details_screen.dart';
4 import 'pages/home_screen.dart';
5 import 'pages/login_screen.dart';
6
7 void main() {
8   runApp(MainApp());
9 }
10
11 class MainApp extends StatelessWidget {
12   @override
13   Widget build(BuildContext context) => MaterialApp(
14     initialRoute: '/',
15     routes: {
16       '/': (context) => HomeScreen(),
17       '/details': (context) => DetailsScreen(),
18       '/login': (context) => LoginScreen(),
19     },
20   );
21 }
```

Named Routes Example (cont.)



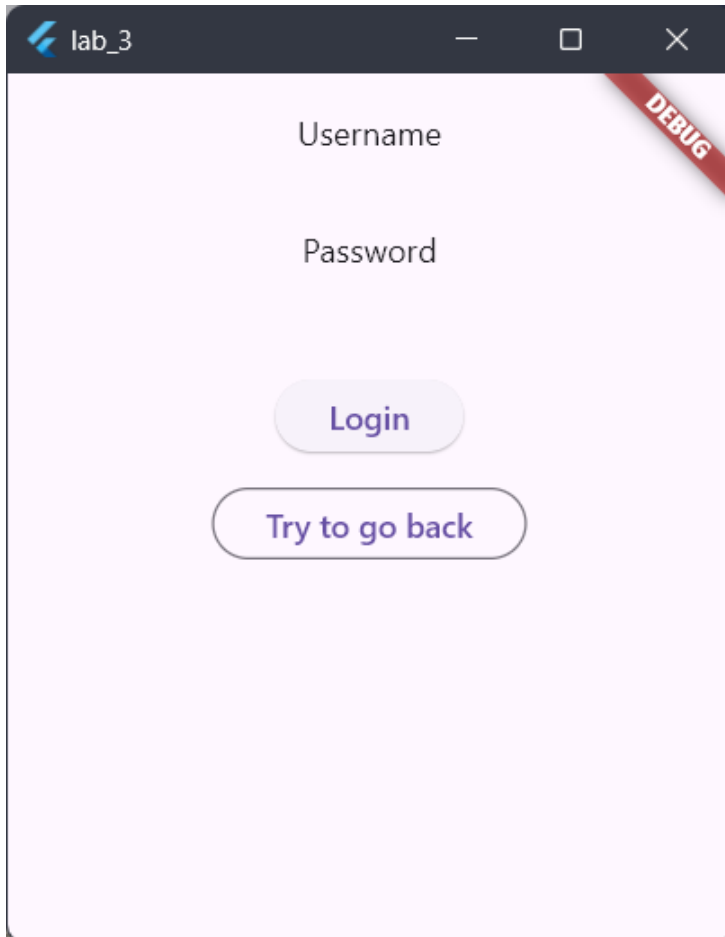
```
1 // pages/home_screen.dart
2 import 'package:flutter/material.dart';
3
4 class HomeScreen extends StatelessWidget {
5   @override
6   Widget build(BuildContext context) => Scaffold(
7     body: Center(
8       child: Column(
9         mainAxisAlignment: MainAxisAlignment.center,
10        children: [
11          ElevatedButton(
12            onPressed: () {
13              Navigator.pushNamed(context, '/details');
14            },
15            child: Text('View Item Details'),
16          ),
17          SizedBox(height: 16.0),
18          ElevatedButton(
19            onPressed: () {
20              Navigator.pushReplacementNamed(context, '/login');
21            },
22            child: Text('Go to Login'),
23          ),
24        ],
25      ),
26    ),
27  );
28 }
```

Named Routes Example (cont.)



```
1 // pages/details_screen.dart
2 import 'package:flutter/material.dart';
3
4 class DetailsScreen extends StatelessWidget {
5   @override
6   Widget build(BuildContext context) => Scaffold(
7     appBar: AppBar(
8       leading: BackButton(),
9       title: Text('Item Details'),
10    ),
11    body: Center(
12      child: ElevatedButton(
13        onPressed: () {
14          Navigator.pop(context);
15        },
16        child: Text('Back to Home'),
17      ),
18    ),
19  );
20 }
```

Named Routes Example (cont.)



```
1 // pages/login_screen.dart
2 import 'package:flutter/material.dart';
3
4 class LoginScreen extends StatelessWidget {
5   @override
6   Widget build(BuildContext context) => Scaffold(
7     body: Center(
8       child: Column(
9         children: [
10           Container(
11             padding: EdgeInsets.all(16.0),
12             child: Text('Username'),
13           ),
14           Container(
15             padding: EdgeInsets.all(16.0),
16             child: Text('Password'),
17           ),
18           SizedBox(height: 32.0),
19           ElevatedButton(
20             onPressed: () {
21               Navigator.pushReplacementNamed(context, '/');
22             },
23             child: Text('Login'),
24           ),
25           SizedBox(height: 16.0),
26           OutlinedButton(
27             onPressed: () {
28               Navigator.pop(context);
29             },
30             child: Text('Try to go back'),
31           ),
32         ],
33       ),
34     ),
35   );
36 }
```

Data Passing & Return Values

Screens rarely exist in isolation. Flutter allows passing data forward when pushing a new route, as well as returning values back asynchronously when a route is popped.

Key Techniques & Methods:

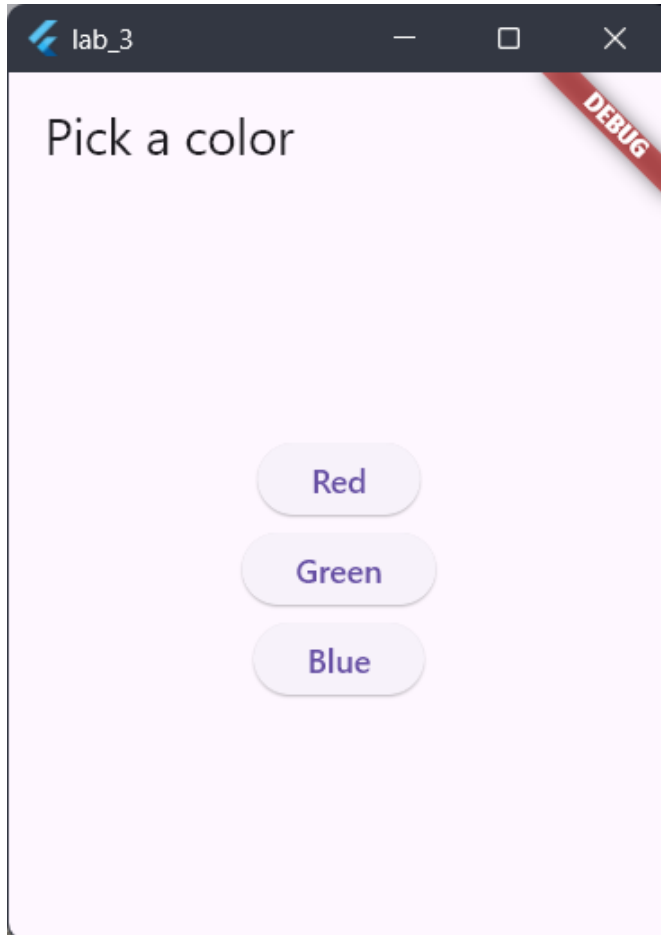
- Passing Arguments Forward: Pass data directly via class constructor parameters or using arguments parameter in `Navigator.pushNamed()`.
- Extracting Modal Arguments: Access route settings on the target screen using `ModalRoute.of(context)!.settings.arguments`.
- Returning Data on Pop: Pass result data back as a parameter in `Navigator.pop(context, resultValue)`.
- Awaiting Return Values: Mark the navigation callback `async` and await `Navigator.push()` to handle the result when the pushed screen closes.

Data Passing Example



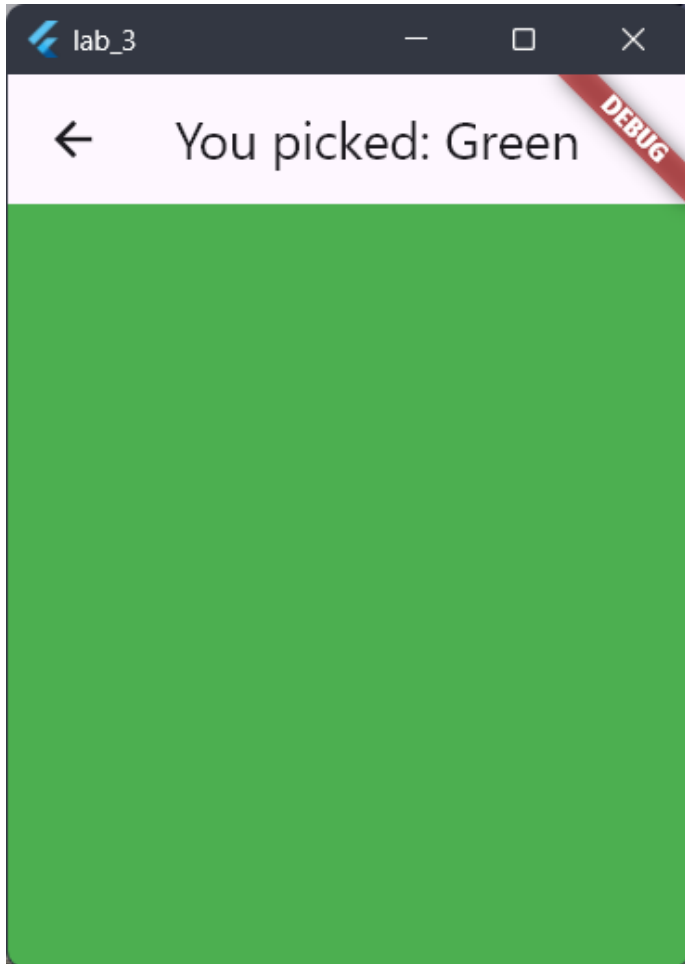
```
1 // main.dart
2 import 'package:flutter/material.dart';
3 import 'pages/color_page.dart';
4 import 'pages/pick_page.dart';
5
6 void main() {
7   runApp(MainApp());
8 }
9
10 class MainApp extends StatelessWidget {
11   @override
12   Widget build(BuildContext context) => MaterialApp(
13     initialRoute: '/',
14     routes: {
15       '/': (context) => PickPage(),
16       '/color': (context) => ColorPage(),
17     },
18   );
19 }
```

Data Passing Example (cont.)



```
1 // pages/pick_page.dart
2 import 'package:flutter/material.dart';
3
4 class PickPage extends StatelessWidget {
5   @override
6   Widget build(BuildContext context) => Scaffold(
7     appBar: AppBar(title: Text('Pick a color')),
8     body: Center(
9       child: Column(
10        mainAxisAlignment: MainAxisAlignment.center,
11        children: [
12          ElevatedButton(
13            onPressed: () {
14              Navigator.pushNamed(
15                context,
16                '/color',
17                arguments: {'name': 'Red', 'color': Colors.red},
18              );
19            },
20            child: Text('Red'),
21          ),
22          SizedBox(height: 8.0),
23          // Add more buttons for other colors
24        ],
25      ),
26    ),
27  );
28 }
```

Data Passing Example (cont.)



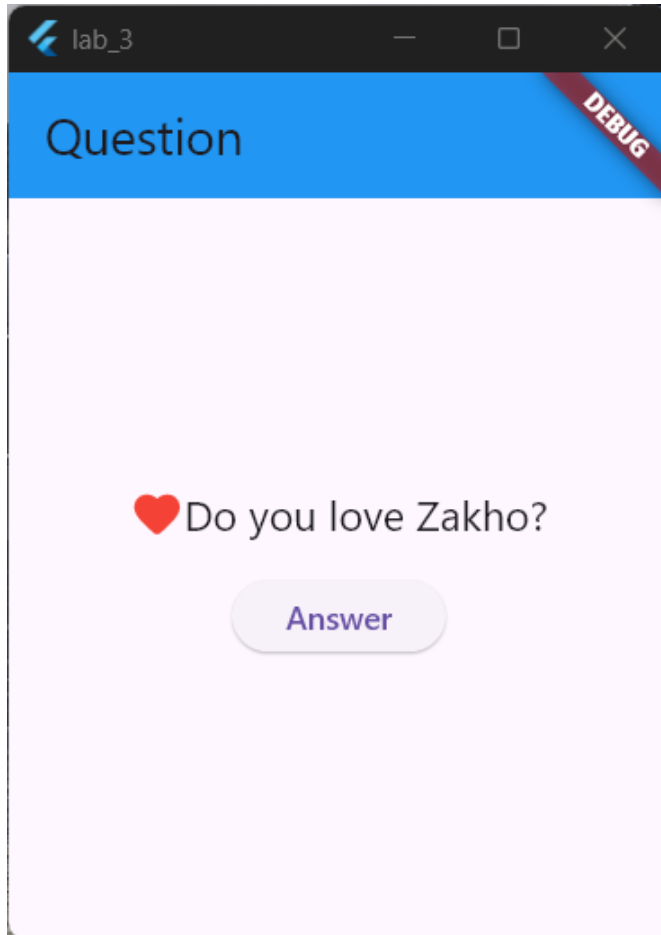
```
1 // pages/color_page.dart
2 import 'package:flutter/material.dart';
3
4 class ColorPage extends StatelessWidget {
5   @override
6   Widget build(BuildContext context) {
7     final pick =
8       ModalRoute.of(context)!.settings.arguments as Map<String, dynamic>;
9     return Scaffold(
10       appBar: AppBar(title: Text("You picked: ${pick['name']}")),
11       backgroundColor: pick['color'],
12     );
13   }
14 }
```

Return Example



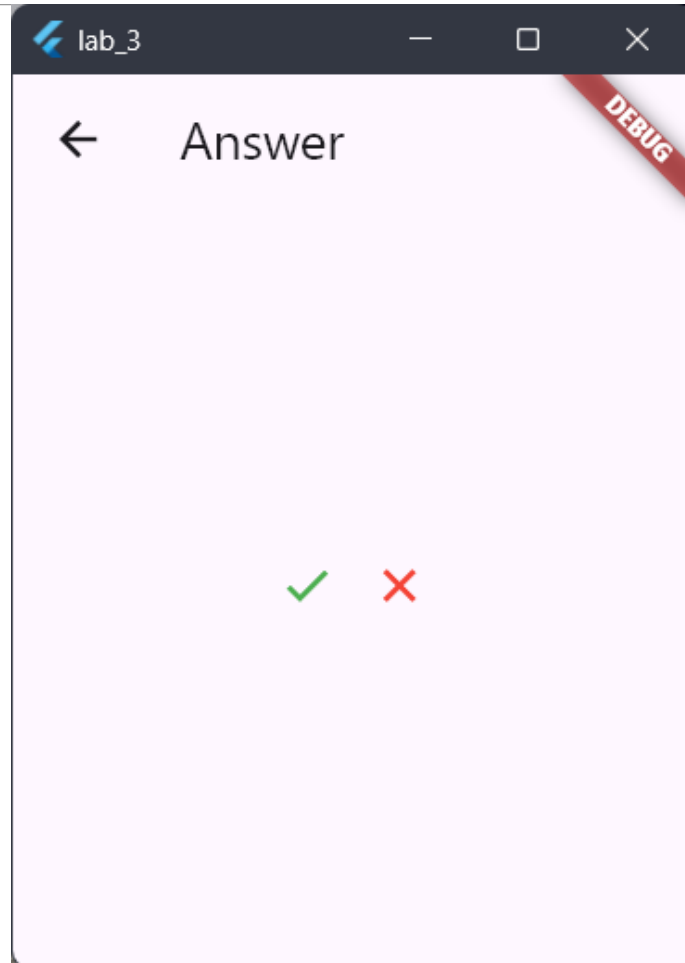
```
1 // main.dart
2 import 'package:flutter/material.dart';
3 import 'pages/question_page.dart';
4 import 'pages/answer_page.dart';
5
6 void main() {
7   runApp(MainApp());
8 }
9
10 class MainApp extends StatelessWidget {
11   @override
12   Widget build(BuildContext context) => MaterialApp(
13     initialRoute: '/question',
14     routes: {
15       '/question': (context) => QuestionPage(),
16       '/answer': (context) => AnswerPage(),
17     },
18   );
19 }
```

Return Example (cont.)



```
1 // pages/question_page.dart
2 import 'package:flutter/material.dart';
3
4 class QuestionPage extends StatelessWidget {
5   @override
6   Widget build(BuildContext context) => Scaffold(
7     appBar: AppBar(title: Text('Question'), backgroundColor: Colors.blue),
8     body: Column(
9       mainAxisAlignment: MainAxisAlignment.center,
10      crossAxisAlignment: CrossAxisAlignment.center,
11      children: [
12        Row(
13          mainAxisAlignment: MainAxisAlignment.center,
14          children: [
15            Icon(Icons.favorite_rounded, size: 24, color: Colors.red),
16            Text('Do you love Zakho?', style: TextStyle(fontSize: 18)),
17          ],
18        ),
19        SizedBox(height: 16.0),
20        ElevatedButton(
21          onPressed: () async {
22            bool answer = await Navigator.pushNamed(context, '/answer') as bool;
23            ScaffoldMessenger.of(context).showSnackBar(
24              SnackBar(
25                content: Text(answer ? 'We all do ❤️' : 'Why not? 😞'),
26                backgroundColor: answer ? Colors.green : Colors.red,
27              ),
28            );
29          },
30          child: Text('Answer'),
31        ),
32      ],
33    ),
34  );
35 }
```

Return Example (cont.)



```
1 // pages/answer_page.dart
2 import 'package:flutter/material.dart';
3
4 class AnswerPage extends StatelessWidget {
5   @override
6   Widget build(BuildContext context) => Scaffold(
7     appBar: AppBar(title: Text('Answer')),
8     body: Center(
9       child: Row(
10        mainAxisAlignment: MainAxisAlignment.center,
11        children: [
12          IconButton(
13            icon: Icon(Icons.check),
14            color: Colors.green,
15            onPressed: () {
16              Navigator.pop(context, true);
17            },
18          ),
19          IconButton(
20            icon: Icon(Icons.close),
21            color: Colors.red,
22            onPressed: () {
23              Navigator.pop(context, false);
24            },
25          ),
26        ],
27      ),
28    ),
29  );
30 }
```

Any Questions?



Lab Work

