

# Named Navigation

## Exercise #02

### Flutter Developer Bootcamp

## Purpose

This Exercise demonstrates a basic Flutter app with multiple screens and navigation between them using named routes. It's a fundamental concept in building Flutter applications, especially those with multiple screens and complex navigation flows.

## Problem

This Flutter Exercise demonstrates a basic app with multiple screens using named routes for navigation. It consists of a home page with buttons to navigate to two other pages. Each secondary page has a button to return to the home page. The navigation is handled by the `Navigator` class.

## How to Solve

1. Checkout the workshop from Git Repo:

```
git clone -b <user-branch> <repo-URL>
```

2. Open the root folder inside VS Code
3. To build the app click the run option in the main method { }
4. **HomePage Class:** Represents the home page with buttons for navigation.
5. **SecondPage Class:** Represents the notification page with a button to navigate back.
6. **ThirdPage Class:** Represents the profile page with a button to navigate back.

# You Will Achieve

This Flutter Exercise defines a simple app with three pages: Home Page, Notification Page, and Profile Page. It uses named routes for navigation. Here's a breakdown of what the app achieves:

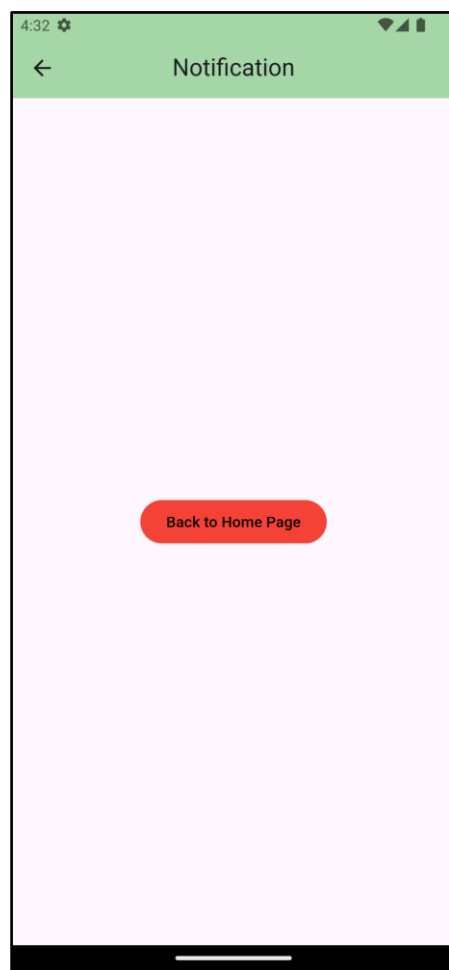
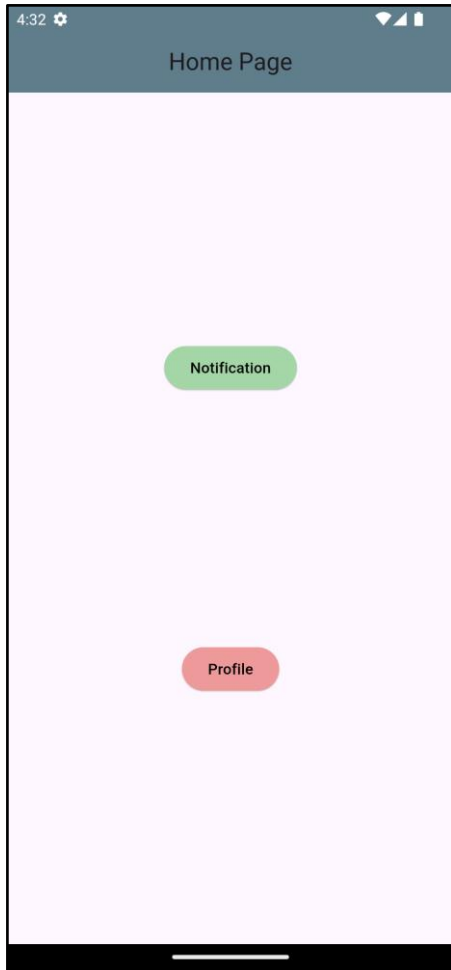
1. **Navigation:** It demonstrates how to navigate between different pages using named routes. The `initialRoute` property is set to `'/'`, which indicates that the initial route of the app is the Home Page. Then, named routes are defined for each page: `'/'` for Home Page,  `'/second'` for Notification Page, and  `'/third'` for Profile Page.
2. **UI:** Each page (`HomePage`, `SecondPage`, and `ThirdPage`) is implemented as a `StatelessWidget` with its own scaffold and UI components. The Home Page has two buttons to navigate to the Notification Page and Profile Page respectively. The Notification Page and Profile Page each have a button to navigate back to the Home Page.
3. **AppBar:** Each page has its own app bar with a title specific to that page.
4. **Button Styling:** The buttons are styled differently using the `style` property of `ElevatedButton.styleFrom`. Colours and other properties are adjusted to provide a visually appealing UI.
5. **Navigation Back:** The buttons on the Notification Page and Profile Page use `Navigator.pop(context)` to navigate back to the previous page, which is the Home Page in this case.

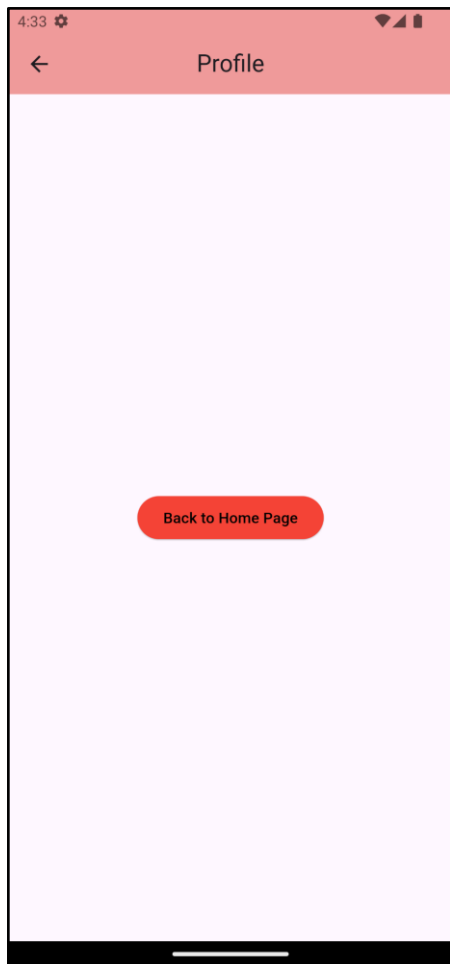
Overall, this code provides a basic structure for a Flutter app with named routes and navigation between multiple pages

○

## Screenshots

## Expected Output





## How to submit your Exercise

Push your project back to the same git branch using command:

<command name>

## Happy Coding!

