**Bài 5**

**Routing trong react native**

**(Navigation)**

Chào các bạn, ở bài trước chúng ta đã nắm được tổng quan về hook trong react native, các bạn có luyện tập ở nhà chưa? Hôm nay chúng ta sẽ học một kiến thức mới trong react native mà dự án nào cũng phải sử dụng, đó là routing hay còn gọi là điều hướng trong react native. Cùng xem những mục tiêu cần đạt trong bài hôm nay nhé:

* Nắm được khái quát về routing trong React Native
* Tạo và cấu hình các thành phần điều hướng như: Stack Navigator, Bottom Tabs, Material Top Tabs,...
* Biết cách sử dụng các phương thức điều hướng chủ yếu như: navigate, goBack,...
* Routing hook
* **Giới thiệu**

Routing là một khái niệm quan trọng trong phát triển ứng dụng di động, giúp chúng ta chuyển đổi giữa các màn hình hoặc thành phần khác nhau trong ứng dụng của chúng ta một cách dễ dàng và hiệu quả.

* **Tại sao chúng ta cần Routing trong ứng dụng React Native?**

Vì trong quá trình xây dựng ứng dụng di động, chúng ta thường có nhiều màn hình cần hiển thị cho người dùng. Ví dụ như trang chủ, trang danh sách, trang chi tiết, trang cài đặt, vân vân. Để chuyển đổi giữa các trang này, chúng ta cần có một cơ chế quản lý và điều hướng một cách hiệu quả. Đó là lúc “**Routing”** trong React Native trở nên cần thiết.

* **Cách thức hoạt động của Routing**

Routing giúp chúng ta xác định các quy tắc và luật để điều hướng giữa các màn hình trong ứng dụng của chúng ta.

Khi người dùng thực hiện một hành động như nhấn nút, thực hiện thao tác vuốt, hoặc điền thông tin vào một biểu mẫu, Routing sẽ xác định màn hình hoặc trang mới sẽ được hiển thị. Điều này giúp chúng ta tạo ra trải nghiệm người dùng mượt mà và tự nhiên.

**Đặt câu hỏi: Vậy thì làm thế nào để sử dụng routing trong dự án react native.**

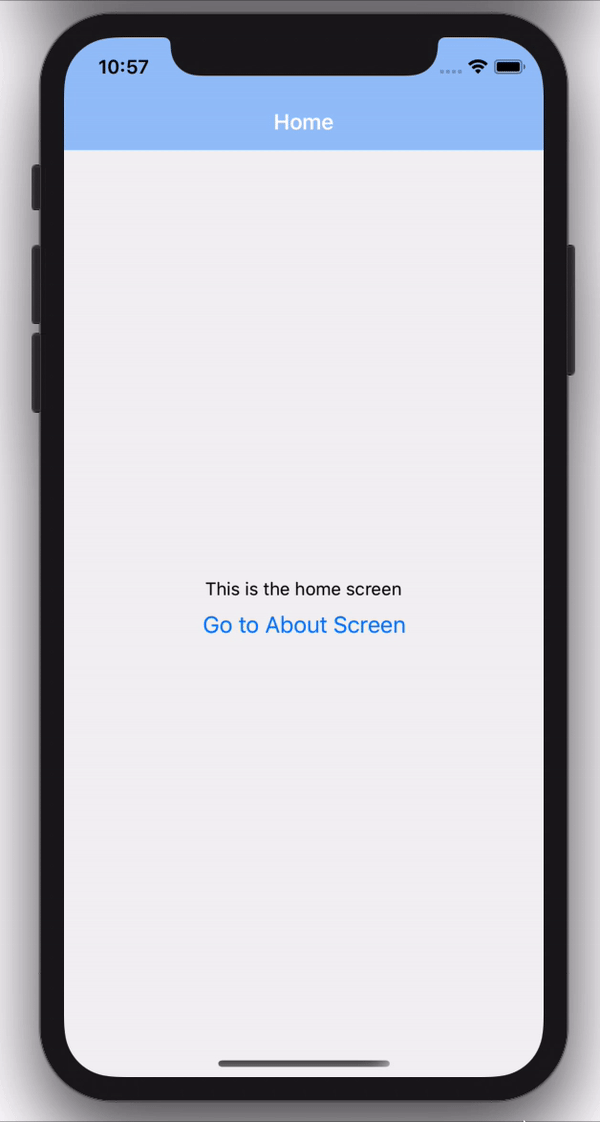
Chúng ta sẽ truy cập reactnavigation.org, đây là một thư viện hỗ trợ điều hướng thường được sử dụng trong react native.

* **Giới thiệu**

React Navigation cung cấp các loại routing phổ biến như Stack (ngăn xếp), Tab (thẻ), Drawer (hộp kéo), và nhiều hơn nữa. Các bạn hãy tìm hiểu thêm tại reactnavigation.org nhé! Còn trong bài học này, mình sẽ hướng dẫn các bạn sử dụng một số loại routing phổ biến trong react navigation sau:

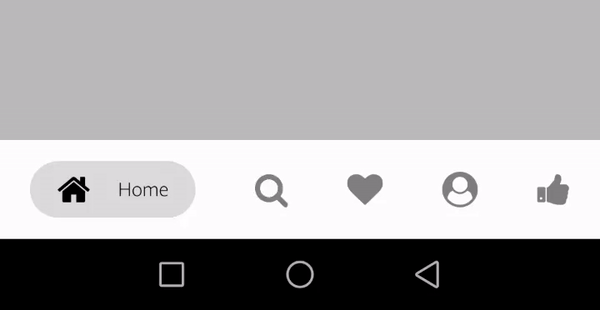
* **Stack Navigator**

Stack Navigator quản lý việc điều hướng giữa các màn hình dưới dạng ngăn xếp (stack). Mỗi khi bạn chuyển đến một màn hình mới, nó sẽ đẩy màn hình đó lên đỉnh ngăn xếp. Khi bạn quay lại, màn hình đầu tiên trong ngăn xếp sẽ được hiển thị. Điều này giúp bạn tạo các luồng điều hướng tuyến tính.



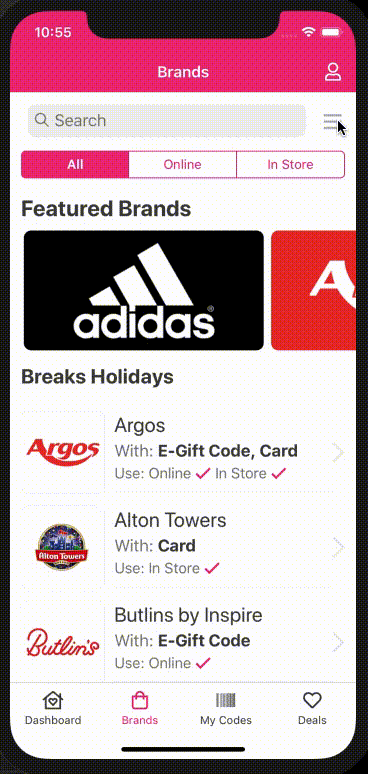
* **Tab Navigator**

Tab Navigator cho phép bạn tạo các tab điều hướng để người dùng có thể chuyển đổi giữa các màn hình khác nhau bằng cách nhấn vào các tab tương ứng. Nó hữu ích khi bạn muốn tổ chức ứng dụng thành các phần riêng biệt.



* **Drawer Navigator**

Drawer Navigator giúp bạn tạo một menu bên trái hoặc bên phải của ứng dụng. Khi người dùng vuốt sang trái hoặc phải, menu sẽ xuất hiện để họ có thể chọn màn hình hoặc thực hiện các hành động khác.



Trong React Native, Chúng ta sẽ tìm hiểu cách điều hướng giữa các màn hình trong React Native sử dụng các phương thức như navigate, goBack và nhiều hơn nữa. Chúng ta hãy cùng tìm hiểu cách sử dụng nhé!

* **Cài đặt và sử dụng thư viện**

Để bắt đầu, hãy cài đặt React Navigation vào dự án của bạn bằng cách sử dụng lệnh

****

Tiếp theo, bạn cần cài đặt các phần tử điều hướng cụ thể phù hợp với loại điều hướng bạn muốn sử dụng, chẳng hạn như Stack Navigator, Bottom Tab Navigator, hoặc Drawer Navigator.

Ví dụ, để sử dụng Stack Navigator, cài đặt như sau:

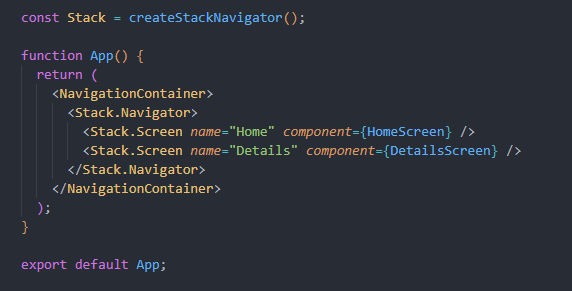
****

Tiếp theo, hãy import những thành phần cần thiết vào tệp App.js hoặc trong file mà bạn muốn quản lý điều hướng:

****

* **Tạo đối tượng Stack Navigator**

Ở đây mình sẽ dùng phần Stack để điều hướng giữa các màn hình cụ thể làvà **DetailsScreen**



* **Tạo đối tượng HomeScreen và DetailsScreen [ Chỗ này code tạo màn hình]**

Ở đây, chúng ta sẽ định nghĩa hai màn hình: HomeScreen và DetailsScreen. Khi bạn muốn chuyển từ màn hình HomeScreen sang màn hình DetailsScreen, bạn có thể sử dụng phương thức navigate như sau:

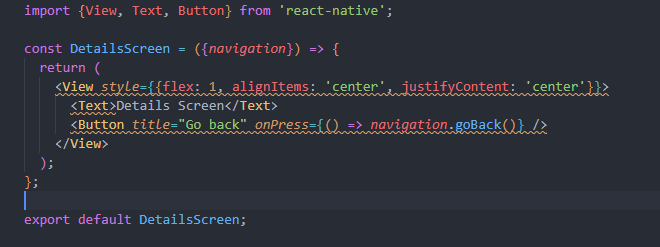
* **\*\*HomeScreen\*\***

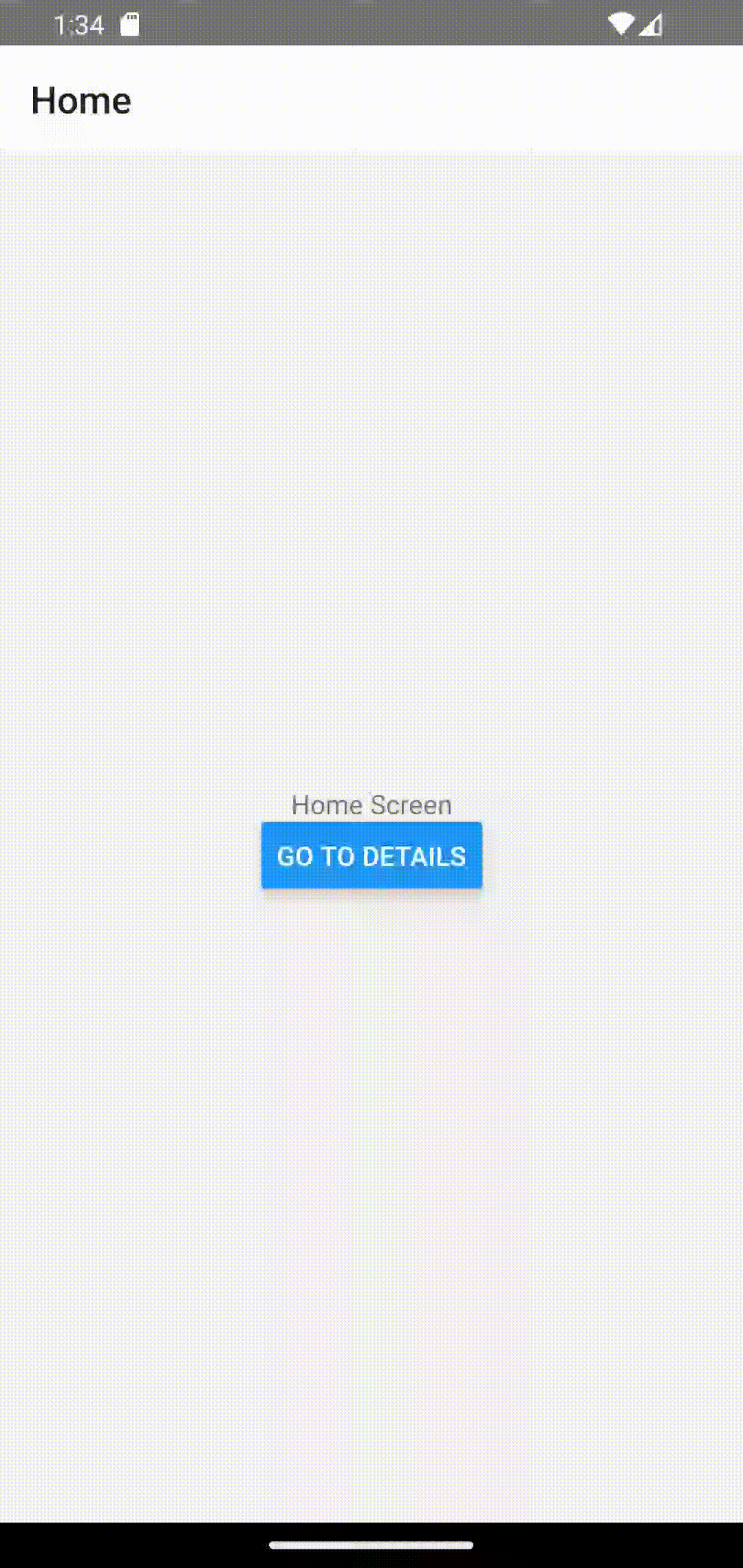
Ở màn hình này, chúng ta sẽ tạo một nút nhấn và sử dụng thuộc tính navigation để thực hiện điều hướng màn hình Home đến màn hình Detail như trong đoạn code.

Sau đó chúng ta thử nhấn vào nút và xem kết quả nhé!

* **\*\*DetailsScreen\*\***

Ở màn hình này, chúng ta sẽ làm tương tự màn hình Home, nhưng sẽ sử dụng thuộc tính goBack() để thực hiện trở về màn hình trước đó, các bạn hãy thử làm đoạn code sau nhé!  
Sau đó chúng ta sẽ thực hiện nhấn vào nút và xem kết quả nhé

****



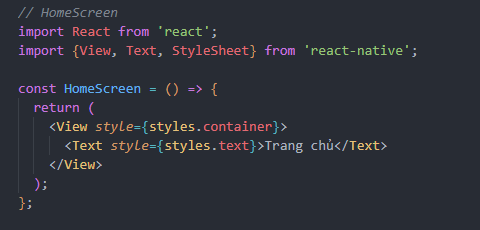
Vậy là chúng ta đã tìm hiểu xong về một ví dụ về điều hướng giữa các màn hình rồi, các bạn hãy tìm hiểu thêm về các phần điều hướng khác như Bottomtab hay Drawer,

* **Giới thiệu**

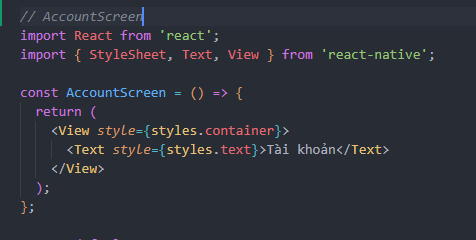
Routing hook là một công cụ quan trọng giúp chúng ta quản lý và điều hướng giữa các màn hình (screens) trong ứng dụng của mình. Trong React Native, chúng ta thường sử dụng một số thư viện routing phổ biến như React Navigation hoặc React Router Native để giúp quản lý việc điều hướng và hiển thị các màn hình.

Ví dụ, chúng ta sẽ tạo một ứng dụng React Native đơn giản với 3 màn hình: “HomeScreen”, “LoginScreen” và “AccountScreen”

**HomeScreen.js**

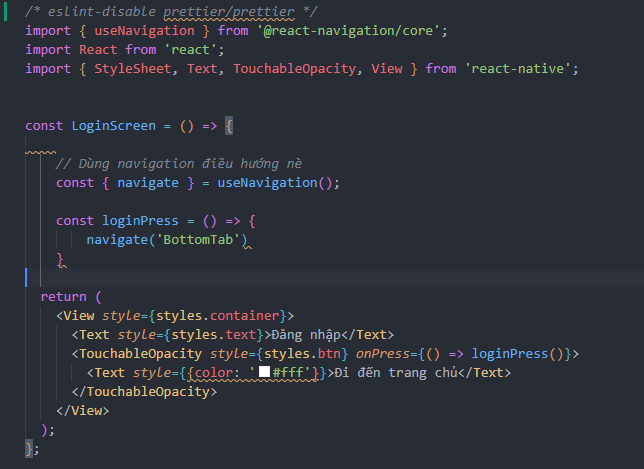
****

**AccountScreen.js**

****

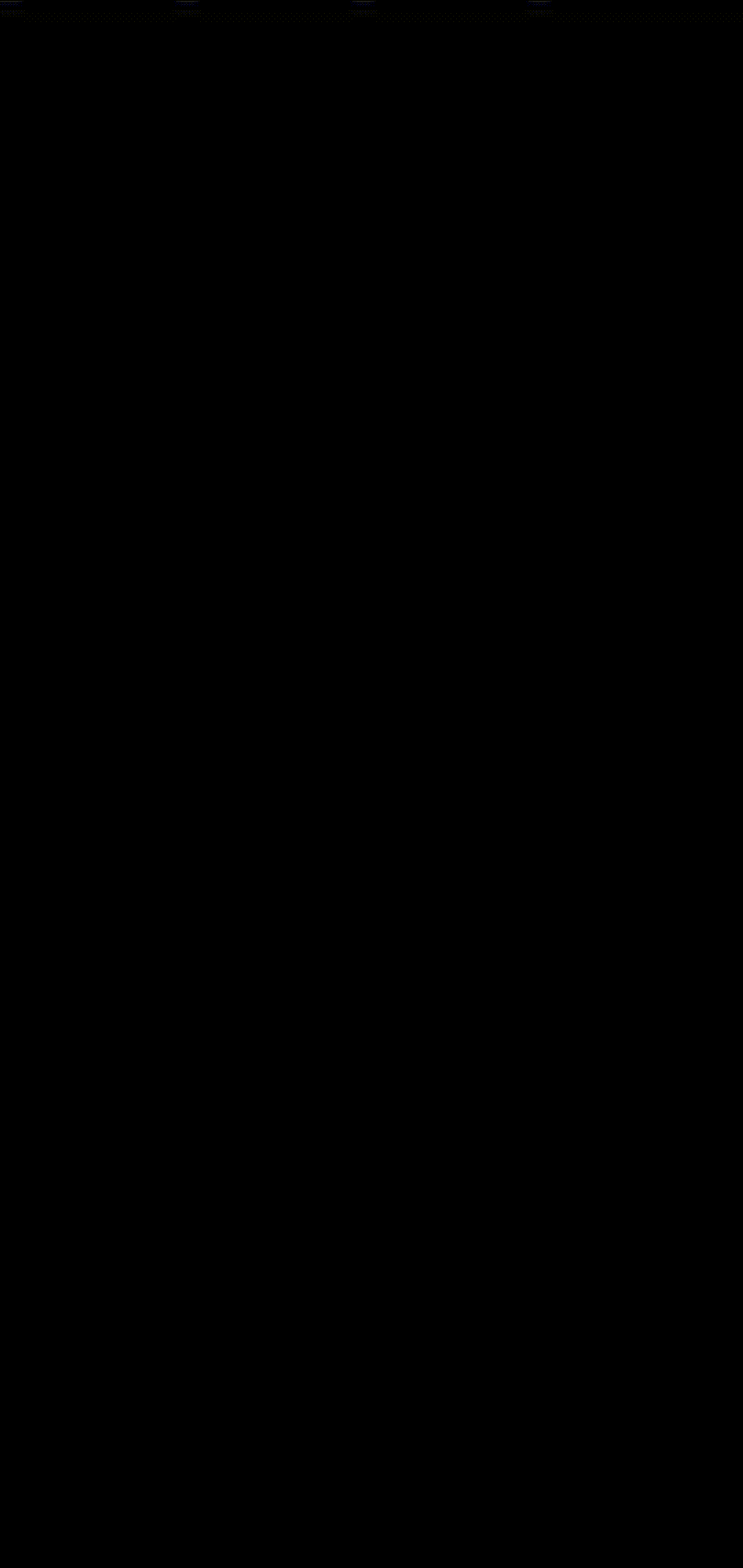
**BottomTab.js**

* Điều này sẽ chuyển đổi giữa các màn hình riêng biệt nhưng vẫn nằm trong một khung hình
* **Tạo routing**
* Tiếp theo, chúng ta sẽ sử dụng Routing Hook **`useNavigation`** từ thư viện React Navigation để điều hướng từ màn hình `**LoginScreen`** đến màn hình **`BottomTab`**



**Cụ thể:**

* Sử dụng **`useNavigation`** để lấy đối tượng điều hướng.
* Trong hàm **`loginPress`,** mình đã sử dụng phương thức **`navigate`** từ đối tượng điều hướng để chuyển đến màn hình **`BottomTab`**
* Khi người dùng nhấn vào **“Đi đến trang chủ”**, màn hình sẽ chuyển đến màn hình **`BottomTab`** mà bạn đã định nghĩa trong hàm **`loginPress`**
* **Chạy ứng dụng**

****

Đó là cách sử dụng "Routing hook" trong React Native để quản lý điều hướng giữa các màn hình. Với React Navigation, bạn có thể tùy chỉnh và điều hướng giữa các màn hình một cách linh hoạt, giúp ứng dụng của bạn trở nên chuyên nghiệp và dễ sử dụng.

**Bài tập**

* **Đề bài :** Ứng dụng React Native với điều hướng bằng Stack Navigator
* **Yêu cầu**

1. Cài đặt React Navigation và các thư viện liên quan cần thiết (Stack {Navigator và Navigation Container).
2. Tạo hai màn hình: HomeScreen và DetailsScreen.
3. Trong HomeScreen, hiển thị một nút "Go to Details". Khi nhấn vào nút này, chuyển đến màn hình DetailsScreen.
4. Trong DetailsScreen, hiển thị một nút "Go back". Khi nhấn vào nút này, quay lại màn hình trước đó (HomeScreen).

**[Kết thúc bài học]**.

* Như đã học, routing là một phần quan trọng trong phát triển ứng dụng di động, giúp chúng ta tạo ra các trang và chuyển đổi giữa chúng một cách dễ dàng. Thư viện React Navigation cung cấp các công cụ mạnh mẽ để tạo và quản lý điều hướng, bao gồm Stack Navigator, Tabs, Drawer Navigator và nhiều loại điều hướng khác.
* Hy vọng các bạn đã có cái nhìn tổng quan về cách quản lý điều hướng và chuyển đổi giữa các màn hình trong ứng dụng di động của mình