# Lab 2

## Mục tiêu

Kết thúc bài thực hành sinh viên có khả năng:

* Làm quen với hooks trong React Native
* Ứng dụng kiến thức được học vào ứng dụng

## Nội dung

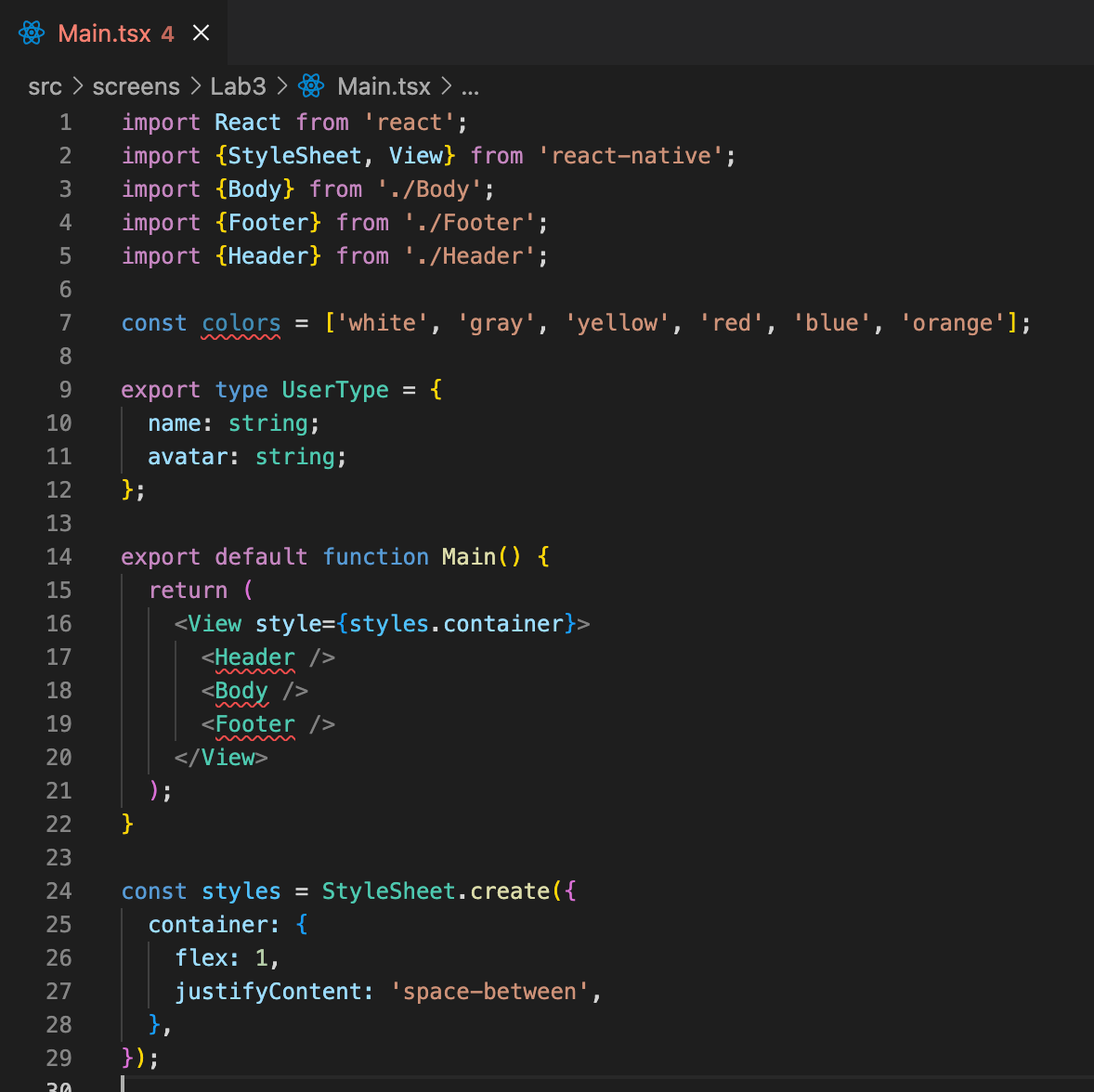
## Bài 1: CẤU TRÚC PROJECT ỨNG DỤNG



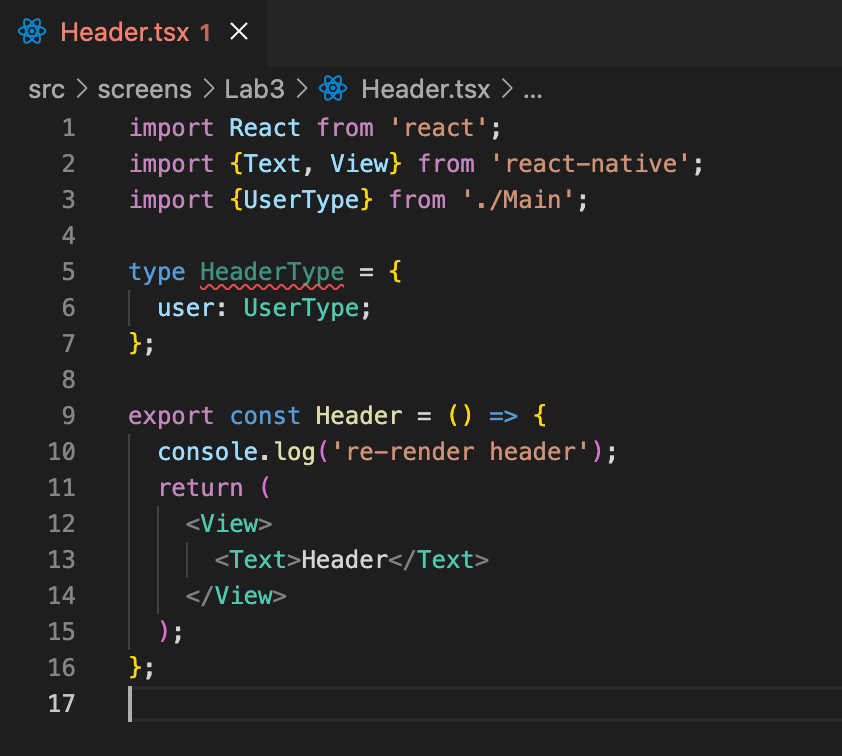
Hướng dẫn:- Tạo mới 4 file bao gồm: **Body.tsx, Footer.tsx, Header.tsx, Main.tsx.**

- File **Main.tsx** sẽ là giao diện chính, chứa 3 file thành phần còn lại

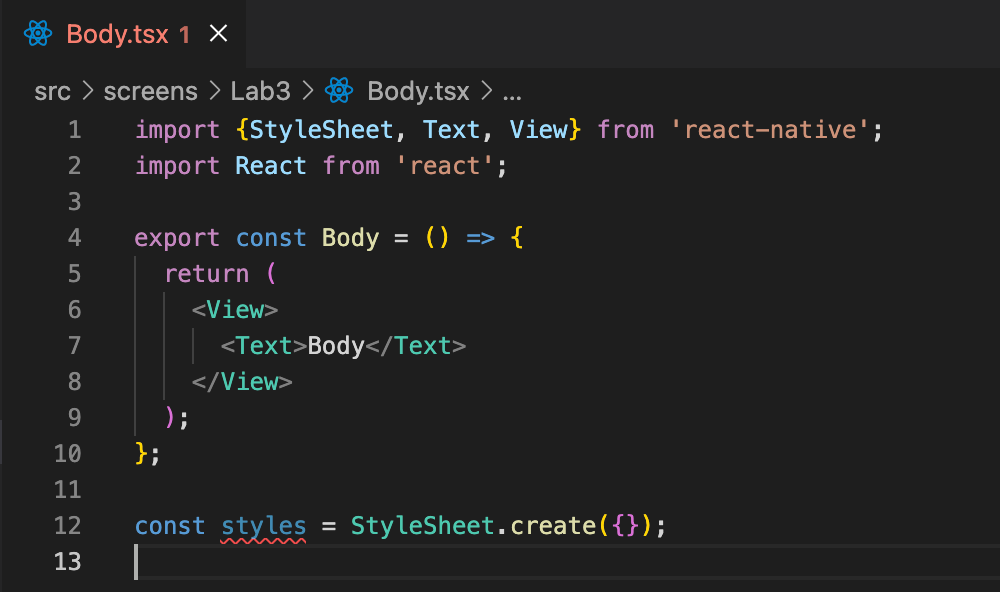
- Code file **Main.tsx** theo hướng dẫn bên dưới đây:



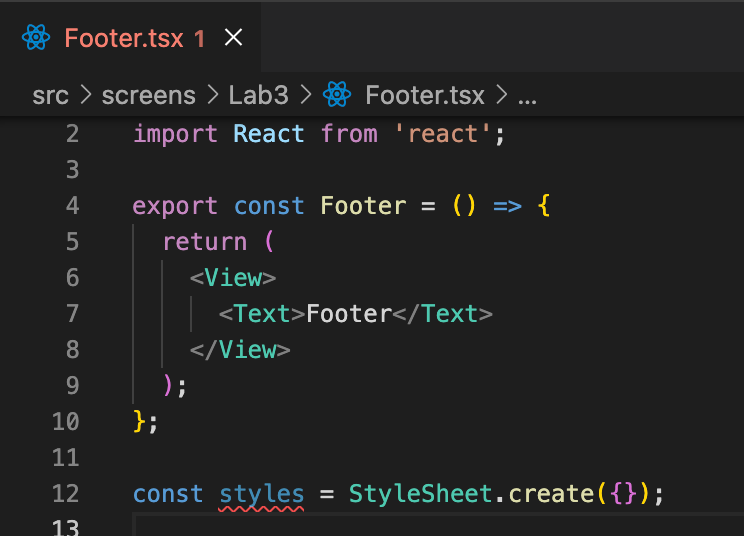
- Code file **Header.tsx** theo hướng dẫn bên dưới đây:



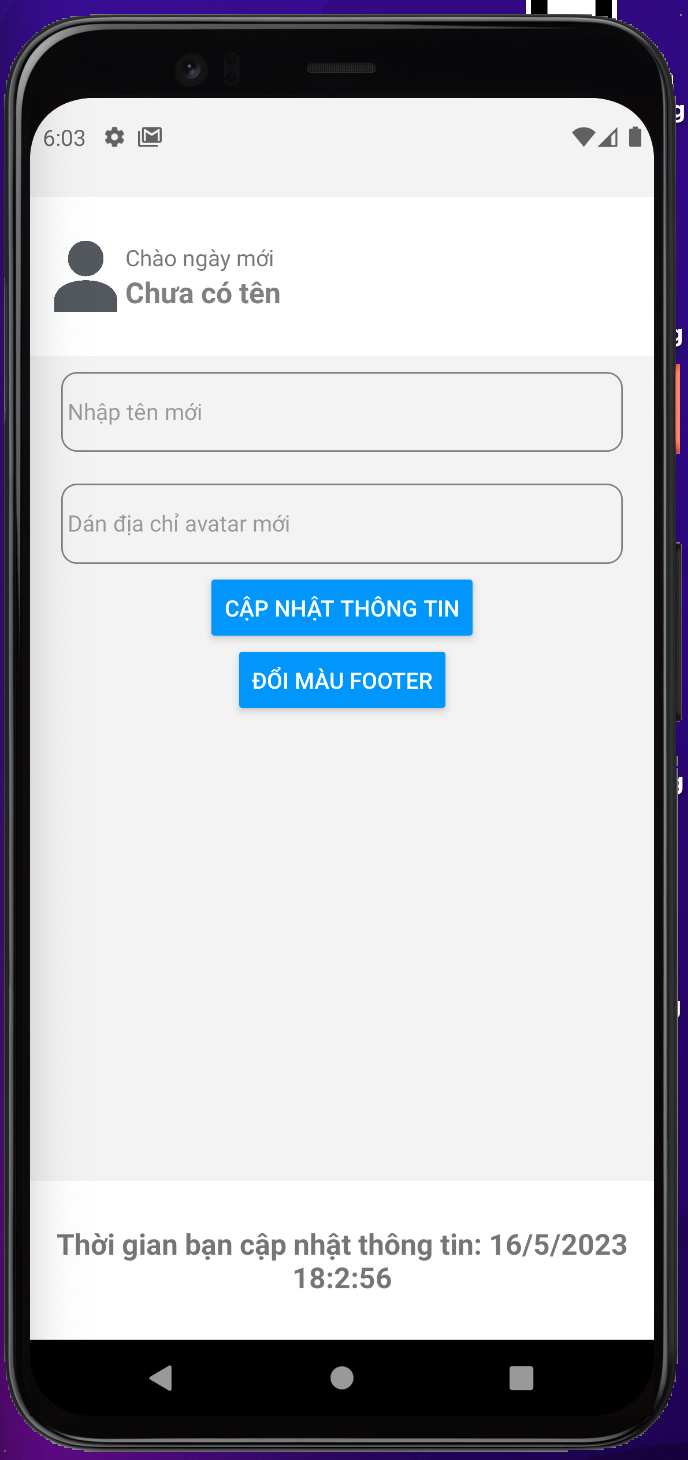
- Code file **Body.tsx** theo hướng dẫn bên dưới đây:



- Code file **Footer.tsx** theo hướng dẫn bên dưới đây:



## Bài 2: XÂY DỰNG ỨNG DỤNG CÓ GIAO DIỆN NHƯ BÊN DƯỚI



Yêu cầu:

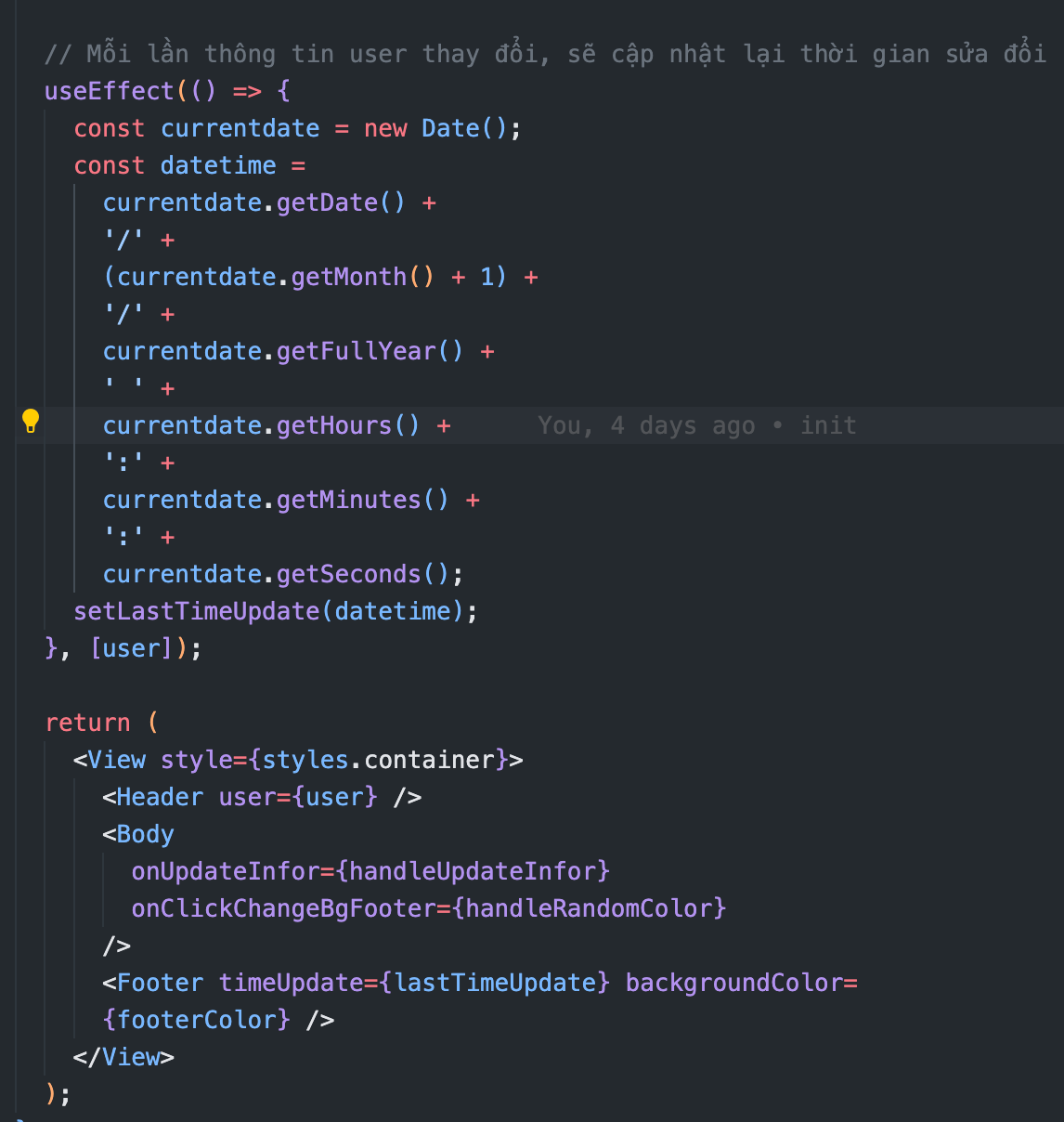
- Bọc memo vào **Header** và **Footer**

- Viết các hàm xử lý có **useCallBack**, **useEffect** xử lý hàm và dữ liệu.

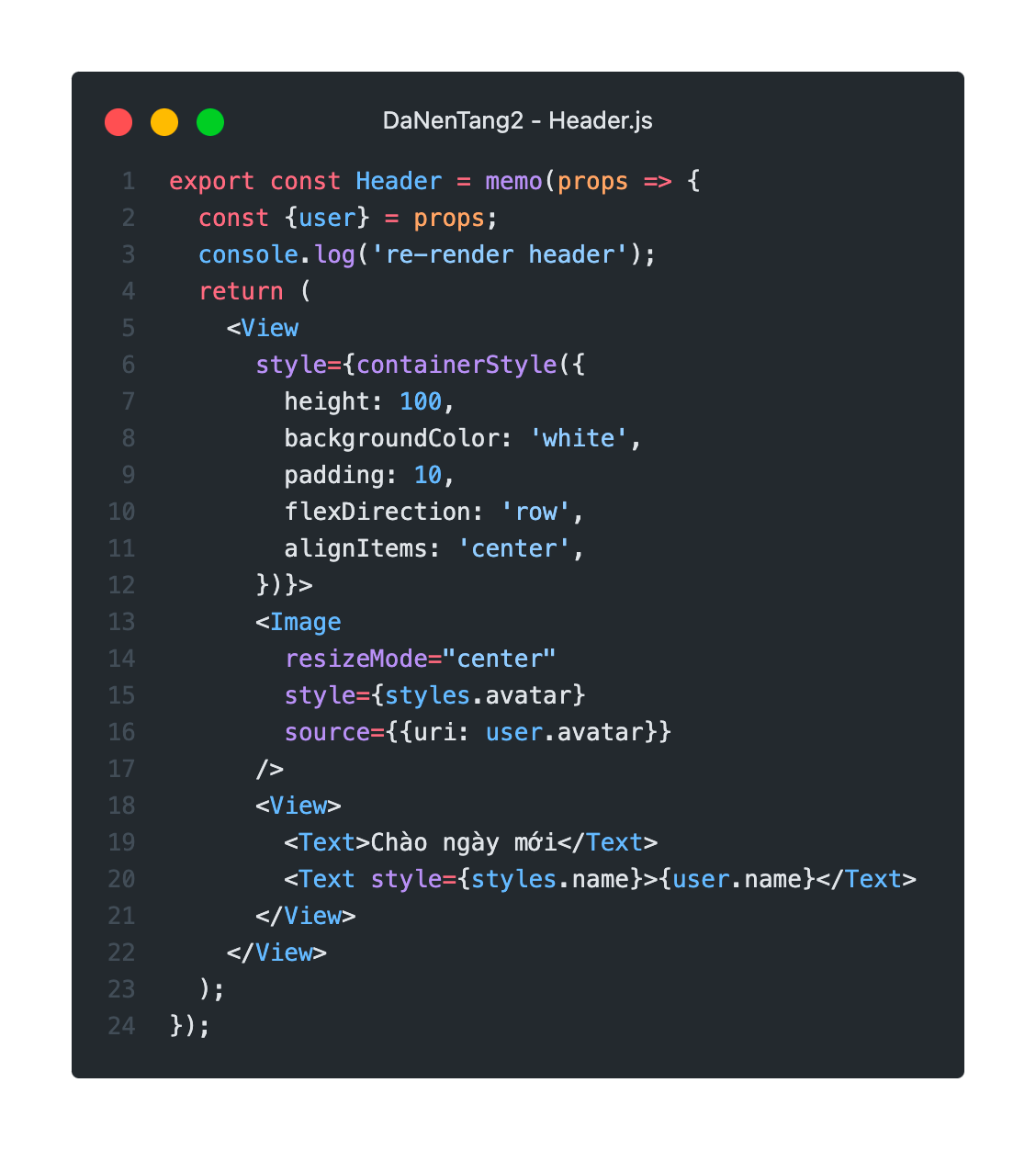
Hướng dẫn:

- Viết các hàm xử lý cho file **Main.tsx**:





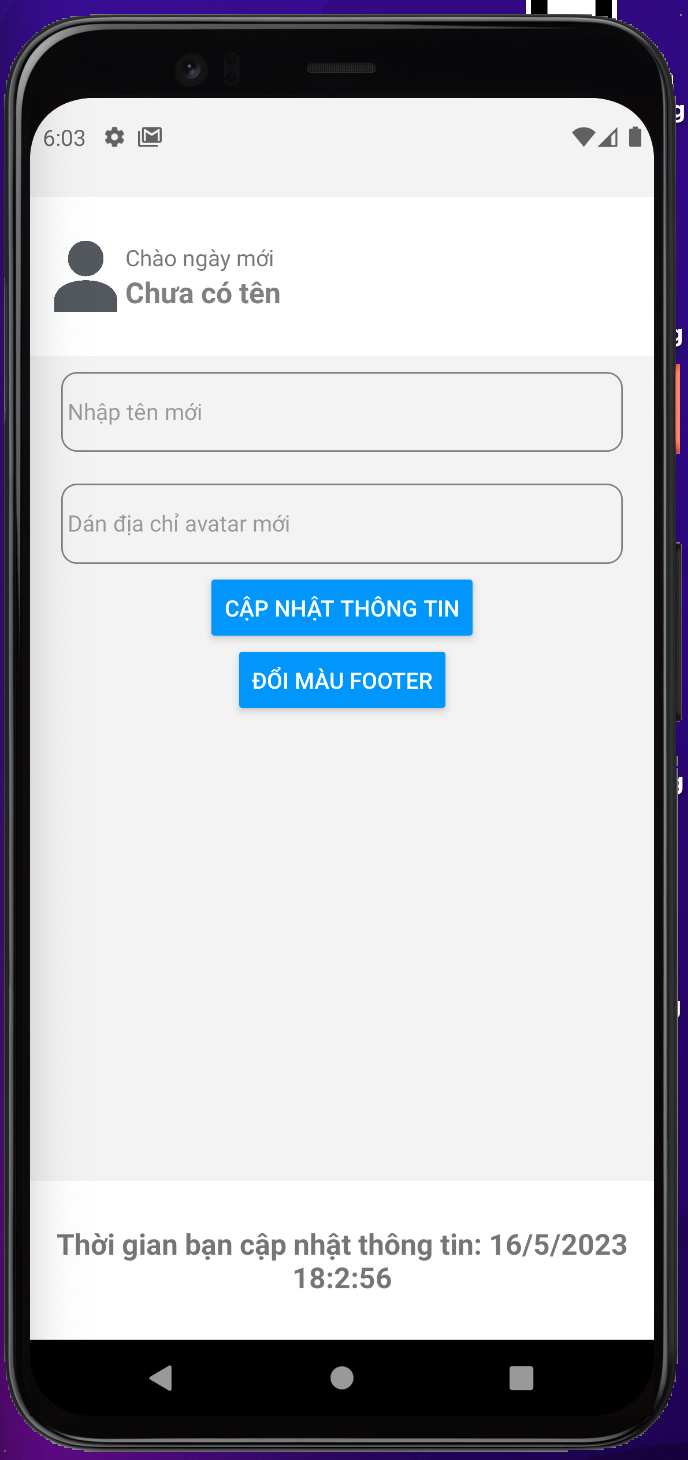
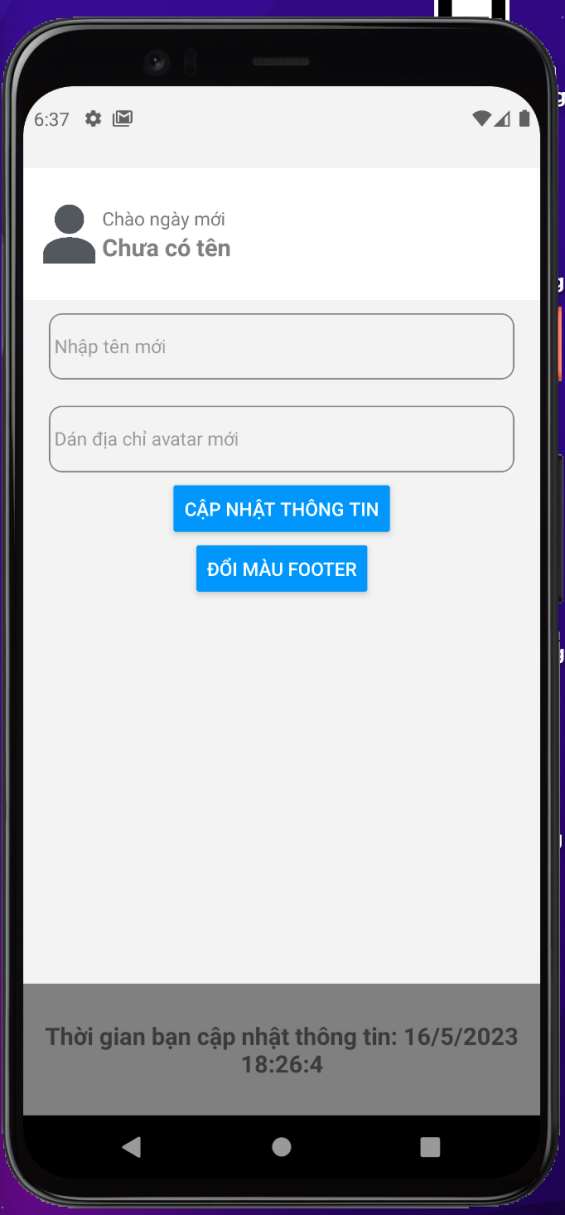
- Viết các hàm xử lý cho file **Header.tsx** có bọc memo:



- Viết các hàm xử lý cho file **Footer.tsx** có bọc memo:



## Bài 3: VIẾT BODY CỦA ỨNG DỤNG CÓ GIAO DIỆN NHƯ SAU:

Yêu cầu:

- Nhấn nút **“ĐỔI MÀU FOOTER”,** background sẽ tự random đổi màu.

- User có thể thay đổi thông tin trên **Header** bằng cách nhập vào thông tin.

- Khi nhập thông tin vào TextInput, component Header và Footer không bị re-render.

- Khi nhấn “Cập nhật thông tin” mới re-render Header và Footer để đẩy dữ liệu lên hiển thị.

Hướng dẫn:

- Tạo 2 state lưu trữ giá trị của name và linkImage.

- Kiểm tra field đã được nhập chưa, nếu chưa nhập thông báo không được để trống.

- Gọi hàm **onUpdateInfor** để cập nhật lại thông tin header.

- Gọi hàm **onClickChangeBgFooter** để cập nhật lại màu nền **Footer.**



Bài 4: GV cho thêm

## \*\*\* Yêu cầu nộp bài:

## Sv nén file bao gồm các yêu cầu đã thực hiện trên, nộp lms đúng thời gian quy định của giảng viên. Không nộp bài coi như không có điểm.

--- Hết