**Họ và tên : Nguyễn Thanh Hoan 21IT551**

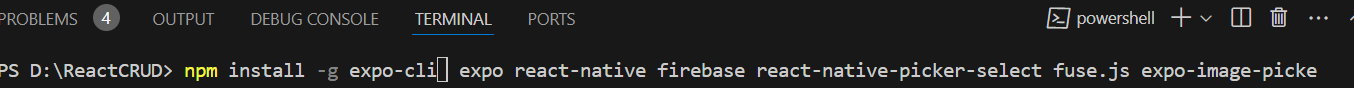
**CÁC BƯỚC TRIỂN KHAI DỰ ÁN**

**1. Cài đặt dự án**

* Đầu tiên, đảm bảo rằng bạn đã cài đặt môi trường phát triển **React Native**. Sử dụng **Expo** để dễ dàng phát triển.



* Cài đặt các gói cần thiết:

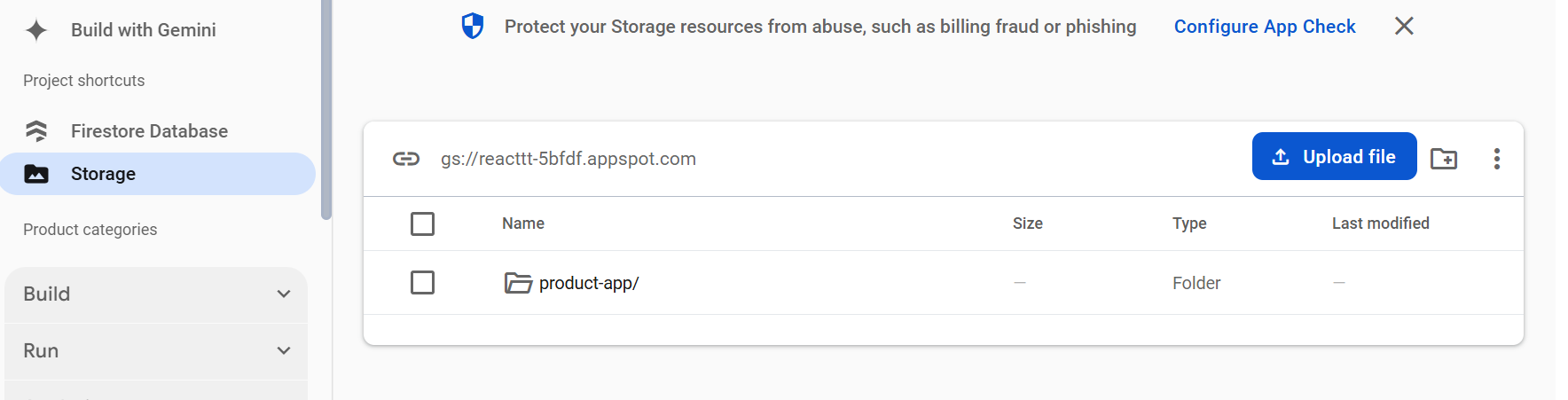


* Tạo và cấu hình Firebase cho dự án. Thêm các cấu hình Firebase vào file



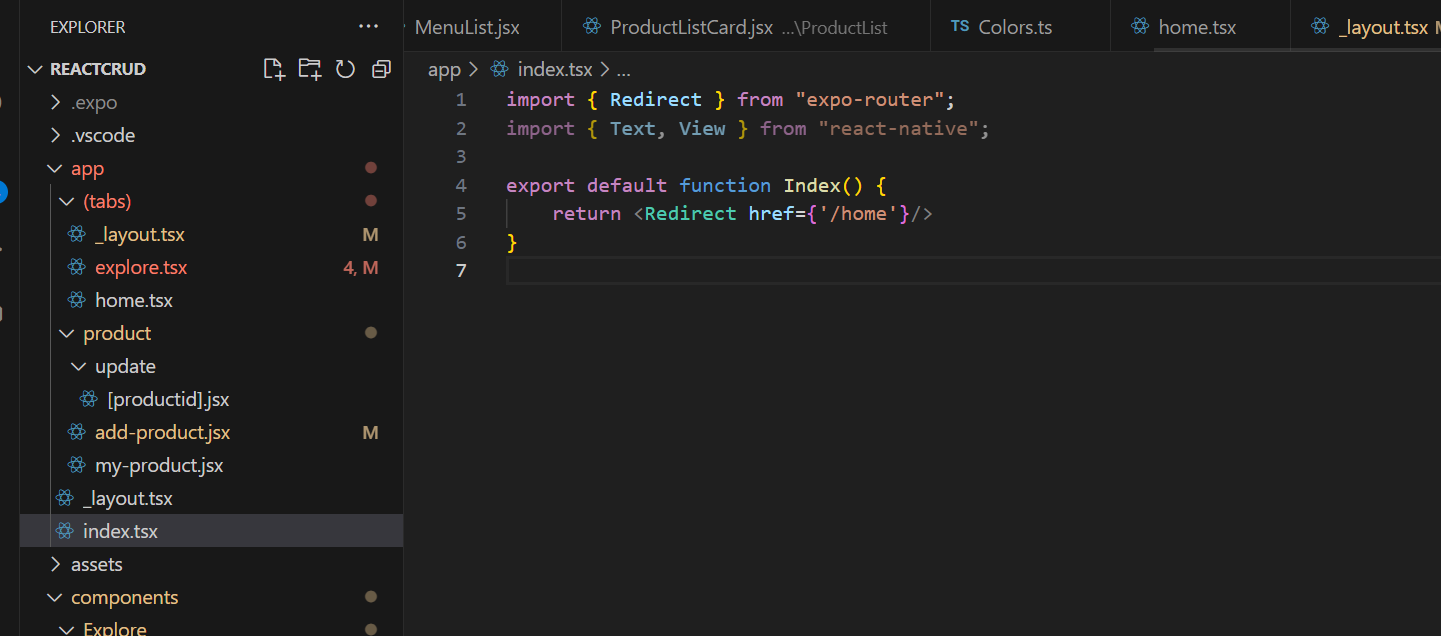
FirebaseConfigs.js:

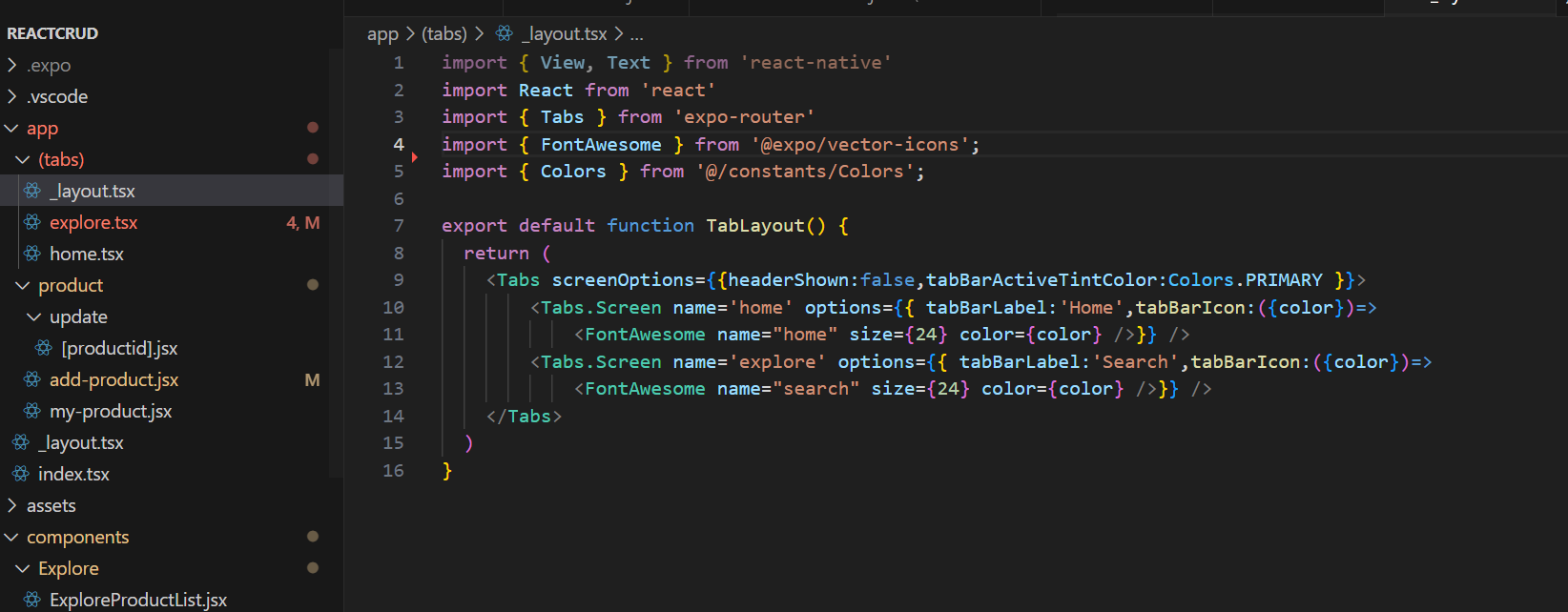


* Trong các chức năng như thêm hoặc cập nhật dữ liệu từ Firestore, bạn sẽ sử dụng đối tượng **db (được khởi tạo từ app) và storage (được khởi tạo từ app)** để tương tác với cơ sở dữ liệu và lưu trữ và quản lý các tệp như hình ảnh và video.
* 

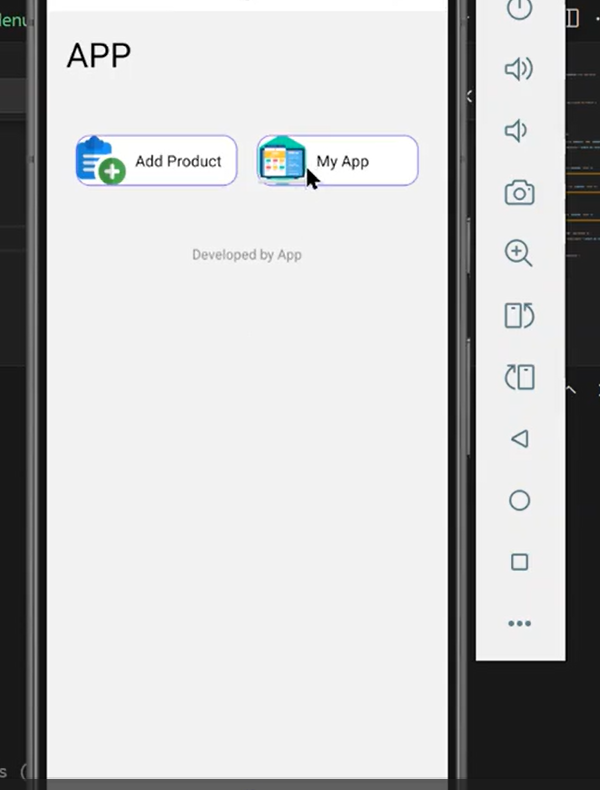
**2. Tạo Tab Home và Search**

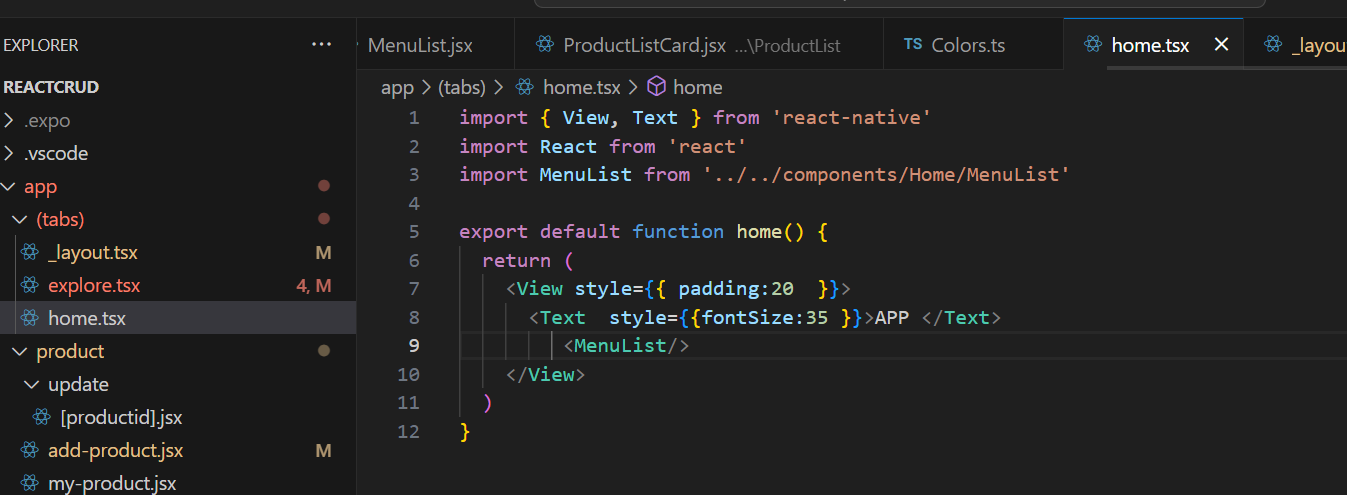
* Redirect sẽ điều hướng người dùng đến đường dẫn /home.



* Tab Home và Search.
* 

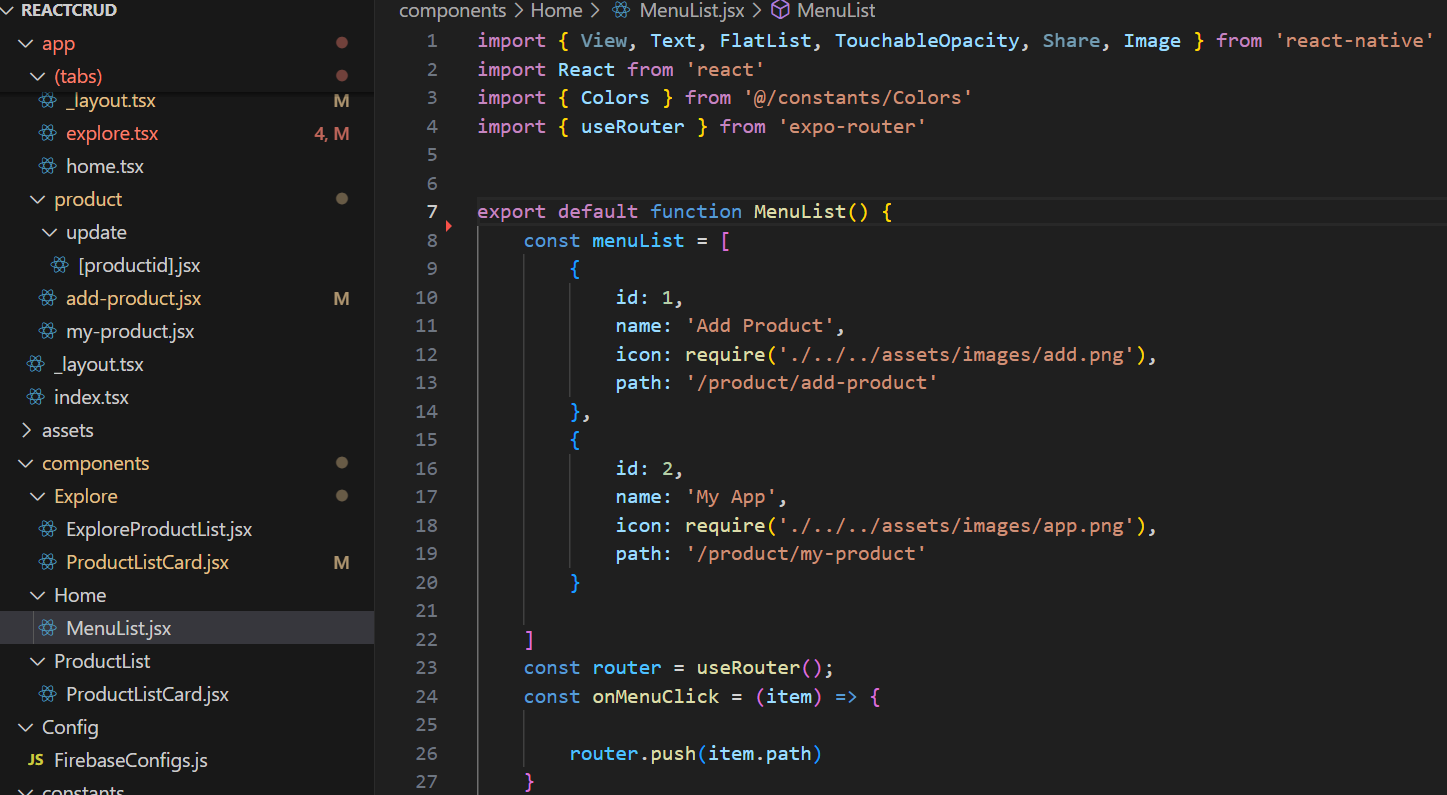
**3. Trang chủ (Home Screen)**

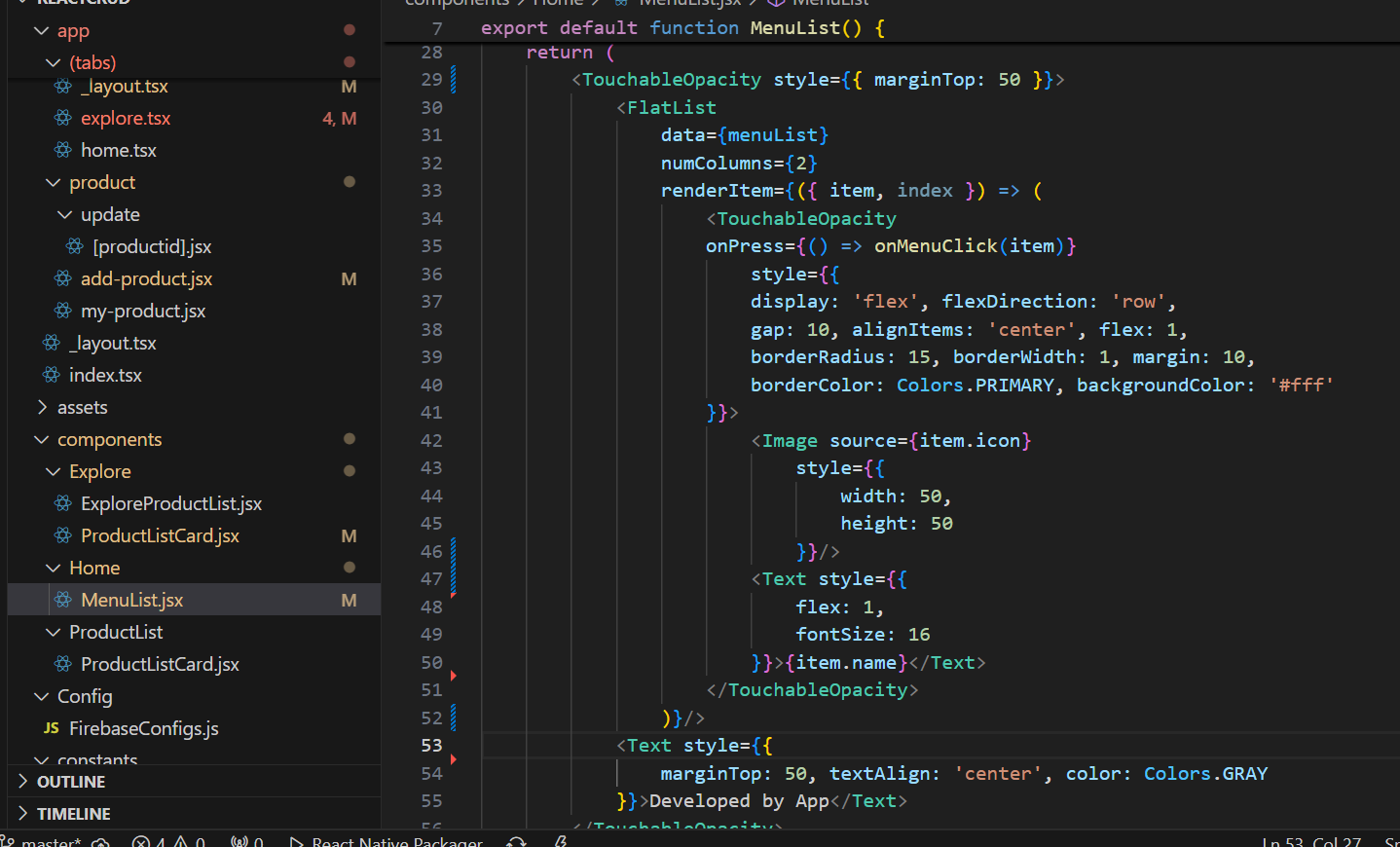
* Đây là nơi hiển thị danh sách các tùy chọn trong MenuList.
* 



**4. Tạo các thành phần giao diện (Components)**

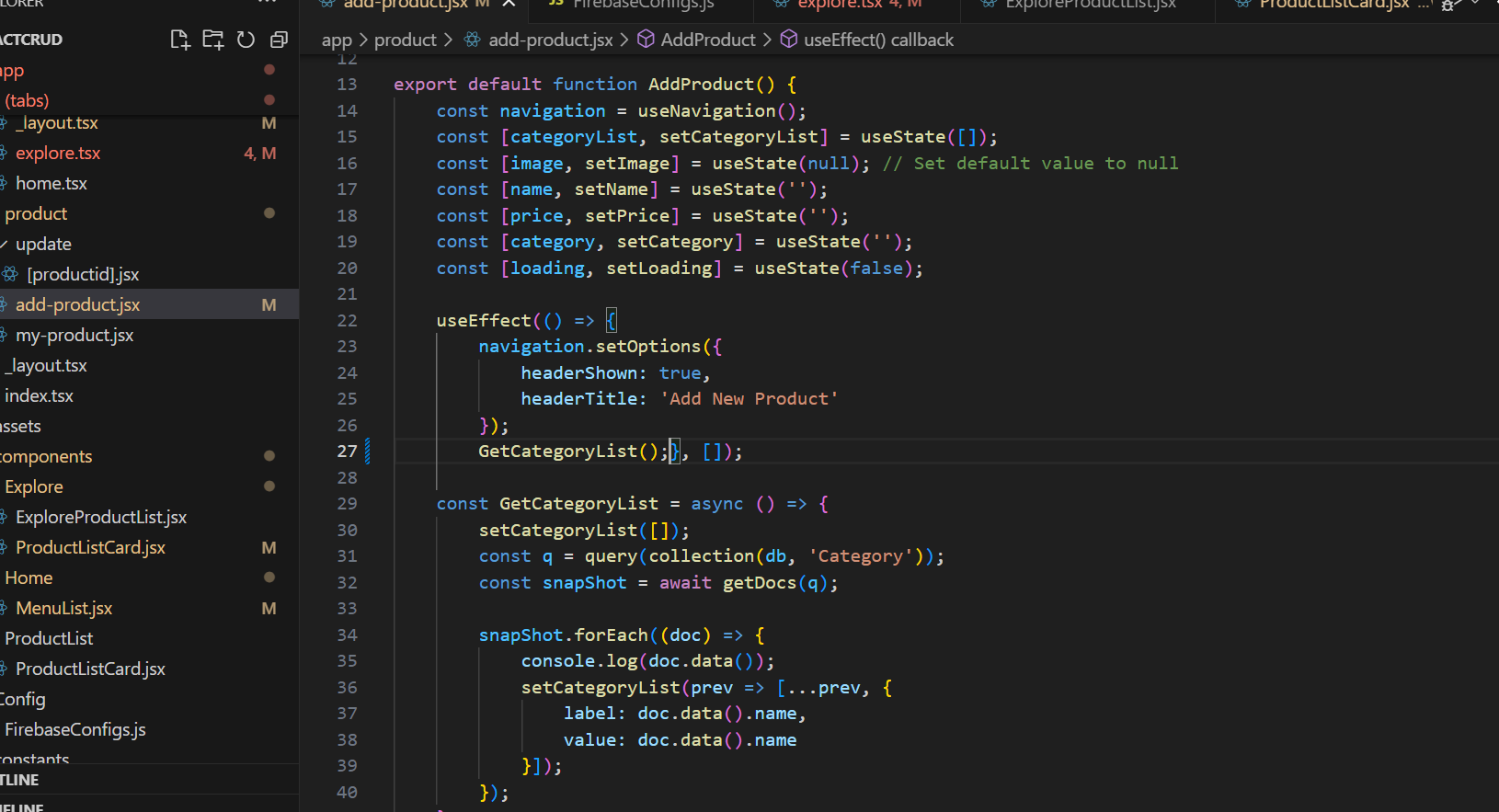
* **MenuList Component**: Dùng để hiển thị danh sách các menu, bao gồm các tùy chọn như “Add Product” và “My Product”.

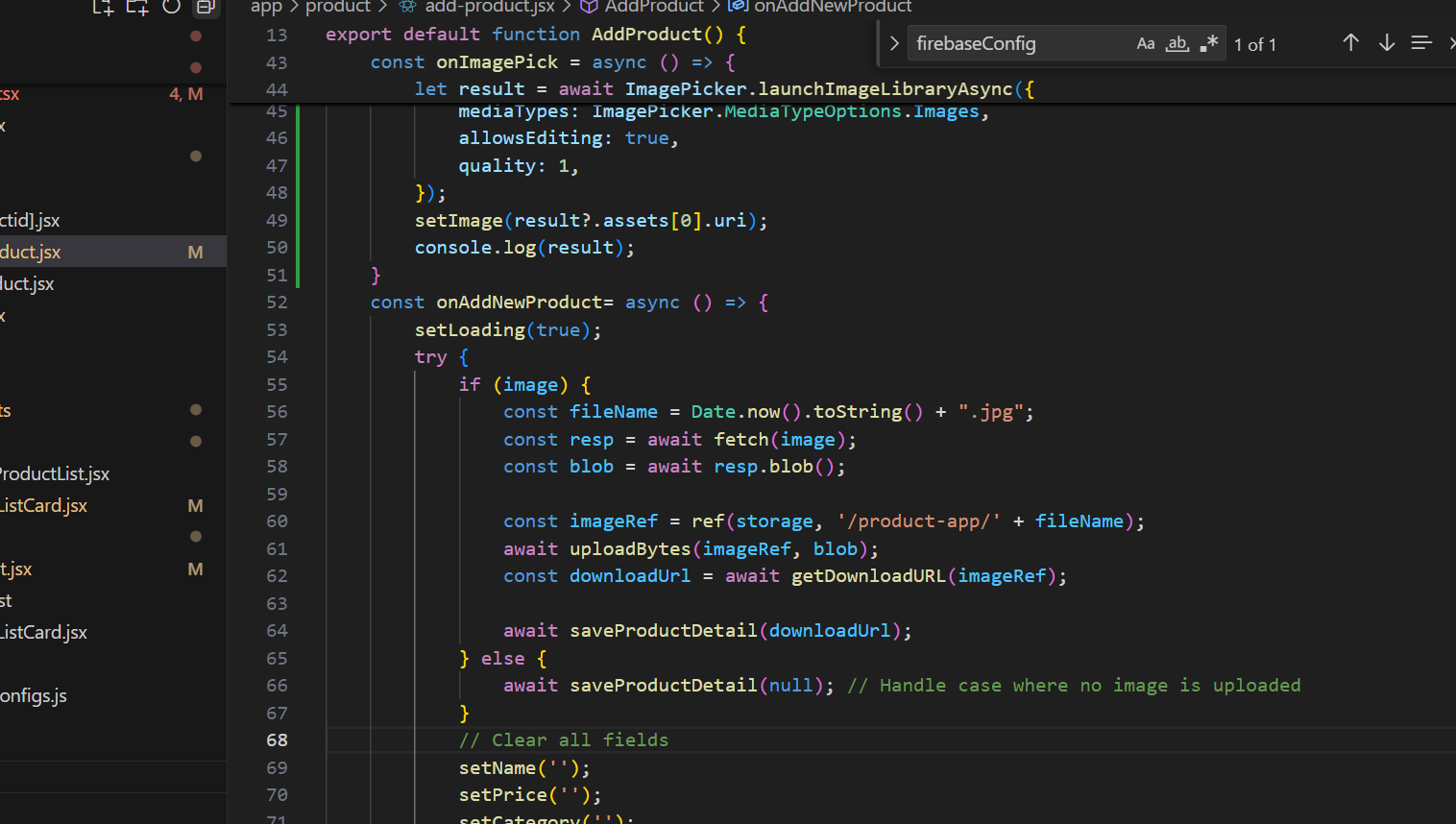


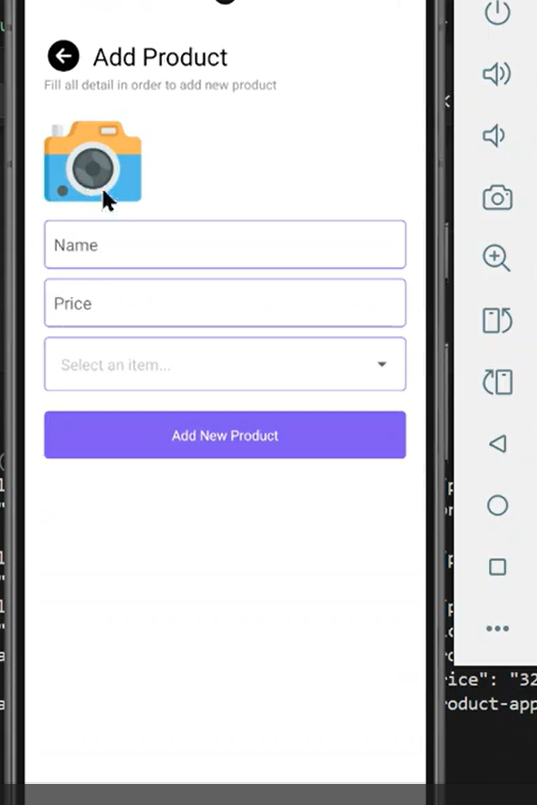


**5. Thêm sản phẩm (Add Product)**

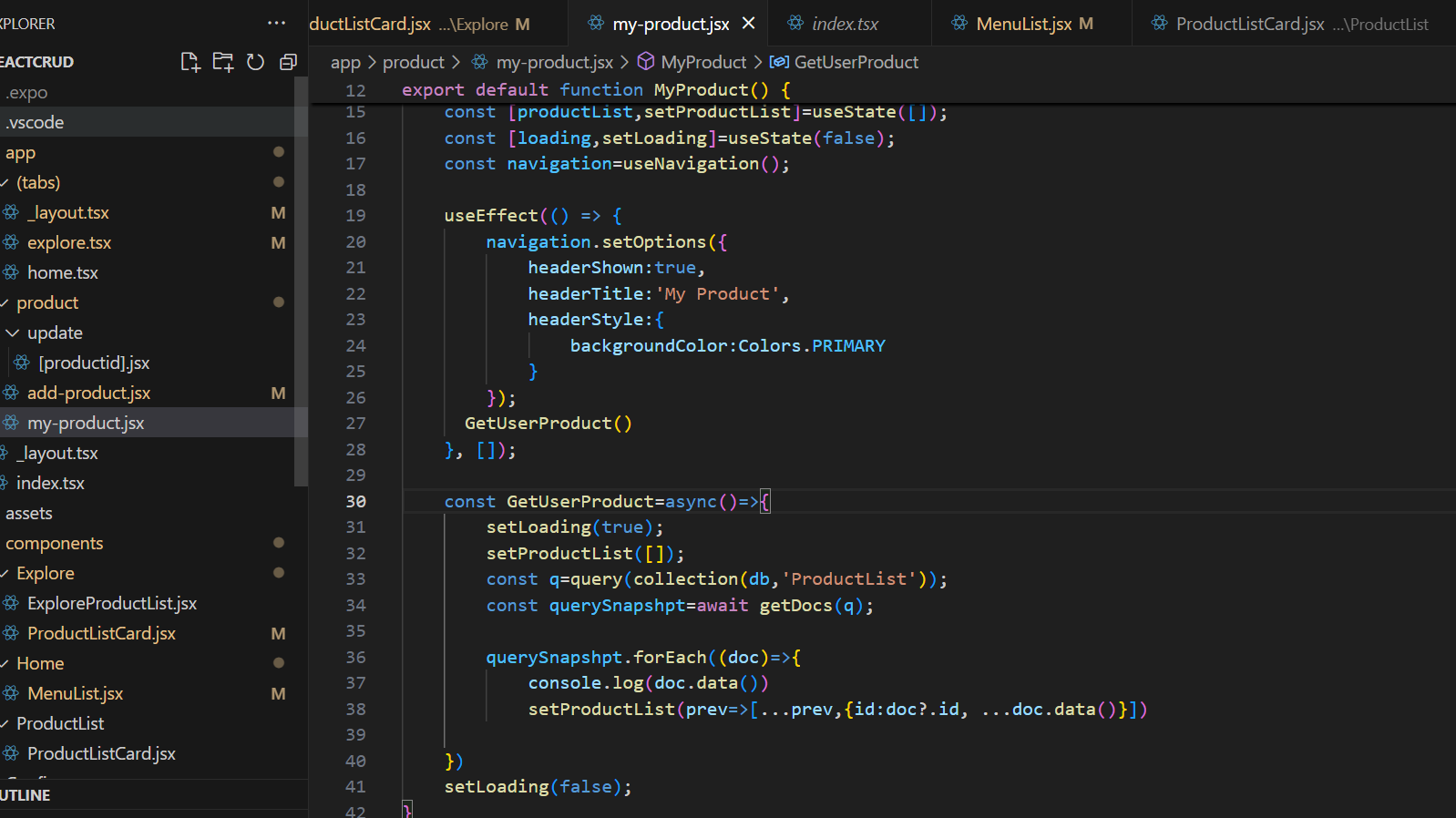
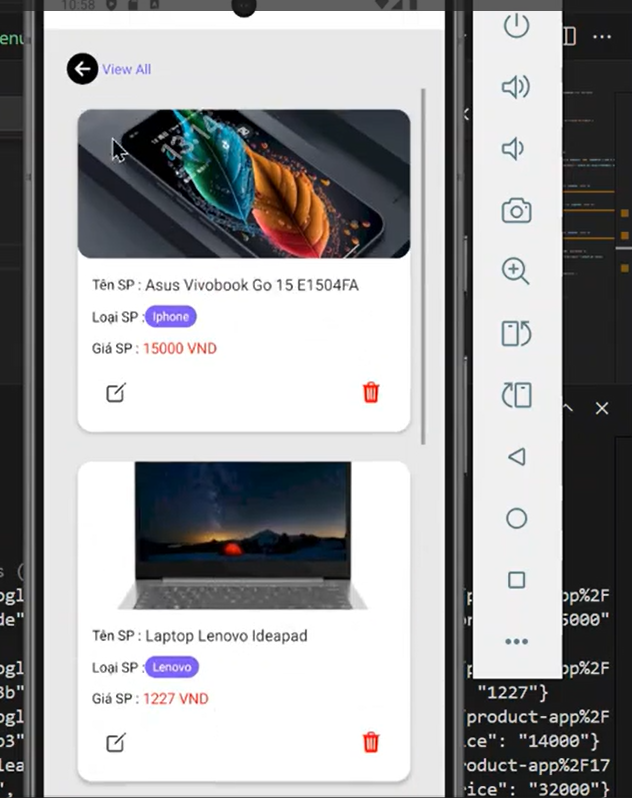
* Tạo file product/add-product.jsx để thêm sản phẩm mới, bao gồm việc chọn ảnh sản phẩm, tên, giá, và danh mục(tạo sẵn dữ liệu sau đó thì dùng RNPickerSelect để chọn ). Dữ liệu sẽ được lưu vào Firebase Firestore.
* Khi component được render lần đầu tiên, tiêu đề trên thanh điều hướng sẽ được thiết lập là "Add New Product" và danh sách danh mục sẽ được lấy về thông qua hàm GetCategoryList(Lấy danh sách các danh mục từ Firestore.) ,Hàm onImagePick: Mở thư viện ảnh để người dùng chọn một hình ảnh, Hàm onAddNewProduct : Thêm một sản phẩm mới vào Firestore và tải lên hình ảnh

****

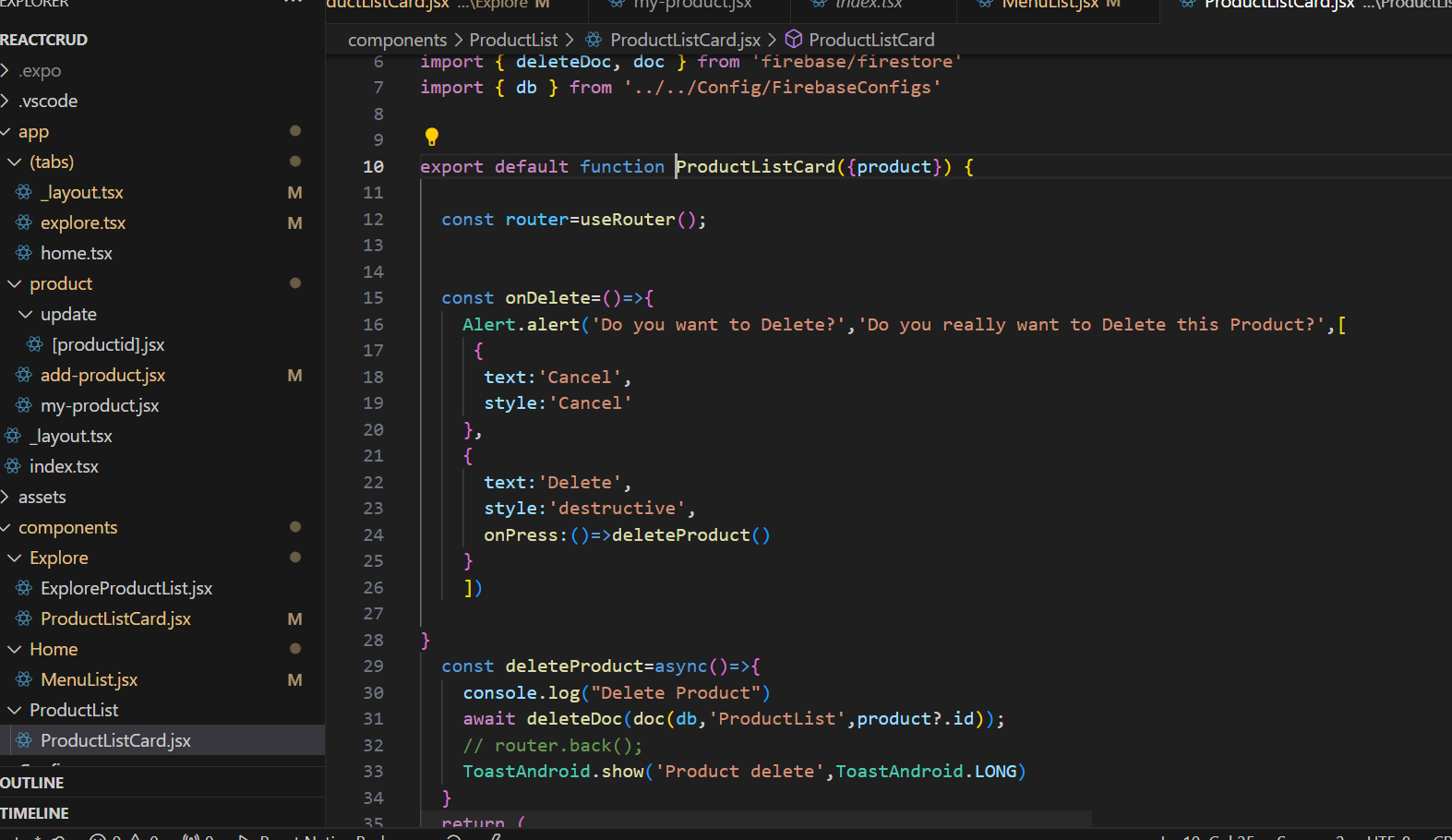
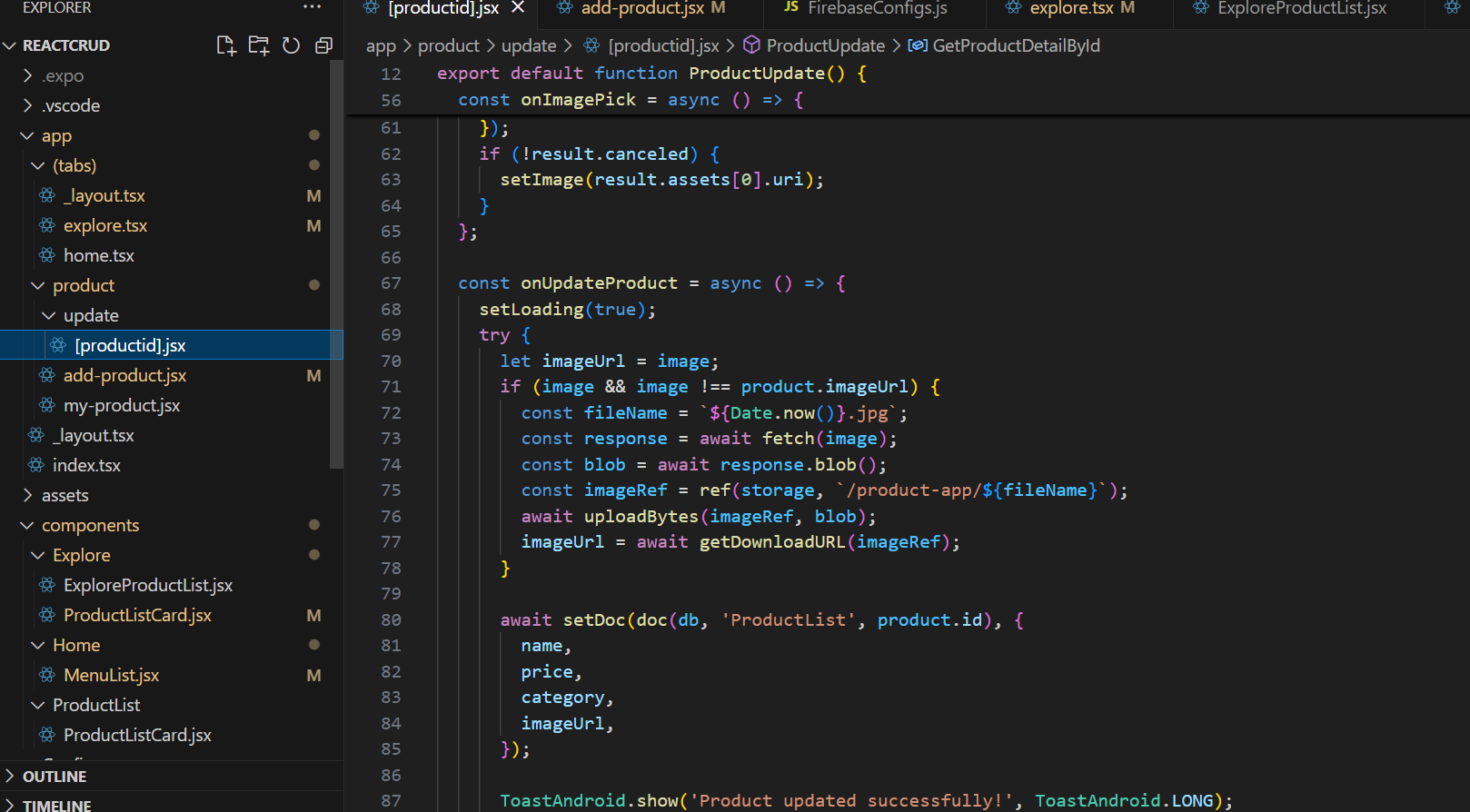
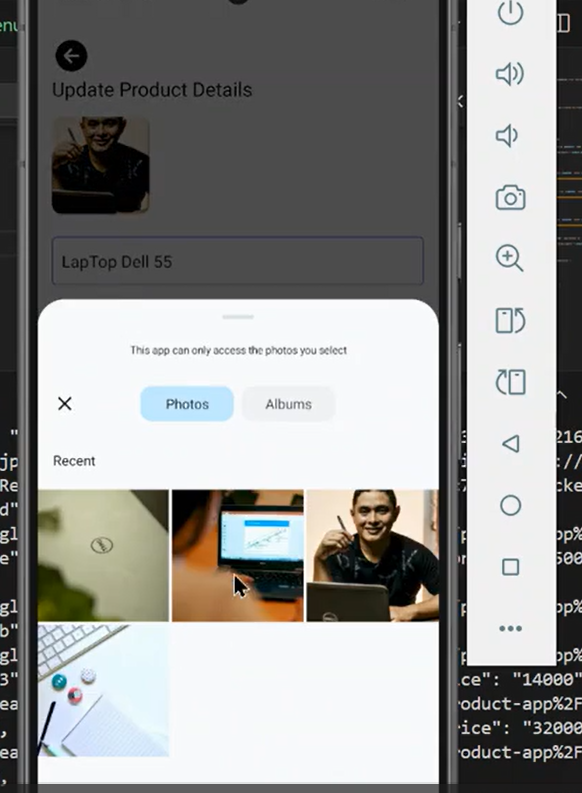
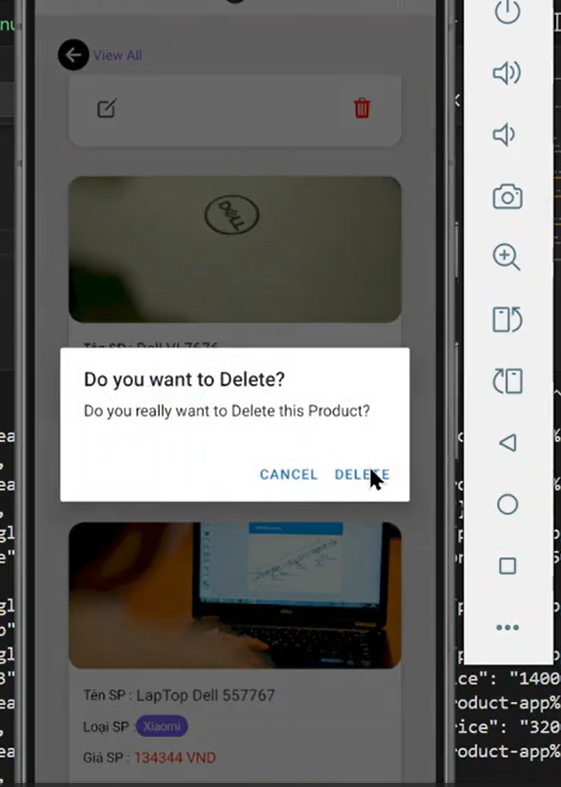
****

* **Giao diện Thêm**
* ****

**6. Danh sách sản phẩm (My Product)**

* **useEffect Hook**: Khi component lần đầu được render, nó thiết lập tùy chọn tiêu đề của màn hình điều hướng (tựa đề "My Product", màu nền). Đồng thời, hàm GetUserProduct được gọi để lấy danh sách sản phẩm từ Firestore.
* Firebase :db, storage: để làm việc với cơ sở dữ liệu và lưu trữ từ Firebase , collection, doc, getDocs, query, setDoc, where: để thao tác với cơ sở dữ liệu Firestore.
* **GetUserProduct**: Đây là hàm bất đồng bộ (async) dùng để lấy dữ liệu từ Firestore. Nó:
  1. Đặt cờ loading thành true để biểu thị rằng dữ liệu đang được tải.
  2. Thực hiện truy vấn (query) tới bộ sưu tập sản phẩm (ProductList).
  3. Sau khi dữ liệu được lấy, hàm setProductList được gọi để cập nhật danh sách sản phẩm với dữ liệu mới.
  4. Cuối cùng, đặt loading thành false sau khi hoàn tất.
* **Nút "Go Back"**: Nút này cho phép người dùng quay lại trang trước bằng cách gọi hàm navigation.goBack().
* **Danh sách sản phẩm**: Được hiển thị bằng FlatList, với dữ liệu được truyền vào từ productList. Mỗi sản phẩm sẽ được hiển thị qua component ProductListCard
* 
* 

**7. Xóa và cập nhật sản phẩm**

* **onDelete** :Hiển thị một hộp thoại cảnh báo (Alert.alert) hỏi người dùng có chắc chắn muốn xóa sản phẩm không.Nếu người dùng chọn "Delete", hàm deleteProduct sẽ được gọi để xóa sản phẩm khỏi Firestore.
* **deleteProduct** :Hàm này thực hiện thao tác xóa sản phẩm từ Firestore bằng cách gọi deleteDoc với ID sản phẩm. ,Sau khi xóa thành công, nó hiển thị thông báo "Product deleted" bằng ToastAndroid
* **Nút chỉnh sửa**: Khi người dùng bấm vào nút này, họ sẽ được chuyển đến trang cập nhật sản phẩm (product/update/[id]) để chỉnh sửa sản phẩm.
* 
* 
* 
* 

**8. Tìm kiếm sản phẩm (Search Product)**

### Bước 1: Cài đặt thư viện fuse.js

Thư viện fuse.js giúp thực hiện tìm kiếm gần giống bằng cách so sánh chuỗi dựa trên độ tương đồng của các ký tự.

* Mở terminal và cài đặt thư viện:

npm install fuse.js

### Bước 2: Thiết kế giao diện cho chức năng tìm kiếm

### Bước 3: Lấy dữ liệu từ Firestore

* Sử dụng hàm getDocs để lấy toàn bộ danh sách sản phẩm từ Firestore.

### Bước 4: Cấu hình Fuse.js để thực hiện tìm kiếm gần giống

* Sau khi đã có danh sách sản phẩm, sử dụng Fuse.js để tìm kiếm dựa trên từ khóa người dùng nhập vào.
* Ví dụ về cấu hình Fuse.js:

### Bước 5: Cập nhật giao diện khi tìm kiếm

* Khi tìm kiếm, bạn có thể hiển thị trạng thái "loading" để thông báo rằng đang thực hiện tìm kiếm, và sau đó cập nhật giao diện để hiển thị kết quả.

### Bước 6: Thêm logic debounce

* Để tránh việc tìm kiếm quá thường xuyên khi người dùng nhập từ khóa, bạn có thể thêm logic "debounce" để trì hoãn việc tìm kiếm cho đến khi người dùng ngừng nhập một khoảng thời gian nhất định (ví dụ: 500ms).

