



BÀI TẬP THỰC HÀNH 4

COMMUNICATING WITH SERVERS

Bài 1: Mobile E-commerce App

1.1. Mục tiêu

Biết cách sử dụng các thư viện để tương tác / làm việc với các API của máy chủ. Hiểu được các nguyên tắc khi giao tiếp giữa một ứng dụng di động với các tài nguyên của máy chủ, bao gồm việc tìm nạp dữ liệu từ máy chủ, quản lý thông tin/xác thực người dùng, quản lý dữ liệu ứng dụng. Hiểu và biết cách sử dụng Modal để trình bày một nội dung phía trên phần còn lại của màn hình. Luyện tập cách kiểm soát khả năng hiển thị tiêu đề của các màn hình khi lồng nhau.

1.2. Mô tả

Trong bài tập này, chúng ta sẽ xây dựng một ứng dụng thương mại điện tử di động. Ứng dụng sẽ bao gồm các màn hình như trang chủ, thể loại, giỏ hàng và tài khoản. Chúng ta sẽ thực hiện việc giao tiếp với máy chủ để tìm nạp và hiển thị các thông tin như sản phẩm, thể loại, giỏ hàng, hồ sơ cá nhân, đồng thời là các xử lý liên quan như xác thực người dùng, chỉnh sửa hồ sơ và cập nhật nội dung giỏ hàng. Nguồn dữ liệu: Dữ liệu API của bài tập này sẽ được lấy từ trang Fake Store API, địa chỉ: <https://fakestoreapi.com/>.

1.3. Yêu cầu

- Nhóm màn hình xác thực (gồm Màn hình đăng nhập và Màn hình đăng ký) có giao diện tương tự như BTTH2 – Bài tập 3. Khi người dùng nhấn đăng nhập, kiểm tra tên người dùng và mật khẩu đã nhập, nếu đúng, thực hiện điều hướng đến nhóm màn hình chính của ứng dụng. Ngược lại, hiển thị một thông báo lỗi.
- Nhóm màn hình chính: Sau khi đăng nhập thành công, người dùng sẽ được chuyển hướng đến nhóm màn hình chính, bao gồm các màn hình sau:
 - Màn hình trang chủ
 - Màn hình thể loại
 - Màn hình giỏ hàng
 - Màn hình tài khoản



- Màn hình trang chủ. Gồm có các thành phần sau:
 - Một văn bản thể hiện khẩu hiệu của ứng dụng (Có thể đặt theo sở thích).
 - Một danh sách các biển hiệu (Banner) của ứng dụng, ngụ ý rằng đây là hình ảnh quảng cáo cho các sản phẩm, các biển hiệu này sẽ tự động chuyển đổi (Băng truyền hình ảnh) sau một khoảng thời gian cố định (Ví dụ 1 giây).
Lưu ý: API sử dụng không cung cấp các hình ảnh biển hiệu. Do đó, các hình ảnh này có thể lấy theo sở thích nhưng cần đảm bảo đủ số lượng với số lượng thể loại có trong dữ liệu và thể hiện đúng với các sản phẩm đang có.
 - Hai danh sách sản phẩm thuộc về hai loại tương ứng là Hot Deals và New Arrivals. Các sản phẩm trong hai danh sách này đều được chia thành hai cột.
Lưu ý: API sử dụng không phân chia các sản phẩm theo loại Hot Deals hay New Arrivals. Do đó, các sản phẩm trong hai danh sách này có thể lấy theo logic tùy ý.
 - Mỗi sản phẩm sẽ bao gồm các thông tin như hình ảnh, tên, giá tiền, kết quả đánh giá (số sao), số lượt đánh giá và một biểu tượng thêm.
 - Khi nhấn vào biểu tượng thêm, nếu sản phẩm chưa có trong giỏ hàng của người dùng, thực hiện việc thêm sản phẩm vào giỏ hàng đồng thời cập nhật số huy hiệu bên dưới Navigator tương ứng với số sản phẩm có trong giỏ hàng. Ngược lại, hiển thị một thông báo cho người dùng biết sản phẩm đã có trong giỏ hàng.
 - Khi nhấn vào một sản phẩm, thực hiện điều hướng đến màn hình chi tiết sản phẩm. Tại đây, thể hiện tên của sản phẩm trên thanh tiêu đề của màn hình, bên dưới là hình ảnh sản phẩm và các thông tin liên quan như tên, mô tả, giá tiền và đánh giá (số sao và số lượt đánh giá).
- Màn hình thể loại. Được dùng để hiển thị danh sách các danh mục sản phẩm và các sản phẩm trong danh mục đã chọn. Bao gồm:
 - Một danh sách các danh mục để người dùng lựa chọn. Danh sách này có hướng ngang và người dùng có thể cuộn danh sách này để lựa chọn loại theo ý muốn. Mỗi danh mục sẽ bao gồm một hình ảnh minh họa và tên danh mục.



Lưu ý: API sử dụng không cung cấp các hình ảnh cho từng danh mục. Do đó, các hình ảnh/vector-icon này có thể lấy theo sở thích nhưng cần đảm bảo phù hợp với loại danh mục tương ứng.

- Khi chọn vào một danh mục, các sản phẩm thuộc danh mục đó sẽ được hiển thị ra bên dưới tương ứng. Danh sách các sản phẩm này có giao diện và chức năng tương tự như ở màn hình trang chủ (Chia hai cột, Hiển thị các thông tin của mỗi sản phẩm, Nhấn vào sẽ điều hướng đến màn hình chi tiết sản phẩm, Nhấn chọn thêm sẽ kiểm tra giỏ hàng để tiến hành thêm hoặc hiển thị thông báo).
- Ngoài các danh mục có trong dữ liệu trả về từ API, cần có thêm một danh mục thể hiện tất cả các sản phẩm đang có (Ví dụ danh mục All). Khi người dùng điều hướng đến màn hình thể loại, danh mục này sẽ được thể hiện đầu tiên (Ngụ ý rằng lúc ban đầu chưa có danh mục nào được chọn nên ta sẽ thể hiện tất cả các sản phẩm).

- Màn hình giỏ hàng:

- Hiển thị danh sách các sản phẩm có trong giỏ hàng tương ứng với người dùng đã đăng nhập.
- Mỗi sản phẩm trong danh sách sẽ gồm có tên sản phẩm, hình ảnh sản phẩm, đơn giá, số lượng, biểu tượng tăng, biểu tượng giảm, biểu tượng xóa và tổng tiền của sản phẩm đó.
- Bên dưới cùng của màn hình thể hiện tổng tiền của tất cả sản phẩm và một nút thanh toán (Chúng ta sẽ không thực hiện chức năng thanh toán ở bài tập này). Tổng tiền và nút thanh toán này sẽ nằm cố định bên dưới cùng của màn hình bất kể có bao nhiêu sản phẩm trong giỏ hàng, trừ trường hợp không có sản phẩm hoặc sản phẩm đã bị xóa hết.
- Khi nhấn vào biểu tượng xóa, hiển thị một Modal để người dùng lựa chọn. Modal sẽ bao gồm tiêu đề, một nút đồng ý và một nút hủy. Nếu người dùng chọn đồng ý, thực hiện việc xóa sản phẩm khỏi giỏ hàng (Bất kể số lượng hiện tại là bao nhiêu) và cập nhập lại tổng tiền, huy hiệu bên dưới Navigator tương ứng. Ngược lại, nếu người dùng chọn hủy, thực hiện việc thoát Modal.



- Khi nhấn vào biểu tượng tăng/giảm, thực hiện việc tăng/giảm số lượng của sản phẩm và cập nhật tổng tiền của sản phẩm tương ứng, tổng tiền của toàn bộ sản phẩm. Trường hợp số lượng sản phẩm chỉ còn một, khi ta nhấn vào biểu tượng giảm sẽ hiển thị Modal xác nhận tương tự như khi ta nhấn vào biểu tượng xóa.
- Trường hợp người dùng không có sản phẩm nào trong giỏ hàng hoặc tất cả sản phẩm trong giỏ đã được xóa hết, hiển thị một văn bản và một nút mua sắm ở giữa màn hình. Khi nhấn vào nút mua sắm này, điều hướng đến màn hình trang chủ.
- Lưu ý: Dữ liệu từ API của một số người dùng có thể có nhiều hơn một giỏ hàng, trong trường hợp này, chỉ thực hiện lấy thông tin giỏ hàng đầu tiên trong danh sách. Ngoài ra, sau khi thực hiện các thay đổi, thông tin về giỏ hàng của người dùng sẽ được cập nhật cho tất cả các màn hình liên quan.
- Màn hình tài khoản. Dùng để hiển thị các thông tin cá nhân của người dùng và các cài đặt khác nếu có, bao gồm các thành phần sau:
 - Một hình ảnh cá nhân của người dùng, bên cạnh là tên (Đầy đủ họ và tên) và ngoài cùng bên phải là một biểu tượng chỉnh sửa. Lưu ý: API sử dụng không cung cấp hình ảnh cá nhân của người dùng. Do đó, có thể lấy hình ảnh theo sở thích cá nhân.
 - Các trường thông tin như tên (Name), tên tài khoản (Username), thư điện tử (Email), số điện thoại (Phone), địa chỉ (Address – Gồm số nhà, tên đường, thành phố). Mỗi trường thông tin sẽ bao gồm một tiêu đề và nội dung tương ứng bên dưới.
 - Một nút đăng xuất (LOG OUT). Khi người dùng nhấn vào nút này, thực hiện đăng xuất khỏi ứng dụng (Quay lại màn hình xác thực).
 - Khi nhấn vào biểu tượng chỉnh sửa, điều hướng đến màn hình chỉnh sửa thông tin người dùng. Tại đây, các trường thông tin sẽ được hiển thị gồm tiêu đề và giá trị thông tin cá nhân đã có tương ứng. Sau khi người dùng chỉnh sửa và chọn vào biểu tượng lưu ở trên thanh tiêu đề, thực hiện việc cập nhật dữ liệu thông tin của người dùng và quay trở lại màn hình tài khoản.



- Ở các màn hình, khi dữ liệu chưa được tìm nạp hoàn tất từ API, hiển thị một biểu tượng đang tải (Loading) cho người dùng nhận biết.
- Gom các components lặp lại / tương tự nhau (Nếu có) thành một custom component để tái sử dụng.
- Tùy chỉnh giao diện của ứng dụng cho bắt mắt và thân thiện với người dùng. Ví dụ, sử dụng các icons để xử lý các TabBarItem, làm nổi bật danh mục được chọn trong danh sách thể loại, v.v.

1.4. Gợi ý

- Tạo mới dự án và thiết lập các điều hướng cơ bản cho ứng dụng.
- Sử dụng React Context để quản lý thông tin đăng nhập và các thông tin chung của toàn bộ ứng dụng.
- Khi người dùng đăng nhập, API sẽ trả về một token nếu đăng nhập thành công, lúc này, thực hiện việc lưu token vào Context. Ở màn hình xác thực, kiểm tra thông tin token của người dùng, nếu có, điều hướng đến nhóm màn hình chính.
- Sử dụng thư viện **fetch** hoặc **axios** để tương tác với các API.
- Màn hình trang chủ:
 - Sử dụng package “**react-native-reanimated-carousel**” hoặc “**react-native-snap-carousel**” để triển khai băng truyền hình ảnh cho các biển hiệu.
 - Tìm nạp dữ liệu sản phẩm từ API và triển khai logic cho hai danh sách sản phẩm bên dưới.
 - Sử dụng FlatList kết hợp với Flexbox Layout để thực hiện việc chia hai cột.
- Màn hình chi tiết sản phẩm:
 - Khi nhấn vào một sản phẩm ở màn hình trang chủ, truyền tham số là sản phẩm được nhấn. Ở màn hình chi tiết, nhận thông tin chi tiết về sản phẩm dưới dạng tham số thông qua “**route.params**” của React Navigation. Đồng thời, lấy tên sản phẩm trích xuất được từ route.params để thiết lập cho tiêu đề.
- Màn hình thể loại:
 - Tạo các biến trạng thái lưu trữ thông tin danh sách các thể loại, danh sách sản phẩm và thể loại được chọn tương ứng.
 - Khi màn hình được hiển thị, tìm nạp tất cả sản phẩm và thể loại từ API.



- Sử dụng ScrollView hoặc FlatList để xử lý danh sách ngang cho các thẻ loại. Mỗi loại trong danh sách là một đối tượng có thể nhấn (Pressable/TouchableOpacity). Khi nhấn chọn vào một thẻ loại, tìm nạp dữ liệu sản phẩm thuộc thẻ loại tương ứng. Trường hợp thẻ loại được chọn là “All”, ta tìm nạp tất cả các sản phẩm.
- Màn hình giỏ hàng:
 - Tạo các biến trạng thái lưu danh sách sản phẩm, tổng tiền.
 - Tìm nạp danh sách các sản phẩm có trong giỏ hàng của người dùng. Ứng với mỗi sản phẩm, thực hiện việc tính tổng tiền bằng số lượng nhân với đơn giá. Sau đó thực hiện tính tổng tiền tất cả sản phẩm.
 - Sử dụng FlatList để hiển thị danh sách sản phẩm có trong giỏ hàng, khi nhấn vào biểu tượng tăng/giảm, thực hiện gọi hàm thay đổi số lượng sản phẩm tương ứng.
 - Trong hàm thay đổi số lượng sản phẩm, tìm sản phẩm cần cập nhật và thực hiện cập nhật sản phẩm với số lượng mới, đồng thời tính toán lại số tiền của sản phẩm cũng như tổng tiền. Sau khi thay đổi, gửi một yêu cầu PUT đến API để cập nhật.
 - Sử dụng Flexbox Layout để cố định văn bản tổng tiền và nút thanh toán ở dưới cùng của màn hình.
 - Tạo một biến trạng thái kiểm soát việc đóng/mở của Modal. Tạo một Modal gồm một văn bản tiêu đề và hai văn bản có thể nhấn. Khi nhấn chọn đồng ý, thực hiện gọi làm xóa sản phẩm với id của sản phẩm được nhấn. Ngược lại, gọi hàm đóng Modal. Khi nhấn vào biểu tượng xóa hoặc khi giảm số lượng sản phẩm từ 1 về 0, ta sẽ hiển thị Modal.
 - Trong hàm xóa sản phẩm, kiểm tra xem có phải sản phẩm cuối cùng không, nếu có, thực hiện việc xóa giỏ hàng và gửi một yêu cầu DELETE đến API để cập nhật. Ngược lại, xóa sản phẩm đã chọn, tính toán lại tổng tiền của các sản phẩm còn lại trong giỏ hàng và gửi một yêu cầu PUT đến API để cập nhật.

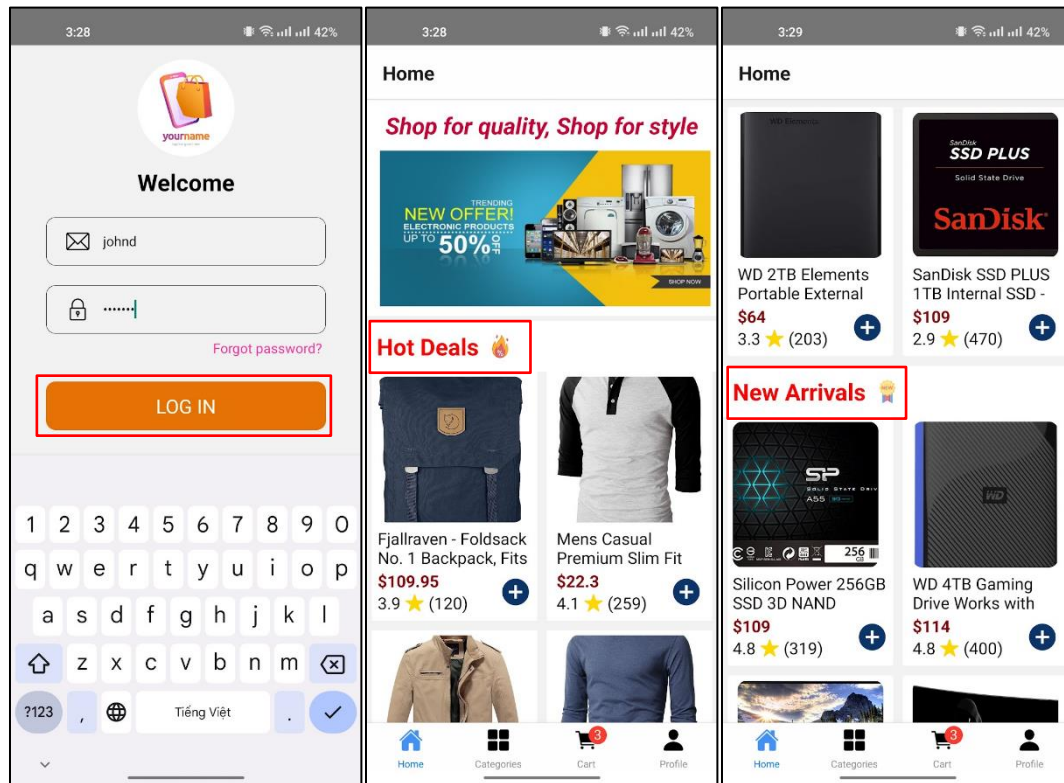


- Màn hình tài khoản:
 - Sử dụng thư viện **jwt-decode** để lấy id của người dùng từ token trả về. Từ id lấy được, thực hiện lệnh gọi API để tìm nạp dữ liệu thông tin cá nhân của người dùng. Khi nhấn vào biểu tượng chỉnh sửa, điều hướng đến màn hình chỉnh sửa cùng với tham số là dữ liệu thông tin người dùng.
 - Trong màn hình chỉnh sửa thông tin người dùng, hiển thị một biểu tượng cập nhật trên thanh Header (sử dụng props headerRight trong options của Screen trong Navigator). Khi nhấn vào biểu tượng này, cập nhật lại thông tin người dùng với dữ liệu mới, gửi một yêu cầu PUT đến API và quay lại màn hình tài khoản.
 - Khi nhấn đăng xuất, thực hiện việc xóa token khỏi Context.
- Đọc kỹ tài liệu được cung cấp trong API để biết cách gọi/truyền dữ liệu phù hợp. Xem thêm về fetch / axios từ trang chủ của React Native / MDN Web Docs để nắm các khái niệm và cách sử dụng.
- Có thể sử dụng công cụ Postman / Insomnia để dễ dàng nắm bắt việc thực hiện các yêu cầu API đến Server.
- Đảm bảo ứng dụng xử lý các lỗi tiềm ẩn khi tương tác với API và cung cấp phản hồi phù hợp cho người dùng.
- Đảm bảo ứng dụng hoạt động khi đăng nhập với các người dùng khác nhau (Ví dụ user1 có 2 giỏ hàng, user2 có 1 giỏ hàng với 3 sản phẩm, user3 không có giỏ hàng). Tìm đọc thông tin các người dùng có giỏ hàng tương ứng trong API để kiểm tra.
- Chú ý vấn đề sử dụng các danh sách có thể cuộn lồng nhau (ScrollView, FlatList).
- Sử dụng hàm “**getFocusedRouteNameFromRoute**” để kiểm soát việc hiển thị thanh header khi điều hướng từ màn hình ngoài vào màn hình chi tiết sản phẩm.
- Sử dụng kết hợp các kỹ thuật như useEffect, Promise, Async/Await để nâng cao hiệu suất và trải nghiệm của người dùng.

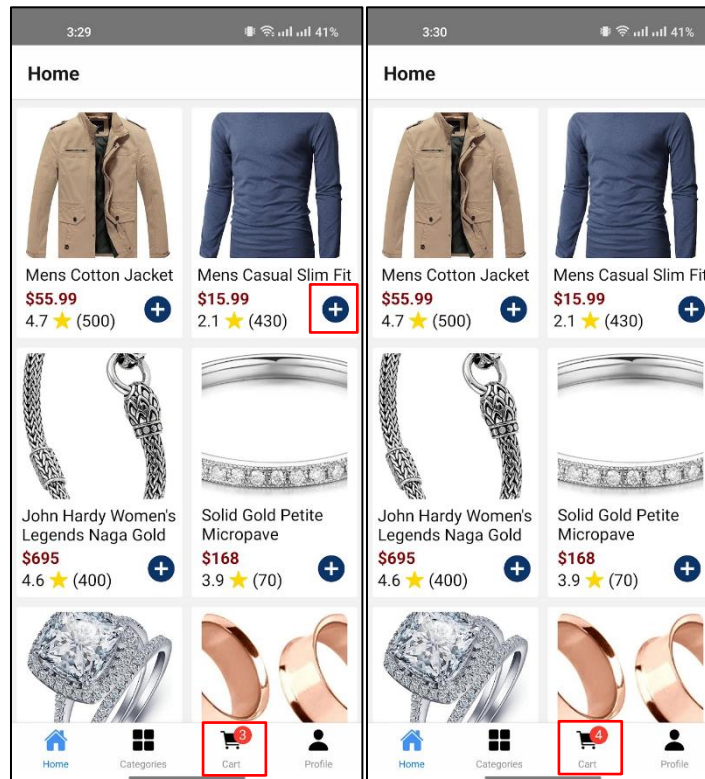


1.5. Hình ảnh ví dụ

- Màn hình trang chủ với khẩu hiệu, biểu ngữ hình ảnh và hai danh sách sản phẩm.

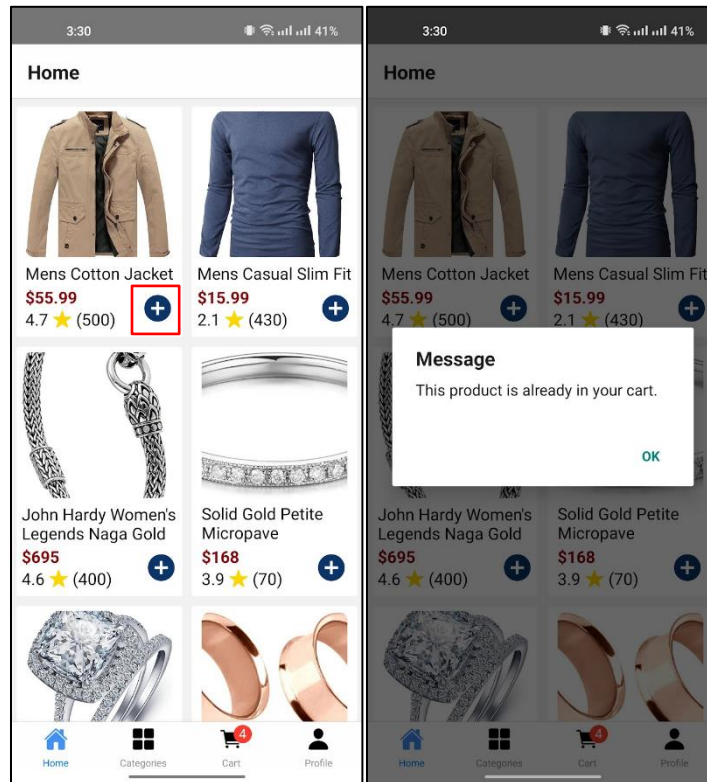


- Thêm sản phẩm vào giỏ hàng, giỏ hàng sẽ được cập nhật nếu đó là sản phẩm mới.

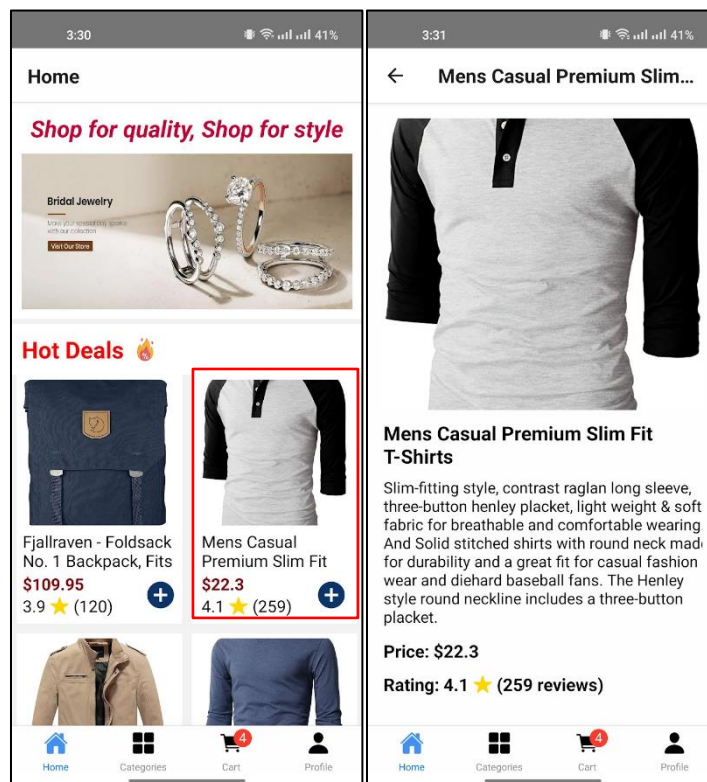




- Thêm sản phẩm vào giỏ hàng, hiển thị thông báo nếu đó là sản phẩm đã có.

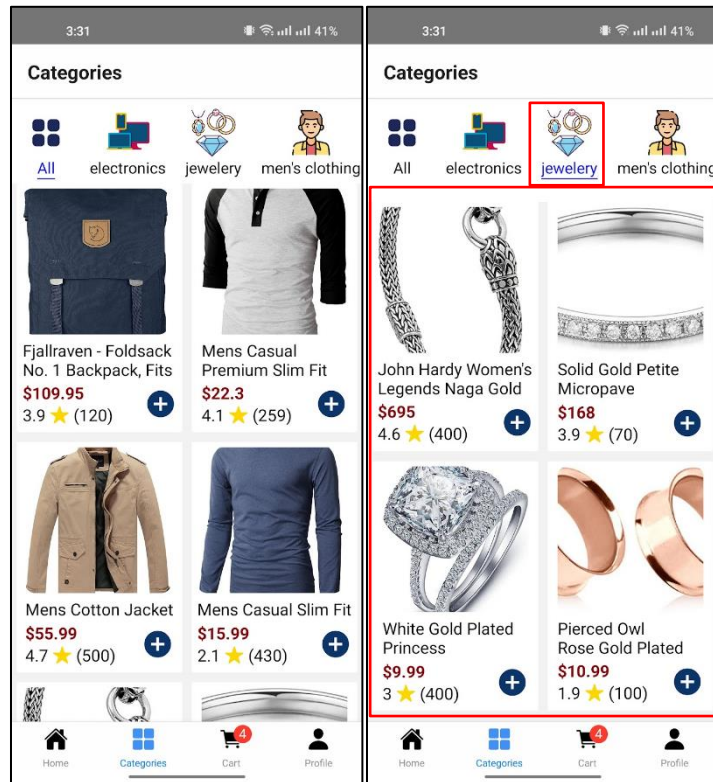


- Điều hướng đến màn hình chi tiết sản phẩm khi nhấn vào một sản phẩm.

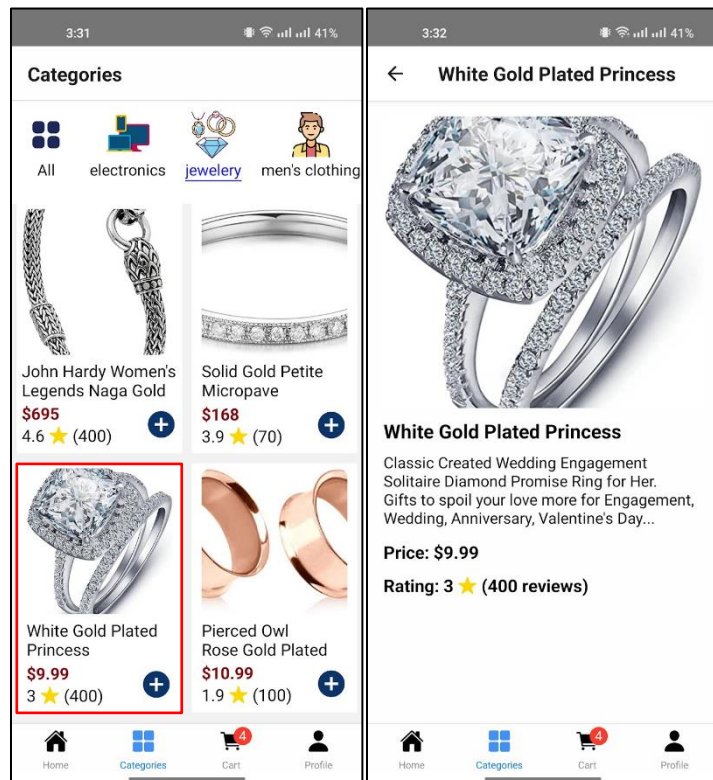




- Màn hình thể loại và danh sách các sản phẩm tương ứng khi chọn một loại.

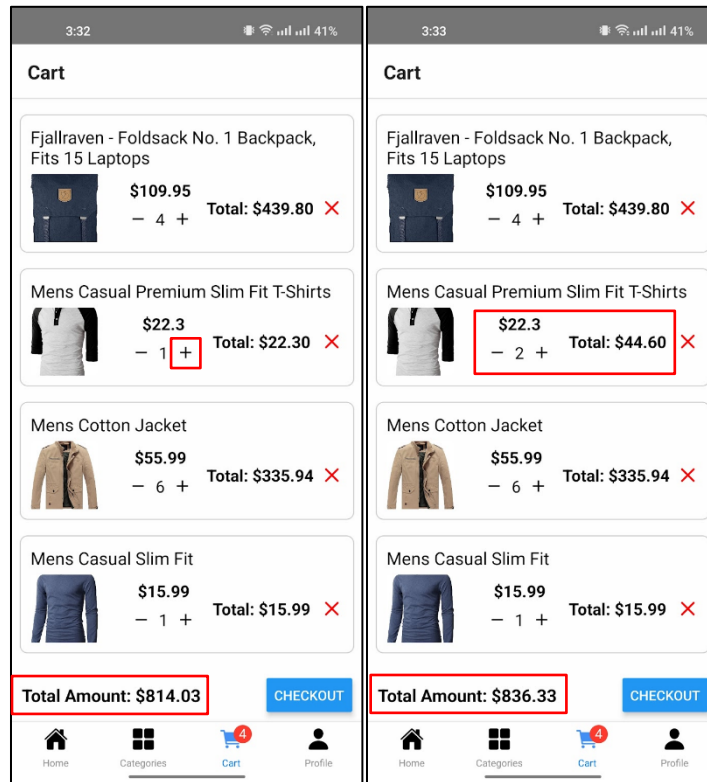


- Điều hướng đến màn hình chi tiết sản phẩm khi nhấn vào một sản phẩm.

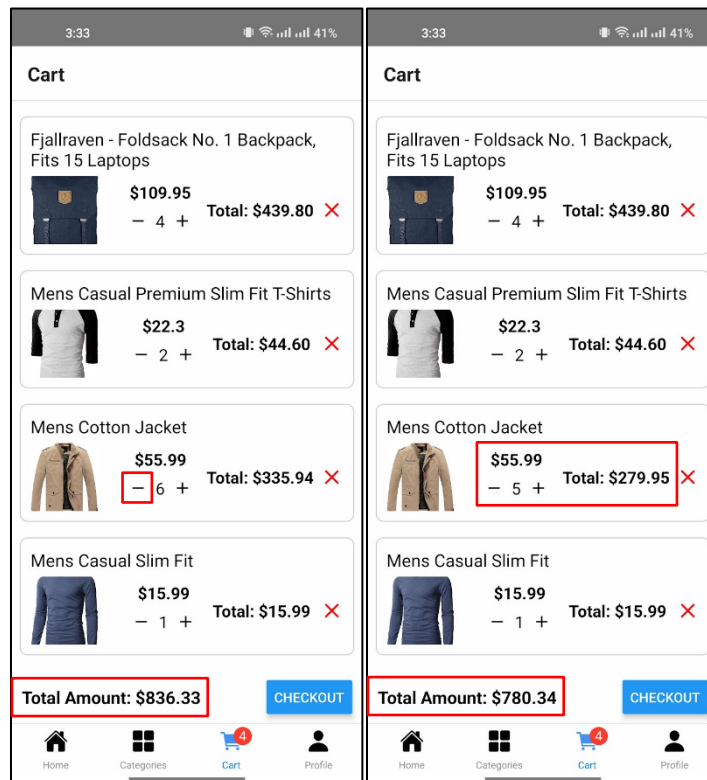




- Màn hình giỏ hàng và khi tăng số lượng một sản phẩm.

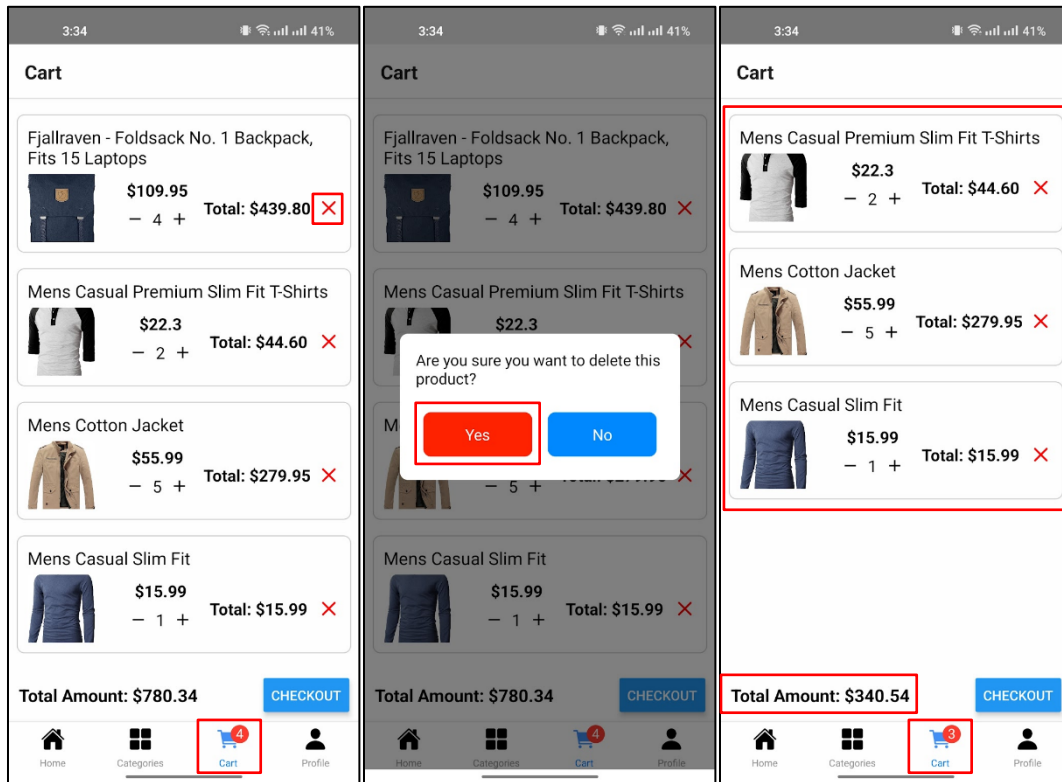


- Màn hình giỏ hàng và khi giảm số lượng một sản phẩm.

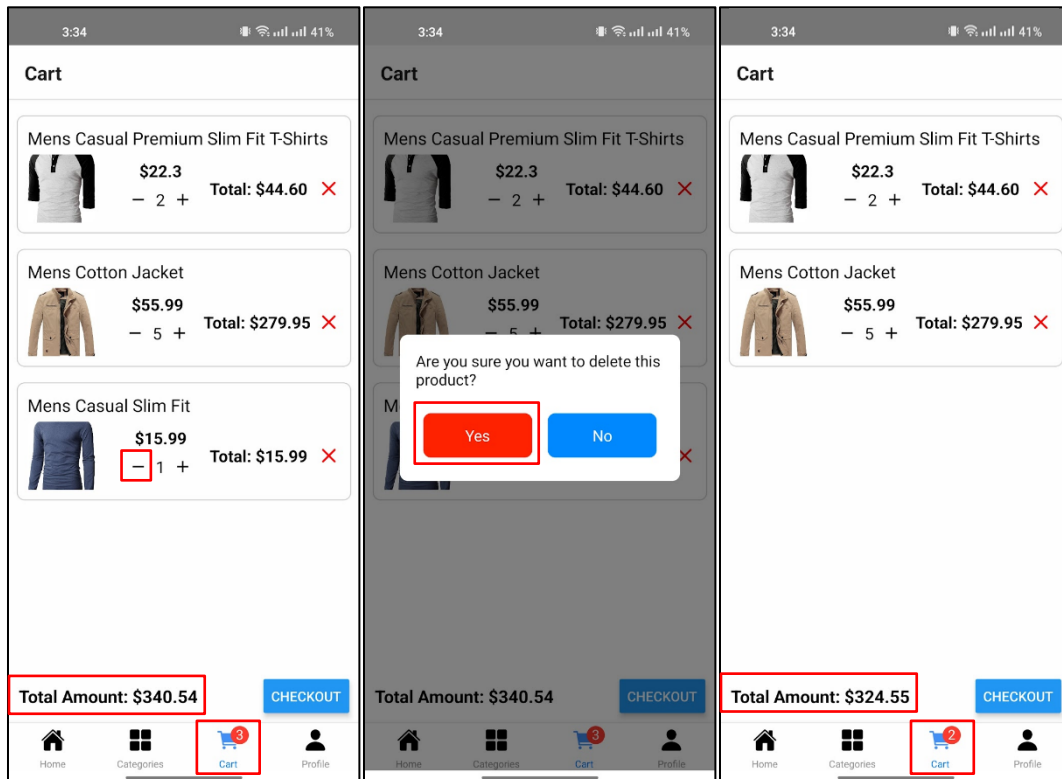




- Khi xóa một sản phẩm, hiển thị Modal và nếu chọn đồng ý, giỏ hàng được cập nhật.

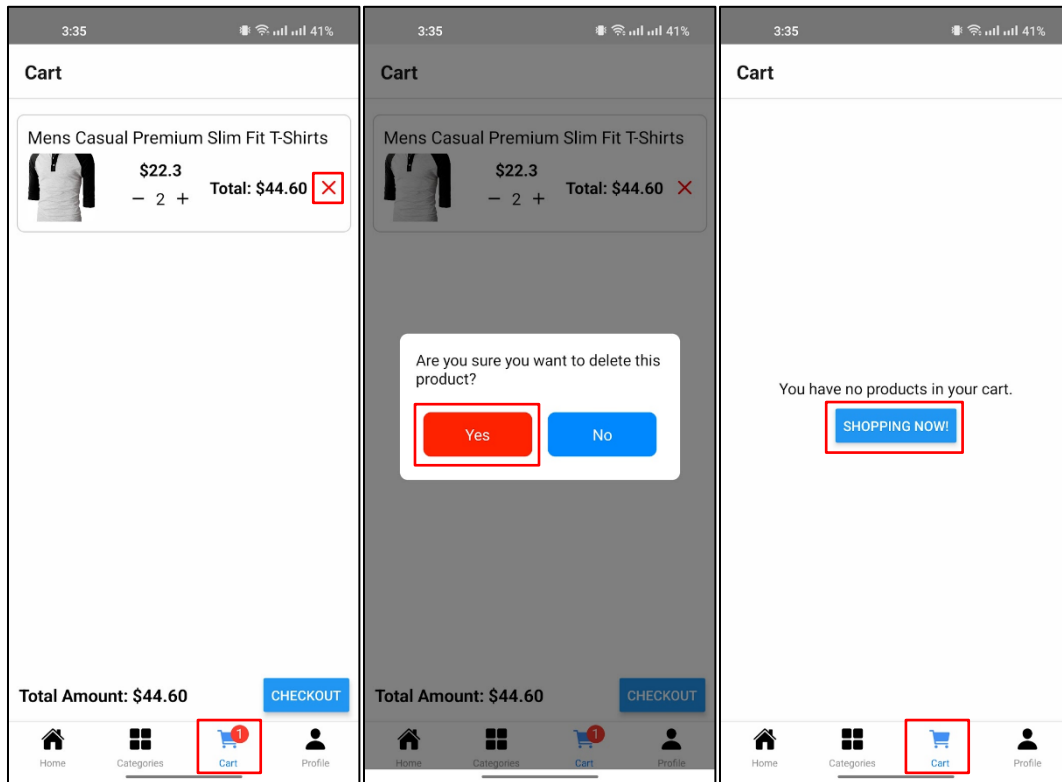


- Modal cũng sẽ được hiển thị nếu thực hiện giảm sản phẩm mà đang có số lượng là 1. Modal có chức năng tương tự như khi chọn từ biểu tượng xóa.

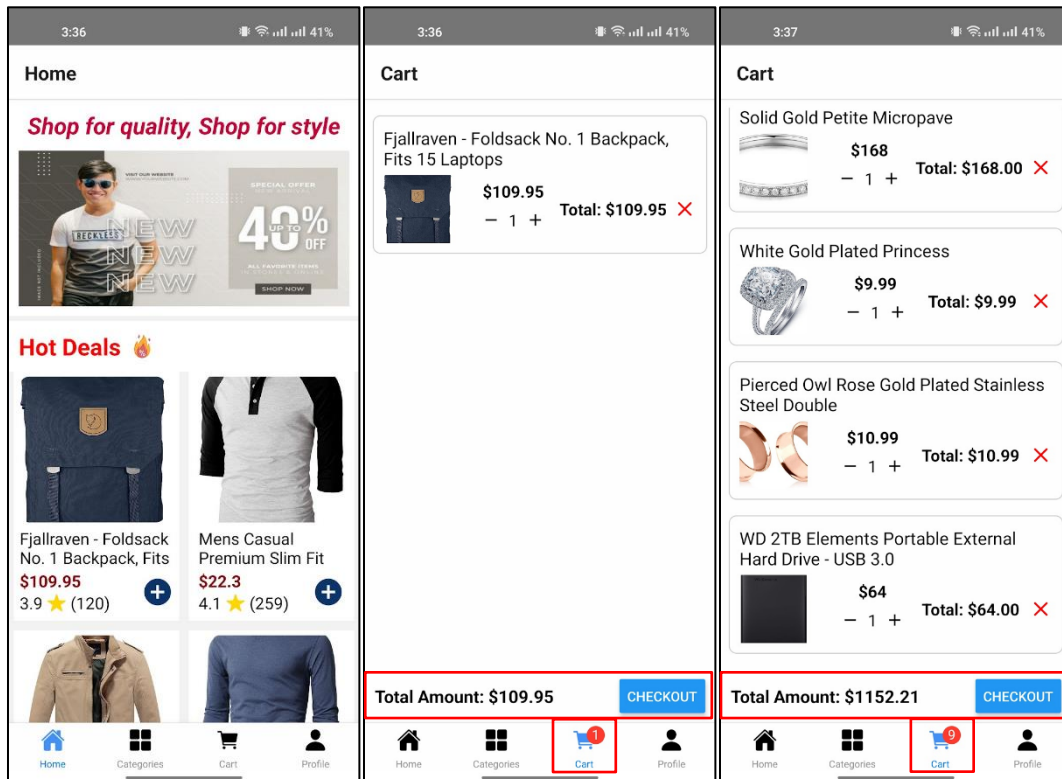




- Khi sản phẩm cuối cùng bị xóa, hiển thị một văn bản và một nút mua sắm.



- Điều hướng về trang chủ khi nhấn vào nút mua sắm. Tổng tiền và nút thanh toán sẽ được nằm cố định ở dưới cùng bất kể số lượng sản phẩm (Trừ khi bằng 0).





- Màn hình tài khoản. Khi nhấn vào biểu tượng chỉnh sửa, chuyển đến màn hình chỉnh sửa thông tin người dùng với các thông tin đã có.

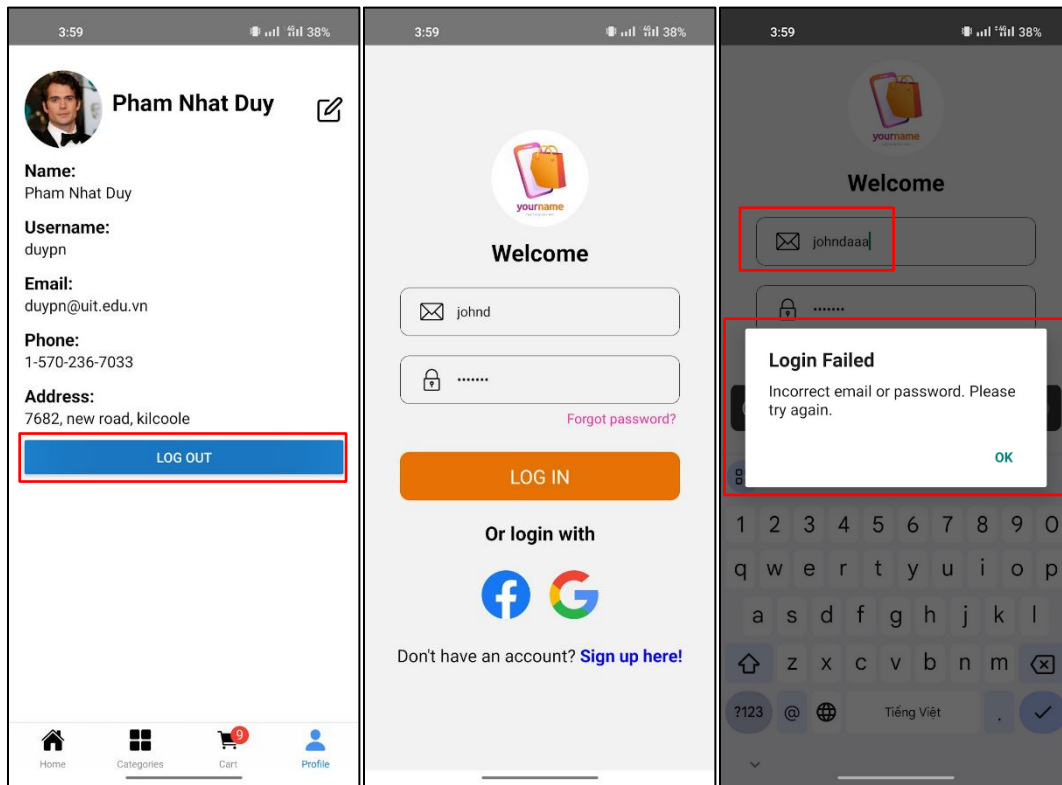
The first screenshot shows the user profile page for 'John Doe'. It includes fields for Name, Username, Email, Phone, and Address. A red box highlights the edit icon (a pencil) in the top right corner. The second screenshot shows the 'EditProfile' screen. It contains input fields for First Name, Last Name, Username, Email, Phone Number, House Number, Street, and City. A red box highlights the entire form area.

- Thực hiện chỉnh sửa và chọn cập nhật, các thông tin sẽ được thay đổi tương ứng.

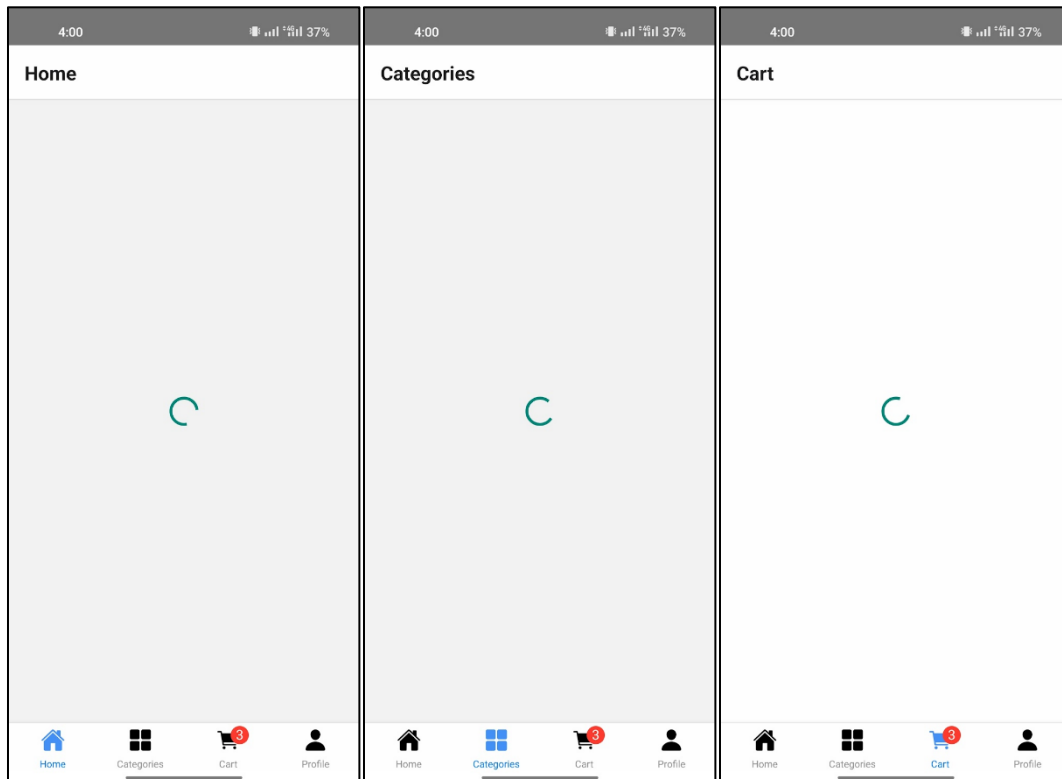
The first screenshot shows the 'EditProfile' screen with a keyboard open. Red boxes highlight the 'First Name' field (containing 'Pham Nhat'), the 'Last Name' field (containing 'Duy'), the 'Username' field (containing 'duypn'), and the 'Email' field (containing 'duypn@uit.edu.vn'). The second screenshot shows the updated profile page for 'Pham Nhat Duy'. Red boxes highlight the new name, the updated username 'duypn', and the updated email 'duypn@uit.edu.vn'.



- Đăng xuất khỏi ứng dụng và thông báo lỗi khi điền sai thông tin đăng nhập (Bên dưới là lỗi trả về từ server với mã lỗi 401).



- Các biểu tượng đang tải ở các màn hình khi chưa tìm nạp xong dữ liệu từ API.





HƯỚNG DẪN NỘP BÀI

- File nộp gồm 2 phần: PDF và ZIP.

- **PDF:**

- Chụp màn hình các mục sau đây và convert sang file pdf, đặt tên: **MSSV_BTTHX.pdf**.
 - Cấu trúc các folders / files của dự án
 - Toàn bộ nội dung code của các file liên quan (tức phần thực hiện) và Màn hình kết quả của các câu. Trong đó, màn hình code phải thể hiện thông tin của cá nhân gồm **Tên và MSSV**.

- **ZIP:**

- Nén source code lại thành file zip, đặt tên: **MSSV_BTTHX.zip**
 - Nếu sử dụng Snack Expo: Export project trực tiếp.
 - Nếu sử dụng VS Code hoặc các IDEs/Text Editors khác: Bỏ chọn thư mục **node_modules** trước khi nén. Trường hợp sử dụng React Native CLI cần xóa thêm thư mục **build** tương ứng với nền tảng trước khi nén. Đường dẫn của thư mục build sẽ là: **Tên_Project/android/app/build** nếu chạy trên Android, **Tên_Project/ios/build** nếu chạy trên IOS.

- **Lưu ý:**

- X là số thứ tự của bài thực hành.
- File PDF không nằm trong file nén ZIP.
- Màn hình kết quả là:
 - Nếu sử dụng Expo: Màn hình được chụp trên thiết bị thật của cá nhân
 - Nếu sử dụng React Native CLI:
 - Màn hình được chụp trên thiết bị thật của cá nhân
 - Hoặc, Màn hình được chụp trên Android Virtual Device (Android Studio) đối với môi trường Android
 - Hoặc, Màn hình được chụp trên Xcode đối với môi trường IOS