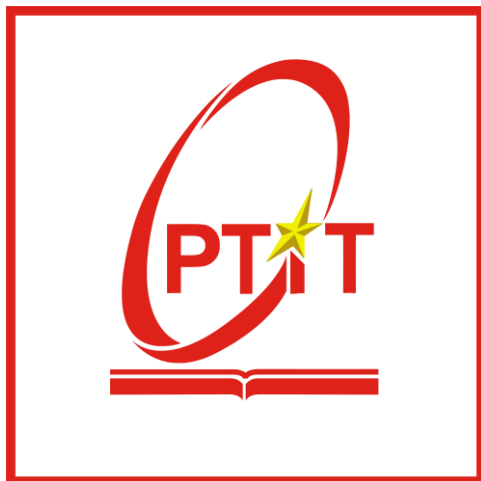


HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

---



## Báo cáo hàng tuần

Môn học: Thực tập cơ sở

Giảng viên: Kim Ngọc Bách

Họ tên: Trần Quang Anh

Mã SV: B22DCCN044

Lớp: E22CQCN04-B

Tuần: 9

# Báo cáo thực tập cơ sở - Tuần 9: Tích hợp redux, cập nhật trang recipe detail

## I. Tích hợp Redux và Container để quản lý state

- Thiết lập Redux cho dự án:
  - Đã tích hợp Redux vào dự án React để quản lý trạng thái (state) tập trung, giúp các thành phần giao diện truy cập và cập nhật dữ liệu một cách linh hoạt. Cấu trúc Redux bao gồm:
    - Store: Được tạo bằng configureStore từ @reduxjs/toolkit, lưu trữ toàn bộ trạng thái của ứng dụng.
    - Reducers: Tạo các slice (ví dụ: recipeFilter) để quản lý danh sách công thức, bộ lọc, và trạng thái người dùng.
    - Actions: Xây dựng các action như fetchRecipes, setFilters, và selectRecipe để xử lý dữ liệu từ API và cập nhật state.
  - Sử dụng react-redux để kết nối các thành phần React với Redux store thông qua Provider và các hook như useSelector, useDispatch.
- Tạo Container components:
  - Đã phát triển các container components (ví dụ: RecipesList, FilterBars) để tách biệt logic xử lý state và giao diện. Các container này:
    - Lấy state từ setFilter của action và reducers để xử lý
    - FilterBars sẽ xử lý dispatch tạo ra dữ liệu filter
    - FilterList sẽ xử lý useSelector để thể hiện dữ liệu
- Cải thiện hiệu suất:
  - Áp dụng createSelector từ @reduxjs/toolkit để memoize các phép chọn state phức tạp, giảm thiểu tính toán không cần thiết khi lọc hoặc hiển thị công thức.

## II. Phát triển trang chi tiết công thức

- Thiết kế giao diện trang chi tiết:
  - Đã xây dựng trang chi tiết công thức (RecipeDetailPage) để hiển thị thông tin đầy đủ của một công thức khi người dùng nhấp vào nút "Xem chi tiết" trên thẻ công thức. Trang này bao gồm:
    - Hình ảnh món ăn: Hiển thị image\_url ở đầu trang với kích thước lớn, tối ưu cho cả desktop và mobile.
    - Tên món ăn: Hiển thị recipe\_name với font chữ lớn, rõ ràng.
    - Thông tin cơ bản: Bao gồm thời gian chuẩn bị (prep), thời gian nấu (cook), số calo (calories), và số lượng nguyên liệu (ingredients.length).

- Danh sách nguyên liệu: Hiển thị dưới dạng danh sách có dấu đầu dòng, với mỗi nguyên liệu được lấy từ mảng ingredients.
- Hướng dẫn nấu: Hiển thị các bước nấu từ trường instructions dưới dạng danh sách được đánh số, đảm bảo dễ đọc.
- Thông tin dinh dưỡng: Trích xuất từ trường nutritions, hiển thị các giá trị như protein, chất béo, carbohydrate (nếu có).
- Giao diện được thiết kế responsive, sử dụng Tailwind CSS để đảm bảo bố cục gọn gàng trên mọi thiết bị.
- Quản lý state cho trang chi tiết:
  - Sử dụng Redux để lưu trữ thông tin công thức được chọn (selectedRecipe). Khi người dùng nhấp vào một công thức, container dispatch action selectRecipe với ID công thức, sau đó gọi API để lấy dữ liệu chi tiết từ server Node.js.
  - Dữ liệu công thức được lưu trong Redux store, cho phép tái sử dụng khi người dùng quay lại trang chi tiết mà không cần gọi lại API.
- Tối ưu trải nghiệm người dùng:
  - Thêm nút "Quay lại" để đưa người dùng về trang danh sách công thức mà không làm mất trạng thái bộ lọc.
  - Xử lý trường hợp công thức không tồn tại (ví dụ: ID không hợp lệ) bằng cách hiển thị thông báo "Công thức không tìm thấy" và gợi ý quay lại trang chính.
  - Tích hợp loading spinner khi dữ liệu chi tiết đang được tải từ server, cải thiện trải nghiệm trong trường hợp kết nối chậm.