

**BỘ THÔNG TIN & TRUYỀN THÔNG  
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**



**PHÁT TRIỂN HỆ THỐNG THƯƠNG  
MẠI ĐIỆN TỬ**

**Đề tài: Website bán hàng thời trang**

**Giảng**            Kim Ngọc Bách

**Nhóm**         : 16

**Sinh viên** : Chu Minh Tiến - B18DCCN529  
               Bùi Thị Thu - B21DCCN697  
               Nguyễn Thu Huyền - B21DCCN445

**Hà Nội - 2025**

## LỜI CẢM ƠN

Lời đầu tiên, nhóm 16 chúng em xin gửi lời cảm ơn sâu sắc đến giảng viên bộ môn-thầy Kim Ngọc Bách đã tận tâm trong việc dạy dỗ, truyền đạt những kiến thức quý báu cho chúng em trong suốt thời gian học tập vừa qua. Trong thời gian tham gia bộ môn Phát triển hệ thống thương mại điện tử, bọn em đã được tiếp thu những kiến thức giúp chúng em hiểu rõ hơn về quy trình phát triển hệ thống thông tin trong thực tế. Cảm ơn thầy đã giúp chúng em tiếp cận một cách dễ hơn với những kiến thức khó nhọc này. Mặc dù kiến thức bộ môn rất trừu tượng nhưng với cách giảng dạy đầy tâm huyết, cùng với những ví dụ minh họa dễ hiểu, cách dẫn dắt logic của thầy đã giúp chúng em có thể hiểu bài và tích lũy cho bản thân kiến thức cần có về phát triển hệ thống thông tin quản lý. Một lần nữa chúng em xin cảm ơn về những bài giảng vô cùng hay và tâm huyết của thầy.

Nhóm chúng em đã cùng nhau cố gắng từ bước phân tích chủ đề đến tìm hiểu, rồi cùng nhau xây dựng lên bài tập lớn về chủ đề này. Tuy nhiên thì chắc chắn chúng em vẫn sẽ còn nhiều hạn chế và khả năng tiếp thu thực tế còn nhiều sai sót. Mặc dù nhóm đã cố gắng hết sức nhưng chắc chắn báo cáo khó có thể tránh khỏi những thiếu sót và nhiều chỗ còn chưa chính xác, mong thầy xem xét và góp ý để bài tập lớn của nhóm được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

## **Mục lục**

I. Tổng quan về dự án .....	4
1. Giới thiệu dự án .....	4
2. Mô hình kinh doanh .....	4
3. Mô hình doanh thu .....	4
II. Cơ sở lý thuyết và Công nghệ sử dụng .....	4
1. HTML, CSS, JavaScript .....	5
2. Tailwind CSS .....	5
3. React.js .....	6
4. Node.js .....	8
5. Express.js .....	8
6. Database MongoDB .....	10
III. Tính năng .....	10
1. Tạo tài khoản .....	10
2. Đăng nhập .....	10
3. Quản lý giỏ hàng .....	10
4. Tìm kiếm / khuyến nghị .....	10
5. Đặt hàng .....	10
6. Thanh toán .....	10
7. Quản lý đổi trả và hoàn tiền .....	11
8. Quản lý sản phẩm .....	11
9. Quản lý đơn hàng .....	11
10. Thống kê và báo cáo .....	11
IV. Kế hoạch thực thi, triển khai dự án .....	12

# I. Tổng quan về dự án

## 1. Giới thiệu dự án

- Tên dự án: Xây dựng website bán hàng thời trang.
- Dự án xây dựng một website thương mại điện tử nhằm tạo ra kênh mua sắm trực tuyến các mặt hàng thời trang. Mục tiêu chính là giúp cung cấp trải nghiệm mua sắm thuận tiện, an toàn và nhanh chóng cho người tiêu dùng đồng thời cũng giúp quản lý hiệu quả danh mục sản phẩm, đơn hàng và khách hàng. Các mặt hàng chủ yếu là sản phẩm thời trang nam/ nữ, phụ kiện; đối tượng khách hàng là người tiêu dùng cá nhân.

## 2. Mô hình kinh doanh

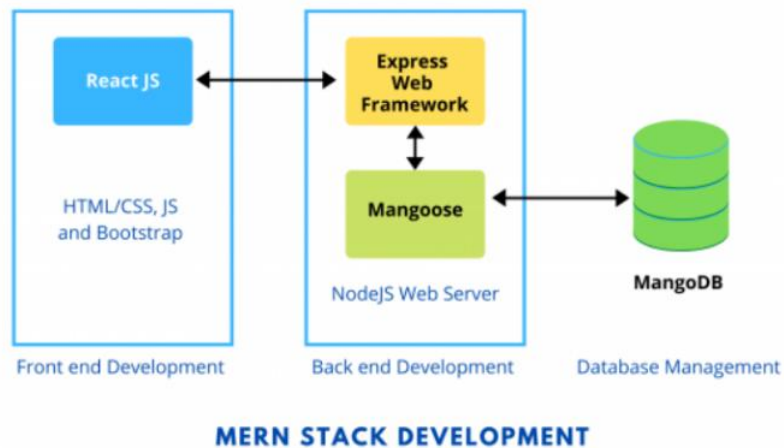
- Dự án áp dụng mô hình kinh doanh B2C (Business to Consumer)
- Mô hình kinh doanh này giúp doanh nghiệp bán sản phẩm trực tiếp đến khách hàng là người tiêu dùng cá nhân qua việc đặt hàng trực tuyến trên website hoặc ứng dụng di động.

## 3. Mô hình doanh thu

- Nguồn doanh thu chính sẽ đến từ bán hàng trực tiếp: Lợi nhuận từ việc chênh lệch giữa giá nhập và giá bán của sản phẩm.
- Hoặc để tăng doanh số và giá trị trung bình đơn hàng, có thể triển khai các chương trình combo ưu đãi như: mua 1 tặng 1, mua theo combo sản phẩm,...

# II. Cơ sở lý thuyết và Công nghệ sử dụng

Trong dự án này, đã nghiên cứu và sử dụng công nghệ MERN Stack để xây dựng trang web. MERN là một thuật ngữ rút gọn của MongoDB, Express.js, React.js và Node.js. Stack MERN là một stack Javascript được thiết kế để giúp phát triển ứng dụng web toàn ngăn xếp dễ dàng hơn và nhanh hơn. Tất cả bốn công nghệ này cung cấp một khuôn khổ hoàn chỉnh cho các nhà phát triển để tạo ra bất kỳ ứng dụng web nào. MERN đang tuân theo kiến trúc 3 tầng truyền thống, bao gồm tầng hiển thị front-end (React.js), tầng ứng dụng (Express.js và Node.js) và tầng cơ sở dữ liệu (MongoDB).



## 1. HTML, CSS, JavaScript

- **HTML (HyperText Markup Language)** là ngôn ngữ đánh dấu siêu văn bản được sử dụng để xây dựng và hiển thị các trang web trên Internet. Nó định nghĩa cấu trúc và ý nghĩa của nội dung trên trang web bằng cách sử dụng các thẻ và các thuộc tính.

- **CSS (Cascading Style Sheets)** là ngôn ngữ được sử dụng để mô tả cách mà các phần tử HTML được hiển thị trên trình duyệt. Nó cho phép lập trình viên điều chỉnh màu sắc, font chữ, khoảng cách, kích thước và nhiều thuộc tính khác của các phần tử trong trang web để tạo ra giao diện khách hàng hấp dẫn và thú vị. CSS giúp tách biệt phần nội dung của trang web và cách mà nó được hiển thị, giúp cho việc thiết kế và bảo trì trang web trở nên dễ dàng hơn.

- **JavaScript** là một ngôn ngữ lập trình phổ biến được sử dụng chủ yếu trong việc phát triển ứng dụng web. Được tích hợp trực tiếp vào trình duyệt web, JavaScript cho phép tương tác động và thay đổi nội dung của trang web một cách linh hoạt và đáp ứng. Nó được sử dụng để thêm các tính năng động như xử lý sự kiện, thao tác với DOM (Document Object Model), tạo và điều khiển hiệu ứng, gửi và nhận dữ liệu từ máy chủ, và nhiều nhiệm vụ khác. JavaScript là một phần không thể thiếu trong phát triển web hiện đại, giúp tạo ra các trang web có trải nghiệm khách hàng tốt hơn và tương tác hơn.

## 2. Tailwind CSS

**Tailwind CSS** giúp xây dựng website một cách nhanh chóng nhất với các thuộc tính CSS đã được gán thành những class riêng, khi dùng chúng ta chỉ có việc gọi ra để dùng. Đây là một utility-first CSS framework, nó cũng giống như Bootstrap, nó có những class built-in mà chúng ta có thể dùng. Tailwind CSS có nhiều các class bao gồm các thuộc tính CSS khác nhau và quan trọng, chúng ta có thể dễ dàng mở rộng tạo mới ra những class bằng chính những class của nó.

### 3. React.js

**React.js** là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, được sử dụng để xây dựng giao diện khách hàng cho các ứng dụng web động. React.js tập trung vào việc tạo ra các thành phần giao diện tái sử dụng (reusable UI components), giúp tăng tính tái sử dụng, hiệu quả và dễ bảo trì của mã nguồn. React.js hoạt động bằng cách sử dụng Virtual DOM và mô hình one-way data binding. Nó xây dựng giao diện khách hàng thành các thành phần, sử dụng JSX để mô tả cấu trúc. Khi dữ liệu hoặc trạng thái thay đổi, React kích thích Reconciliation để cập nhật chỉ những phần thay đổi trong DOM thực sự, tối ưu hóa hiệu suất ứng dụng web.

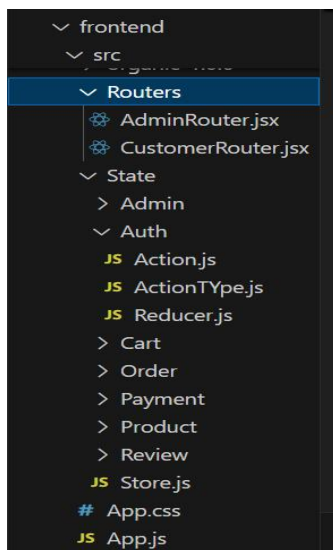
#### **Một số điểm nổi bật về React.js:**

- + Thành phần (Component): React.js xây dựng trên một kiến trúc dựa trên thành phần, trong đó mỗi phần của giao diện khách hàng được xem xét là một thành phần độc lập. Điều này giúp tạo ra mã nguồn gọn gàng, dễ hiểu và dễ bảo trì.
- + Virtual DOM (DOM Ảo): React.js sử dụng Virtual DOM để tối ưu hóa hiệu suất của ứng dụng. Virtual DOM là một bản sao của DOM thực sự, được React.js sử dụng để thực hiện các thay đổi trên giao diện khách hàng một cách hiệu quả mà không cần phải cập nhật DOM thực sự.
- + Tính năng JSX: JSX là một phần mở rộng của JavaScript cho phép viết HTML trong JavaScript, tạo ra mã nguồn dễ đọc và dễ bảo trì. JSX kết hợp HTML với JavaScript, giúp tạo ra các thành phần giao diện một cách dễ dàng.
- + Cộng đồng lớn và hỗ trợ mạnh mẽ: React.js có một cộng đồng lớn và mạnh mẽ, điều này có nghĩa là có nhiều tài liệu, hướng dẫn và thư viện bổ sung sẵn có để hỗ trợ việc phát triển ứng dụng React.

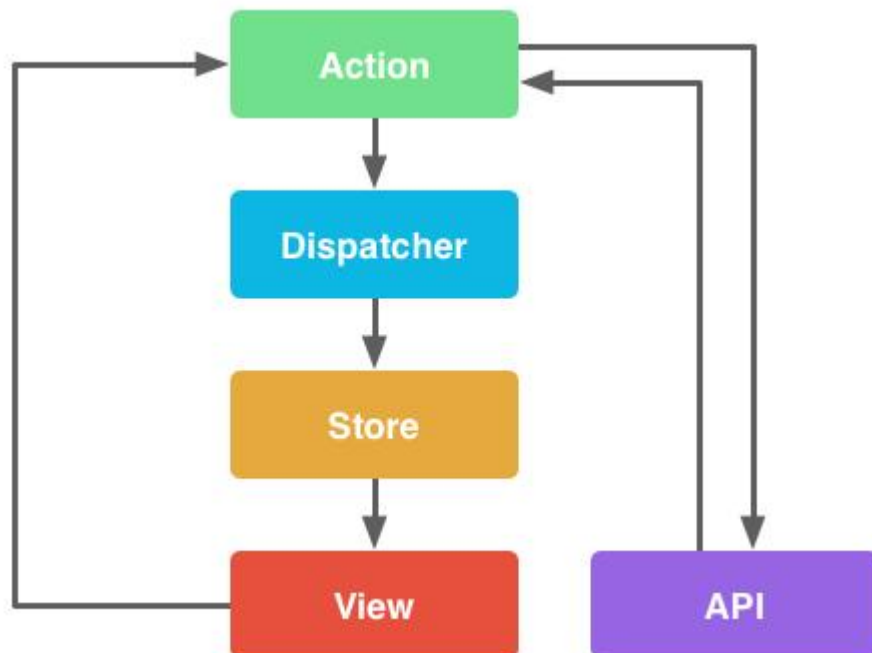
React.js chạy độc lập, song song với phía Server, để xây dựng tầng Giao diện người dùng. Như đã nói, đây là lý do chúng tôi cần CORS và cấu hình CORS một cách cụ thể, thuận tiện việc gọi API, cũng như đảm bảo tính bảo mật.

**Dưới đây là kiến trúc phía Client:**

- Cấu trúc file, folder:



- Kiến trúc Front-end sẽ là một tập các Route, được định tuyến đến các Page. Ngoài ra, nhóm sử dụng thêm thư viện Redux cho việc quản lý các State. Nhìn chung, kiến trúc phía Front-end của đề tài có thể mô tả đơn giản như mô hình dưới đây.



- Routes sẽ có tới 3 loại, phụ thuộc vào Authentication, Authorization của người dùng

+ Route: dành cho Khách, không cần đăng nhập

- + PrivateRoute: dành cho Người mua, Người bán, cần đăng nhập
- + AdminRoute: chỉ cho phép tài khoản Quản trị viên

## 4. Node.js

**Node.js** là một nền tảng runtime mã nguồn mở xây dựng trên JavaScript Engine (V8 Engine) của Google Chrome. Nó cho phép lập trình viên viết mã JavaScript để chạy ở phía máy chủ mà không chỉ ở môi trường trình duyệt web. Node.js cung cấp một môi trường thực thi cho việc phát triển các ứng dụng mạng hiệu quả và mạnh mẽ.

### Một số điểm nổi bật về Node.js:

- + Non-blocking và Asynchronous I/O: Node.js sử dụng mô hình không chặn (nonblocking) và I/O bất đồng bộ (asynchronous I/O), cho phép xử lý nhiều yêu cầu một cách hiệu quả và đồng thời mà không phải chờ đợi kết quả của mỗi yêu cầu trước đó hoàn thành.
- + Sự mở rộng dễ dàng: Node.js cho phép xây dựng các ứng dụng có khả năng mở rộng tốt nhờ vào mô hình sự kiện và luồng không đồng bộ. Điều này làm cho việc xử lý hàng trăm hoặc thậm chí hàng nghìn kết nối cùng một lúc trở nên khả thi và hiệu quả.
- + Phát triển nhanh chóng: Với cộng đồng lớn và sự phát triển liên tục, Node.js cung cấp nhiều công cụ và thư viện để giúp phát triển ứng dụng nhanh chóng và dễ dàng. Các framework như Express.js cũng giúp giảm bớt công việc lặp đi lặp lại và tăng hiệu suất phát triển.
- + Phù hợp cho ứng dụng thời gian thực: Node.js thích hợp cho việc xây dựng các ứng dụng thời gian thực như ứng dụng chat, trò chơi trực tuyến, các ứng dụng phát trực tuyến và nhiều ứng dụng khác cần phản hồi nhanh và xử lý đa khách hàng.

## 5. Express.js

**Express.js** là một framework phổ biến được sử dụng để xây dựng ứng dụng web và API thông qua Node.js. Nền tảng được xem là một phương thức xử lý các yêu cầu HTTP, quản lý các tuyến đường, xử lý phần mềm trung gian và nhiều tính năng khác để phát triển hiệu quả ứng dụng web. [7]

**Express.js** là một mã nguồn mở hoàn toàn miễn phí. Điều này cho phép các nhà phát triển xây dựng web và API ứng dụng mà không phải trả bất kỳ khoản phí nào khi sử dụng framework này. Công nghệ mang đến điều kiện thuận lợi cho cả những dự án lớn và nhỏ có nhu cầu sử dụng Express.js mà không cần phải lo lắng về các chi phí phát triển.

### Một số điểm nổi bật về Express.js:

- + Tốc độ phát triển: Express.js sở hữu cấu trúc linh hoạt và đơn giản, giúp phát triển trang web trở nên nhanh chóng hơn. Việc có sẵn các tính năng xử lý các yêu cầu HTTP, quản lý các tuyến đường và phần mềm trung gian giúp nhà phát triển tập trung vào việc phát triển chức năng chính của trang web mà không cần phải lo lắng về cấu trúc cơ bản.



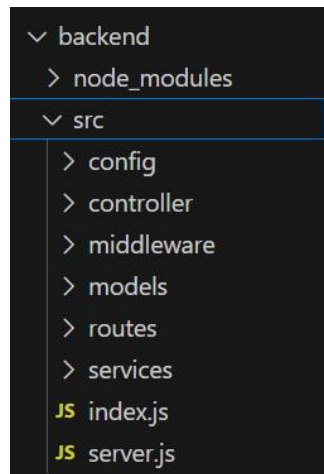
+ Hỗ trợ phần mềm trung gian: Express.js cho phép tích hợp nhiều phần mềm trung gian từ bên ngoài, mở rộng khả năng của trang web thông qua các tính năng bổ sung như khách hàng xác thực, xử lý lỗi, quản lý phiên bản và nhiều tính năng khác mà không cần thiết viết lại từ đầu.

+ Hỗ trợ API và định tuyến: Express.js được sử dụng rộng rãi để xây dựng RESTful API, giúp tạo các API điểm cuối một cách nhanh chóng và dễ dàng. Ngoài ra, công việc quản lý định tuyến trong Express cũng rất linh hoạt, giúp phân chia trang web thành các module và quản lý URL một cách dễ dàng.

+ Cộng đồng mạnh mẽ: Express.js có cộng đồng lớn với rất nhiều tài liệu, khả năng mở rộng gói đa dạng và cơ hội hỗ trợ từ cộng đồng. Những điều này đều có tác dụng xây dựng trang web nhanh chóng và hiệu quả.

### Dưới đây là kiến trúc phía Server:

- Khởi tạo dự án, Cấu trúc file, folder



- Khởi tạo ứng dụng, cấu hình Port và kết nối với Cơ sở dữ liệu

```
const express = require('express');

//app
const app = express();

//port
const port = process.env.PORT || 8000;
app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});

//db
mongoose.connect(process.env.DATABASE, (error) => {
  if (error) throw error;
  console.log('DB connected!');
});
```

## 6. Database MongoDB

**MongoDB** là một hệ thống quản lý cơ sở dữ liệu NoSQL mã nguồn mở phổ biến, được thiết kế để xử lý, lưu trữ và truy xuất dữ liệu một cách linh hoạt và hiệu quả. MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh. MongoDB có nhiều ưu điểm như khả năng lưu trữ dữ liệu phân tán, linh hoạt trong cấu trúc dữ liệu, có thể mở rộng, tốc độ truy vấn nhanh và hỗ trợ các tính năng như indexing, replication, sharding và map-reduce.

## III. Tính năng

### 1. Tạo tài khoản

- Hệ thống cho phép người dùng tạo tài khoản bằng cách nhập các thông tin sau: username, password, email.

### 2. Đăng nhập

- Hệ thống cho phép người dùng đăng nhập bằng cách nhập username và password.

### 3. Quản lý giỏ hàng

- Hệ thống cho phép người dùng thêm hoặc xóa các mặt hàng đã có trong giỏ hàng.
- Hệ thống cho phép người dùng xem thông tin các mặt hàng có trong giỏ bao gồm: tên, số lượng, giá tiền của từng mặt hàng và tổng giá trị của tất cả các mặt hàng trong giỏ.

### 4. Tìm kiếm / khuyến nghị

- Hệ thống cho phép người dùng tìm kiếm mặt hàng theo tên mặt hàng và sắp xếp các mặt hàng theo thứ tự giá bán tăng dần hoặc giảm dần.
- Hệ thống cho phép người dùng tìm kiếm những mặt hàng bán chạy nhất và các mặt hàng đang được giảm giá.
- Hệ thống hiển thị cho người dùng các mặt hàng phù hợp nhất với người dùng dựa trên lịch sử mua hàng.

### 5. Đặt hàng

- Chỉ có người dùng đã đăng nhập mới được phép thực hiện chức năng này.
- Hệ thống cho phép người dùng thực hiện đặt hàng online.
- Hệ thống cho phép người dùng theo dõi tình trạng đơn hàng.

### 6. Thanh toán

- Chỉ có người dùng đã đăng nhập mới được phép thực hiện chức năng này.

- Hệ thống cho phép người dùng thanh toán theo 2 cách. Một là, thanh toán khi nhận hàng. Hai là, thanh toán qua thẻ tín dụng.

## **7. Quản lý đổi trả và hoàn tiền**

Khách hàng được phép đổi trả khi:

- Sản phẩm bị lỗi, bản.
- Gửi sai mẫu mã, size.
- Sản phẩm không đúng mô tả.

## **8. Quản lý sản phẩm**

Quản trị viên có giao diện riêng để quản lý toàn bộ sản phẩm.

Tính năng bao gồm:

- Thêm mới sản phẩm
- Cập nhật hoặc xóa sản phẩm.
- Phân loại theo danh mục

## **9. Quản lý đơn hàng**

- Người dùng có thể xem danh sách đơn hàng đã đặt, theo dõi tiến độ giao hàng, hủy đơn nếu chưa thanh toán.
- Admin có thể xem toàn bộ đơn hàng, cập nhật trạng thái: “Chờ xác nhận” → “Đang giao hàng” → “Đã giao” hoặc “Đã hủy”.
- Xử lý các yêu cầu trả hàng-hoàn tiền.

## **10. Thống kê và báo cáo**

Quản trị viên có thể xem các số liệu báo cáo:

- Tổng doanh thu theo ngày/tháng/năm.
- Sản phẩm bán chạy nhất.

## IV. Kế hoạch thực thi, triển khai dự án

- Phạm vi dự án: Xây dựng hệ thống quản lý cửa hàng thời trang với 2 role:

+ Khách hàng: Xem sản phẩm, đặt hàng, thanh toán, đánh giá.

+ Admin: Quản lý sản phẩm, đơn hàng, thống kê.

- Chi tiết phân công theo tuần

a, Tuần 1: Phân tích & Thiết kế hệ thống

- Mục tiêu: Hoàn thiện tài liệu thiết kế, wireframe, và cấu trúc cơ bản.

Thành viên	Công việc chi tiết
Cả nhóm	- Brainstorming tính năng chính - Phân tích yêu cầu người dùng - Thiết kế Use Case Diagram và User Flow
Front-end	- Cài đặt React + Tailwind CSS + Router • Trang đăng nhập/đăng ký • Trang quản trị (CRUD sản phẩm)
Back-end	- Thiết kế ERD MongoDB (: User, Product, Order, Review) - Cấu hình dự án Node.js + Express

b, Tuần 2: Phát triển hệ thống back end và front-end

- Mục tiêu: Hoàn thành 80% API backend và giao diện cốt lõi.

Thành viên	Công việc chi tiết
Cả nhóm	Test luồng đặt hàng/thanh toán
Back-end	- API Product: CRUD, upload ảnh (Cloudinary), lọc theo danh mục - API Cart: Thêm/xóa sản phẩm, tính tổng tiền
Front-end	- Xây dựng: • Trang chủ (hiển thị sản phẩm từ API) • Trang chi tiết sản phẩm

c, Tuần 3: Kết nối front-end và backend

- Mục tiêu: Kết nối frontend-backend, triển khai chức năng thanh toán

Thành viên	Công việc chi tiết
Cả nhóm	- Fix bug hiển thị thông báo lỗi
Front-end	- Trang giỏ hàng - Trang thanh toán - Trang quản trị (Dashboard đơn hàng)
Back-end	- API Order: Tạo đơn hàng, lịch sử đơn hàng - API Review: Thêm/xem đánh giá

d, Tuần 4: Hoàn thiện và viết báo cáo

- Mục tiêu: Tối ưu UX/UI, deploy và chuẩn bị báo cáo.

Thành viên	Công việc chi tiết
Cả nhóm	- Viết tài liệu hướng dẫn - Tổng hợp báo cáo (Google Docs) - Chuẩn bị slide
Front-end	- Responsive mobile
Back-end	Deploy backend (Render/Vercel)