

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN 2**



**ĐỒ ÁN HỌC PHẦN
TÊN HỌC PHẦN: NHẬP MÔN CÔNG NGHỆ PHẦN MỀM**

Tên đề tài: Mini App Quản Lý Đơn Hàng Cho Shop Online

Họ tên sinh viên:

Huỳnh Trần Thu Thảo – N23DCPT105

Nguyễn Thị Thu Tâm – N23DCPT103

Dương Khánh Ngọc – N23DCPT095

Lớp: D23CQPTUD01-N

Ngày nộp: 24/12/2025

Sinh viên ký tên:

Giảng viên ký tên:

TP.HCM, ngày 24 tháng 12 năm 2025

MỤC LỤC

LỜI MỞ ĐẦU	2
1. Tổng quan đề tài	3
1.1. Giới thiệu	3
1.2. Công nghệ sử dụng	3
1.3. Mô hình hoạt động	3
2. Phân tích và thiết kế hệ thống.....	3
2.1. Thiết kế cơ sở dữ liệu.....	3
2.2. Thiết kế kiến trúc	4
3. Triển khai các chức năng chính.....	4
3.1. Quản lý khách hàng.....	4
3.2. Quản lý sản phẩm.....	6
3.3. Quản lý giỏ hàng.....	7
3.4. Quản lý đơn hàng.....	7
4. Báo cáo thống kê và cải tiến.....	9
4.1. Kiểm thử Đơn vị (Unit Test).....	9
4.2. Kiểm thử Giao diện & Tích hợp (UI/Integration Test).....	10
4.3. Kiểm thử API (API Test)	12
4.4. Báo cáo tỷ lệ bao phủ kiểm thử.....	19
5. Báo cáo doanh số, trạng thái bán hàng	20
6. Đề xuất cải tiến	21
6.1. Tích hợp thanh toán thực tế	21
6.2. Tích hợp vận chuyển.....	21
LỜI KẾT	22

LỜI MỞ ĐẦU

Trong bối cảnh thương mại điện tử đang phát triển mạnh mẽ, thói quen mua sắm của người tiêu dùng đã chuyển dịch đáng kể từ phương thức truyền thống sang các nền tảng trực tuyến. Đối với các chủ cửa hàng vừa và nhỏ, việc sở hữu một hệ thống quản lý bán hàng riêng biệt, tập trung và hiệu quả là vô cùng cấp thiết để tối ưu hóa quy trình vận hành cũng như nâng cao trải nghiệm khách hàng.

Nhận thức được nhu cầu đó, nhóm chúng em đã thực hiện đề tài "Mini App Quản Lý Đơn Hàng Cho Shop Online". Mục tiêu của đề án là xây dựng một hệ thống bán hàng hoàn chỉnh theo mô hình Single Vendor (Đơn nhà cung cấp), bao gồm giao diện mua sắm thân thiện cho khách hàng và trang quản trị trực quan cho chủ cửa hàng.

Khác với các sàn thương mại điện tử phức tạp, hệ thống này tập trung sâu vào việc xử lý logic đơn hàng, tính toán doanh thu và quản lý kho hàng. Đề án được xây dựng trên nền tảng các công nghệ hiện đại như Node.js, Express.js cho Backend, React.js, Tailwind CSS cho Frontend và cơ sở dữ liệu MySQL.

Để hoàn thành đề án này, bên cạnh sự nỗ lực nghiên cứu và tìm hiểu của các thành viên trong nhóm, chúng em đã nhận được sự quan tâm và hướng dẫn tận tình từ giảng viên bộ môn. Chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến Thầy Châu Văn Vân – người đã trực tiếp truyền đạt kiến thức, định hướng và đưa ra những lời khuyên quý báu giúp chúng em giải quyết các vấn đề khó khăn trong quá trình thực hiện đề tài.

Báo cáo này sẽ trình bày chi tiết về quá trình phân tích, thiết kế hệ thống, triển khai các chức năng cốt lõi cũng như kết quả kiểm thử toàn diện từ Unit Test đến UI Test nhằm đảm bảo tính ổn định của ứng dụng.

Dù đã rất cố gắng, nhưng do giới hạn về mặt kiến thức và thời gian, đề án chắc chắn không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự đóng góp ý kiến từ thầy để đề tài ngày càng hoàn thiện hơn.

1. Tổng quan đề tài

1.1. Giới thiệu

- **Tên đề tài:** Mini App Quản Lý Đơn Hàng Cho Shop Online
- **Mục tiêu:** Xây dựng một hệ thống bán hàng hoàn chỉnh bao gồm giao diện mua sắm cho khách hàng và trang quản trị cho chủ cửa hàng

1.2. Công nghệ sử dụng

- **Backend:** Node.js, Express.js
- **Database:** MySQL Workbench
- **Frontend:** React.js, Tailwind CSS
- **Kiểm thử:** Jest (Unit Test), Cypress (UI/Integration Test), Thunder Client (API Test).

1.3. Mô hình hoạt động

Khác với các sàn thương mại điện tử như Shopee hay Lazada, hệ thống này được xây dựng theo mô hình single vendor (Đơn nhà cung cấp).

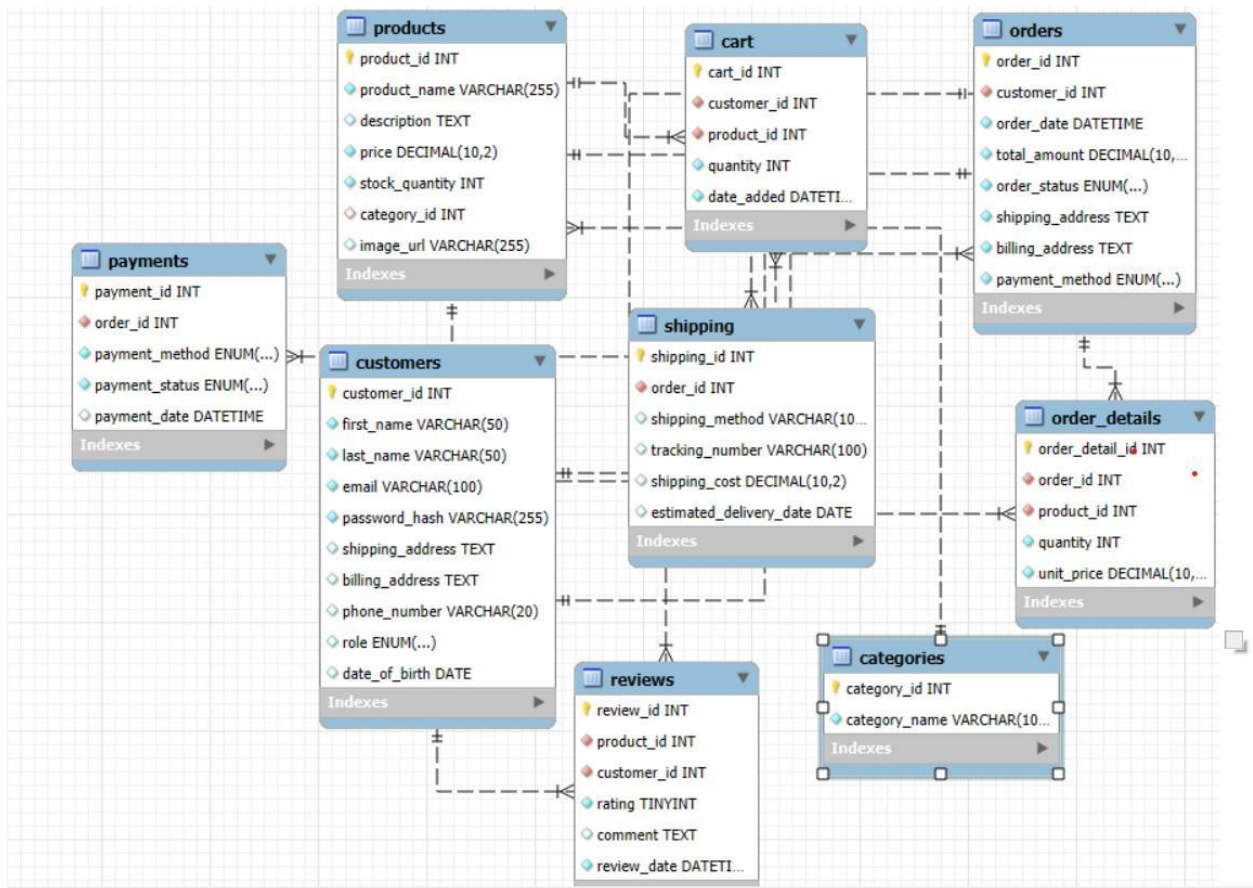
- **Đặc điểm:** Hệ thống chỉ có duy nhất một đơn vị bán hàng (Admin/Chủ shop).
- **Phân quyền:**
 - **Admin (Chủ shop):** Người duy nhất có quyền nhập liệu sản phẩm, quản lý kho, xử lý trạng thái đơn hàng và xem báo cáo doanh thu.
 - **Customer (Khách hàng):** Người dùng thực hiện các thao tác đăng ký tài khoản, xem sản phẩm và đặt mua hàng.
- **Lý do lựa chọn:** Mô hình này phù hợp với quy mô đồ án môn học, giúp tập trung sâu vào việc xử lý logic đơn hàng, tính toán doanh thu và trải nghiệm người dùng thay vì phân tán vào việc quản lý đa gian hàng phức tạp.

2. Phân tích và thiết kế hệ thống

2.1. Thiết kế cơ sở dữ liệu

Mô tả: Hệ thống sử dụng cơ sở dữ liệu quan hệ MySQL với các bảng chính: Customers, Products, Categories, Orders, Order_Details, Cart.

- Sơ đồ ERD:



2.2. Thiết kế kiến trúc

- Mô hình Client-Server: Frontend (Port 5173) giao tiếp với Backend (Port 8080) thông qua HTTP Requests (Axios).

3. Triển khai các chức năng chính

3.1. Quản lý khách hàng

- **Chức năng:** Đăng ký, đăng nhập, cập nhật hồ sơ cá nhân.
- **Backend:** API /auth/register, /auth/login, /auth/profile.
- **Giao diện:**

Đăng Ký Tài Khoản

Tên

Họ

Ngày sinh

mm/dd/yyyy



Email

Mật khẩu

Tạo Tài Khoản

Đăng Nhập

Email

Mật khẩu

Đăng Nhập

Thông Tin Tài Khoản

Họ và Tên:

D Nguyễn Văn

Email:

customer@gmail.com

Ngày sinh:

1/1/2005

Số điện thoại:

0123456789

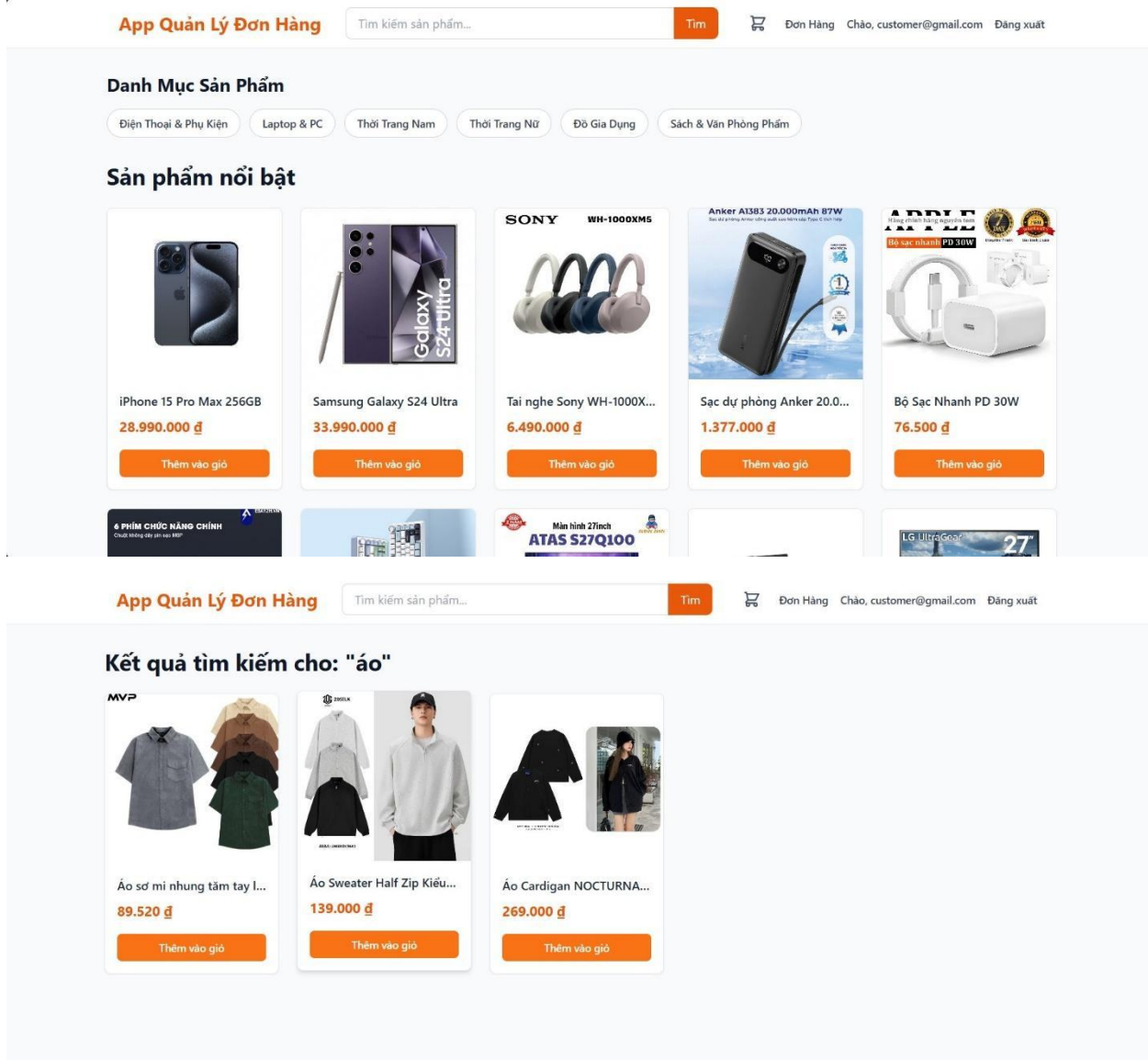
Địa chỉ mặc định:

123 Đường A, TPHCM

Cập Nhật Thông Tin

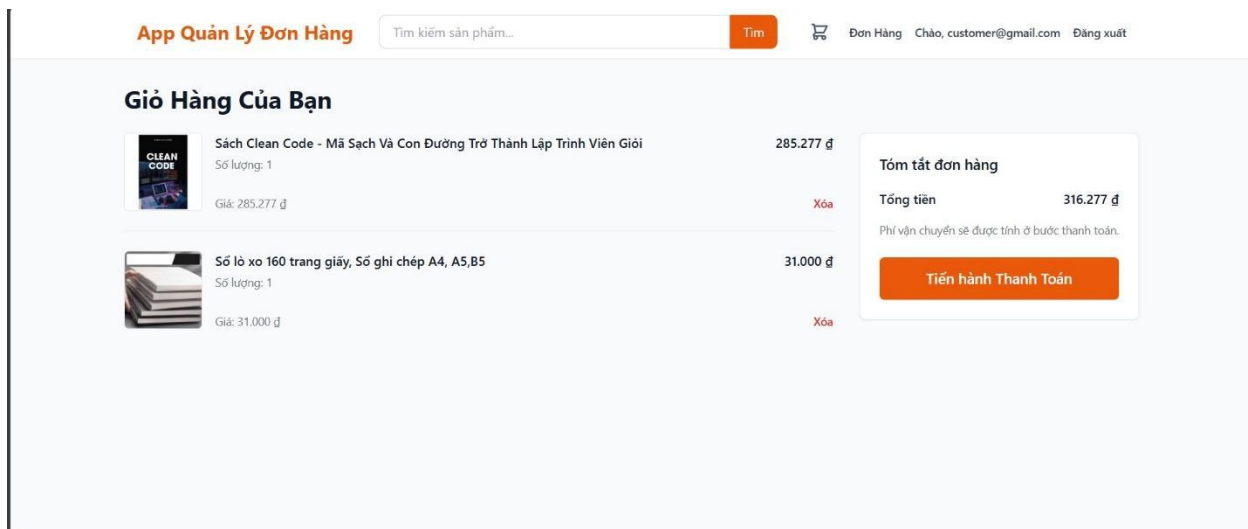
3.2. Quản lý sản phẩm

- **Chức năng:** Hiện thị danh sách, tìm kiếm theo tên, lọc theo danh mục.
- **Backend:** API GET /api/products, GET /api/products?search=....
- **Giao diện:**



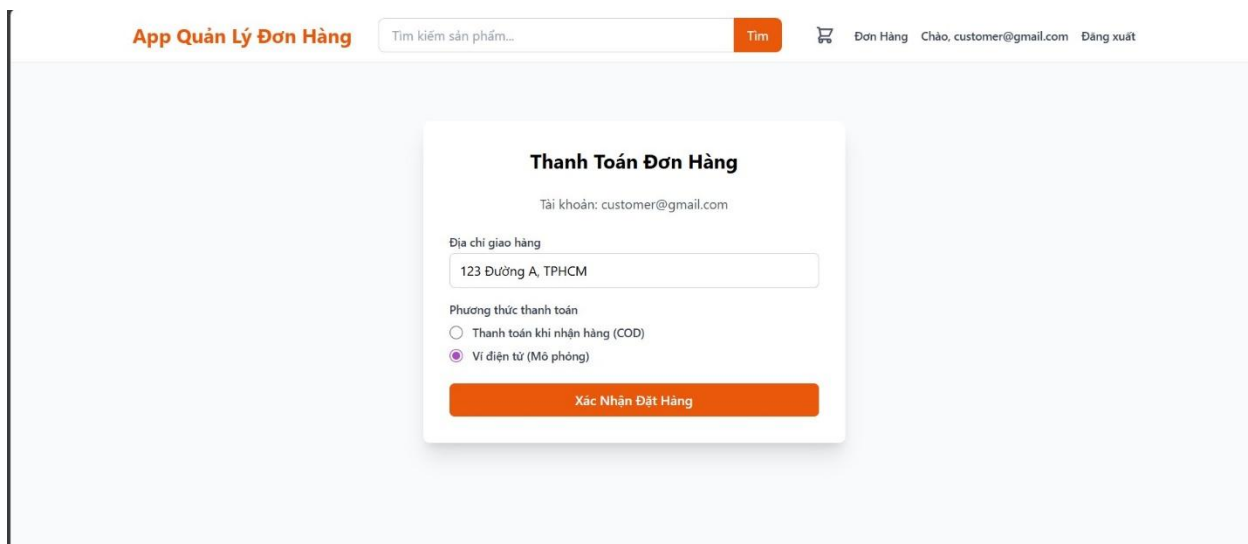
3.3. Quản lý giỏ hàng

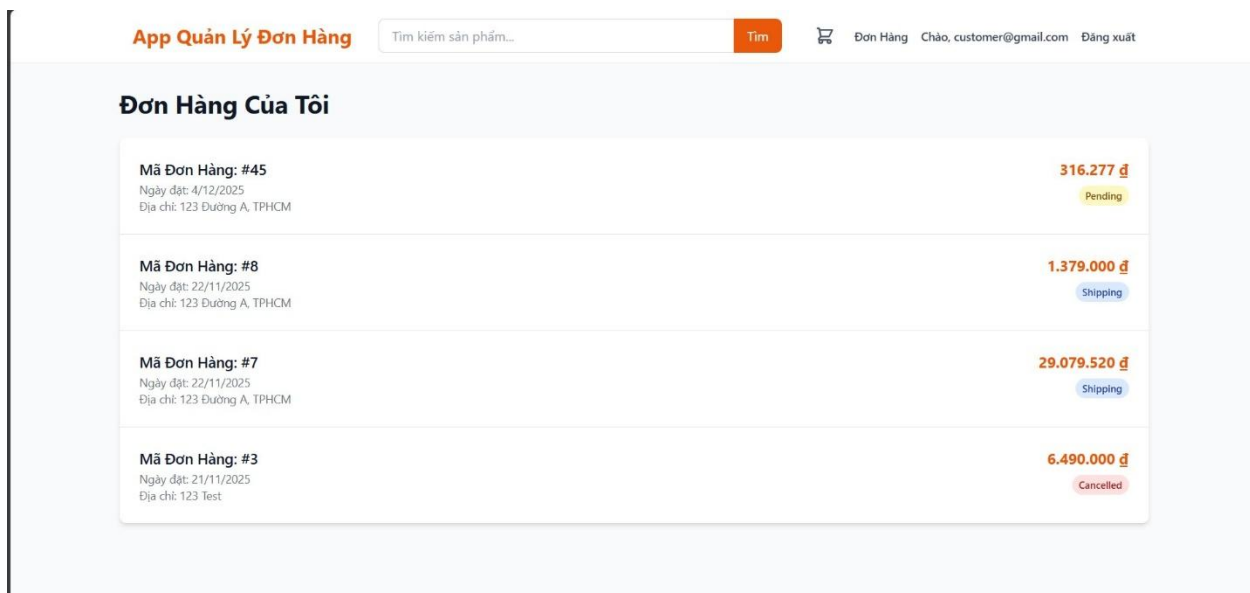
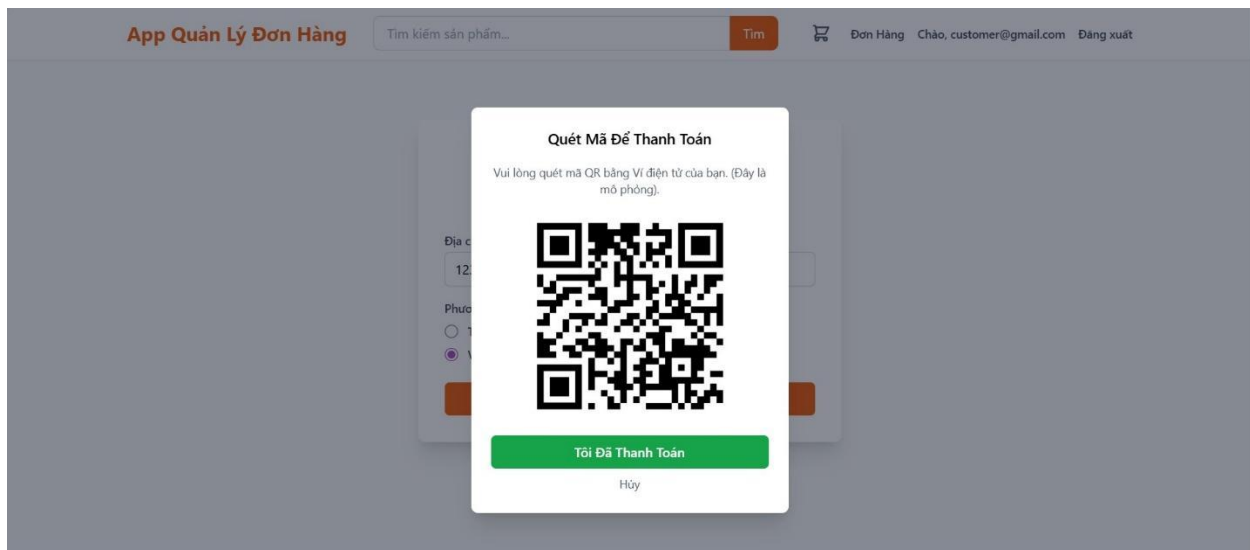
- **Chức năng:** Thêm sản phẩm, xem giỏ hàng, tính tổng tiền tự động.
- **Backend:** API /api/cart.
- **Giao diện:**



3.4. Quản lý đơn hàng

- **Chức năng:** Nhập địa chỉ giao hàng, chọn phương thức thanh toán (COD/Ví điện tử), tạo đơn hàng.
- **Backend:** API /api/orders
- **Giao diện:**





4. Báo cáo thống kê và cải tiến

4.1. Kiểm thử Đơn vị (Unit Test)

- **Công cụ:** Jest
- **Số lượng:** 5 Test Case
- **Thông qua:** 5 (100%)
- **Hình ảnh:**

```
PS C:\Users\huynh\OneDrive\Desktop\Programming\WEBSITE CNPM\Backend> npm test

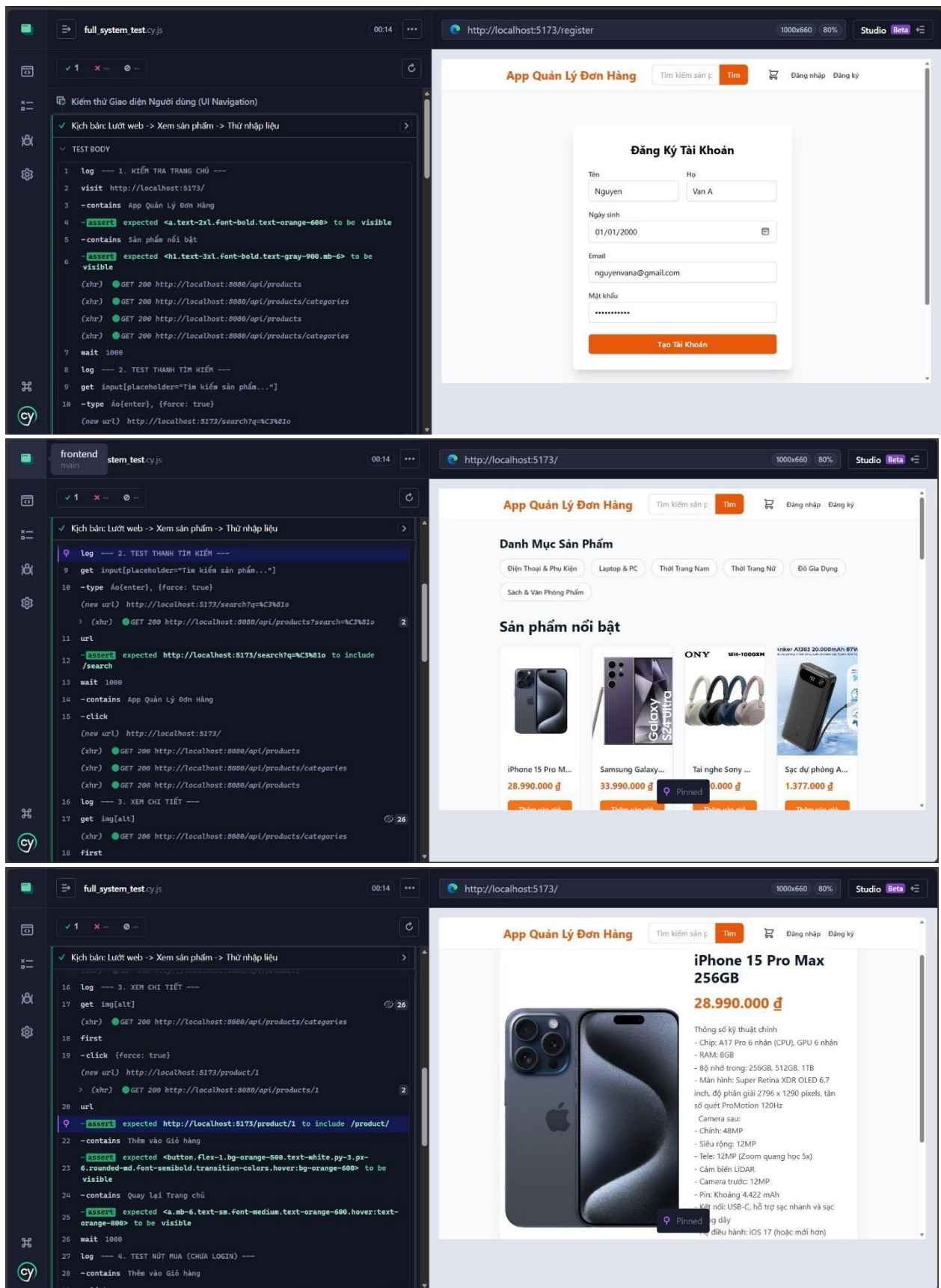
> backend@1.0.0 test
> jest

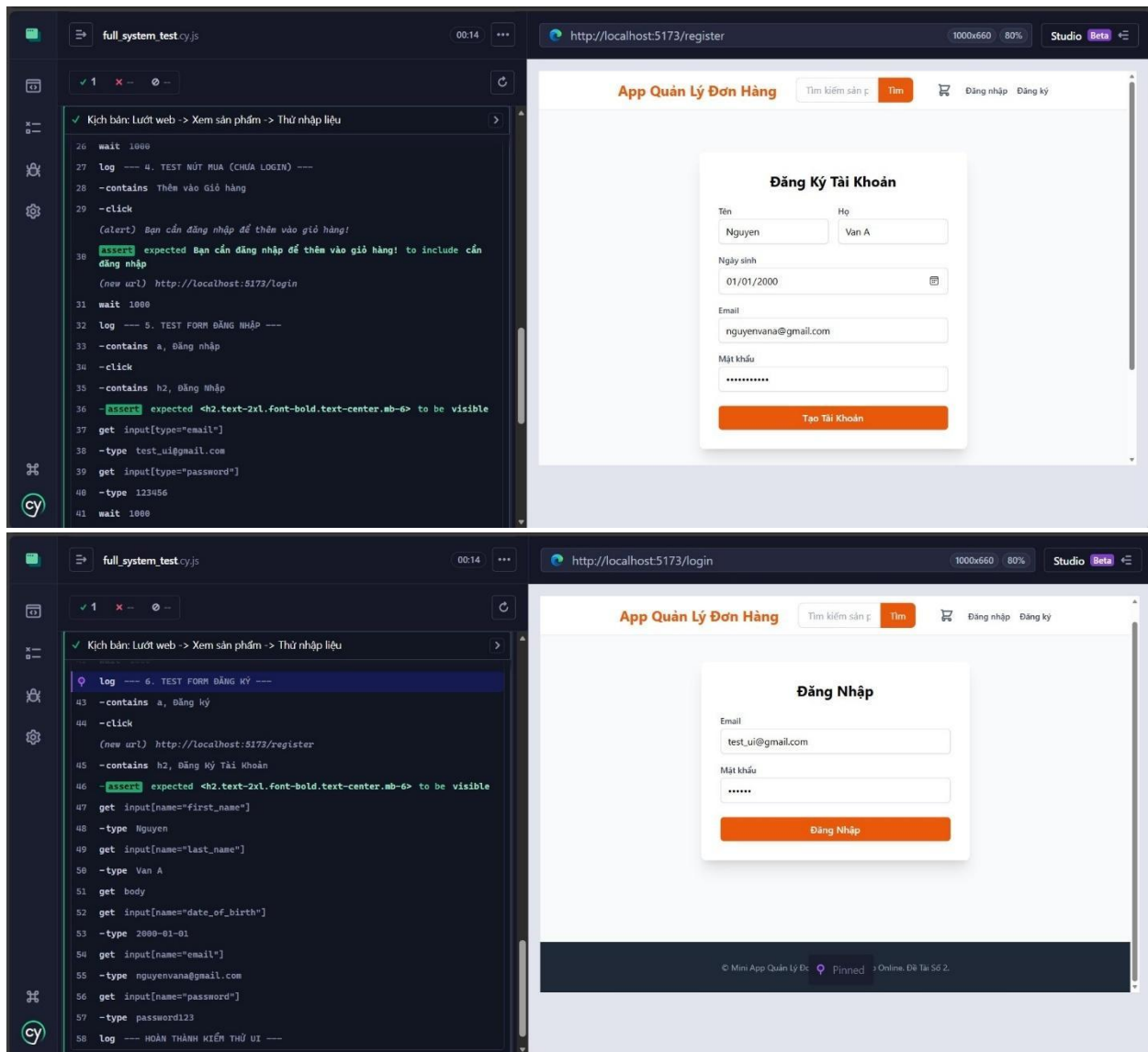
PASS tests/orderUtils.test.js
  Unit Test: Hàm calculateTotal (Tính tổng tiền)
    ✓ TC-01: Tính đúng tổng tiền cho giỏ hàng có 2 món (3 ms)
    ✓ TC-02: Trả về 0 nếu giỏ hàng rỗng (1 ms)
    ✓ TC-03: Báo lỗi nếu dữ liệu sai (Giá âm) (8 ms)
  Unit Test: Hàm validateOrderInput (Kiểm tra đầu vào)
    ✓ TC-04: Báo lỗi nếu thiếu địa chỉ giao hàng (1 ms)
    ✓ TC-05: Hợp lệ nếu đủ thông tin và đúng phương thức (1 ms)

Test Suites: 1 passed, 1 total
Tests:       5 passed, 5 total
Snapshots:   0 total
Time:        0.57 s, estimated 1 s
Ran all test suites.
```

4.2. Kiểm thử Giao diện & Tích hợp (UI/Integration Test)

- Công cụ: Cypress
- Hình ảnh:





4.3. Kiểm thử API (API Test)

- **Phương thức:** Thunder Client
- **Số lượng:** 4 Collections (Auth, Products, Cart, Orders)
- **Thông qua:** 4 (100%)

Collections:

❖ Auth (Quản lý khách hàng)

1. Đăng ký (POST /register)

POST	http://localhost:8080/api/auth/register	Send	Status: 201 Created Size: 222 Bytes Time: 174 ms
Query	Headers	Auth	Body Tests Pre Run
JSON	XML	Text	Form Form-encode GraphQL Binary
JSON Content			Format
<pre> 1 { 2 "first_name": "A", 3 "last_name": "Nguyễn Văn", 4 "email": "customer123@gmail.com", 5 "password": "password123", 6 "date_of_birth": "2005-01-01" 7 } </pre>			Response Headers Cookies Results Docs {}
<pre> 1 { 2 "message": "Đăng ký thành công!", 3 "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6NCwicm9sZSI6ImN1c3RvbWVYIiwiaWF0IjoxNzYzNmMxMDQ3LCJleHAiOjE3NjMzQ2NDd9.Le6pRbGgOfU7Z8_I-dB5y1-2MjAcvm0zLF8XqBkk3Q8", 4 "user_id": 4 5 } </pre>			

Trường hợp trùng email thì sẽ không thể đăng ký được

POST	http://localhost:8080/api/auth/register	Send	Status: 400 Bad Request Size: 51 Bytes Time: 5 ms
Query	Headers	Auth	Body Tests Pre Run
JSON	XML	Text	Form Form-encode GraphQL Binary
JSON Content			Format
<pre> 1 { 2 "first_name": "ABC", 3 "last_name": "Nguyễn Văn", 4 "email": "customer123@gmail.com", 5 "password": "password123", 6 "date_of_birth": "2005-01-01" 7 } </pre>			Response Headers Cookies Results Docs {}
<pre> 1 { 2 "message": "Email này đã được sử dụng." 3 } </pre>			

2. Đăng nhập (POST /login)

POST	http://localhost:8080/api/auth/login	Send	Status: 200 OK Size: 243 Bytes Time: 138 ms
Query	Headers	Auth	Body Tests Pre Run
JSON	XML	Text	Form Form-encode GraphQL Binary
JSON Content			Format
<pre> 1 { 2 "email": "customer@gmail.com", 3 "password": "password123" 4 } 5 </pre>			Response Headers Cookies Results Docs {}
<pre> 1 { 2 "message": "Đăng nhập thành công!", 3 "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6Mywicm9sZSI6ImN1c3RvbWVYIiwiaWF0IjoxNzYzNmZyOjMjA2LCJleHAiOjE3NjMzNDZ9.ikEHwtOUTcbV9wqPgpr1sPN94Fiqkv2muqJm8MF1KBg", 4 "user_id": 3, 5 "role": "customer" 6 } </pre>			

Trường hợp đăng nhập nhưng sai mật khẩu hoặc email

The screenshot shows a REST client interface with a POST request to `http://localhost:8080/api/auth/login`. The request body is a JSON object: `{ "email": "customer@gmail.com", "password": "password12345" }`. The response status is `401 Unauthorized` with a size of 44 bytes and a time of 130 ms. The response body is a JSON object: `{ "message": "Sai email hoặc mật khẩu." }`.

3. Cập nhật thông tin người dùng (PUT /profile)

Chỉ có thể cập nhật thông tin sau khi người dùng đăng nhập

The screenshot shows a REST client interface with a PUT request to `http://localhost:8080/api/auth/profile`. The request body is a JSON object: `{ "first_name": "D", "last_name": "Nguyễn Văn", "shipping_address": "123 Đường A, TPHCM", "phone_number": "0123456789" }`. The response status is `200 OK` with a size of 283 bytes and a time of 34 ms. The response body is a JSON object: `{ "message": "Cập nhật thông tin thành công!", "user": { "customer_id": 3, "email": "customer@gmail.com", "first_name": "D", "last_name": "Nguyễn Văn", "shipping_address": "123 Đường A, TPHCM", "phone_number": "0123456789", "role": "customer", "date_of_birth": "2004-12-31T17:00:00.000Z" } }`.

Trường hợp người dùng chưa đăng nhập

The screenshot shows a REST client interface with a PUT request to `http://localhost:8080/api/auth/profile`. The request body is a JSON object: `{ "first_name": "ABC", "last_name": "Nguyễn Văn", "shipping_address": "123 Đường ADE, TPHCM", "phone_number": "0123456789" }`. The response status is `401 Unauthorized` with a size of 52 bytes and a time of 6 ms. The response body is a JSON object: `{ "message": "Chưa đăng nhập, không có token." }`.

❖ Products (Quản lý sản phẩm)

1. Lấy danh sách sản phẩm (GET /products)

GET Send

Query Headers Auth Body Tests Pre Run

Query Parameters

☐ parameter value

Status: 200 OK Size: 19.32 KB Time: 17 ms

Response Headers Cookies Results Docs {} ≡

```
1 [
2   {
3     "product_id": 1,
4     "product_name": "iphone 15 Pro Max 256GB",
5     "description": "Thông số kỹ thuật chính\n- Chip: A17
Pro 6 nhân (CPU), GPU 6 nhân\n- RAM: 8GB\n- Bộ nhớ
trong: 256GB, 512GB, 1TB\n- Màn hình: Super Retina
XDR OLED 6.7 inch, độ phân giải 2796 x 1290 pixels
, tần số quét ProMotion 120Hz\n- Camera sau:\n-
Chính: 48MP\n- Siêu rộng: 12MP\n- Tele: 12MP (Zoom
quang học 5x)\n- Cảm biến LiDAR\n- Camera trước:
12MP\n- Pin: Khoảng 4.422 mAh\n- Kết nối: USB-C,
hỗ trợ sạc nhanh và sạc không dây\n- Hệ điều hành:
iOS 17 (hoặc mới hơn)\n- Kháng nước, bụi: IP68 ",
6     "price": "28990000.00",
7     "stock_quantity": 34,
8     "image_url": "https://product.hstatic.net/200000409445
Response Chart
```

2. Tìm kiếm sản phẩm (GET /products)

Nhập từ khóa bàn phím

GET Send

Query Headers Auth Body Tests Pre Run

Query Parameters

☒ search

☐ parameter value

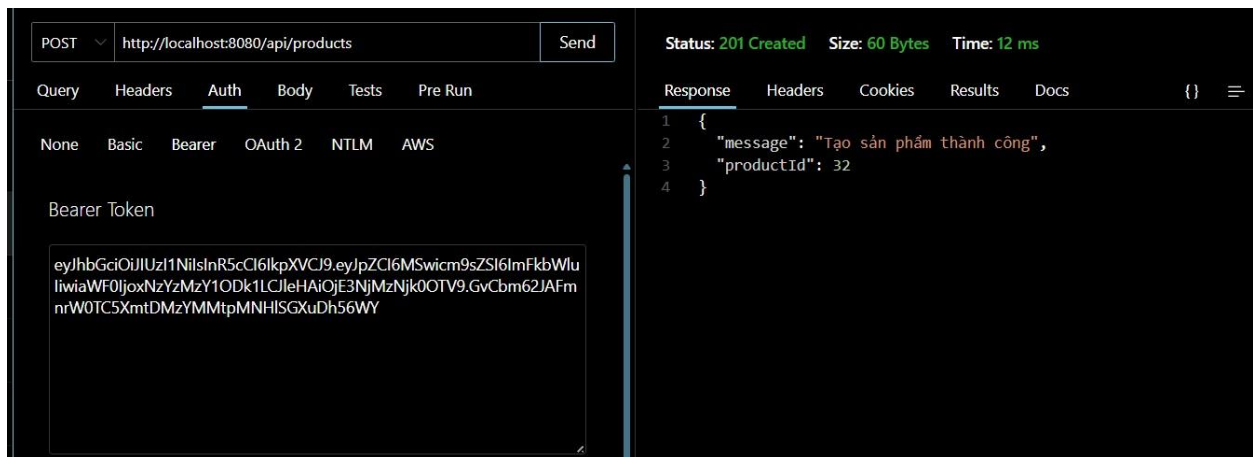
Status: 200 OK Size: 535 Bytes Time: 6 ms

Response Headers Cookies Results Docs {} ≡

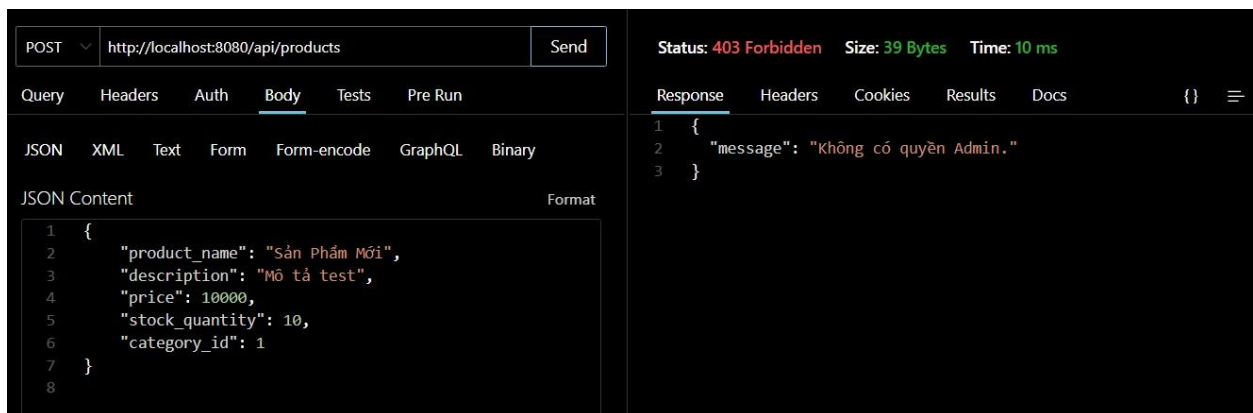
```
1 [
2   {
3     "product_id": 7,
4     "product_name": "Bàn phím cơ Aula F75",
5     "description": "THÔNG TIN SẢN PHẨM Aula F75\n- LAYOUT
75%\n- MẠCH XUÔI \n- LED RGB \n- KEYCAP PBT\n-
GASKET MOUNT\n- PIN 4000 MAH\n- 3 MODE KẾT NỐI (TYPE
C , BT 5.0, 2.4GHZ) \n- CÓ APP CHÍNH LED ĐI KÈM\n-
FULL FOAM\n- Switch : 2 loại Star Vector( Linear ) ,
switch silent Outemu V2",
6     "price": "999000.00",
7     "stock_quantity": 14,
8     "image_url": "https://bizweb.dktcdn.net/thumb/1024x1024
/100/438/322/products/main-07.jpg?v=1761806374973",
9     "category_name": "Laptop & PC"
10  }
11 ]
Response Chart
```

3. Thêm sản phẩm mới vào danh sách sản phẩm (POST /products)

Chỉ có thể thêm sản phẩm mới khi là admin



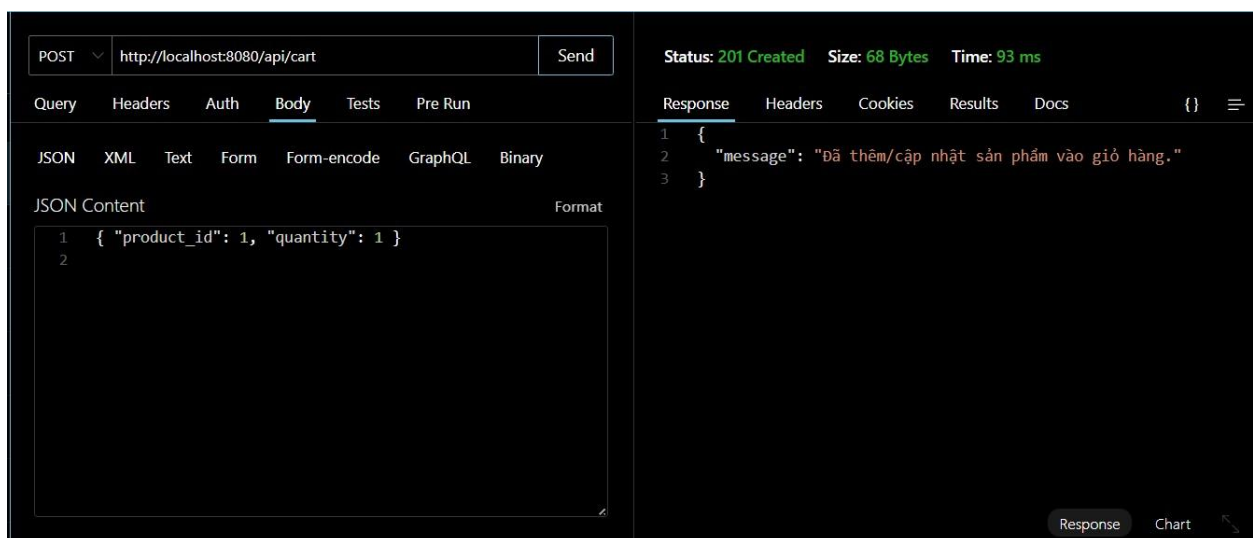
Trường hợp không phải là admin



❖ Cart (Quản lý giỏ hàng)

1. Thêm sản phẩm vào giỏ (POST /cart)

Chỉ có thể thêm sản phẩm vào giỏ sau khi người dùng đã đăng nhập



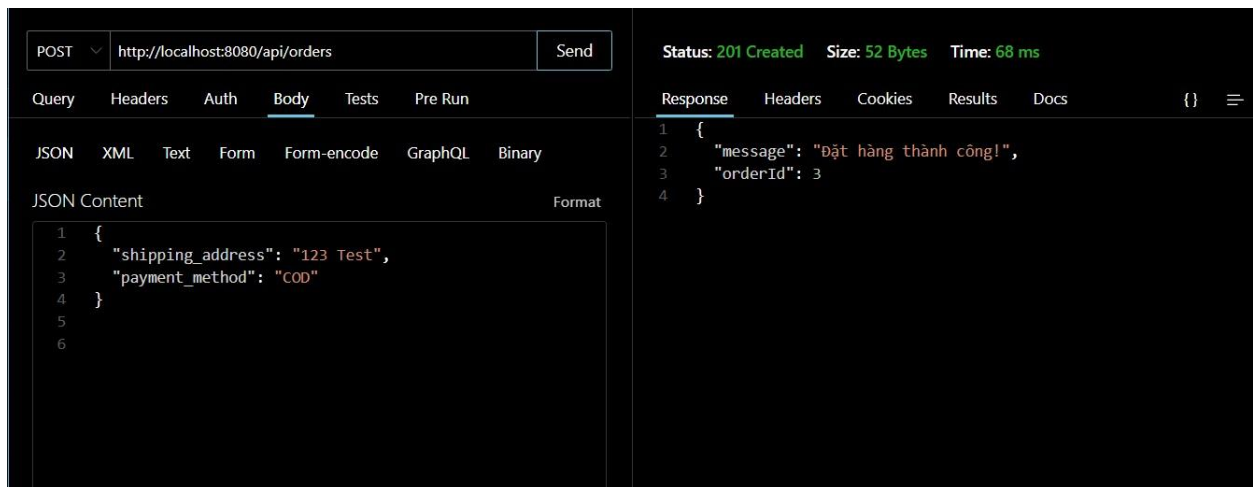
Trường hợp chưa đăng nhập



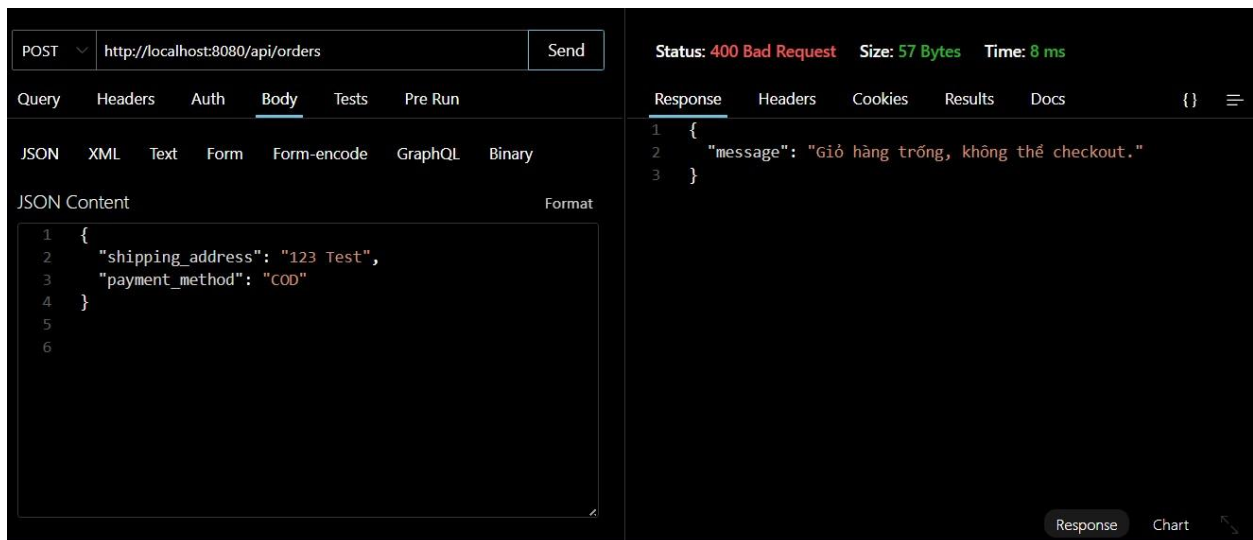
❖ Orders (Quản lý đơn hàng)

1. Checkout (POST /orders)

Chỉ có thể checkout khi giỏ hàng có ít nhất 1 vật phẩm trở lên

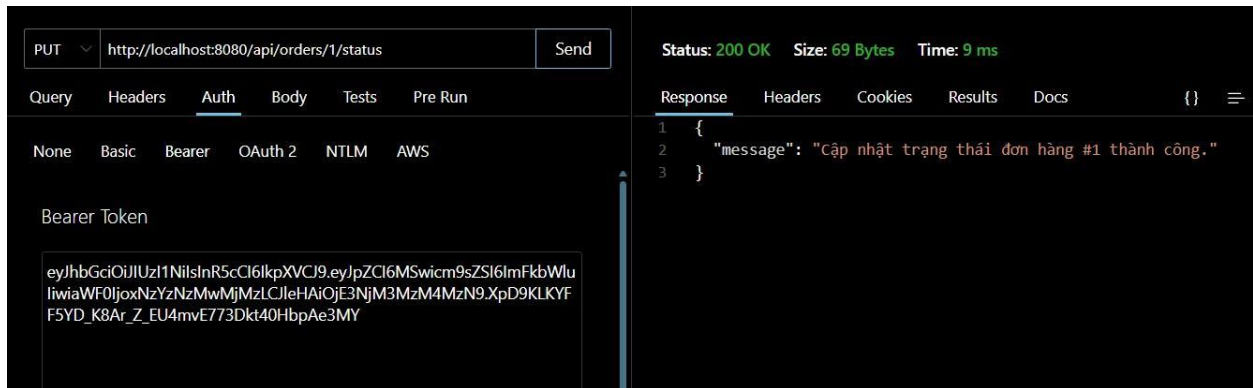


Trường hợp giỏ hàng rỗng

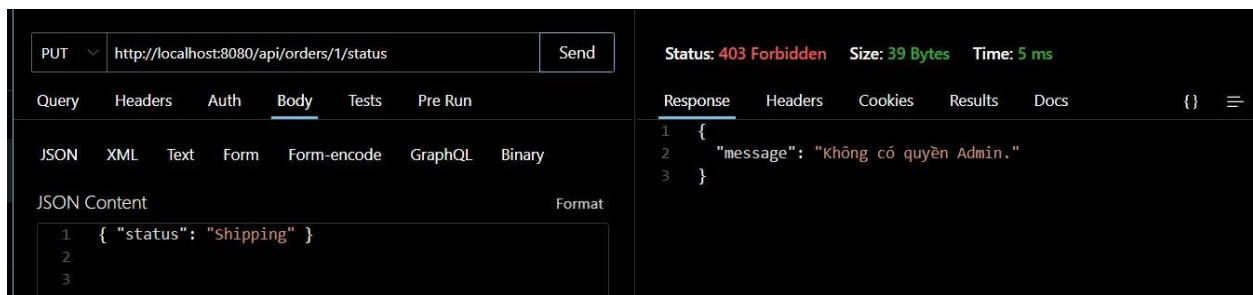


2. Đổi status đơn hàng (PUT /status)

Chỉ có thể cập nhật status khi là admin

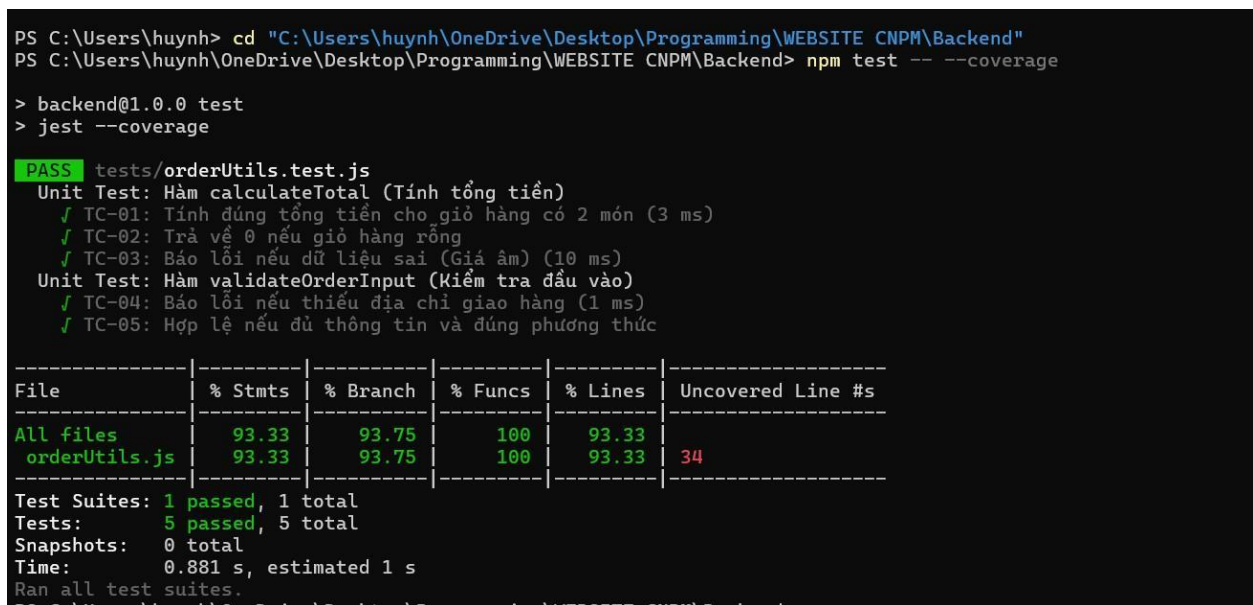


Trường hợp không phải là admin



4.4. Báo cáo tỷ lệ bao phủ kiểm thử

- **Công cụ:** Jest Coverage
- **Hình ảnh:**



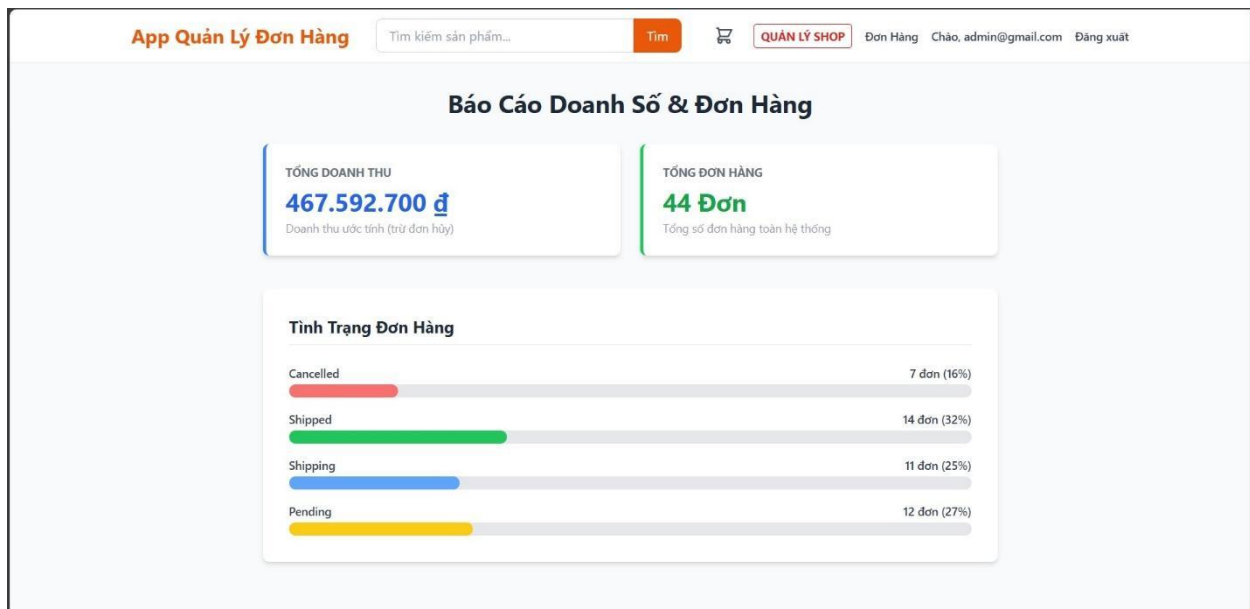


Kết quả:

- **Độ bao phủ hàm (Funcs): Đạt 100% (2/2 hàm).** Tất cả các hàm xử lý logic nghiệp vụ cốt lõi đều đã được đưa vào kịch bản kiểm thử.
- **Độ bao phủ câu lệnh (Stmts) và Dòng (Lines): Đạt 93.33%.** Hầu hết các dòng mã thực thi đều đã được kiểm tra tính đúng đắn.
- **Độ bao phủ nhánh (Branch): Đạt 93.75%.** Các trường hợp điều kiện biên, các luồng if-else xử lý lỗi dữ liệu đầu vào đều đã được bao quát.

Kết luận: Mã nguồn đạt chất lượng cao, đảm bảo tính ổn định và giảm thiểu tối đa các lỗi logic trong quá trình vận hành thực tế (như tính sai tiền, lỗi dữ liệu đầu vào).

5. Báo cáo doanh số, trạng thái bán hàng



6. Đề xuất cải tiến

6.1. Tích hợp thanh toán thực tế

- Tích hợp API của VNPAY hoặc Momo.
- Xây dựng hệ thống IPN (Instant Payment Notification) để nhận phản hồi tự động từ ngân hàng khi khách chuyển khoản thành công, thay vì xác nhận thủ công.

6.2. Tích hợp vận chuyển

- Hiện tại: trạng thái đơn hàng được cập nhật thủ công bởi Admin.
- Cải tiến: Kết nối API với Giao Hàng Nhanh (GHN) hoặc Giao Hàng Tiết Kiệm (GHTK) để đồng bộ trạng thái đơn hàng theo thời gian thực

LỜI KẾT

Sau quá trình nghiên cứu và triển khai, nhóm đã hoàn thiện đồ án "**Mini App Quản Lý Đơn Hàng Cho Shop Online**" với đầy đủ các tính năng cơ bản đáp ứng nhu cầu của một quy trình bán hàng trực tuyến.

Các kết quả đạt được:

- **Về mặt chức năng:** Xây dựng thành công hệ thống phân quyền rõ ràng giữa Admin và Customer. Người dùng có thể thực hiện trọn vẹn quy trình từ đăng ký, xem sản phẩm, quản lý giỏ hàng đến đặt hàng. Admin có thể quản lý sản phẩm và theo dõi báo cáo doanh thu trực quan.
- **Về mặt chất lượng:** Hệ thống đã trải qua quy trình kiểm thử nghiêm ngặt với **Jest** (Unit Test), **Cypress** (UI/Integration Test) và **Thunder Client** (API Test). Kết quả kiểm thử cho thấy độ bao phủ hàm đạt **100%** và độ bao phủ câu lệnh đạt **93.33%**, đảm bảo giảm thiểu tối đa các lỗi logic trong vận hành.

Hạn chế và hướng phát triển: Mặc dù hệ thống đã hoạt động ổn định, nhóm nhận thấy vẫn còn những điểm cần cải thiện để ứng dụng trở nên thực tế hơn. Trong tương lai, nhóm đề xuất tích hợp thêm các tính năng nâng cao như:

- **Thanh toán trực tuyến:** Kết nối API với VNPay hoặc Momo để xây dựng hệ thống IPN, tự động xác nhận thanh toán thay vì quy trình thủ công hiện tại.
- **Vận chuyển:** Tích hợp API của các đơn vị như Giao Hàng Nhanh (GHN) hoặc Giao Hàng Tiết Kiệm (GHTK) để đồng bộ trạng thái giao hàng theo thời gian thực.

Chúng em xin gửi lời cảm ơn chân thành đến thầy Châu Văn Vân đã tận tình chỉ bảo, giúp đỡ nhóm hoàn thành đồ án này. Nhóm rất mong nhận được những ý kiến đóng góp quý báu để hoàn thiện sản phẩm tốt hơn nữa.