

BÁO CÁO THÔNG KÊ VÀ CẢI TIẾN

1. Báo cáo kết quả kiểm thử

1.1. 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.
```

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

- **Công cụ:** Cypress
- **Hình ảnh:**

full_system_test.cy.js

```

    ✓ Kích bản: Lướt web -> Xem sản phẩm -> Thử nhập liệu
      ✓ TEST BODY
        1 log --- 1. KIỂM TRA TRANG CHỦ ---
        2 visit http://localhost:5173/
        3 -contains App Quản Lý Đơn Hàng
        4 -assert expected <a.text-2xl.font-bold.text-orange-600> to be visible
        5 -contains Sản phẩm nổi bật
        6 -assert expected <h1.text-3xl.font-bold.text-gray-900.mb-6> to be visible
        (xhr) GET 200 http://localhost:8080/api/products
        (xhr) GET 200 http://localhost:8080/api/products/categories
        (xhr) GET 200 http://localhost:8080/api/products
        (xhr) GET 200 http://localhost:8080/api/products/categories
        7 wait 1000
        8 log --- 2. TEST THANH TÌM KIẾM ---
        9 get input[placeholder="Tìm kiếm sản phẩm..."]
        10 -type Alo[enter], {force: true}
        (new url) http://localhost:5173/search?q=%C3%A1o
      
```

frontend_main_stem_test.cy.js

```

    ✓ Kích bản: Lướt web -> Xem sản phẩm -> Thử nhập liệu
      ✓ log --- 2. TEST THANH TÌM KIẾM ---
      9 get input[placeholder="Tìm kiếm sản phẩm..."]
      10 -type Alo[enter], {force: true}
      (new url) http://localhost:5173/search?q=%C3%A1o
      > (xhr) GET 200 http://localhost:8080/api/products?search=%C3%A1o
      11 url
      -assert expected http://localhost:5173/search?q=%C3%A1o to include /search
      12 wait 1000
      13 -contains App Quản Lý Đơn Hàng
      14 -click
      (new url) http://localhost:5173/
      (xhr) GET 200 http://localhost:8080/api/products
      (xhr) GET 200 http://localhost:8080/api/products/categories
      (xhr) GET 200 http://localhost:8080/api/products
      15 log --- 3. XEM CHI TIẾT ---
      16 get img[alt]
      (xhr) GET 200 http://localhost:8080/api/products/categories
      17 first
      
```

full_system_test.cy.js

```

    ✓ Kích bản: Lướt web -> Xem sản phẩm -> Thử nhập liệu
      ✓ log --- 3. XEM CHI TIẾT ---
      17 get img[alt]
      (xhr) GET 200 http://localhost:8080/api/products/categories
      18 first
      -click {force: true}
      (new url) http://localhost:5173/product/1
      > (xhr) GET 200 http://localhost:8080/api/products/1
      19 url
      -assert expected http://localhost:5173/product/1 to include /product/
      20 -contains Thêm vào Giỏ hàng
      -assert expected <button.flex-1.bg-orange-600.text-white.py-3.px-6.rounded-md.font-semibold.transition-colors.hover:bg-orange-600> to be visible
      21 -contains Quay Lại Trang chủ
      -assert expected <a.mb-6.text-sm.font-medium.text-orange-600.hover:text-orange-600> to be visible
      22 wait 1000
      23 log --- 4. TEST NÚT MUA (CHƯA LOGIN) ---
      24 -contains Thêm vào Giỏ hàng
      
```

http://localhost:5173/register

App Quản Lý Đơn Hàng

Đăng Ký Tài Khoản

Tên: Nguyen, Họ: Van A
Ngày sinh: 01/01/2000
Email: nguyenvana@gmail.com
Mật khẩu: *****

Tạo Tài Khoản

http://localhost:5173/

App Quản Lý Đơn Hàng

Danh Mục Sản Phẩm

Điện Thoại & Phụ Kiện, Laptop & PC, Thời Trang Nam, Thời Trang Nữ, Đồ Gia Dụng, Sách & Văn Phòng Phẩm

Sản phẩm nổi bật

- iPhone 15 Pro Max: 28.990.000 đ
- Samsung Galaxy S24 Ultra: 33.990.000 đ
- Tai nghe Sony WH-1000XM: 1.377.000 đ
- Sạc dự phòng Anker A1383 20.000mAh: 87K

http://localhost:5173/

App Quản Lý Đơn Hàng

iPhone 15 Pro Max 256GB

28.990.000 đ

Thông số kỹ thuật chính

- Chip: A17 Pro 6 nhân (CPU), GPU 6 nhân
- RAM: 8GB
- Bộ nhớ trong: 256GB, 512GB, 1TB
- Màn hình: Super Retina XDR OLED 6.7 inch, độ phân giải 2796 x 1290 pixels, tần số quét ProMotion 120Hz
- Camera sau:

 - Chính: 48MP
 - Siêu rộng: 12MP (Zoom quang học 5x)
 - Cảm biến LiDAR
 - Camera trước: 12MP
 - Pin: Khoảng 4.422 mAh
 - Kết nối: USB-C, hỗ trợ sạc nhanh và sạc không dây
 - Diễn hành: iOS 17 (hoặc mới hơn)

The screenshot displays two separate test runs in the Cypress UI Test Runner:

- Top Run:** Title: full_system_test.cy.js, Status: 1 Passed. It shows a successful login flow. The log output includes steps like "Kích bản: Lướt web -> Xem sản phẩm -> Thú nhập liệu", "TEST NÚT MUA (CHƯA LOGIN)", and "assert expected Bạn cần đăng nhập để thêm vào giỏ hàng! to include cần đăng nhập". The browser window shows the "Đăng Ký Tài Khoản" (Register Account) page.
- Bottom Run:** Title: full_system_test.cy.js, Status: 1 Passed. It shows a successful registration flow. The log output includes steps like "TEST FORM ĐĂNG KÝ" and "assert expected <h2.text-2xl.font-bold.text-center.mb-6> to be visible". The browser window shows the "Đăng Nhập" (Login) page.

1.3. Kiểm thử API (API Test)

- **Phương thức:** Thunder Client
- **Số lượng:** 4 Collections (Auth, Products, Cart, Orders)
- **Thông qua:** 4 (100%)
 - ❖ **Auth (Quản lý khách hàng)**
 - 1. **Đăng ký (POST /register)**

The screenshot shows a POST request to `http://localhost:8080/api/auth/register`. The request body is a JSON object:

```

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 }

```

The response status is **201 Created**, size is **222 Bytes**, and time is **174 ms**. The response body contains a message and a token:

```

1 {
2   "message": "Đăng ký thành công!",
3   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
4 .eyJpZCI6NCwicm9sZSI6ImN1c3RvbWYiIwiaWF0IjoxNzYzMxMD
5 Q3LCJleHAiOjE3NjMzMzQ2NDd9.Le6pRbGgOfU7Z8_I-dB5yl
6 -2MjAcvmbzLF8XqBkk3Q8",
7   "user_id": 4
8 }

```

Trường hợp trùng email thì sẽ không thể đăng ký được

The screenshot shows a POST request to `http://localhost:8080/api/auth/register`. The request body is identical to the previous one:

```

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 }

```

The response status is **400 Bad Request**, size is **51 Bytes**, and time is **5 ms**. The response body contains a message:

```

1 {
2   "message": "Email này đã được sử dụng."
3 }

```

2. Đăng nhập (POST /login)

The screenshot shows a POST request to `http://localhost:8080/api/auth/login`. The request body is a JSON object:

```

1 {
2   "email": "customer@gmail.com",
3   "password": "password123"
4 }

```

The response status is **200 OK**, size is **243 Bytes**, and time is **138 ms**. The response body contains a message, a token, and user information:

```

1 {
2   "message": "Đăng nhập thành công!",
3   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
4 .eyJpZCI6Mywicm9sZSI6ImN1c3RvbWYiIwiaWF0IjoxNzYzMzY0Mj
5 A2LCJleHAiOjE3NjMzMjcdMDZ9
6 .iKEHwtOUTcbV9wqPgpr1sPN94Fijkv2muqJm8MF1KBg",
7   "user_id": 3,
8   "role": "customer"
9 }

```

Trường hợp đăng nhập nhưng sai mật khẩu hoặc email

The screenshot shows a POST request to `http://localhost:8080/api/auth/login`. The request body is a JSON object:

```
1 {
2   "email": "customer@gmail.com",
3   "password": "password12345"
4 }
```

The response status is **401 Unauthorized**, size is **44 Bytes**, and time is **130 ms**. The response body is:

```
1 {
2   "message": "Sai email hoặc mật khẩu."
3 }
```

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 PUT request to `http://localhost:8080/api/auth/profile`. The request body is a JSON object:

```
1 {
2   "first_name": "D",
3   "last_name": "Nguyễn Văn",
4   "shipping_address": "123 Đường A, TPHCM",
5   "phone_number": "0123456789"
6 }
```

The response status is **200 OK**, size is **283 Bytes**, and time is **34 ms**. The response body is:

```
1 {
2   "message": "Cập nhật thông tin thành công!",
3   "user": {
4     "customer_id": 3,
5     "email": "customer@gmail.com",
6     "first_name": "D",
7     "last_name": "Nguyễn Văn",
8     "shipping_address": "123 Đường A, TPHCM",
9     "phone_number": "0123456789",
10    "role": "customer",
11    "date_of_birth": "2004-12-31T17:00:00.000Z"
12  }
13 }
```

Trường hợp người dùng chưa đăng nhập

The screenshot shows a PUT request to `http://localhost:8080/api/auth/profile`. The request body is a JSON object:

```
1 {
2   "first_name": "ABC",
3   "last_name": "Nguyễn Văn",
4   "shipping_address": "123 Đường ADE, TPHCM",
5   "phone_number": "0123456789"
6 }
```

The response status is **401 Unauthorized**, size is **52 Bytes**, and time is **6 ms**. The response body is:

```
1 {
2   "message": "chưa đăng nhập, không có token."
3 }
```

❖ Products (Quản lý sản phẩm)

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

The screenshot shows the Postman interface with a successful API call to `http://localhost:8080/api/products`. The response status is `200 OK`, size is `19.32 KB`, and time taken is `17 ms`. The response body is a JSON object representing a product:

```
[{"product_id": 1, "product_name": "iPhone 15 Pro Max 256GB", "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\n, tần số quét ProMotion 120Hz\n- Camera sau: 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", "price": "28990000.00", "stock_quantity": 34, "image_url": "https://product.hstatic.net/200000409445"}
```

2. Tìm kiếm sản phẩm (GET /products)

Nhập từ khóa bàn phím

The screenshot shows the Postman interface with a successful search query to `http://localhost:8080/api/products?search=bàn phím`. The response status is `200 OK`, size is `535 Bytes`, and time taken is `6 ms`. The response body is a JSON object representing a product found by the search term:

```
[{"product_id": 7, "product_name": "Bàn phím cơ Aula F75", "description": "THÔNG TIN SẢN PHẨM Aula F75\n- LAYOUT 75%\n- MẠCH XUỐI\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", "price": "999000.00", "stock_quantity": 14, "image_url": "https://bizweb.dktcdn.net/thumb/1024x1024/100/438/322/products/main-07.jpg?v=1761806374973", "category_name": "Laptop & PC"}]
```

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

Chỉ có thể thêm sản phẩm mới khi là admin

POST <http://localhost:8080/api/products>

Status: 201 Created Size: 60 Bytes Time: 12 ms

```

1 {
2   "message": "Tạo sản phẩm thành công",
3   "productId": 32
4 }
```

Trường hợp không phải là admin

POST <http://localhost:8080/api/products>

Status: 403 Forbidden Size: 39 Bytes Time: 10 ms

```

1 {
2   "message": "Không có quyền Admin."
3 }
```

❖ 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

POST <http://localhost:8080/api/cart>

Status: 201 Created Size: 68 Bytes Time: 93 ms

```

1 {
2   "message": "Đã thêm/cập nhật sản phẩm vào giỏ hàng."
3 }
```

Trường hợp chưa đăng nhập

The screenshot shows a POST request to `http://localhost:8080/api/cart`. The 'Auth' tab is selected, showing 'Bearer Token' and a text input field labeled 'enter token'. The response status is 401 Unauthorized, with a message: "Chưa đăng nhập, không có token."

POST		http://localhost:8080/api/cart				Send
Query	Headers	Auth	Body	Tests	Pre Run	
None	Basic	Bearer	OAuth 2	NTLM	AWS	
Bearer Token						
enter token						
Token Prefix		Bearer				

Status: 401 Unauthorized Size: 52 Bytes Time: 5 ms

Response Headers Cookies Results Docs { } ≡

```
1 {
2   "message": "Chưa đăng nhập, không có token."
3 }
```

❖ Orders (Quản lý đơn hàng)

1. Checkout (POST /orders)

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

The screenshot shows a POST request to `http://localhost:8080/api/orders`. The 'Body' tab is selected, showing JSON content with a shipping address and payment method. The response status is 201 Created, with a message: "Đặt hàng thành công!" and an orderId of 3.

POST		http://localhost:8080/api/orders				Send
Query	Headers	Auth	Body	Tests	Pre Run	
JSON	XML	Text	Form	Form-encode	GraphQL	Binary
JSON Content						Format
{ "shipping_address": "123 Test", "payment_method": "COD" }						

Status: 201 Created Size: 52 Bytes Time: 68 ms

Response Headers Cookies Results Docs { } ≡

```
1 {
2   "message": "Đặt hàng thành công!",
3   "orderId": 3
4 }
```

Trường hợp giỏ hàng rỗng

The screenshot shows a POST request to `http://localhost:8080/api/orders`. The 'Body' tab is selected, showing JSON content with a shipping address and payment method. The response status is 400 Bad Request, with a message: "Giỏ hàng trống, không thể checkout."

POST		http://localhost:8080/api/orders				Send
Query	Headers	Auth	Body	Tests	Pre Run	
JSON	XML	Text	Form	Form-encode	GraphQL	Binary
JSON Content						Format
{ "shipping_address": "123 Test", "payment_method": "COD" }						

Status: 400 Bad Request Size: 57 Bytes Time: 8 ms

Response Headers Cookies Results Docs { } ≡

```
1 {
2   "message": "Giỏ hàng trống, không thể checkout."
3 }
```

2. Đổi status đơn hàng (PUT /status)

Chỉ có thể cập nhật status khi là admin

PUT http://localhost:8080/api/orders/1/status Send

Query Headers Auth Body Tests Pre Run

None Basic Bearer OAuth 2 NTLM AWS

Bearer Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJpZCI6MSwicm9sZSI6ImFkbWluIiwiaWF0IjoxNzYzNzNmMjMzMjEhAioJE3NjM3MzM4MzN9.XpD9KLKYF5YD_K8Ar_Z_EU4mvE773Dkt40HbpAe3MY

Status: 200 OK Size: 69 Bytes Time: 9 ms

Response Headers Cookies Results Docs { } ≡

```
1 {
2   "message": "Cập nhật trạng thái đơn hàng #1 thành công."
3 }
```

Trường hợp không phải là admin

PUT http://localhost:8080/api/orders/1/status Send

Query Headers Auth Body Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 { "status": "Shipping" }
```

Status: 403 Forbidden Size: 39 Bytes Time: 5 ms

Response Headers Cookies Results Docs { } ≡

```
1 {
2   "message": "Không có quyền Admin."
3 }
```

2. Báo cáo tỷ lệ bao phủ kiểm thử

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

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

> backend@1.0.0 test
> jest --coverage

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
    ✓ TC-03: Báo lỗi nếu dữ liệu sai (Giá âm) (10 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

-----|-----|-----|-----|-----|-----|
File      | %Stmts | %Branch | %Funcs | %Lines | Uncovered Line #s |
-----|-----|-----|-----|-----|-----|
All files | 93.33  | 93.75   | 100    | 93.33  |                |
orderUtils.js | 93.33  | 93.75   | 100    | 93.33  | 34               |
-----|-----|-----|-----|-----|-----|

Test Suites: 1 passed, 1 total
Tests:      5 passed, 5 total
Snapshots:  0 total
Time:       0.881 s, estimated 1 s
Ran all test suites.
PS C:\Users\huynh\OneDrive\Desktop\Programming\WEBSITE CNPM\Backend>
```

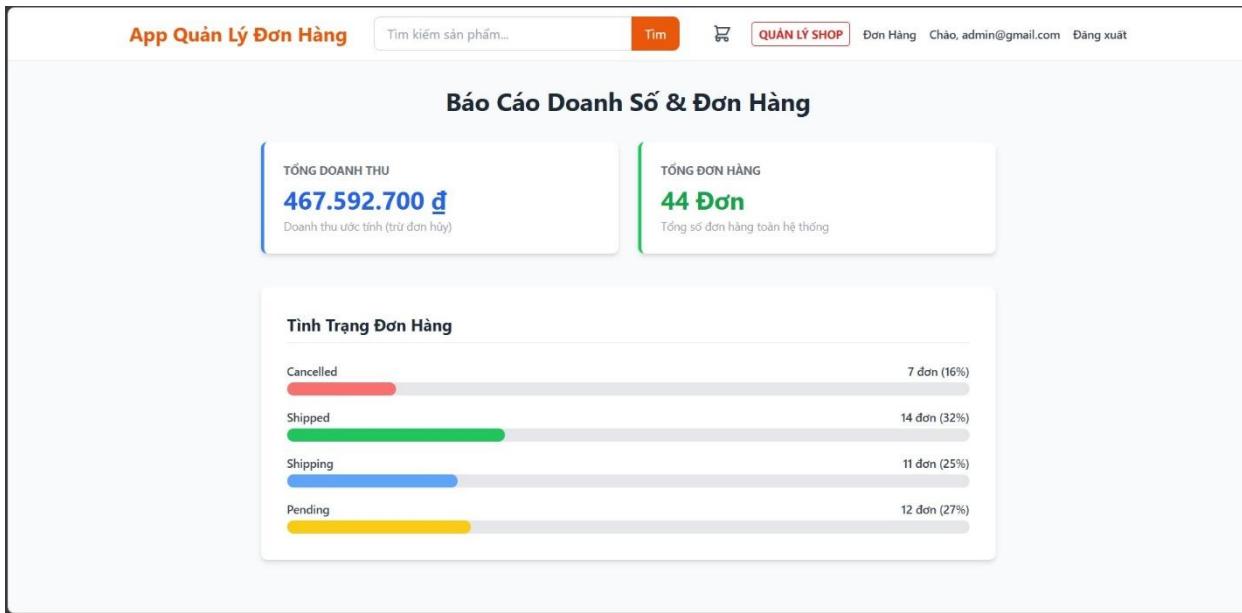


Kết quả:

- 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).

3. Báo cáo doanh số, trạng thái bán hàng



4. Đề xuất cải tiến

4.1. 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.

4.2. 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