## 2.1. Tool và thư viện sử dụng:

* **Testing Tool: Jest**
* **HTTP Request Simulation: Supertest**
* **Database ORM: Prisma**

## 2.2 test những function/class/file nào? vì sao bỏ không test những  function/file/class còn lại?

* **Test all function in category.js file**
* **Test all function in customer\_order\_product.js file**
* **Test all function in customer\_orders.js file**
* **Test all function in mainImages.js file**
* **Test all function in productImages.js file**
* **Test all function in slugs.js file**
* **Test all function in users.js file**
* **Test all function in wishlist.js file**
* **Test all function in Product.js file except Get All Products Old because it had not been exported in the controllers/Products.js**
* **Test function searchProducts in controllers/Search.js**

## 2.3 bộ test case:

**- cần thể hiện các thông tin: tên file/tên class**

**- với mỗi test case, cần thể hiện: mã của từng test case/mục tiêu của test/input/expected output/ghi chú.**

### 1. Controllers/catergory.js

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Target** | **Input** | **Expected Output** | **Note** |
| **TC01** | **Verify that a category is returned when the ID exists** | **Product ID, image data  ({ productID: testProduct.id, image: "new-image.gif" })** | **200 status, response contains the category data, and the category exists in the database.** | **Verifies database state matches the expected output.** |
| **TC02** | **Verify that a 404 error is returned for a non-existent category ID.** | **Non-existent Category ID (`"nonexistent-id"`)** | **404 status, response contains `{ error: "Category not found" }`, and no record exists in the database.** | **Ensures proper handling of non-existent IDs.** |
| **TC03** | **Verify that all categories are retrieved successfully.** | **No input** | **200 status, response contains an array of categories, and the array includes the test category.** | **Verifies the response includes all categories in the database.** |
| **TC04** | **Verify that a new category can be created successfully.** | **New category data**  **(`{ name: "Another Test Category" }`)** | **201 status, response contains the created category data, and the category exists in the database.** | **Verifies database state after creation.** |
| **TC05** | **Verify that an existing category can be updated successfully.** | **Category ID (`testCategory.id`), updated data (`{ name: "Updated Category" }`)** | **200 status, response contains the updated category data, and the database reflects the updated name.** | **Verifies database state after update** |
| **TC6** | **Verify that a 404 error is returned when updating a non-existent category.** | **Non-existent Category ID (`"nonexistent-id"`), updated data (`{ name: "Updated Category" }`)** | **404 status, response contains `{ error: "Category not found" }`.** | **Ensures proper handling of non-existent IDs during update.** |
| **TC7** | **Verify that a category can be deleted successfully.** | **Category ID (`categoryToDelete.id`)** | **204 status, no content in response, and the category no longer exists in the database.** | **Verifies database state after deletion.** |
| **TC08** | **Verify adding, updating, and deleting a record in the database.** | **New category data (`{ name: "Add Test Category" }`), updated data (`{ name: "Updated Test Category" }`)** | **Category is created, updated, and deleted successfully, with the database reflecting the correct state at each step.** | **Comprehensive test for CRUD operations.** |

### 2. Controllers/customer\_order\_product.test.js

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Target** | **Input** | **Expected Output** | **Note** |
| **TC01** | **Verify that a new order-product link can be created successfully** | **Valid order-product data: { customerOrderId, productId, quantity }** | **HTTP status 201, JSON response with created order-product link** | **Ensures Prisma [create](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) method is called with correct data.** |
| **TC02** | **Verify that an existing order-product link can be updated successfully** | **Valid update data: { customerOrderId, productId, quantity }** | **HTTP status 200, JSON response with updated order-product link** | **Ensures Prisma update method updates the record correctly.** |
| **TC03** | **Verify that updating a non-existent order-product link returns a 404 error** | **Non-existent [id](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), update data: { quantity }** | **HTTP status 404, JSON response: { error: "Order not found" }** | **Ensures the function handles non-existent records gracefully.** |
| **TC04** | **Verify that all links for a given [customerOrderId](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) can be deleted** | **Valid [customerOrderId](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)** | **HTTP status 204, no content in response** | **Ensures Prisma [deleteMany](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) method deletes all records for the given [customerOrderId](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).** |
| **TC05** | **Verify that all product links for a given [customerOrderId](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) can be retrieved** | **Valid [customerOrderId](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)** | **HTTP status 200, JSON response with an array of order-product links** | **Ensures Prisma [findMany](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) method retrieves all records for the given [customerOrderId](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).** |
| **TC06** | **Verify that retrieving product links for a non-existent [customerOrderId](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) returns an empty array** | **Non-existent**  **[customerOrderId](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)** | **HTTP status 200, JSON response: []** | **Ensures the function handles non-existent [customerOrderId](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) gracefully.** |
| **TC07** | **Verify that all orders are grouped by customer with product details** | **No input required** | **HTTP status 200, JSON response with grouped orders and product details** | **Ensures Prisma [findMany](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) method retrieves grouped data with relationships.** |

### 3. Controllers/customer\_orders.js

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Target** | **Input** | **Expected Output** | **Note** |
| **TC01** | **Verify fetching an order by ID when the order exists** | **Valid [id](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) of an existing order** | **HTTP status 200, JSON response with order details** | **Ensures the function retrieves the correct order from the database.** |
| **TC02** | **Verify 404 response for non-existent order ID** | **Non-existent [id](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)** | **HTTP status 404, JSON response: { error: "Order not found" }** | **Ensures the function handles non-existent orders gracefully.** |
| **TC03** | **Verify fetching all orders** | **No input** | **HTTP status 200, JSON response with an array of all orders** | **Ensures the function retrieves all orders from the database.** |
| **TC04** | **Verify creating a new order** | **Valid order data: { name, lastname, email, ... }** | **HTTP status 201, JSON response with created order details** | **Ensures the function creates a new order and saves it to the database.** |
| **TC05** | **Verify updating an existing order** | **Valid [id](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) of an existing order, updated data: { status, total, email, ... }** | **HTTP status 200, JSON response with updated order details** | **Ensures the function updates the order correctly in the database.** |
| **TC06** | **Verify 404 response when updating a non-existent order** | **Non-existent [id](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), updated data: { status }** | **HTTP status 404, JSON response: { error: "Order not found" }** | **Ensures the function handles updates for non-existent orders gracefully.** |
| **TC07** | **Verify deleting an order** | **Valid [id](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) of an existing order** | **HTTP status 204, no content in response** | **Ensures the function deletes the order from the database.** |
| **TC08** | **Verify adding, updating, and deleting a record in the database** | **Add: { name, lastname, email, ... }, Update: { status, total }, Delete: [id](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)** | **Add: Order exists in DB, Update: Updated order exists in DB, Delete: Order removed** | **Ensures the function handles the full lifecycle of a record (add, update, delete).** |

### 4. Controllers/mainImages.js

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Target** | **Input** | **Expected Output** | **Note** |
| **TC01** | **Verify that a category is returned when the ID exists** | **Product ID, image data  ({ productID: testProduct.id, image: "new-image.gif" })** | **200 status, response contains the category data, and the category exists in the database.** | **Verifies database state matches the expected output.** |
| **TC02** | **Verify that a 404 error is returned for a non-existent category ID.** | **Non-existent Category ID (`"nonexistent-id"`)** | **404 status, response contains `{ error: "Category not found" }`, and no record exists in the database.** | **Ensures proper handling of non-existent IDs.** |
| **TC03** | **Verify that all categories are retrieved successfully.** | **No input** | **200 status, response contains an array of categories, and the array includes the test category.** | **Verifies the response includes all categories in the database.** |
| **TC04** | **Verify that a new category can be created successfully.** | **New category data**  **(`{ name: "Another Test Category" }`)** | **201 status, response contains the created category data, and the category exists in the database.** | **Verifies database state after creation.** |
| **TC05** | **Verify that an existing category can be updated successfully.** | **Category ID (`testCategory.id`), updated data (`{ name: "Updated Category" }`)** | **200 status, response contains the updated category data, and the database reflects the updated name.** | **Verifies database state after update** |
| **TC6** | **Verify that a 404 error is returned when updating a non-existent category.** | **Non-existent Category ID (`"nonexistent-id"`), updated data (`{ name: "Updated Category" }`)** | **404 status, response contains `{ error: "Category not found" }`.** | **Ensures proper handling of non-existent IDs during update.** |
| **TC7** | **Verify that a category can be deleted successfully.** | **Category ID (`categoryToDelete.id`)** | **204 status, no content in response, and the category no longer exists in the database.** | **Verifies database state after deletion.** |
| **TC08** | **Verify adding, updating, and deleting a record in the database.** | **New category data (`{ name: "Add Test Category" }`), updated data (`{ name: "Updated Test Category" }`)** | **Category is created, updated, and deleted successfully, with the database reflecting the correct state at each step.** | **Comprehensive test for CRUD operations.** |

### 5. Controllers/ productImages.js

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Target** | **Input** | **Expected Output** | **Note** |
| **TC001** | **Verify that the function returns a 400 error when no files are uploaded** | **No [files](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) field in the request** | **HTTP status 400, JSON response: { message: "Nema otpremljenih fajlova" }** | **Ensures the function handles requests without files gracefully.** |
| **TC002** | **Verify that the function returns a 400 error when req.files is empty** | **[files](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) field is an empty object** | **HTTP status 400, JSON response: { message: "Nema otpremljenih fajlova" }** | **Ensures the function handles empty [files](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) objects gracefully.** |
| **TC003** | **Verify successful file upload** | **Valid file object in req.files** | **HTTP status 200, JSON response: { message: "Fajl je uspešno otpremljen" }** | **Ensures the function calls the [mv](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) method with the correct path and handles success.** |
| **TC004** | **Verify that the function returns a 500 error when the [mv](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) function fails** | **Valid file object in req.files, [mv](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) function throws an error** | **HTTP status 500, JSON response: Error object** | **Ensures the function handles errors from the [mv](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) method gracefully.** |

### 6. Controllers/ product.js

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Target** | **Input** | **Expected Output** | **Note** |
| **D1** | **Không xóa được sản phẩm có liên kết với đơn hàng** | **request.params.id = '1' (sản phẩm có liên kết)** | **status 400, JSON: { error: 'Cannot delete product because of foreign key constraint.' }** | **Pass** |
| **D2** | **Xóa sản phẩm nếu không có liên kết** | **request.params.id = '1' (không có liên kết)** | **status 204, gọi prisma.product.delete** | **Pass** |
| **D3** | **Xử lý lỗi cơ sở dữ liệu khi xóa** | **request.params.id = '1', giả lập lỗi DB** | **status 500, JSON: { error: 'Error deleting product' }** | **Pass** |
| **U1** | **Cập nhật sản phẩm thành công** | **params.id = PRODUCT\_ID, body = UPDATE\_DATA** | **status 200, JSON: sản phẩm đã cập nhật** | **Pass** |
| **U2** | **Trả về 404 nếu không tìm thấy sản phẩm** | **findUnique trả về null** | **status 404, JSON: { error: 'Product not found' }** | **Pass** |
| **U3** | **Hỗ trợ cập nhật một phần dữ liệu** | **body = { title: 'Partially Updated', price: 150 }** | **Gọi update với dữ liệu partial** | **Pass** |
| **U4** | **Xử lý lỗi khi cập nhật** | **Xử lý lỗi khi cập nhật** | **status 500, JSON: { error: 'Error updating product' }** | **Pass** |
| **U5** | **Duy trì quan hệ với category** | **body chứa categoryId mới** | **Gọi update với categoryId mới** | **Pass** |
| **S1** | **Trả về danh sách sản phẩm theo từ khóa tìm kiếm** | **req.query.keyword = 'test'** | **status 200, JSON chứa sản phẩm có title, description, hoặc slug khớp từ khóa** | **Pass** |
| **S2** | **Trả về tất cả sản phẩm nếu không có từ khóa** | **req.query.keyword = undefined** | **status 200, JSON chứa toàn bộ sản phẩm** | **Pass** |
| **S3** | **Xử lý lỗi từ Prisma** | **Prisma findMany lỗi** | **status 500, JSON: { error: 'Internal Server Error' }** | **Pass** |
| **A1** | **Lấy danh sách tất cả sản phẩm với category** | **Không cần input** | **status 200, JSON: danh sách sản phẩm với category.name** | **Pass** |
| **A2** | **Xử lý lỗi khi gọi findMany** | **Prisma lỗi** | **status 500, JSON: { error: 'Internal Server Error' }** | **Pass** |
| **B1** | **Lấy sản phẩm theo id thành công** | **req.params.id = '1', sản phẩm tồn tại** | **status 200, JSON: thông tin sản phẩm** | **Pass** |
| **B2** | **Trả về 404 nếu không tìm thấy sản phẩm theo id** | **req.params.id = '1', không có sản phẩm** | **status 404, JSON: { error: 'Product not found' }** | **Pass** |
| **B3** | **Xử lý lỗi khi truy vấn theo id** | **Prisma findUnique lỗi** | **status 500, JSON: { error: 'Internal Server Error' }** | **Pass** |

### 7. Controllers/ search.js

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Target** | **Input** | **Expected Output** | **Note** |
| **SP1** | **Return 400 error if no search query is provided** | **req.query = {}** | **status 400, JSON: { error: "Query parameter is required" }, database should not be queried** | **Pass** |
| **SP2** | **Search products by title or description** | **req.query = { query: "Test" }, Prisma returns matching mock data** | **status 200, JSON: [mockProducts], findMany called with OR on title and description** | **Pass** |
| **SP3** | **Return empty array if no matching products found** | **req.query = { query: "NonExistent" }, Prisma returns []** | **status 200, JSON: []** | **Pass** |
| **SP4** | **Handle database error during search** | **req.query = { query: "Test" }, Prisma findMany throws an error** | **status 500, JSON: { error: "Error searching products" }, error logged to console.error** | **Pass** |

### 8. Controllers/slugs.js

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Target** | **Input** | **Expected Output** | **Note** |
| **TC01** | **Verify that product details are returned when the slug exists** | **Valid [slug](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): "actual-test-product-slug-v2"** | **HTTP status 200, JSON response with product details** | **Ensures the function retrieves the correct product from the database.** |
| **TC02** | **Verify that a 404 error is returned when the slug does not exist** | **Non-existent [slug](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): "this-slug-definitely-does-not-exist"** | **HTTP status 404, JSON response: { error: "Product not found" }** | **Ensures the function handles non-existent slugs gracefully.** |

### 9. Controllers/ users.js

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Target** | **Input** | **Expected Output** | **Note** |
| **TC01** | **Verify that user data is returned when the email exists** | **Valid [email](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): "test@example.com"** | **HTTP status 200, JSON response with user details: { email: "test@example.com", role: "user" }** | **Ensures the function retrieves the correct user from the database.** |
| **TC02** | **Verify that a 404 error is returned when the email does not exist** | **Non-existent [email](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): "nonexistent@example.com"** | **HTTP status 404, JSON response: { error: "User not found" }** | **Ensures the function handles non-existent emails gracefully.** |
| **TC03** | **Verify adding, updating, and deleting a user in the database** | **Add: { email: "addtest@example.com", password: "testpass", role: "user" }, Update: { email: "updated@example.com" }, Delete: [id](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)** | **Add: User exists in DB, Update: Updated user exists in DB, Delete: User removed** | **Ensures the function handles the full lifecycle of a user record (add, update, delete).** |

### 10. Controllers/wishlist.js

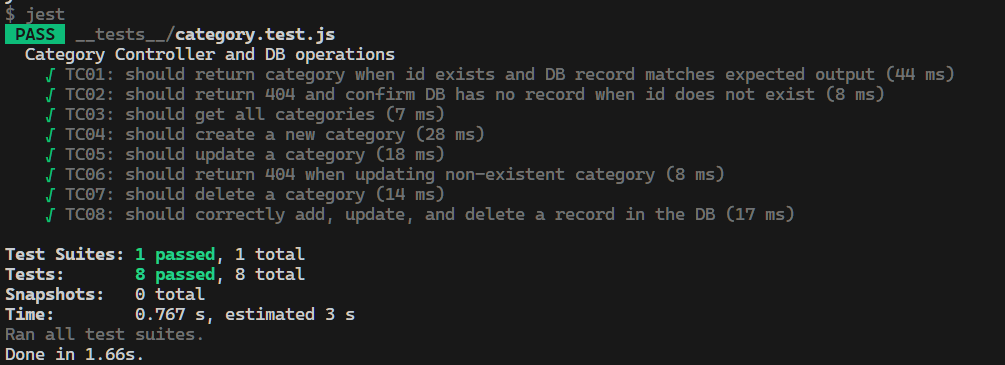
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Target** | **Input** | **Expected Output** | **Note** |
| **TC01** | **Verify creating a new wishlist item** | **{ userId: "test-user-id-1", productId: "test-prod-wishlist-id-2" }** | **HTTP status 201, JSON response with created wishlist item** | **Ensures the function creates a new wishlist item and saves it to the database.** |
| **TC02** | **Verify creating a duplicate wishlist item (if no unique constraint)** | **{ userId: "test-user-id-1", productId: "test-prod-wishlist-id-1" }** | **HTTP status 201, JSON response with created wishlist item** | **Ensures the function allows duplicate wishlist items if schema permits.** |
| **TC03** | **Verify error when creating a wishlist item with an invalid user ID** | **{ userId: "non-existent-user", productId: "test-prod-wishlist-id-1" }** | **HTTP status 500, JSON response: { error: "Error creating wish item" }** | **Ensures the function handles invalid user IDs gracefully.** |
| **TC04** | **Verify error when creating a wishlist item with an invalid product ID** | **{ userId: "test-user-id-1", productId: "non-existent-product" }** | **HTTP status 500, JSON response: { error: "Error creating wish item" }** | **Ensures the function handles invalid product IDs gracefully.** |
| **TC05** | **Verify fetching wishlist items for a specific user** | [userId: "test-user-id-1"](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) | **HTTP status 200, JSON response with an array of wishlist items** | **Ensures the function retrieves all wishlist items for the specified user.** |
| **TC06** | **Verify fetching an empty wishlist for a user with no items** | [userId: "test-user-id-2"](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) | **HTTP status 200, JSON response: []** | **Ensures the function handles users with no wishlist items gracefully.** |
| **TC07** | **Verify fetching an empty wishlist for a non-existent user ID** | [userId: "non-existent-user"](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) | **HTTP status 200, JSON response: []** | **Ensures the function handles non-existent user IDs gracefully.** |
| **TC08** | **Verify fetching a specific wishlist item** | [userId: "test-user-id-1", productId: "test-prod-wishlist-id-1"](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) | **HTTP status 200, JSON response with the specific wishlist item** | **Ensures the function retrieves the correct wishlist item for the user and product.** |
| **TC09** | **Verify fetching an empty result for a non-existent wishlist item** | [userId: "test-user-id-1", productId: "test-prod-wishlist-id-2"](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) | **HTTP status 200, JSON response: []** | **Ensures the function handles non-existent wishlist items gracefully.** |
| **TC10** | **Verify fetching an empty result for a non-existent user ID** | [userId: "non-existent-user", productId: "test-prod-wishlist-id-1"](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) | **HTTP status 200, JSON response: []** | **Ensures the function handles non-existent user IDs gracefully.** |
| **TC11** | **Verify fetching an empty result for a non-existent product ID** | [userId: "test-user-id-1", productId: "non-existent-product"](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) | **HTTP status 200, JSON response: []** | **Ensures the function handles non-existent product IDs gracefully.** |
| **TC12** | **Verify deleting a specific wishlist item** | [userId: "test-user-id-1", productId: "test-prod-wishlist-id-1"](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) | **HTTP status 204, no content in response** | **Ensures the function deletes the specified wishlist item from the database.** |
| **TC13** | **Verify deleting a non-existent wishlist item** | [userId: "test-user-id-1", productId: "test-prod-wishlist-id-2"](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) | **HTTP status 204, no content in response** | **Ensures the function handles deletion of non-existent wishlist items gracefully.** |
| **TC14** | **Verify deleting a wishlist item for non-existent user or product IDs** | [userId: "non-existent-user", productId: "test-prod-wishlist-id-1"](vscode-file://vscode-app/c:/Users/Manh%20Luong/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) | **HTTP status 204, no content in response** | **Ensures the function handles deletion for non-existent user or product IDs gracefully.** |
| **TC15** | **Verify fetching all wishlist items globally** | **No input** | **HTTP status 200, JSON response with an array of all wishlist items** | **Ensures the function retrieves all wishlist items from the database.** |
| **TC16** | **Verify fetching an empty result when no wishlist items exist globally** | **No input** | **HTTP status 200, JSON response: []** | **Ensures the function handles an empty wishlist database gracefully.** |

## 2.4. link dự án trên github:

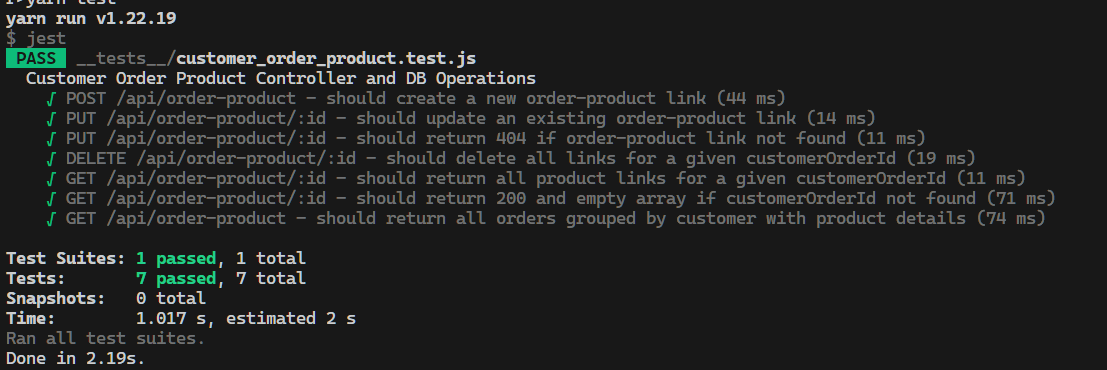
**<https://github.com/Kuzma02/Electronics-eCommerce-Shop-With-Admin-Dashboard-NextJS-NodeJS>**

## 2.5. Báo cáo kết quả chạy test + ảnh chụp minh chứng kết quả chạy test

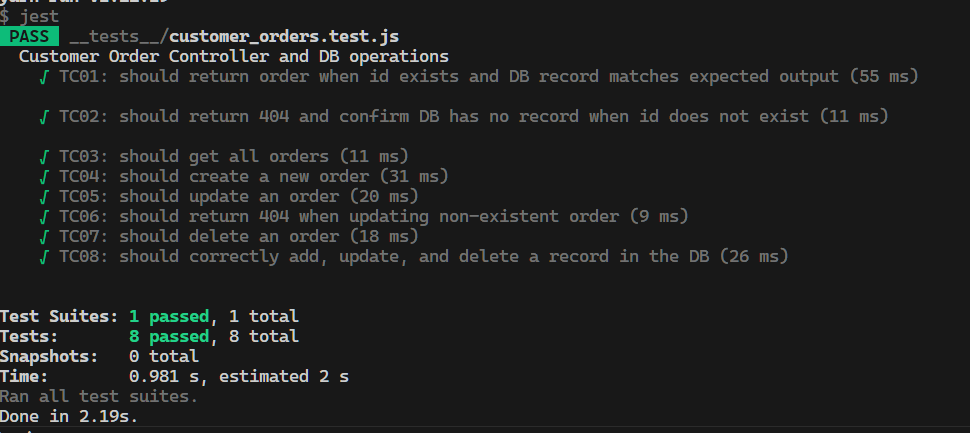
### 2.4.1. category.test.js

****

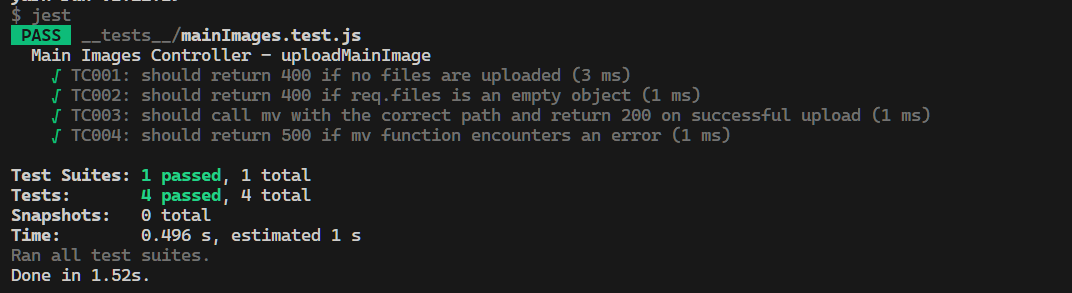
### 2.4.2. customer\_order\_product.test.js

****

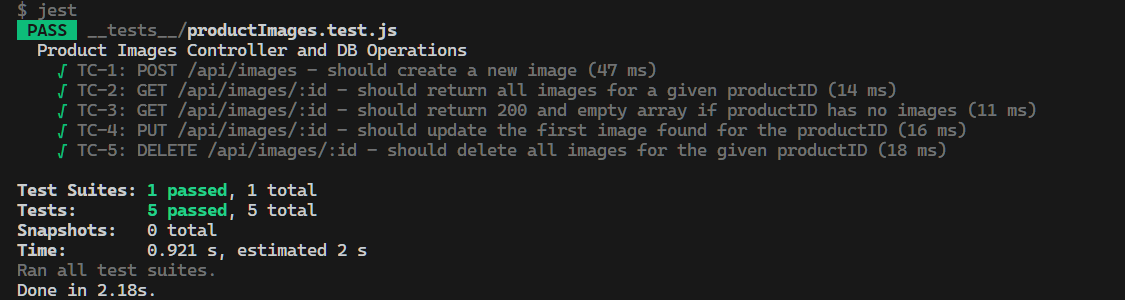
### 2.4.3. customer\_orders.test.js

****

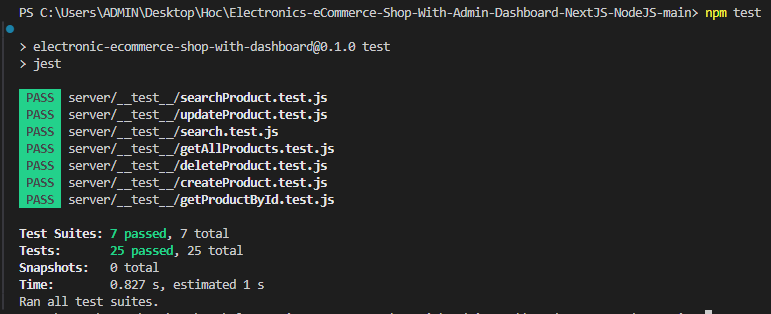
### 2.4.4. mainImages.test.js

****

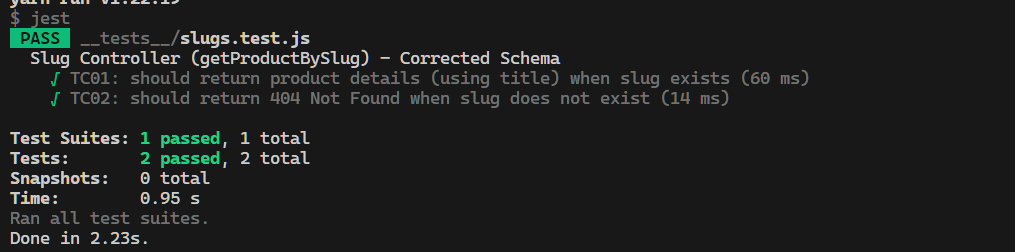
### 2.4.5. productImages.test.js

****

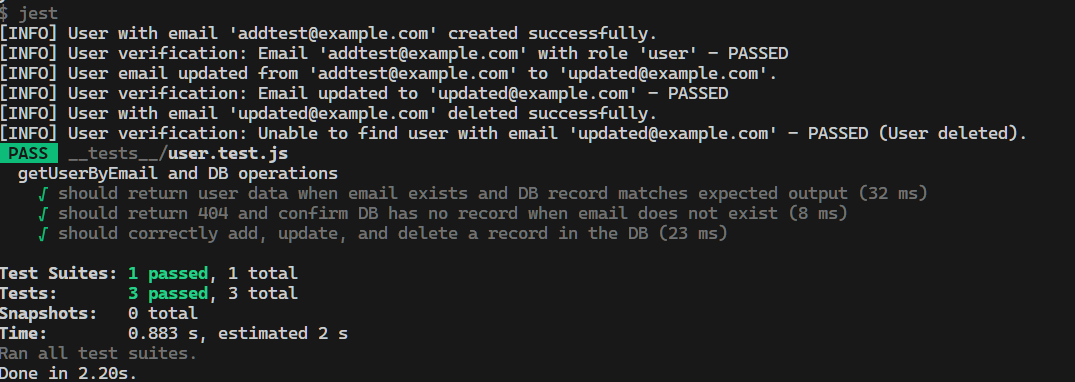
### 2.4.6. products.test.js + search.test.js

****

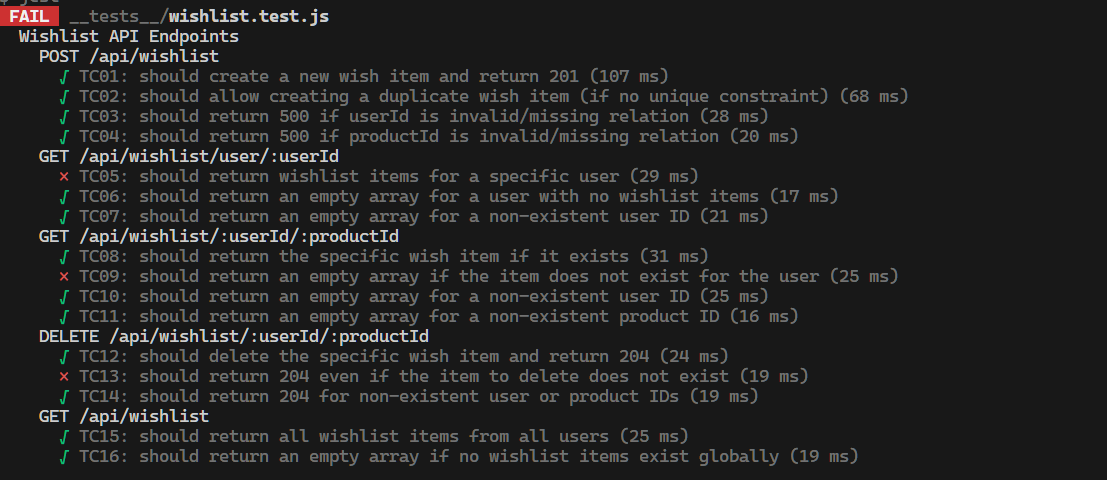
### 2.4.7. slugs.test.js

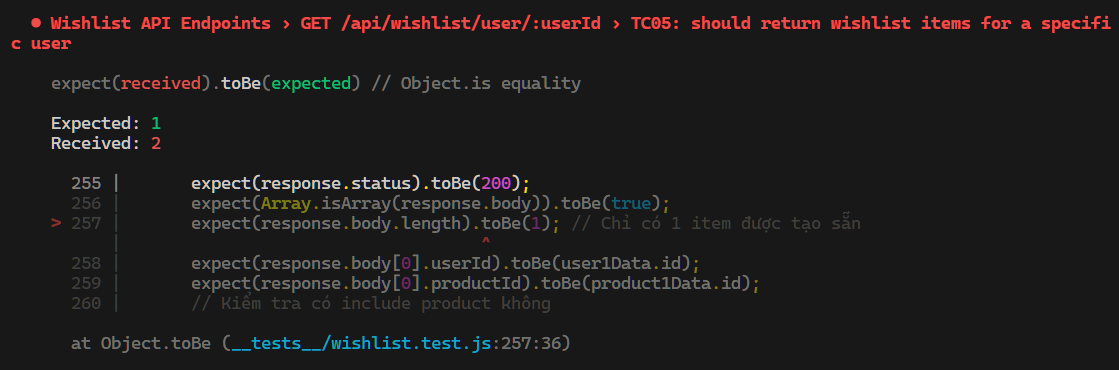
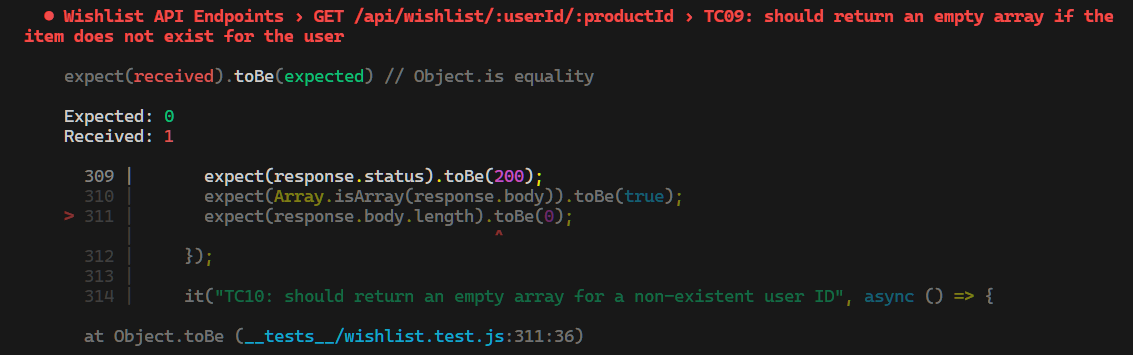
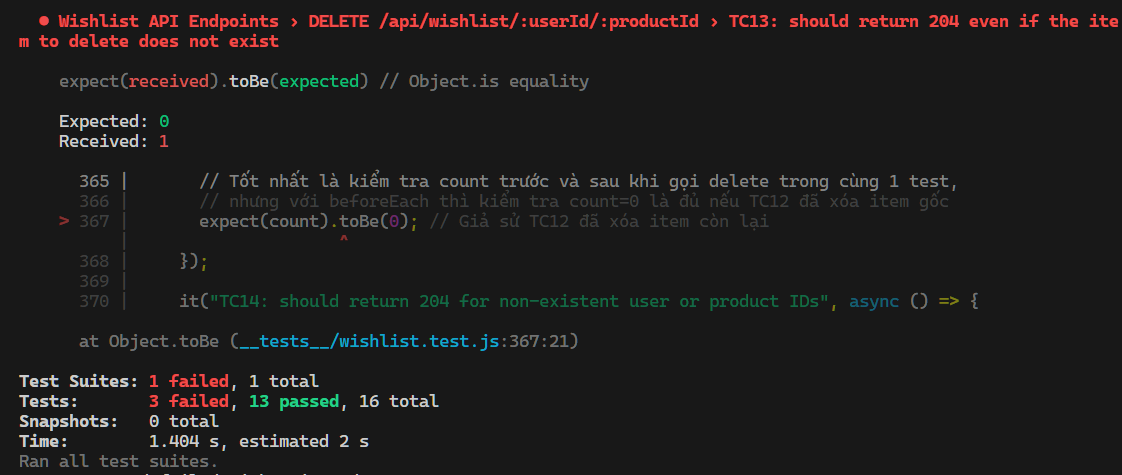
****

### 2.4.8. user.test.js



### 2.4.9. wishlist.test.js



## 2.6. Báo cáo kết quả độ phủ + ảnh chụp minh chứng kết quả độ phủ (chạy tool ra)

