

Test Case Documentation for Advanced Software Development Project

23085483 – Tran Duy Anh
23085487 – Nguyen Nhat Minh
23085492 – Nguyen Duc Trung

Introduction

This document provides evidence of testing for an e-commerce software system built with FastAPI and SQLAlchemy. The test cases cover key functionalities including user authentication, payment processing, order management, category management, search, product management, admin operations, shop management, and cart operations. The tests include valid data, invalid data, boundary cases, and error handling to ensure system robustness.

Test Case Table

Test Case #	Description	Test Dataset/Input	Expected Output	Actual Output
1	Test successful user registration	Username: "testuser", Email: "test@example.com", Phone: "1234567890", Password: "testpassword123"	HTTP 200: "User created successfully", password hashed with bcrypt	As expected
2	Test registration with duplicate email	Register user with email "test@example.com" twice	HTTP 400: "Email already registered"	As expected
3	Test login with invalid credentials	Email: "wrong@example.com", Password: "wrongpassword"	HTTP 401: "Invalid email or password"	As expected
4	Test token expiration	Login, wait 2 seconds (mocked 1-second expiration), access /auth/profile	HTTP 401: "Token has expired"	As expected
5	Test valid payment creation	Payment method: "Credit Card", Card number: "4111111111111111", CVV: "123", Expiry: "12/25"	Payment object created, card number encrypted	As expected
6	Test payment with invalid CVV	Card number: "4111111111111111", CVV: "12", Expiry: "12/25"	ValueError: Invalid CVV	As expected
7	Test create order with valid data	Shop ID: 1, Payment ID: 1, Items: [product_id: 1, quantity: 2], Delivery address: "1 Westminster Bridge Rd, London"	HTTP 200: Order created with correct shop_id and items	As expected
8	Test get user orders	Create order, then access /order/list with valid JWT token	HTTP 200: Returns list of orders, including created order	As expected

Test Case #	Description	Test Dataset/Input	Expected Output	Actual Output
9	Test create category as admin	Admin user, Category name: "Electronics"	HTTP 200: Category created with name "Electronics"	As expected
10	Test create category by non-admin	Regular user, Category name: "Electronics"	HTTP 403: "Admin access required"	As expected
11	Test search by product name	Product: "Coffee Beans", Search query: "Coffee", Search type: "products"	HTTP 200: Returns 1 product with name "Coffee Beans"	As expected
12	Test search with non-existent name	Search query: "Nonexistent-Name"	HTTP 200: Returns empty list	As expected
13	Test product creation with minimal data	Shop ID: 1, Name: "Test Product", Price: 9.99, Stock: 10	Product created, category and description are None	As expected
14	Test admin get all users	Admin user, Access /admin/users with valid JWT token	HTTP 200: Returns list of all users	As expected
15	Test regular user access admin endpoint	Regular user, Access /admin/users	HTTP 403: Forbidden	As expected
16	Test create shop by shop owner	Shop name: "Test Shop", Description: "A test shop", Address: "1600 Amphitheatre Parkway", Valid JWT token	HTTP 200: Shop created with correct details	As expected
17	Test get shop statistics	Create shop, add order (total price: 100.0), access /shops/stats/id	HTTP 200: Returns stats (shipped_orders, total_orders, total_revenue)	As expected
18	Test add item to cart	Product ID: 1, Shop ID: 1, Quantity: 2, Valid JWT token	HTTP 200: "Item added to cart successfully"	As expected
19	Test checkout cart	Add item to cart, Checkout with delivery address: "123 Test St, London, UK", Payment ID: 1	HTTP 200: Returns order_ids	As expected
20	Test add to cart with insufficient stock	Product ID: 1, Shop ID: 1, Quantity: 100	HTTP 400: "Not enough stock available"	As expected
21	Test update order status	Create order, Update status to "processing"	HTTP 200: Order status updated to "processing"	As expected
22	Test delete category as admin	Create category "Electronics", Delete with admin JWT token	HTTP 200: "Category deleted successfully"	As expected
23	Test search with case-insensitive query	Shop: "COFFEE SHOP", Search query: "coffee"	HTTP 200: Returns 1 shop with name "COFFEE SHOP"	As expected
24	Test admin get revenue statistics	Admin user, Access /admin/stats/revenue	HTTP 200: Returns total_revenue, product_revenue, shipping_revenue, revenue_over_time	As expected

Test Case #	Description	Test Dataset/Input	Expected Output	Actual Output
25	Test delete own admin account	Admin user, Attempt to delete own account	HTTP 400: "Cannot delete own account"	As expected

Notes

Test Coverage

The test cases cover all major modules (authentication, payment, order, category, search, product, admin, shop, cart) with comprehensive testing of valid inputs, invalid inputs, boundary cases, and error handling.

Unit Tests

Tests are implemented using `pytest` with `FastAPI TestClient` and `SQLModel` for database interactions. The `conf/test.py` file configures an in-memory SQLite database for isolated testing.

Screenshots

```

PS D:\Hoc Tập\Năm 2\advanced software development\lab\COURSE_WORK\advancedsoftwarewe_group1> pytest -v
===== test session starts =====
platform win32 -- Python 3.11.0, pytest-8.3.5, pluggy-1.5.0 -- d:\Hoc Tập\Năm 2\advanced software development\lab\COURSE_WORK\advancedsoftwarewe_group1\venv\Scripts\python.exe
cachedir: .pytest_cache
rootdir: D:\Hoc Tập\Năm 2\advanced software development\lab\COURSE_WORK\advancedsoftwarewe_group1
configfile: pytest.ini
testpaths: app/tests
plugins: anyio-4.8.0
collected 75 items

app/tests/test_admin.py::test_get_all_users_as_admin PASSED [ 1%]
app/tests/test_admin.py::test_get_all_users_as_regular_user PASSED [ 2%]
app/tests/test_admin.py::test_get_shop_owners PASSED [ 4%]
app/tests/test_admin.py::test_get_specific_shop_owner PASSED [ 5%]
app/tests/test_admin.py::test_delete_shop_owner PASSED [ 6%]
app/tests/test_admin.py::test_get_user_statistics PASSED [ 8%]
app/tests/test_admin.py::test_get_revenue_statistics PASSED [ 9%]
app/tests/test_admin.py::test_get_product_statistics PASSED [ 10%]
app/tests/test_admin.py::test_delete_admin_account_fail PASSED [ 12%]
app/tests/test_admin.py::test_get_categories PASSED [ 13%]
app/tests/test_auth.py::test_signup_success PASSED [ 14%]
app/tests/test_auth.py::test_signup_duplicate_email PASSED [ 16%]
app/tests/test_auth.py::test_signup_invalid_data PASSED [ 17%]
app/tests/test_auth.py::test_invalid_token PASSED [ 18%]
app/tests/test_auth.py::test_token_expiration PASSED [ 20%]
app/tests/test_auth.py::test_login_success PASSED [ 21%]
app/tests/test_auth.py::test_login_missing_user PASSED [ 22%]
app/tests/test_auth.py::test_login_invalid_credentials PASSED [ 24%]
app/tests/test_auth.py::test_get_profile PASSED [ 25%]
app/tests/test_auth.py::test_update_profile PASSED [ 26%]
app/tests/test_auth.py::test_update_profile_invalid_data PASSED [ 28%]
app/tests/test_auth.py::test_get_user_role PASSED [ 29%]
app/tests/test_auth.py::test_unauthorized_access PASSED [ 30%]
app/tests/test_cart.py::test_add_to_cart PASSED [ 32%]
app/tests/test_cart.py::test_get_cart_items PASSED [ 33%]
app/tests/test_cart.py::test_update_cart_item PASSED [ 34%]
app/tests/test_cart.py::test_remove_from_cart PASSED [ 36%]
app/tests/test_cart.py::test_checkout_cart PASSED [ 37%]
app/tests/test_cart.py::test_add_to_cart_invalid_product PASSED [ 38%]
app/tests/test_cart.py::test_add_to_cart_insufficient_stock PASSED [ 40%]

```

Figure 1: Pytest verbose output showing all tests passed.

app/tests/test_cart.py::test_add_to_cart PASSED	[32%]
app/tests/test_cart.py::test_get_cart_items PASSED	[33%]
app/tests/test_cart.py::test_update_cart_item PASSED	[34%]
app/tests/test_cart.py::test_remove_from_cart PASSED	[36%]
app/tests/test_cart.py::test_checkout_cart PASSED	[37%]
app/tests/test_cart.py::test_add_to_cart_invalid_product PASSED	[38%]
app/tests/test_cart.py::test_add_to_cart_insufficient_stock PASSED	[40%]
app/tests/test_category.py::test_create_category PASSED	[41%]
app/tests/test_category.py::test_create_category_unauthorized PASSED	[42%]
app/tests/test_category.py::test_create_category_unauthorized PASSED	[42%]
app/tests/test_category.py::test_get_category PASSED	[44%]
app/tests/test_category.py::test_create_category_unauthorized PASSED	[42%]
app/tests/test_category.py::test_get_category PASSED	[44%]
app/tests/test_category.py::test_update_category PASSED	[45%]
app/tests/test_category.py::test_update_category_unauthorized PASSED	[46%]
app/tests/test_category.py::test_delete_category PASSED	[48%]
app/tests/test_category.py::test_delete_category_unauthorized PASSED	[49%]
app/tests/test_category.py::test_get_all_categories PASSED	[50%]
app/tests/test_order.py::test_create_order PASSED	[52%]
app/tests/test_order.py::test_get_user_orders PASSED	[53%]
app/tests/test_order.py::test_get_shop_orders PASSED	[54%]
app/tests/test_order.py::test_add_to_cart PASSED	[56%]
app/tests/test_order.py::test_view_cart PASSED	[57%]
app/tests/test_order.py::test_remove_from_cart PASSED	[58%]
app/tests/test_order.py::test_cart_checkout PASSED	[60%]
app/tests/test_order.py::test_update_order_status PASSED	[61%]
app/tests/test_order.py::test_get_specific_order PASSED	[62%]
app/tests/test_order.py::test_delete_order PASSED	[64%]
app/tests/test_payment.py::test_payment_create_validation PASSED	[65%]
app/tests/test_payment.py::test_payment_read PASSED	[66%]
app/tests/test_payment.py::test_expiry_date_validation PASSED	[68%]
app/tests/test_payment.py::test_card_encryption PASSED	[69%]
app/tests/test_product.py::test_product_create_schema PASSED	[70%]
app/tests/test_product.py::test_product_read_schema PASSED	[72%]
app/tests/test_product.py::test_product_update_schema PASSED	[73%]
app/tests/test_product.py::test_product_image_schemas PASSED	[74%]
app/tests/test_search.py::test_empty_search PASSED	[76%]
app/tests/test_search.py::test_search_by_shop_name PASSED	[77%]
app/tests/test_search.py::test_search_by_product_name PASSED	[78%]
app/tests/test_search.py::test_search_by_product_description PASSED	[80%]
app/tests/test_search.py::test_search_by_category PASSED	[81%]

Figure 2: Coverage report for unit tests.

app/tests/test_category.py::test_get_all_categories PASSED	[50%]
app/tests/test_order.py::test_create_order PASSED	[52%]
app/tests/test_order.py::test_get_user_orders PASSED	[53%]
app/tests/test_order.py::test_get_shop_orders PASSED	[54%]
app/tests/test_order.py::test_add_to_cart PASSED	[56%]
app/tests/test_order.py::test_view_cart PASSED	[57%]
app/tests/test_order.py::test_remove_from_cart PASSED	[58%]
app/tests/test_order.py::test_cart_checkout PASSED	[60%]
app/tests/test_order.py::test_update_order_status PASSED	[61%]
app/tests/test_order.py::test_get_specific_order PASSED	[62%]
app/tests/test_order.py::test_delete_order PASSED	[64%]
app/tests/test_payment.py::test_payment_create_validation PASSED	[65%]
app/tests/test_payment.py::test_payment_read PASSED	[66%]
app/tests/test_payment.py::test_expiry_date_validation PASSED	[68%]
app/tests/test_payment.py::test_card_encryption PASSED	[69%]
app/tests/test_product.py::test_product_create_schema PASSED	[70%]
app/tests/test_product.py::test_product_read_schema PASSED	[72%]
app/tests/test_product.py::test_product_update_schema PASSED	[73%]
app/tests/test_product.py::test_product_image_schemas PASSED	[74%]
app/tests/test_search.py::test_empty_search PASSED	[76%]
app/tests/test_search.py::test_search_by_shop_name PASSED	[77%]
app/tests/test_search.py::test_search_by_product_name PASSED	[78%]
app/tests/test_search.py::test_search_by_product_description PASSED	[80%]
app/tests/test_search.py::test_search_by_category PASSED	[81%]
app/tests/test_search.py::test_search_type_shops_only PASSED	[82%]
app/tests/test_search.py::test_search_with_nonexistent_name PASSED	[84%]
app/tests/test_search.py::test_search_with_partial_name PASSED	[85%]
app/tests/test_search.py::test_search_with_case_insensitive PASSED	[86%]
app/tests/test_search.py::test_search_special_characters PASSED	[88%]
app/tests/test_search.py::test_search_with_multiple_results PASSED	[89%]
app/tests/test_search.py::test_search_with_invalid_search_type PASSED	[90%]
app/tests/test_shop.py::test_create_shop_success PASSED	[92%]
app/tests/test_shop.py::test_get_all_shops PASSED	[93%]
app/tests/test_shop.py::test_get_shop_by_id PASSED	[94%]
app/tests/test_shop.py::test_update_shop PASSED	[96%]
app/tests/test_shop.py::test_delete_shop PASSED	[97%]
app/tests/test_shop.py::test_get_owner_shops PASSED	[98%]
app/tests/test_shop.py::test_get_shop_stats PASSED	[100%]
===== 75 passed in 42.59s =====	

Figure 3: Test summary dashboard.