TRƯỜNG ĐẠI HỌC SÀI GÒN

KHOA CÔNG NGHỆ THÔNG TIN



KIỂM THỬ PHẦN MỀM

**BàiTập Lớn - Kiểm Thử Phần Mềm**

**Ứng dụng Đăng nhập & Quản lý Sản phẩm**

**(Version 1.0)**

GVHD:

Từ Lãng Phiêu

SV:

SV1 - 3123410314 - Kiều Tấn Tài

SV2 -

SV3 -

SV4 -

TP. HỒ CHÍ MINH, THÁNG 2/2024

**Mục lục**

1. [Giới thiệu về Dự án](#_bookmark0) 1
2. **Giới thiệu về dự án**
   1. **Tổng quan**

Dự án **FloginFE\_BE** là một ứng dụng web hoàn chỉnh bao gồm:

* **Chức năng Login:** Hệ thống đăng nhập với validation đầy đủ
* **Chức năng Product:** Quản lý sản phẩm (CRUD operations)
* **Frontend:** React 18+
* **Backend:** Spring Boot 3.2+
* **Testing:** Phát triển theo phương pháp TDD
  1. **Công nghệ sử dụng**
     1. **Frontend**
* React 18+ - Framework JavaScript
* React Testing Library - Testing cho React
* Jest - Testing framework
* Axios - HTTP client
* CSS3 - Styling với animations
  + 1. **Backend**
* Spring Boot 3.2+ - Framework Java
* Java 17+
* JUnit 5 - Testing framework
* Mockito - Mock framework
* Maven - Build tool
* Spring Data JPA – Database operations
  1. **Cấu trúc dự án**

**FloginFE\_BE/**

* **frontend/** - React Application
  + src/
    - components/ - Login, Productg components
    - services/ - API services
    - tests/ - Test files
  + package.json
* **backend/** - Spring Boot API
  + src /
    - main/java/com/flogin/
      * controller/ - AccountController, ProductController
      * service/ - AccountService, ProductService
      * dto/ - Data Transfer Objects
      * entity/ - Database entities
      * repository / - Data access
    - test/java/ - Test files
  + pom.xml

1. **Câu 1: Phân tích và thiết kế Test Cases**
   1. **Login**
      1. **Yêu cầu**
      2. **Phân tích đầy đủ yêu cầu chức năng của Login chi tiết**
         * Validation rules cho username handling
           + **Độ dài**: 3 - 50 ký tự.
           + **ký tự hợp lệ**: a-z, A-Z, 0-9, dấu gạch ngang(-), dấu chấm (.), dấu gạch dưới (\_).
           + **Không được chứa ký tự đặc biệt ngoài ký tự trên.**
           + **Không cho phép để trống.**
         * Validation rules cho password
           + **Độ dài:** 6 - 100 ký tự.
           + **Phải có ít nhất:** 1 ký tự chữ (a-z hoặc A-Z), 1 chữ số (0 – 9)
           + **Không cho phép để trống.**
         * Authentication flow
           + Người dùng nhập username và password
           + hệ thống kiểm tra hợp lệ của dữ liệu (validation rules trên).
           + Nếu dữ liệu không hợp lệ -> trả về lỗi và không gửi request xác thực
           + Nếu hợp lệ, gửi request tới server/backend để xác thực thông tin đăng nhập.
           + Backend kiểm tra username & password trong database:

Đúng: trả về token/session, cho phép truy cập tài khoản.

Sai: trả về lỗi, không cho truy cập.

* + - * + Hệ thống ghi nhận trạng thái phiên làm việc, chuyển huontgws vào hệ thống cho phiên thành công.
      * Error handling
        + **Username error:**  Hiển thị thông báo “Username không tồn tại hoặc sai”.
        + **Password error:** Hiển thị thông báo “Mật khẩu không đúng.”
        + **Invalid form Username or Password:** Thông báo rõ trường nào sai kèm hướng dẫn (ví dụ: “Username chỉ cho phép 3-50 ký tự, chỉ a-z, A-Z, 0-9, -,.,\_”).
        + **Empty value:** Thông báo “không được để trống”.
        + **Server error:** Thông báo lỗi “Có lỗi xảy ra, vui lòng thử lại sau.”
        + **Prevent try again when fail 3 times:** Thông báo “Tài khoản bị khóa tạm thời do nhập sai quá nhiều lần”.
    1. **Liệt kê và mô tả ít nhất 10 test scenarios cho Login** 
       1. **Happy path:**
          - Scenario: Đăng nhập thành công
          - Detail: Nhập đúng username và password hợp lệ
          - Expected Result: Báo lỗi “Password không được để trống”, không đăng nhập
       2. **Negative test:**
          - Scenario: Username để trống
          - Detail: Để trống trường username, nhập password hợp lệ
          - Expected Result: Báo lỗi “Username không được để trống”
       3. **Negative test:**
          - Scenario: Password để trống
          - Detail: Nhập trường username hợp lệ, để trống trường password
          - Expected Result: Báo lỗi “Username hoặc password không đúng”
       4. **Negative test:**
          - Scenario: Username chứa ký tự đặc biệt
          - Detail: Nhập username có ký tự lạ như !, @, #, password hợp lệ
          - Expected Result: Hiển thị lỗi “Username không hợp lệ”, không gửi yêu cầu lên server
       5. **Negative test:**
          - Scenario: Password không đáp ứng yêu cầu
          - Detail: Password toàn số, toàn chữ, hoặc dưới 6 ký tự, trên 100 ký tự
          - Expected Result: Báo lỗi “Password không hợp lệ”, không đăng nhập
       6. **Boundary test:**
          - Scenario: Username dưới 3 ký tự
          - Detail: Nhập 1–2 ký tự hợp lệ ở username, password hợp lệ
          - Expected Result: Báo lỗi “Username quá ngắn”
       7. **Boundary test:**
          - Scenario: Username trên 50 ký tự
          - Detail: Nhập username >50 ký tự, password hợp lệ
          - Expected Result: Báo lỗi “Username quá dài”
       8. **Boundary test:**
          - Scenario: Password dưới 6 ký tự
          - Detail: Nhập username là 'user name', password hợp lệ
          - Expected Result: Báo lỗi “Password quá ngắn”
       9. **Boundary test:**
          - Scenario: Password >100 ký tự
          - Detail: Username hợp lệ, password dài hơn 100 ký tự
          - Expected Result: Báo lỗi “Password quá dài”
       10. **Edge case:**
           - Scenario: Username có chứa space
           - Detail: Nhập username là 'user name', password hợp lệ
           - Expected Result: Hiển thị lỗi “Username không hợp lệ”
       11. **Negative test:**
           - Scenario:  Username đúng, password sai
           - Detail: Nhập username có tài khoản thật, password sai
           - Expected Result: Báo lỗi “Username hoặc password không đúng”
       12. **Negative/Edge test:**
           - Scenario: Nhập sai password nhiều lần
           - Detail: Thực hiện nhập sai password liên tục nhiều lần với cùng 1 username
           - Expected Result: Nếu vượt ngưỡng quy định, tài khoản bị khóa tạm thời hoặc hiện captcha xác thực
       13. **Edge case:**
           - Scenario: Username và password hợp lệ nhưng không có trong hệ thống
           - Detail: Nhập thông tin chưa đăng ký
           - Expected Result: Báo lỗi “Username hoặc password không đúng”
    2. **Phân loại test scenarios theo mức độ ưu tiên và giải thích**
       1. **Critical**
          - đăng nhập thành công (chức năng hệ thống)
          - Username để trống
          - Password để trống => 2 field quan trọng ko được phép để trống
          - Username sai định dạng
          - Password sai định dạng => ngăn input lỗi, injection
          - Username hợp lệ, password sai hoặc ngược lại => Lỗi hay xảy ra với user, cần xác thực và đưa ra thông báo rõ ràng
       2. **High**
          - username < 3 ký tự hoặc username > 50 ký tự
          - password < 6 ký tự hoặc password > 100 ký tự => Giới hạn độ dài dữ liệu
          - username, password có khoảng trống => edge case, tránh lỗi xử lý string
          - sai quá nhiều lần => tránh brute - force
       3. **Medium**
          - username, password đúng định dạng nhưng không tồn tại=> thông báo chính xác cho user, ít ảnh hưởng đến hệ thống
    3. **Thiết kế test cases chi tiết**

|  |  |
| --- | --- |
| Test case ID | TC \_LOGIN \_001 |
| Test Name | Đăng nhập thành công với cridentials hợp lệ |
| Priority | Critical |
| Preconditions | - User account exists  - Application’s running |
| Test Steps | 1. Navigate to login page  2. Enter valid username  3. Enter valid password  4. Click login button |
| Test Data | Username: testuser  Password: Test123 |
| Expected result | - Success message displayed  - Token stored  - Redirected to dashboard |
| Actual result | (Để trống) |
| Status | Not run |

|  |  |
| --- | --- |
| Test case ID | TC \_LOGIN \_002 |
| Test Name | Đăng nhập thất bại với username để trống |
| Priority | Critical |
| Preconditions | - Application’s running |
| Test Steps | 1. Navigate to login page  2. Enter empty  3. Enter valid password  4. Click login button |
| Test Data | Username:  Password: Test123 |
| Expected result | - Error message displayed |
| Actual result | (Để trống) |
| Status | Not run |

|  |  |
| --- | --- |
| Test case ID | TC \_LOGIN \_003 |
| Test Name | Đăng nhập thành thất bại với password sai định dạng |
| Priority | Critical |
| Preconditions | - User account exists  - Application’s running |
| Test Steps | 1. Navigate to login page  2. Enter valid username  3. Enter invalid password  4. Click login button |
| Test Data | Username: testuser  Password: 1)23 |
| Expected result | - Error message displayed |
| Actual result | (Để trống) |
| Status | Not run |

|  |  |
| --- | --- |
| Test case ID | TC \_LOGIN \_004 |
| Test Name | Đăng nhập thất bại với mật khẩu sai |
| Priority | Critical |
| Preconditions | - User account exists  - Application’s running |
| Test Steps | 1. Navigate to login page  2. Enter valid username  3. Enter wrong password  4. Click login button |
| Test Data | Username: testuser  Password: Test1234 |
| Expected result | - Error message displayed |
| Actual result | (Để trống) |
| Status | Not run |

|  |  |
| --- | --- |
| Test case ID | TC \_LOGIN \_005 |
| Test Name | Đăng nhập thất bại với username sai |
| Priority | Critical |
| Preconditions | - Application’s running |
| Test Steps | 1. Navigate to login page  2. Enter wrong username  3. Enter valid password  4. Click login button |
| Test Data | Username: testuserA  Password: Test123 |
| Expected result | - Error message displayed |
| Actual result | (Để trống) |
| Status | Not run |

* 1. **Product**
     1. **Yêu cầu**

1. **Phân tích đầy đủ yêu cầu chức năng product CRUDthích**
   * **Create**
     + Nhập thông tin sản phẩm: Product name, price, quantity, description, category.
     + Validation:
       - Product Name: Không rỗng, 3–100 ký tự.
       - Price: > 0, ≤ 999,999,999.
       - Quantity: ≥ 0, ≤ 99,999.
       - Description: ≤ 500 ký tự.
       - Category: Nằm trong danh mục cho phép.
     + Hành động: Kiểm tra hợp lệ dữ liệu, lưu sản phẩm vào database.
     + Phản hồi: báo thành công hoặc mô tả lỗi cụ thể.
   * **Read**
     + **Xem danh sách sản phẩm:** Cho phép lọc, tìm kiếm theo tên, category,..
     + **Xem chi tiết sản phẩm:** xem đầy đủ thông tin một sản phẩm đã lưu.
     + **Phản hồi:** Hiển thị đúng thông tin, xử lý trường hợp sản phẩm không tồn tại.
   * **Update**
     + Chọn sản phẩm cần sửa, nhập dữ liệu mới
     + Validation: giống phần create
     + Lưu lại dữ liệu mới vào database (nếu hợp lệ).
     + Phản hồi: báo thành công hoặc lỗi.
   * **Delete**
     + Chọn sản phẩm cần xóa (theo ID)
     + Điều kiện:
     + Hành động:
       - Sản phẩm phải tồn tại
       - Cót thể có quy định phụ: không xóa được nếu đang liên quan order,v.v
     + Phản hồi: Thông báo thành công hoặc lỗi (nếu không tồn tại)
2. **Liệt kê và mô tả ít nhất 10 Test scenarios cho Product**
3. ****Happy path: Thêm sản phẩm mới thành công****
   * Nhập đầy đủ các trường hợp lệ (name, price, quantity, description, category).
   * ****Kết quả:**** Sản phẩm được lưu vào hệ thống.
4. ****Happy path: Xem chi tiết sản phẩm****
   * Chọn sản phẩm đã tồn tại, truy cập trang chi tiết.
   * ****Kết quả:**** Hiển thị đúng đầy đủ thông tin.
5. ****Happy path: Cập nhật sản phẩm thành công****
   * Chỉnh sửa tên/giá/quantity của sản phẩm hợp lệ.
   * ****Kết quả:**** Sản phẩm được cập nhật.
6. ****Happy path: Xóa sản phẩm thành công****
   * Xóa sản phẩm đang có trong hệ thống.
   * ****Kết quả:**** Sản phẩm bị xóa; không tìm thấy khi query lại.
7. ****Negative: Thêm sản phẩm với tên rỗng****
   * Để trống trường tên khi thêm.
   * ****Kết quả:**** Báo lỗi “Tên sản phẩm không được rỗng”.
8. ****Negative: Nhập giá âm hoặc bằng 0****
   * Nhập price là 0 hoặc số âm khi thêm/cập nhật.
   * ****Kết quả:**** Báo lỗi “Giá phải lớn hơn 0”.
9. ****Negative: Category không hợp lệ****
   * Nhập category không thuộc danh mục cho phép.
   * ****Kết quả:**** Báo lỗi “Category không hợp lệ”.
10. ****Boundary: Tên sản phẩm dưới 3 ký tự****
    * Nhập name 1 hoặc 2 ký tự.
    * ****Kết quả:**** Báo lỗi “Tên quá ngắn”.
11. ****Boundary: Quantity vượt quá 99,999****
    * Nhập số lượng = 100,000 hoặc nhiều hơn.
    * ****Kết quả:**** Báo lỗi “Số lượng vượt quá lớn nhất cho phép”.
12. ****Edge: Thêm 2 sản phẩm trùng tên****
    * Thêm sản phẩm mới có cùng tên với sản phẩm đã tồn tại.
    * ****Kết quả:**** (Tùy policy, thông thường báo lỗi “Tên đã tồn tại”, hoặc chấp nhận nếu chỉ cần phân biệt theo ID.)
13. ****Edge: Xóa sản phẩm không tồn tại****
    * Thử xóa bằng ID không có trong database.
    * ****Kết quả:**** Báo “Không tìm thấy sản phẩm”.
14. ****Negative: Mô tả vượt quá 500 ký tự****
    * Nhập description dài hơn 500 ký tự.
    * ****Kết quả:**** Báo lỗi “Description quá dài”.
15. **Phân loại test scenarios theo mức độ ưu tiên và giải thích**
    * **Critical**
      + - Thêm sản phẩm mới thành công (1)
        - Xem chi tiết sản phẩm (2)
        - Cập nhật sản phẩm thành công (3)
        - Xóa sản phẩm thành công (4)
        - Thêm sản phẩm với tên rỗng (5)
        - Nhập giá âm hoặc bằng 0 (6)
        - Category không hợp lệ (7)

****Giải thích:**** Đây là các thao tác CRUD chính và các validation quan trọng giúp hệ thống không sinh dữ liệu lỗi, đảm bảo tính toàn vẹn dữ liệu.

* + **High**
    - * Tên sản phẩm quá ngắn (8)
      * Quantity quá lớn (9)
      * Mô tả quá dài (12)

**Giải thích:** Kiểm tra biên dữ liệu đầu vào, hạn chế lỗi phát sinh, tránh tràn dữ liệu.

* + **Medium**
    - * Thêm hai sản phẩm trùng tên (10)
      * Xóa sản phẩm không tồn tại (11)

****Giải thích:**  Edge case/negative test, ít xảy ra nhưng vẫn nên kiểm tra để cải thiện trải nghiệm người dùng và tránh lỗi "ẩn".**

* + 1. **Thiết kế Test cases chi tiết**

|  |  |
| --- | --- |
| Test case ID | TC \_PRODUCT \_001 |
| Test Name | Tạo sản phẩm mới thành công |
| Priority | Critical |
| Preconditions | - User đã đăng nhập  - User có quyền tạo sản phẩm |
| Test Steps | 1. Navigate to Product page  2. Click “Add new product”  3. Enter product information  4. Click Save button |
| Test Data | Name: Laptop Dell  Price: 15000000  Quantity: 10  Category: Electronics |
| Expected result | - Product created successfully  - Success mesage displayed  - Product appears in list |
| Actual result | (Để trống) |
| Status | Not run |

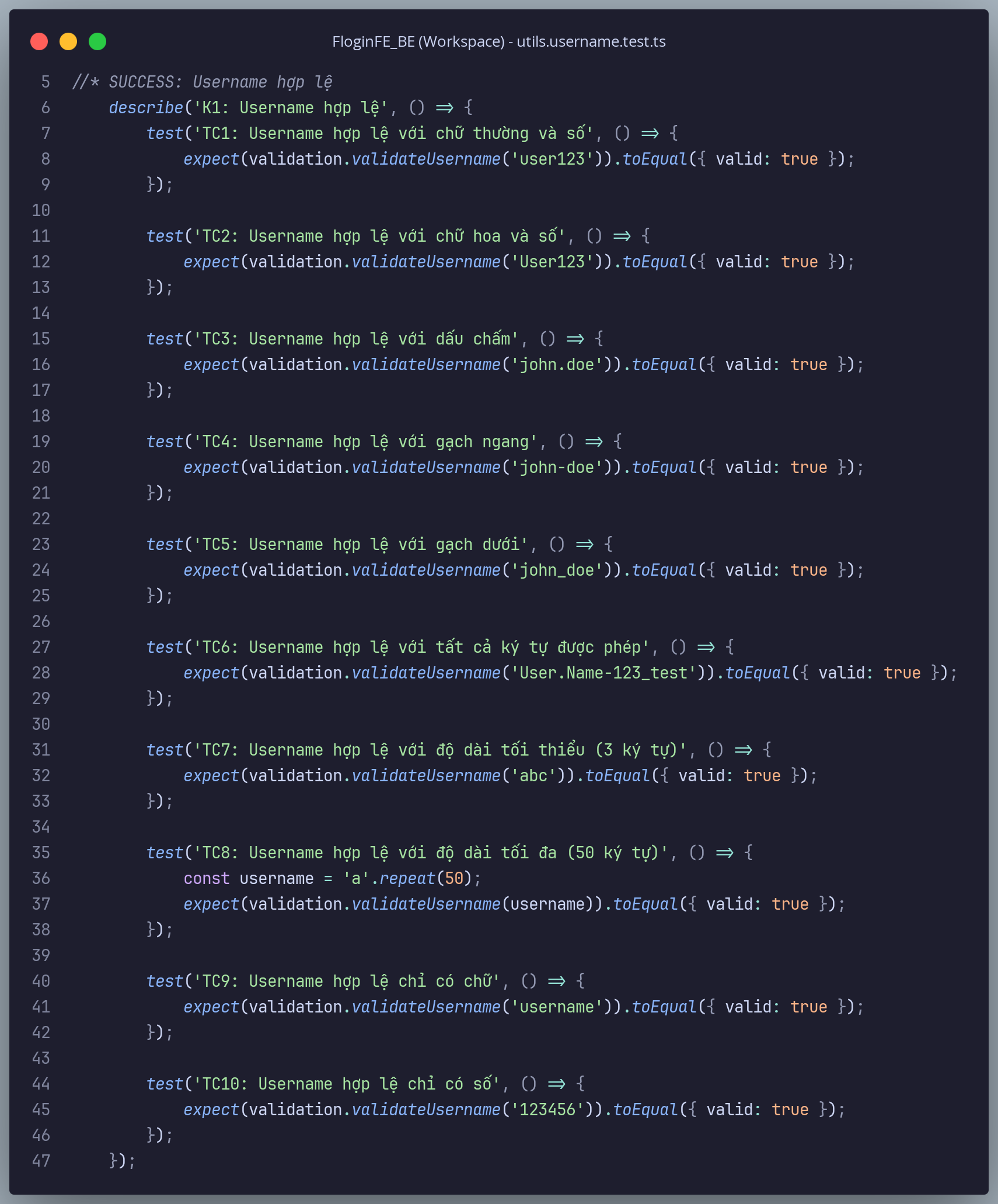
|  |  |
| --- | --- |
| Test case ID | TC \_PRODUCT \_002 |
| Test Name | Cập nhật sản phẩm mới thành công |
| Priority | Critical |
| Preconditions | - User đã đăng nhập  - User có quyền cập nhật sản phẩm |
| Test Steps | 1. Navigate to Product page  2. Click “Update product”  3. Enter new product information  4. Click Save button |
| Test Data | Name: Laptop Dell Xps  Price: 25000000  Quantity: 15  Category: Electronics |
| Expected result | - Product updated successfully  - Success mesage displayed  - New Product’s information appears in list |
| Actual result | (Để trống) |
| Status | Not run |

|  |  |
| --- | --- |
| Test case ID | TC \_PRODUCT \_003 |
| Test Name | Xóa phẩm mới thành công |
| Priority | Critical |
| Preconditions | - User đã đăng nhập  - User có quyền xóa sản phẩm |
| Test Steps | 1. Navigate to Product page  2. Click “Remove product” of Product wanna remove |
| Test Data | Name: Laptop Dell  Price: 15000000  Quantity: 10  Category: Electronics  IsDeleted: true |
| Expected result | - Product removed successfully  - Success mesage displayed  - Product disappears in list |
| Actual result | (Để trống) |
| Status | Not run |

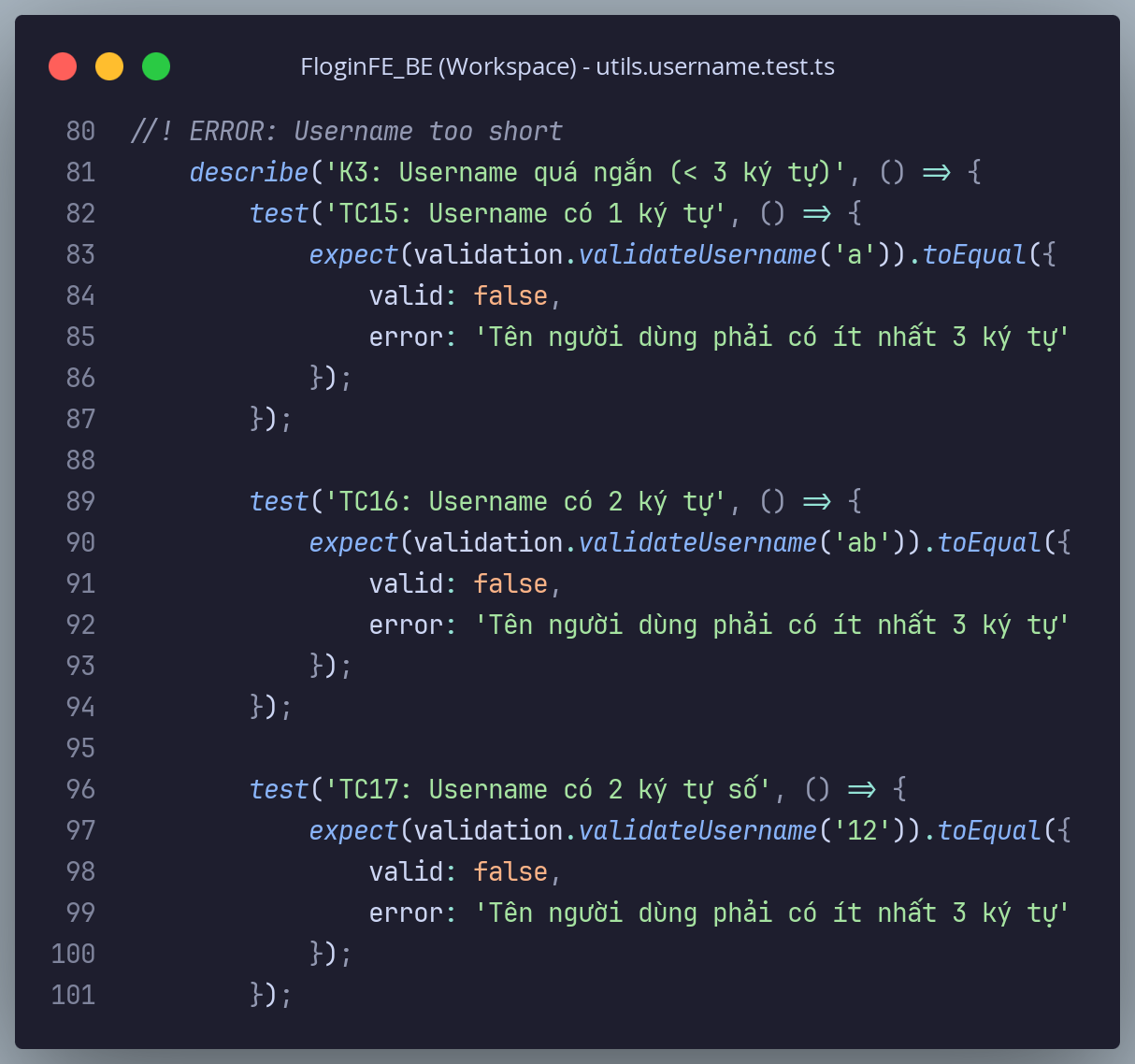
|  |  |
| --- | --- |
| Test case ID | TC \_PRODUCT \_004 |
| Test Name | Xem chi tiết sản phẩm thành công |
| Priority | Critical |
| Preconditions | - User đã đăng nhập  - User có quyền xem sản phẩm |
| Test Steps | 1. Navigate to Product page  2. Click to product |
| Test Data | Name: Laptop Dell  Price: 15000000  Quantity: 10  Category: Electronics |
| Expected result | - Detail Product popup in UI |
| Actual result | (Để trống) |
| Status | Not run |

|  |  |
| --- | --- |
| Test case ID | TC \_PRODUCT \_005 |
| Test Name | Thêm sản phẩm thất bại với tên rỗng |
| Priority | Critical |
| Preconditions | - User đã đăng nhập  - User có quyền thêm sản phẩm |
| Test Steps | 1. Navigate to Product page  2. Click “Add new product”  3. Enter product information  4. Click Save button |
| Test Data | Name:  Price: 15000000  Quantity: 10  Category: Electronics |
| Expected result | - Error when create product  - Error mesage displayed  - Product not appears in list |
| Actual result | (Để trống) |
| Status | Not run |

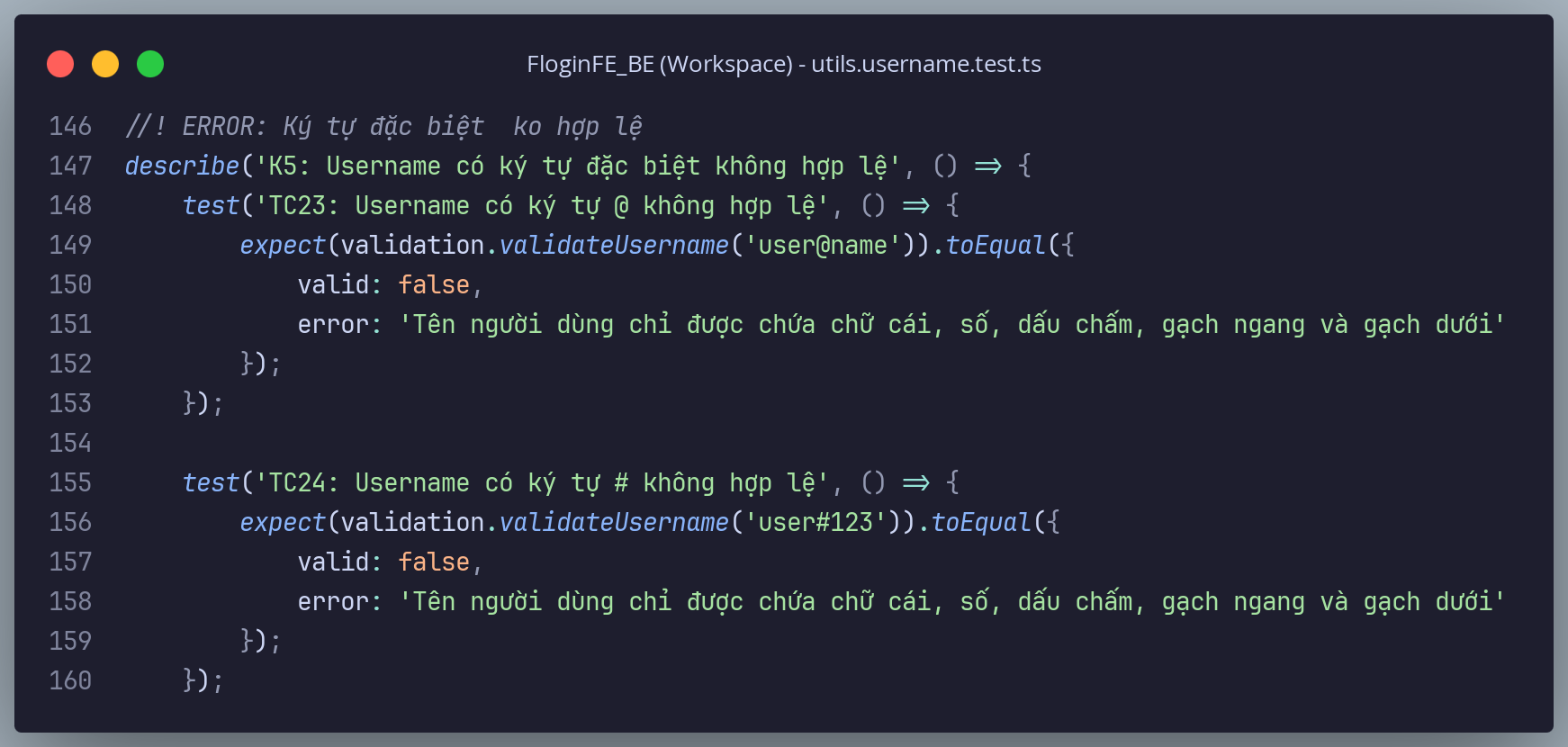
1. **Câu 2: Unit Testing và Test-Driven Development**
   1. **Login - Unit Test FE và BE**
      1. **FE Unit Tests - Validation Login**
2. Validate Username





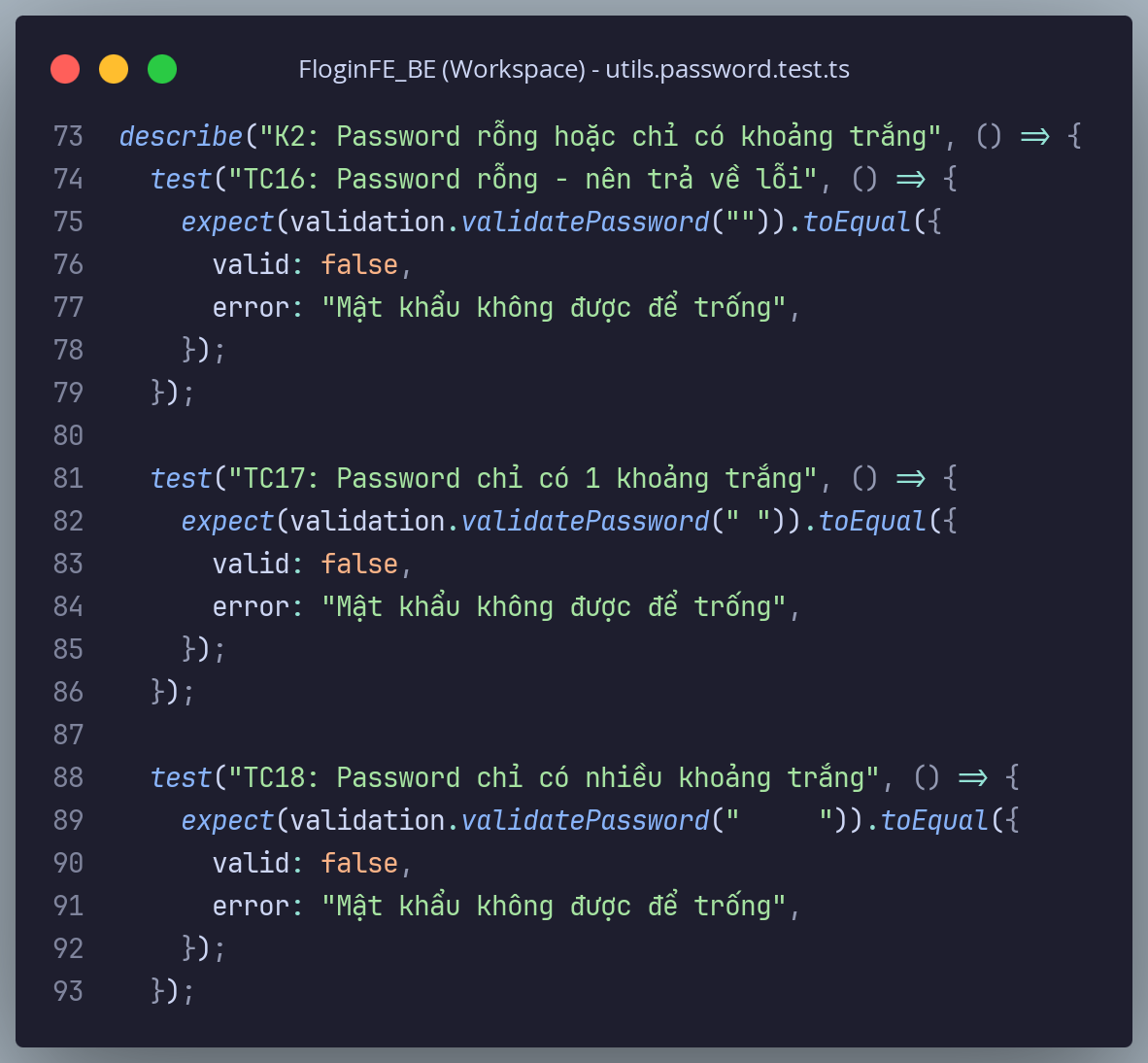






1. Validate Password

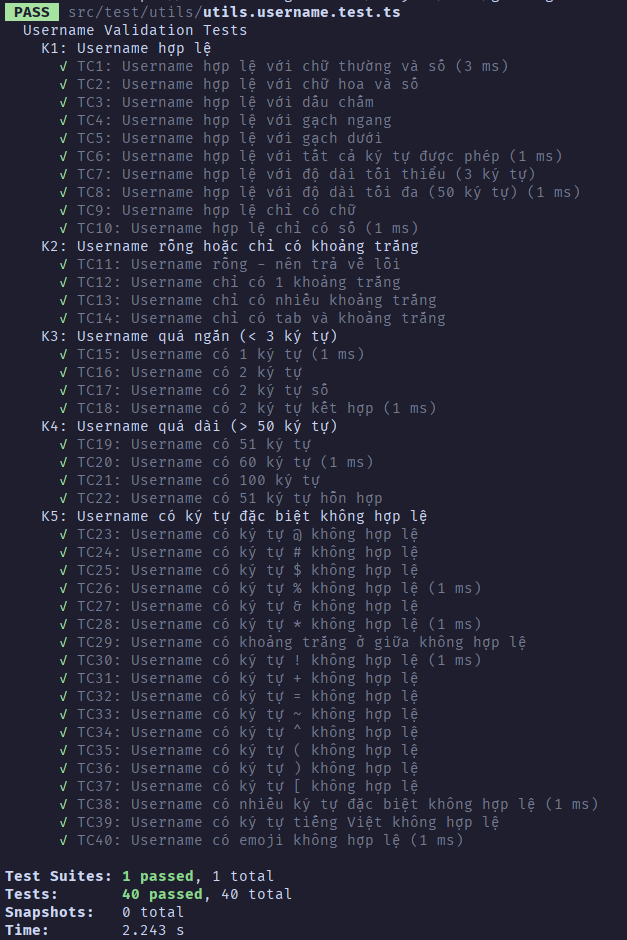


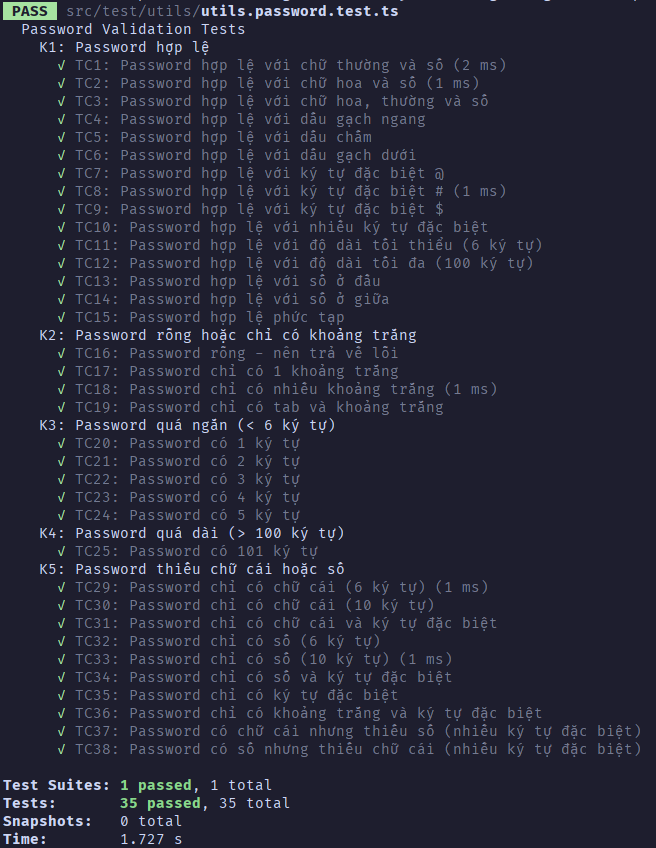






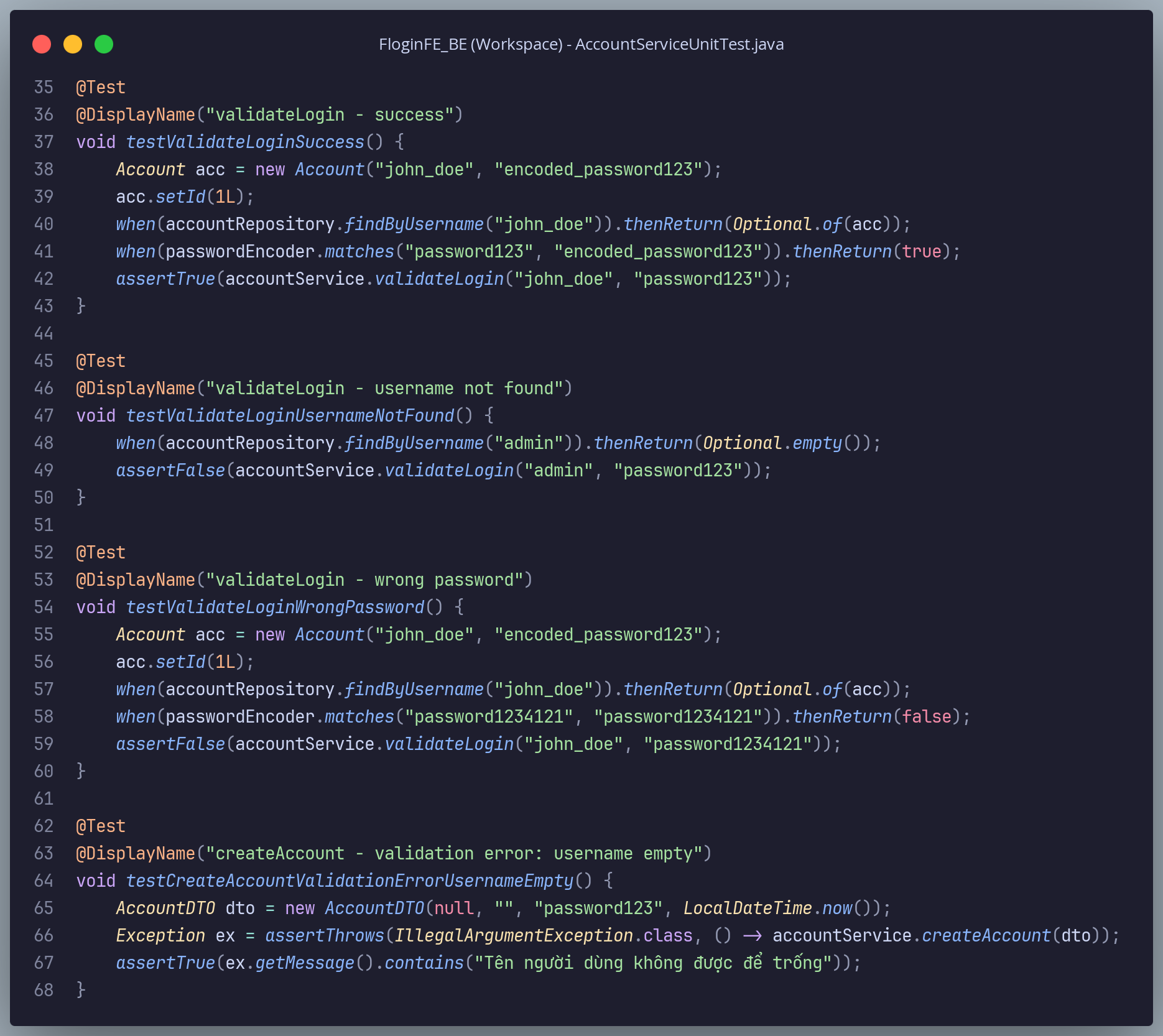
\* RESULT:





* + 1. **BE Unit Tests - Login Service**

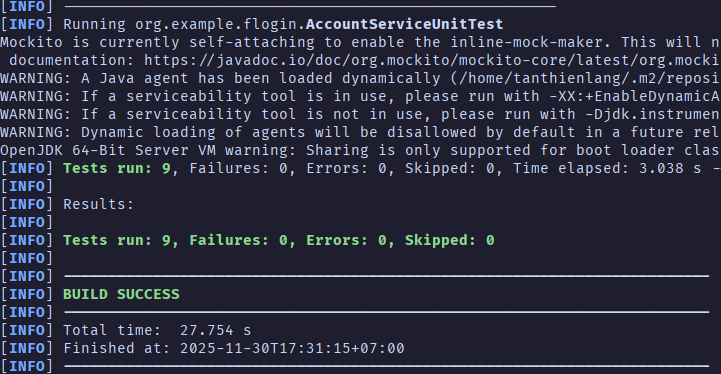
1. Test Authenticate với các scenarios
   * Login thành công
   * Login với username không tồn tại
   * Login với password sai
   * Validation errors



1. Test validation riêng lẻ



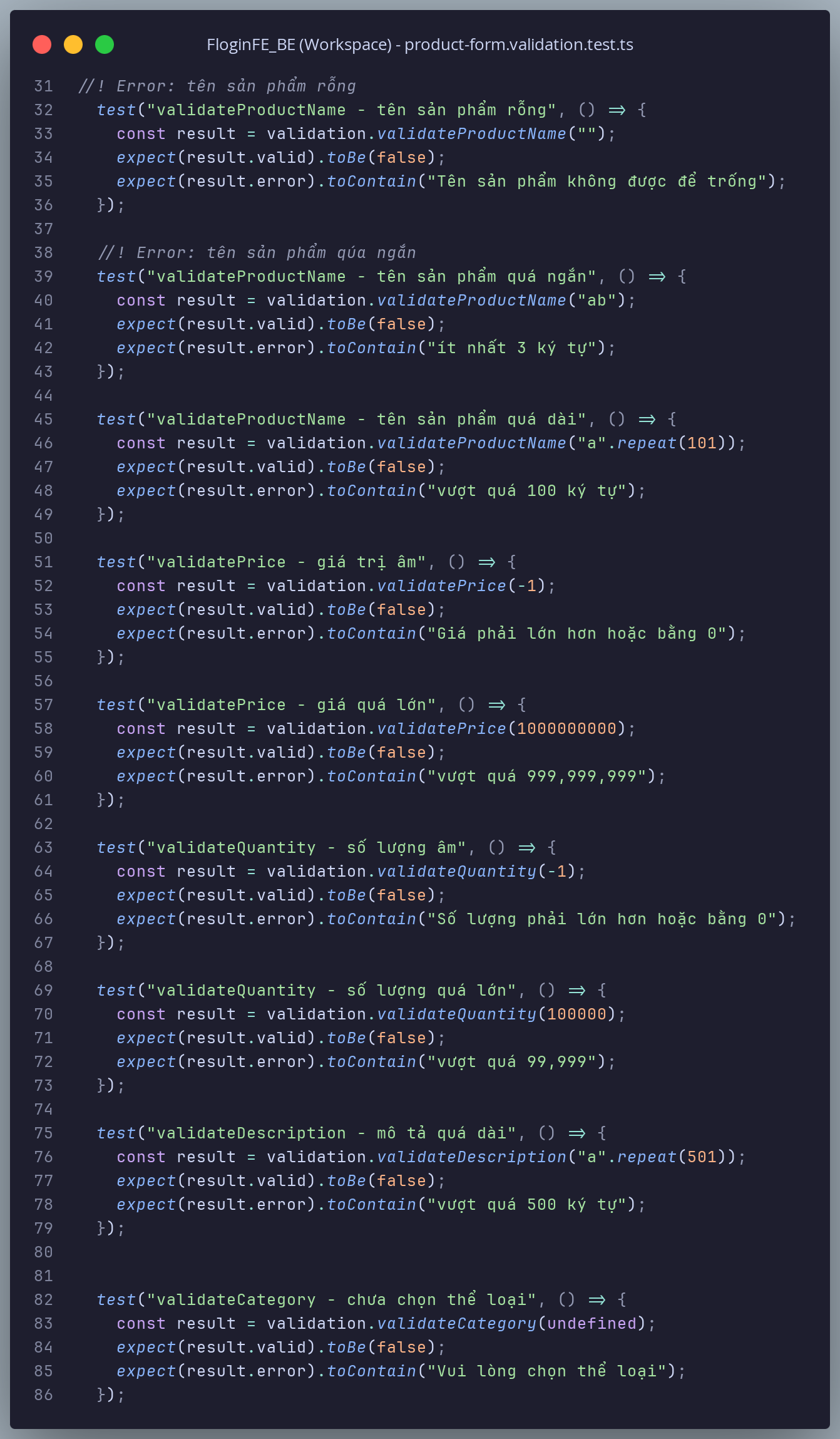
\* RESULT:



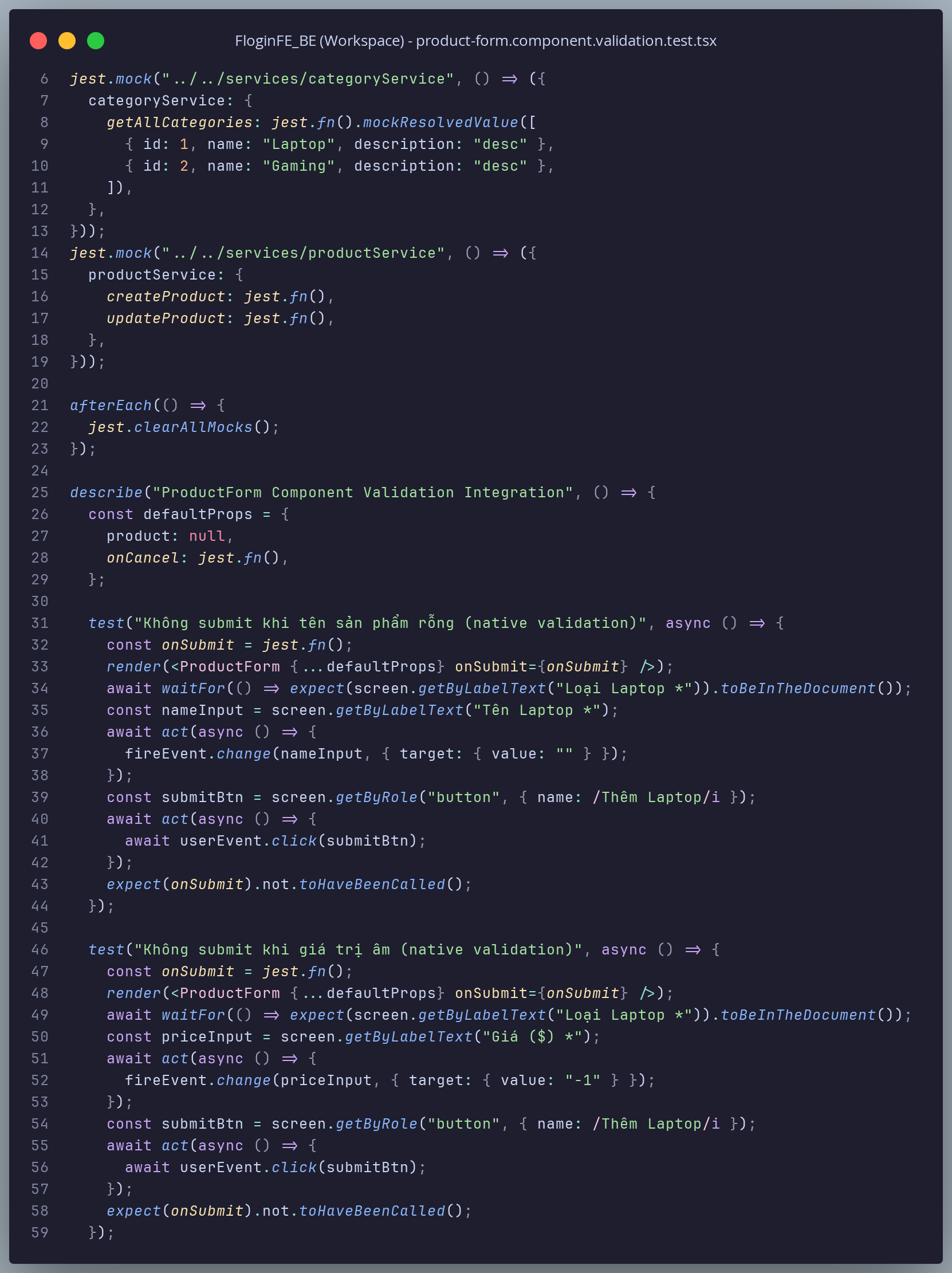
* 1. **Product - Unit Test FE và BE**
     1. **FE Unit Tests - Validation Product**

1. Validate Product
   * Test product name validation
   * Test price validation
   * Test quantity validation
   * Test description length
   * Test category validation





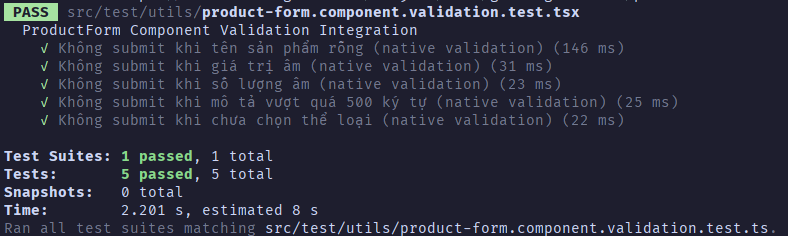
1. Product form component





\* RESULT:



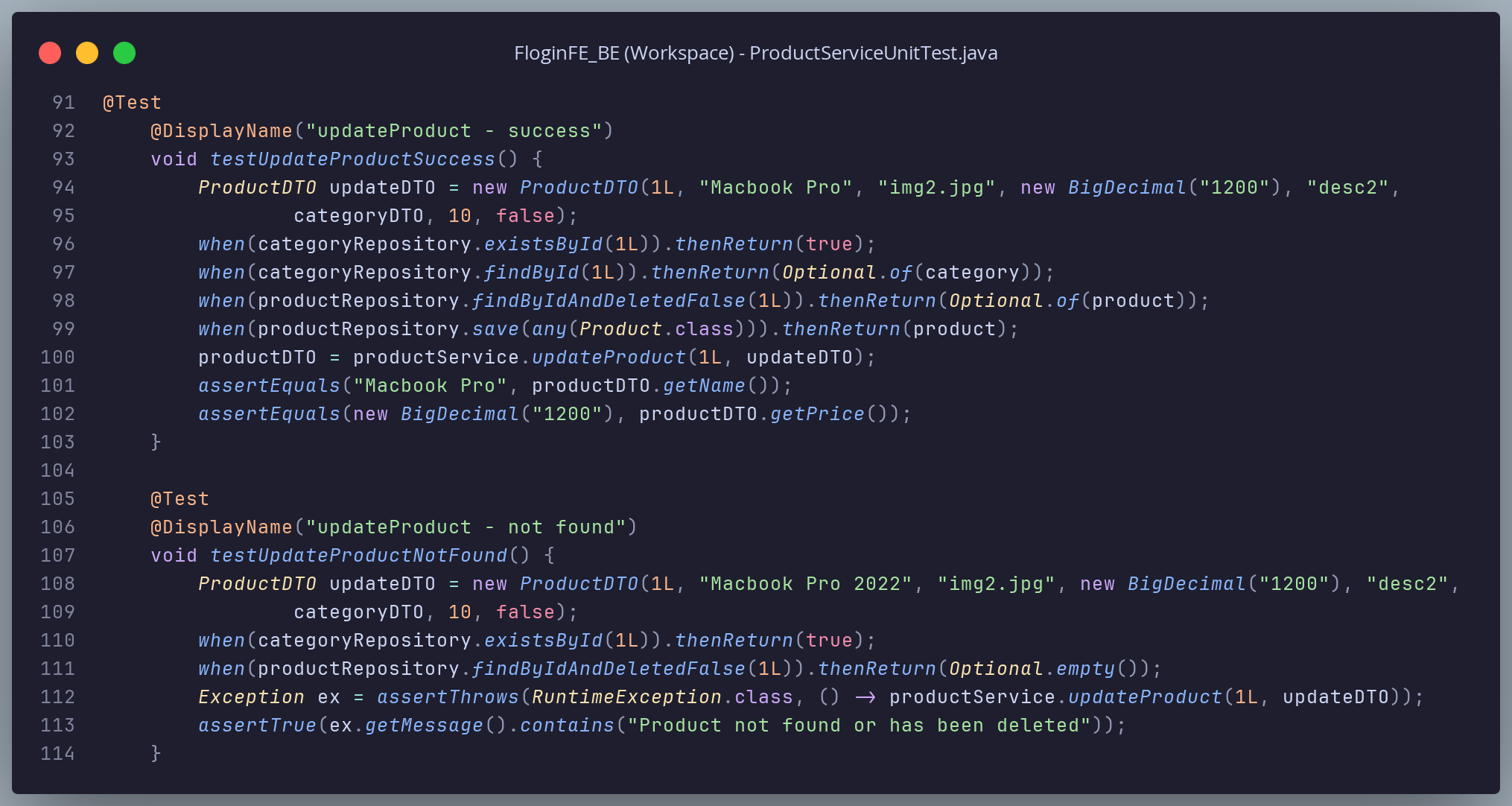


* + 1. **BE Unit Tests - Product Service**

1. Test CRUD operations
   * Test createProduct
   * Test getProduct
   * Test updateProduct
   * Test deleteProduct
   * Test getAll với pagination





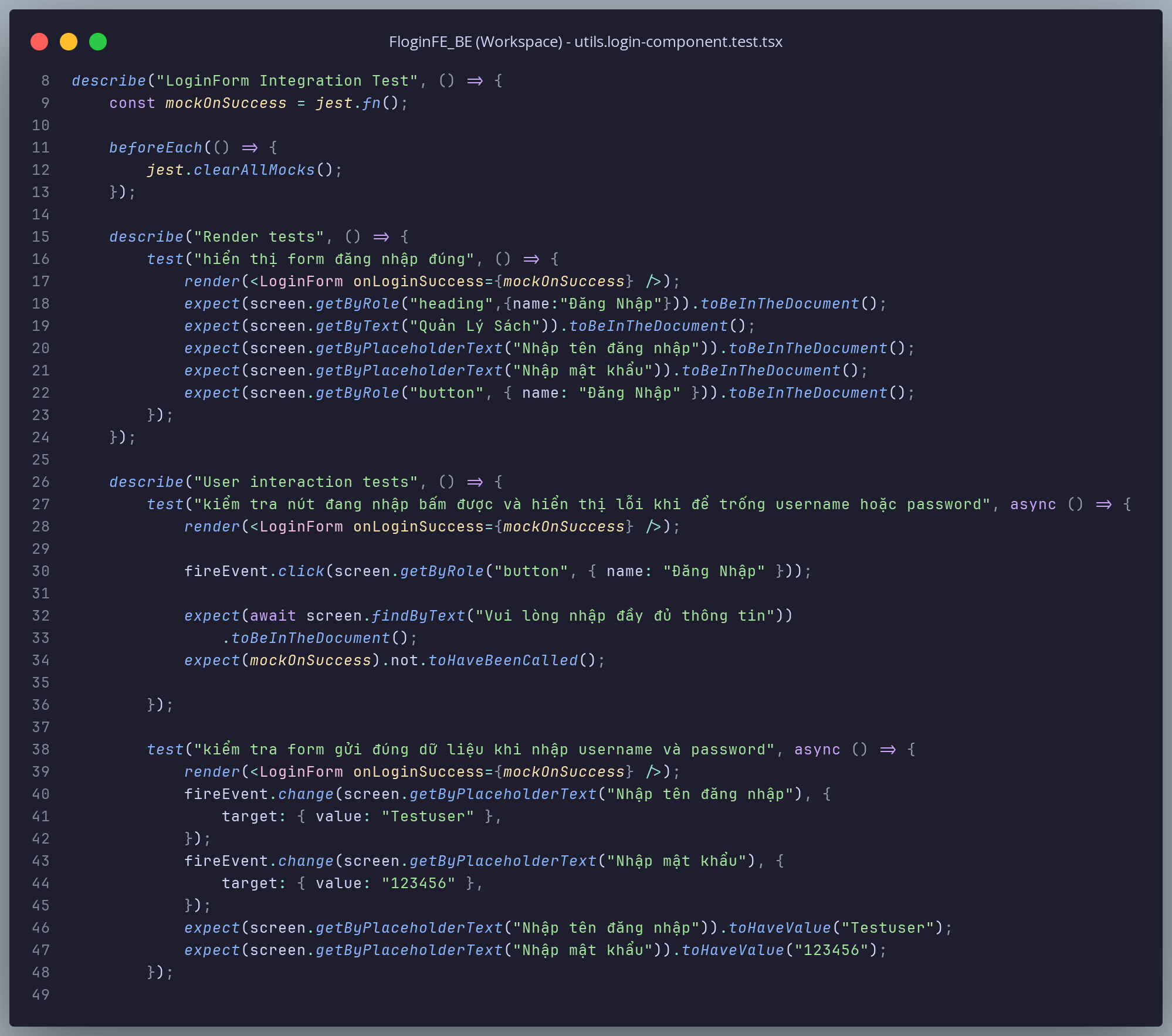
****

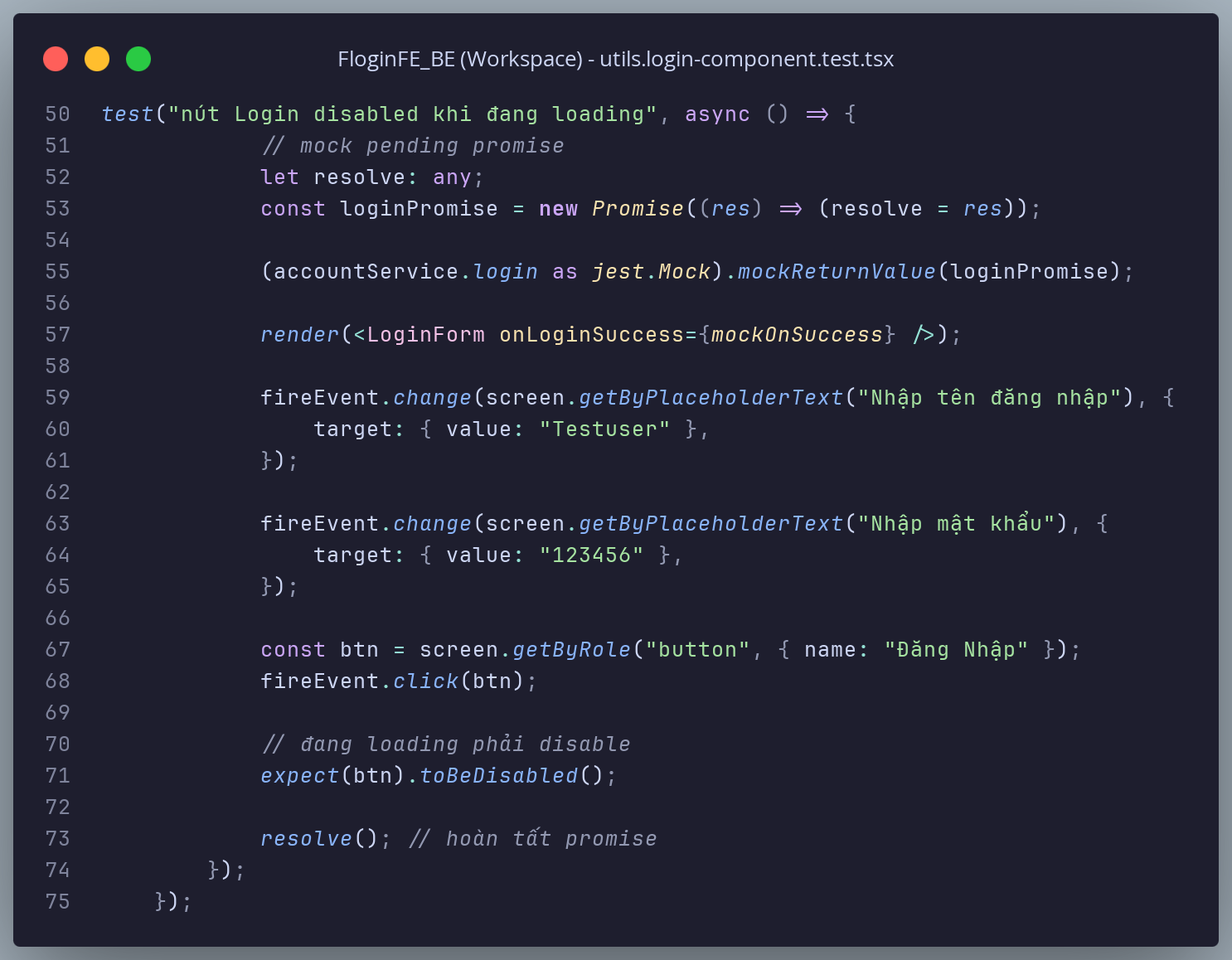


\* RESULT:



1. **Câu 3: Integration Testingimplementation**
   1. **Login - Integration Testing**
      1. **FE component Integration**
2. Test rendering và user interactions
3. Test form submission và API calls
4. Test error handling và success messages

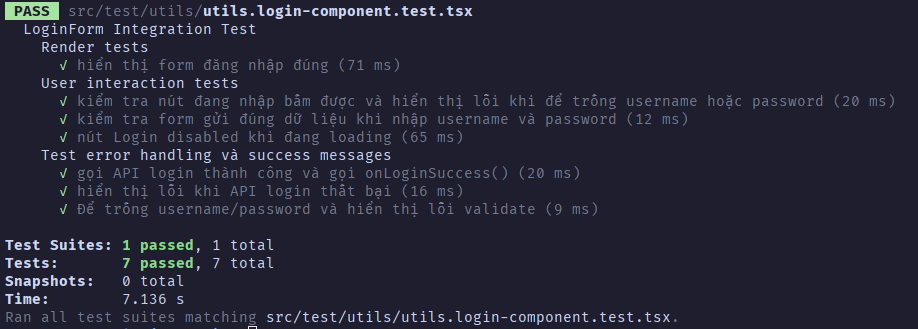








\* RESULT:



* + 1. **Backend API Integration**

1. Test POST /api/accounts/login endpoint



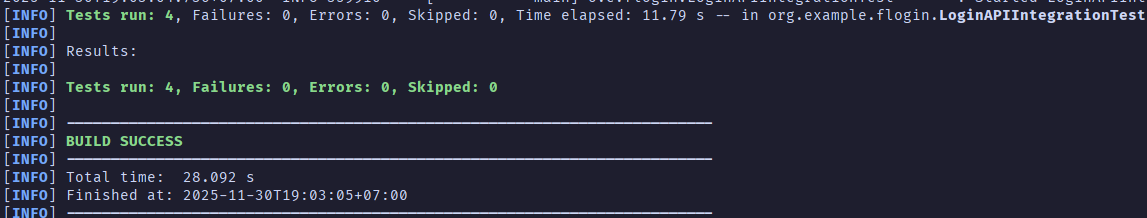
1. Test response structure và status codes



1. Test CORS và headers



\* RESULT:



* 1. **Product - Integration Testing**
     1. **FE component Integration**

1. Test ProductList component với API
2. Test ProductFrom component
3. Test ProductDetail component

\* RESULT:

* + 1. **BE API Integration**

1. Test POST /api/products
2. Test GET /api/products
3. Test GET /api/products/{id}
4. Test PUT /api/products/{id}
5. Test DELETE /api/products/{id}

\* RESULT:

1. **Câu 4: Mock Testing**
   1. **Login - Mock Testing**
      1. **FE mocking**
2. Mock accountService.Login
3. Test với mocked successful / failed responses
4. Verify mock calls

\* RESULT:

* + 1. **BE mocking**

1. Mock accountService với @MockBean
2. Test controller với mocked service
3. Verify mock Interactions

\* RESULT:

* 1. **Product - Mock Testing**
     1. **FE mocking**

1. Mock CRUD operations
2. Test success và failure scenarios
3. verify all mock calls

\* RESULT:

* + 1. **BE mocking**

1. Mock Product Repository
2. Test service layer với mocked repository
3. Verify repository interactions

\* RESULT:

1. **Câu 5: Automation Testing & CI/CD**
   1. **Login - E2E Automation Testing**
      1. **Setup & config**
      2. **E2E Test Scenarios cho Login**
      3. **CI/CD Integration cho Login Tests**
   2. **Product - E2E Automation Testing**
      1. **Setup page object model**
      2. **E2E Test Scenarios cho Product**
      3. **CI/CD Integration**
2. **Phần mở rộng**
   1. **Performance Testing**
      1. **Yêu cầu**
3. Setup JMeter || K6 cho performance testing
4. Viết performance tests cho Login API
5. Viết performance tests cho Product API
6. Phân tích kết quả và đưa ra recommendations
   1. **Security Testing**
      1. **yêu cầu**
7. Test common vulnerabilities
8. Test input validation & sanitization
9. Security best practices implementation