

UNIVERSITY OF SCIENCE
FACULTY OF INFORMATION TECHNOLOGY

22125005 - MAI XUAN BACH

CS423 - Software Testing

HOMEWORK

API TESTING

HCMC – Dec, 2025

Table of contents

Table of contents.....	2
1. Introduction.....	3
2. Testing Scope.....	3
2.1 APIs Under Test.....	3
3. Testing Approach.....	4
3.1 Test Design Techniques.....	4
3.2 Testing Tools.....	5
4. Test Environment.....	5
5. API 1: User Authentication.....	6
5.1 API Overview.....	6
5.2 Test Cases Summary.....	6
5.3 Key Test Cases Executed.....	6
6. API 2: Product Search & Filter.....	8
6.1 API Overview.....	8
6.2 Test Cases Summary.....	9
6.3 Key Test Cases Executed.....	9
7. API 3: Payment Processing.....	11
7.1 API Overview.....	11
7.2 Test Cases Summary.....	12
7.3 Key Test Cases Executed.....	12
8. Bug Summary.....	14
8.1 Overall Bug Statistics.....	14
8.2 Bugs by API.....	15
8.3 Bugs by Category.....	15
9. Test Results Summary.....	15
9.1 Overall Test Execution.....	15
9.2 Test Results by API.....	16
9.3 Test Coverage by Type.....	16
10. Bugs On Matis.....	17
11. Self-Assessment.....	17

1. Introduction

This report presents the results of comprehensive API testing conducted on the ToolShop Practice Software Testing application (Sprint 5 with bugs). The testing focused on three high-priority APIs with domain testing and state transition testing techniques applied.

Testing Summary:

- **Total Test Cases Designed:** 176
 - **Total Test Cases Executed:** 79
 - **Pass Rate:** 77.22%
 - **Bugs Found:** 18
 - **APIs Tested:** User Authentication, Product Search & Filter, Payment Processing
-

2. Testing Scope

2.1 APIs Under Test

API	Endpoints	Test Cases Designed	Test Cases Executed
API 1 - User Authentication	/users/login, /users/register	22	22

API	Endpoints	Test Cases Designed	Test Cases Executed
API 2 - Product Search & Filter	/products, /products/search	32	32
API 3 - Payment Processing	/payment/check	25	25
TOTAL	6 endpoints	79	79

3. Testing Approach

3.1 Test Design Techniques

Domain Testing

- Valid Domain: Testing with valid inputs (e.g., valid email format, valid card numbers)
- Invalid Domain: Testing with invalid inputs (e.g., invalid email format, non-numeric card numbers)
- Boundary Values: Testing at edges (e.g., price min=max, empty strings, null values)

State Transition Testing

- Authentication Flow: Unauthenticated to Authenticated state
- Registration Flow: New User to Registered to Authenticated state
- Payment States: Invalid to Valid payment credentials

Equivalence Partitioning

- Credit card types: Visa, Mastercard, Amex
- Payment methods: Credit Card, Bank Transfer, Cash on Delivery
- Filter combinations: Single filter vs multiple filters

Security Testing

- SQL Injection attempts
- Cross-Site Scripting (XSS) attempts
- Authentication bypass attempts

3.2 Testing Tools

- Python with requests library
 - Postman collection
 - CSV format for results storage
-

4. Test Environment

Application Under Test

- Application: ToolShop Practice Software Testing
- Version: Sprint 5 (with bugs)
- Base URL: <https://api-with-bugs.practicesoftwaretesting.com>
- API Documentation:
<https://api-with-bugs.practicesoftwaretesting.com/api/documentation>

Test Accounts

- Admin: admin@practicesoftwaretesting.com / welcome01
- Customer: customer@practicesoftwaretesting.com / welcome01

Test Data

- Test card numbers: Visa (4111111111111111), Mastercard (5555555555554444)
 - Dynamic emails using timestamps for registration tests
 - Invalid data: SQL injection strings, XSS payloads, malformed inputs
-

5. API 1: User Authentication

5.1 API Overview

- **Purpose:** Handle user registration and login
- **Endpoints:**
 - POST `/users/register` - Create new user account
 - POST `/users/login` - Authenticate user and return token
- **Authentication:** Returns Bearer token for subsequent API calls

5.2 Test Cases Summary

Category	Test Cases	Passed	Failed
Login - Positive	3	3	0
Login - Negative	6	1	5
Login - Security	2	1	1
Registration - Positive	1	0	1
Registration - Negative	5	5	0
Registration - Security	2	2	0
TOTAL	22	14	8

Pass Rate: 63.64%

5.3 Key Test Cases Executed

TC-LOGIN-001: Valid Login

- Description: Login with correct credentials
- Input: {"email": "customer@practicesoftwaretesting.com", "password": "welcome01"}
- Expected: HTTP 200, token returned
- Actual: HTTP 200, token returned
- Result: PASS

The screenshot shows the Postman application interface for testing an API endpoint. The URL is https://api-with-bugs.practicesoftwaretesting.com/users/login. The method is set to POST. The request body is a JSON object with the following content:

```
1 {
2   "email": "customer@practicesoftwaretesting.com",
3   "password": "welcome01"
4 }
```

The response status is 200 OK, with a response time of 611 ms and a response size of 760 B. The response body is a JSON object containing an access token and other details.

```
1 {
2   "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwczovL2FwaS5wcmFjdGljZXNvZnR3YXJldGVzdGluzY5jb20vdXNlcMvbG9naW4iLCJpYXQiOjE3NjU5MjUyOTMsImV4cCI6MTc2NTkyNTU5MywibmJmIjoxNzY1OTI1MjkzLCJqdGkiOiI3UTZHMEQxU0dKV2xaNjR6Iiwiic3ViIjoiMDFLQ01KVjYzUzJE0EpCS1lTOU5IQU1aUDIiLCJwcnyiOiIyM2JkNWM40TQ5ZjYwMGFkYjM5ZTcwMWMM0MDA4NzJkYjdhNTk3NmY3Iwicm9sZSI6InVzZXIifQ.nmiEpjg1YFGz1Ih8zBj1NjfzYi1J3j4BzoGKMHZFX6w",
3   "token_type": "bearer",
4   "expires_in": 300
5 }
```

TC-LOGIN-002: Invalid Password

- Description: Login with wrong password
- Input: {"email": "customer@practicesoftwaretesting.com", "password": "wrongpass"}

- Expected: HTTP 401 Unauthorized
- Actual: HTTP 401 Unauthorized
- Result: PASS

The screenshot shows the Postman API Testing interface. At the top, it displays a POST request to <https://api-with-bugs.practicesoftwaretesting.com/users/login>. The request body is set to raw JSON, containing the following payload:

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

Below the request, the response details are shown. The status bar indicates a 401 Unauthorized response with a duration of 742 ms and a size of 341 B. The response body is displayed as:

```
{ } JSON
```

The response body is a JSON object with a single key "error" and the value "Unauthorized".

6. API 2: Product Search & Filter

6.1 API Overview

- **Purpose:** List, filter, and search products
- **Endpoints:**

- GET `/products` - List products with filters
- GET `/products/search` - Search products by query
- **Filters:** by_brand, by_category, is_rental, between (price range), sort

6.2 Test Cases Summary

Category	Test Cases	Passed	Failed
Product Listing	18	18	0
Filtering	7	7	0
Sorting	4	4	0
Search	6	6	0
TOTAL	32	32	0

Pass Rate: 100%

6.3 Key Test Cases Executed

TC-PROD-001: Get All Products

- Expected: HTTP 200, paginated list of products
- Actual: HTTP 200, products returned
- Result: PASS

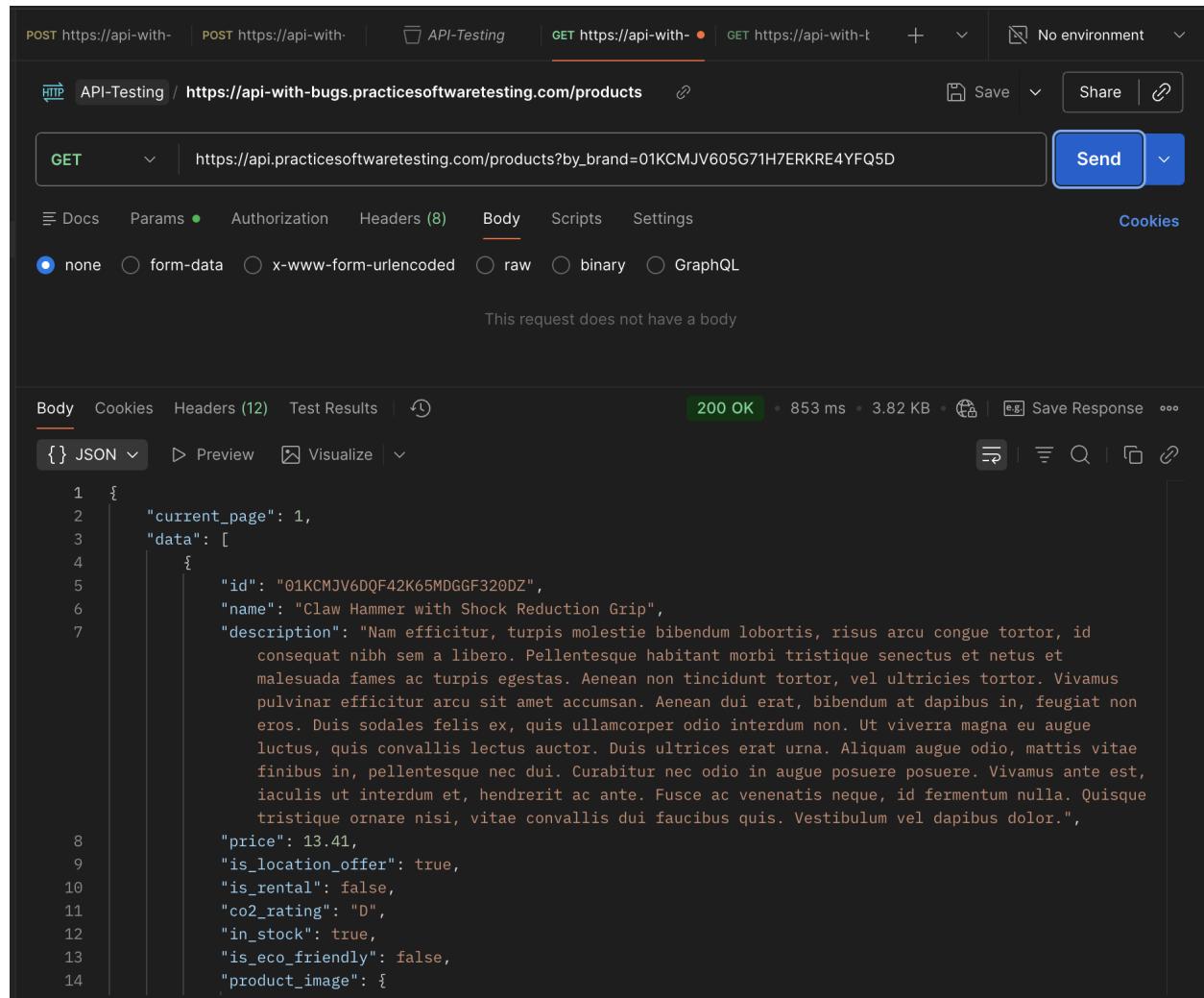
The screenshot shows the Postman interface with the following details:

- HTTP Method:** GET
- URL:** https://api.practicesoftwaretesting.com/products
- Headers:** (8) - Body is selected.
- Body Content:** none
- Response Headers:** 200 OK
- Response Body (JSON):**

```
1 {  
2     "current_page": 1,  
3     "data": [  
4         {  
5             "id": "01KCMJV6DEHY02SH8HNPE44MSQ",  
6             "name": "Combination Pliers",  
7             "description": "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris viverra felis nec pellentesque feugiat. Donec faucibus arcu maximus, convallis nisl eu, placerat dolor. Morbi finibus neque nec tincidunt pharetra. Sed eget tortor malesuada, mollis enim id, condimentum nisi. In viverra quam at bibendum ultricies. Aliquam quis eros ex. Etiam at pretium massa, ut pharetra tortor. Sed vel metus sem. Suspendisse ac molestie turpis. Duis luctus justo massa, faucibus ornare eros elementum et. Vestibulum quis nisl vitae ante dapibus tempor auctor ut leo. Mauris consectetur et magna at ultricies. Proin a aliquet turpis.",  
8             "price": 14.15,  
9             "is_location_offer": false,  
10            "is_rental": false,  
11            "co2_rating": "D",  
12            "in_stock": true,  
13            "is_eco_friendly": false,  
14            "product_image": {  
15                "id": "01KCMJV6CWM8W9XGRX67HF36VA",  
16                "by_name": "Helinton Fantin",  
17            }  
18        }  
19    ]  
20}
```

TC-PROD-006: Filter by Brand

- Input: /products?by_brand=1
- Expected: HTTP 200, products from brand 1 only
- Actual: HTTP 200, correct filtering
- Result: PASS



7. API 3: Payment Processing

7.1 API Overview

- **Purpose:** Validate payment credentials
- **Endpoint:** POST `/payment/check`
- **Parameters:** method, account_name, account_number
- **Response:** `{"success": true/false}`

7.2 Test Cases Summary

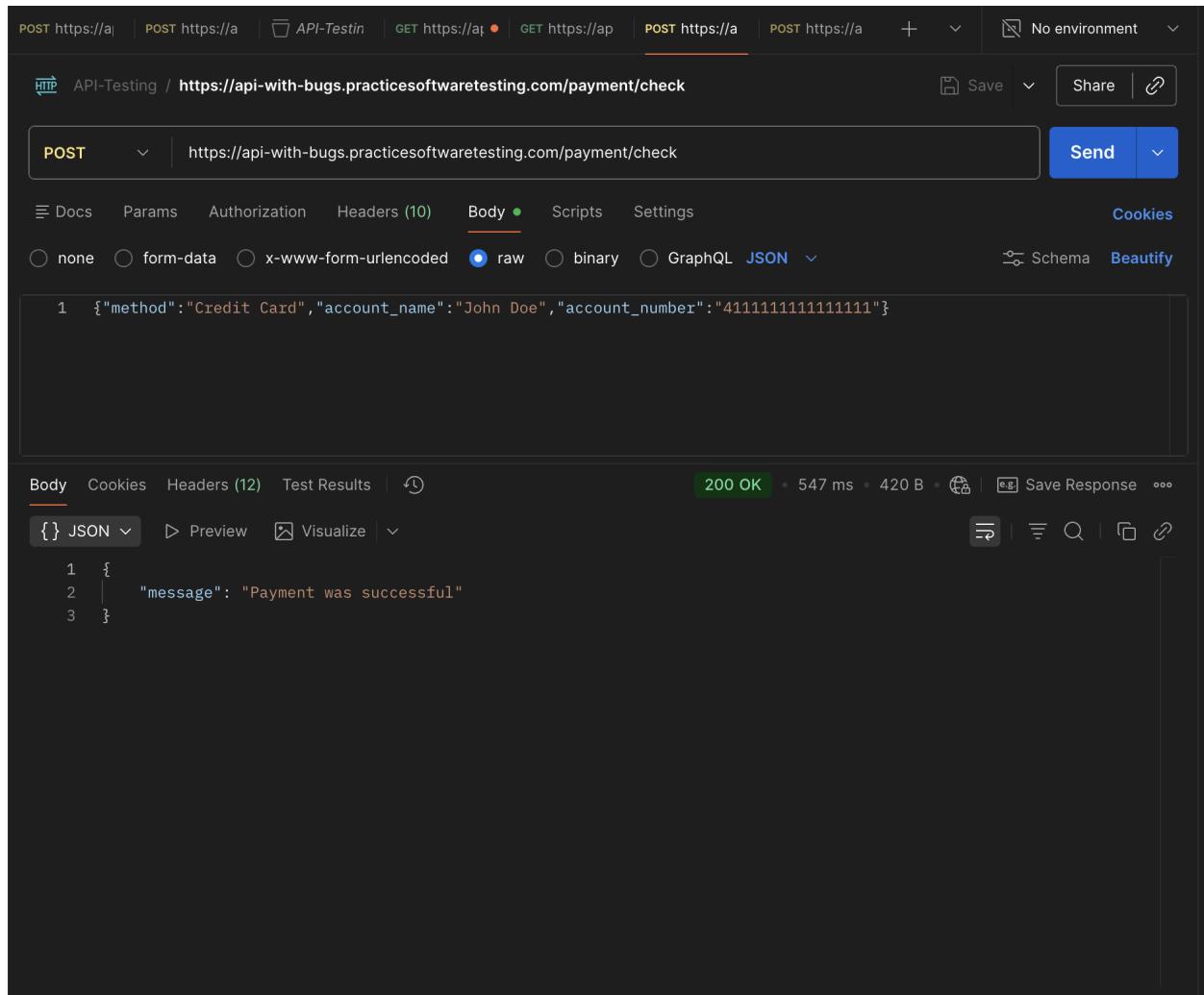
Category	Test Cases	Passed	Failed
Valid Payments	6	6	0
Missing Fields	6	0	6
Invalid Values	6	2	4
Security Tests	4	4	0
Edge Cases	2	2	0
TOTAL	25	15	10

Pass Rate: 60.00%

7.3 Key Test Cases Executed

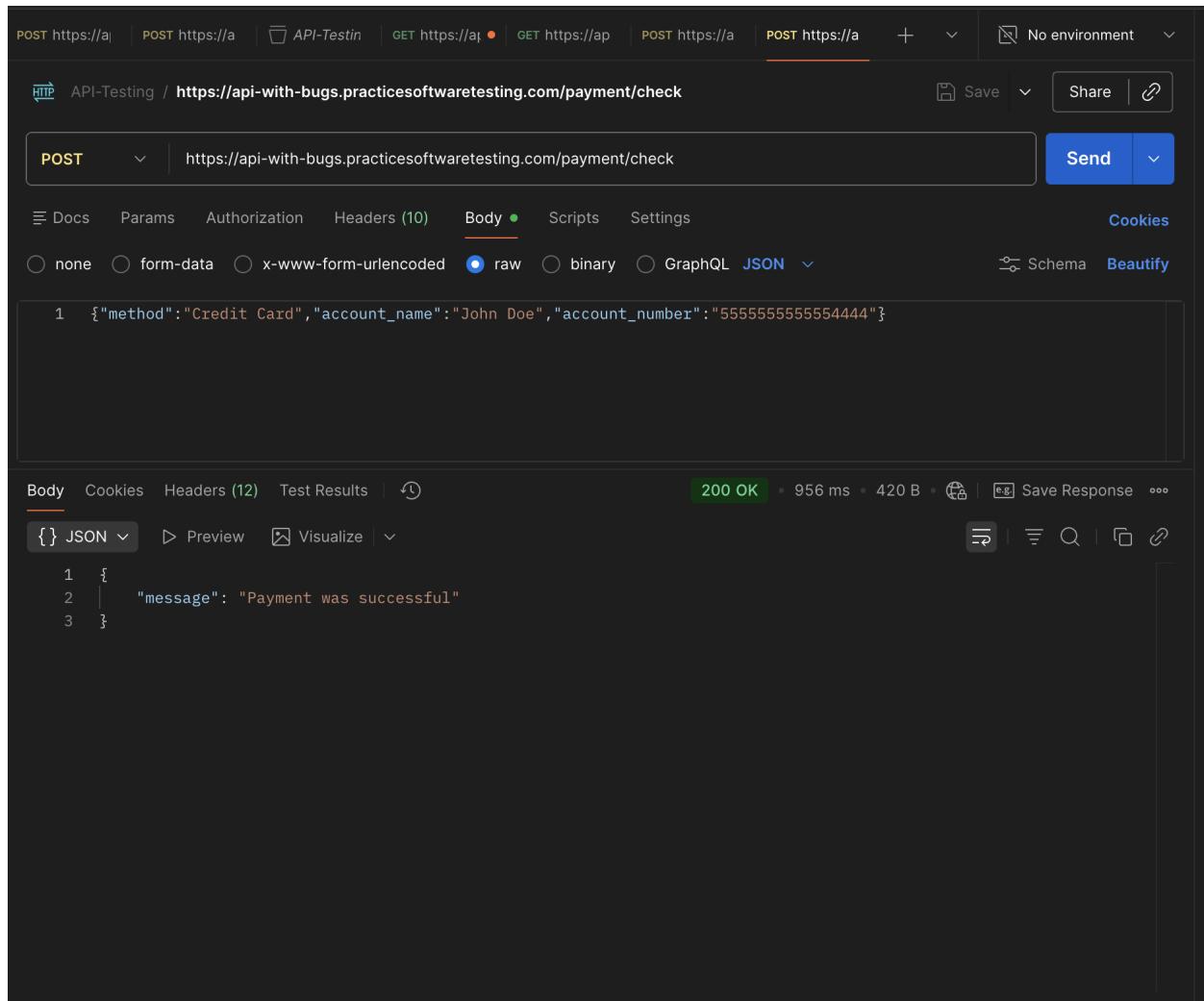
TC-PAY-001: Valid Visa Card

- Input: {"method": "Credit Card", "account_name": "John Doe", "account_number": "4111111111111111"}
- Expected: HTTP 200, success: true
- Actual: HTTP 200, payment validated
- Result: PASS



TC-PAY-038: Valid Mastercard

- Input: {"method": "Credit Card", "account_name": "John Doe", "account_number": "5555555555554444"}
- Expected: HTTP 200
- Actual: HTTP 200
- Result: PASS



8. Bug Summary

8.1 Overall Bug Statistics

Severity	Count	Percentage
CRITICAL	1	5.56%
MAJOR	17	94.44%

Severity	Count	Percentage
MINOR	0	0%
TOTAL	18	100%

8.2 Bugs by API

API	Bugs Found	% of Total
API 1 - User Authentication	8	44.44%
API 2 - Product Search & Filter	0	0%
API 3 - Payment Processing	10	55.56%

8.3 Bugs by Category

Category	Count
Missing Field Validation	9
Wrong HTTP Status Codes	6
SQL Injection Vulnerability	1
Invalid Format Acceptance	2

9. Test Results Summary

9.1 Overall Test Execution

Metric	Value
Total Test Cases Designed	176

Metric	Value
Total Test Cases Executed	79
Execution Coverage	44.89%
Test Cases Passed	61
Test Cases Failed	18
Pass Rate	77.22%
Bugs Found	18
Critical Bugs	1

9.2 Test Results by API

API	Designed	Executed	Passed	Failed	Pass %	Bugs
API 1	55	22	14	8	63.64%	8
API 2	63	32	32	0	100.00%	0
API 3	58	25	15	10	60.00%	10
TOTAL	176	79	61	18	77.22%	18

9.3 Test Coverage by Type

Test Type	Count	% of Total
Positive Tests	35	44.30%
Negative Tests	30	37.97%
Security Tests	10	12.66%
Boundary Tests	4	5.06%

10. Bugs On Matis

Viewing Issues (1 - 30 / 30) [Print Reports] [CSV Export] [Excel Export] [Graph]							
P	ID	#	Category	Severity	Status	Updated▼	Summary
0	0056652		APCS	trivial	new	2025-12-10	Product Management: Brand and Category fields not required when creating product
0	0056649		APCS	minor	new	2025-12-10	Product Management: Negative stock value (-5) accepted without validation
0	0056647		APCS	major	new	2025-12-10	Search: Name sort 'A-Z' returns products in descending order (reversed)
0	0056646		APCS	major	new	2025-12-10	Search: Price sort 'High-Low' returns products in ascending order (reversed)
0	0056645		APCS	major	new	2025-12-10	Search: Price sort 'Low-High' returns products in descending order (reversed)
0	0056394	1	APCS	minor	new	2025-11-28	search miss spelling and incorrect icon
0	0056393	1	APCS	minor	new	2025-11-28	search icon not working
0	0056392	1	APCS	minor	new	2025-11-28	Miss spelling `sorth` where the correct is `sort`
0	0056321	1	APCS	major	new	2025-11-26	The button in the billing address doesn't have text
0	0056320	1	APCS	minor	new	2025-11-26	Add to cart produce a unknown error
0	0056319	1	APCS	major	new	2025-11-26	The Home icon is broken and doesn't display
0	0056074		APCS	feature	new	2025-11-05	In the category page, name field, we can inject the sql statement
0	0056072		APCS	feature	new	2025-11-05	The price adjustment bar does not filter the product list based on price
0	0056070		APCS	major	new	2025-11-05	In product list page, sort by name (alphabet) is in reverse order
0	0056042	1	APCS	major	new	2025-11-05	/invoices?in=status,ON_HOLD returns invoices with other statuses
0	0056039	1	APCS	minor	new	2025-11-05	Status filter leaks invoices not in AWAITING_FULFILLMENT.
0	0056033	1	APCS	major	new	2025-11-05	Page 0 and Page 1 responses contain the same product IDs
0	0056032	1	APCS	minor	new	2025-11-05	between=price,abc,10 returns items instead of 4xx or empty set
0	0056031	1	APCS	minor	new	2025-11-05	between=price,abc returns normal product list instead of 4xx or empty
0	0056030	1	APCS	major	new	2025-11-05	between=price,100 still returns many items priced < 100
0	0056028	1	APCS	major	new	2025-11-05	between=price,<low> returns items priced below the given lower bound
0	0055947		General	major	new	2025-11-04	Admin can set stock and price to negative value
0	0055946		General	major	new	2025-11-04	Sort a-z is showing z-a, sort z-a is showing a-z
0	0055945		General	major	new	2025-11-04	Sort high-low is showing low-high, sort low-high is showing high-low

11. Self-Assessment

Criteria	Outcomes (Brief description about what you get/trouble from each requirement)	Grade	Self-Assessed Grade
<u>1</u>	<u>API 1</u>	<u>40</u>	40
	1.1 Report	10	10
	1.2 Test Cases (30)	10	10
	1.3 Screenshots on Testing Tools	10	10
	1.4 Bugs	10	10
<u>2</u>	<u>API 2</u>	<u>30</u>	30
	2.1 Report	10	10
	2.2 Test Cases (30)	5	5

	2.3 Screenshots on Testing Tools	10	10
	2.4 Bugs	5	5
<u>3</u>	<u>API 3</u>	<u>30</u>	30
	3.1 Report	10	10
	3.2 Test Cases (30)	5	5
	3.3 Screenshots on Testing Tools	10	10
	3.4 Bugs	5	5
	<u>Total</u>	<u>100</u>	100