NeuroNation API QA Challenge

Author: - Sonal Kajrolkar

Role: - QA Engineer

Table of Contents

❖ **Bug Report** 2

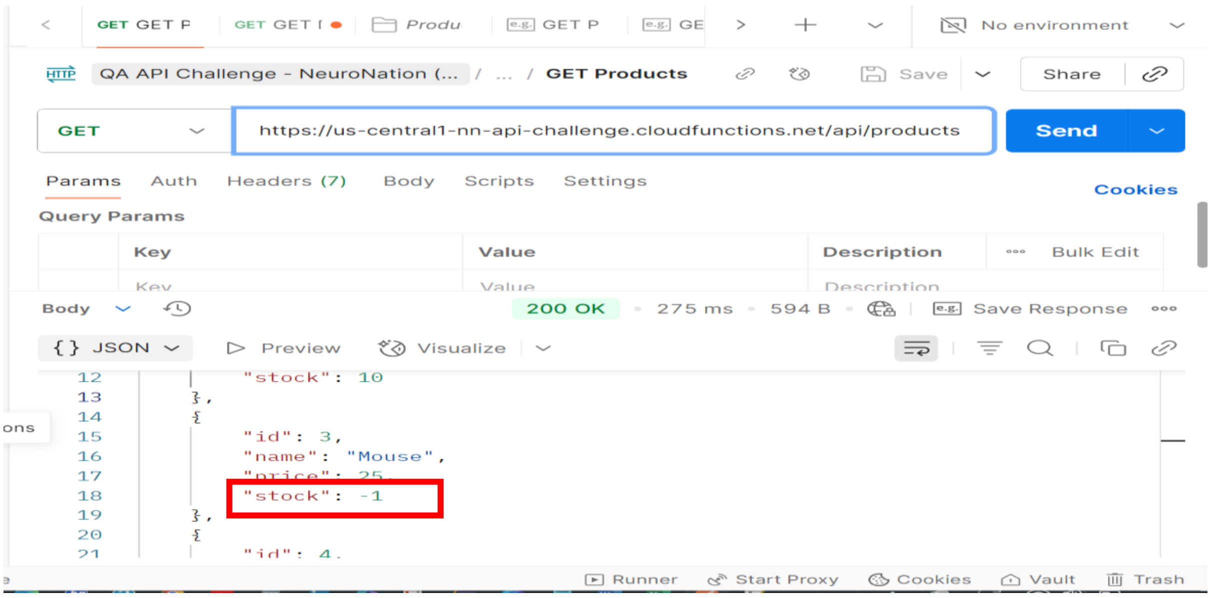
❖ **Bug Summary Table** 15

❖ **Recommendation Summary** 16

❖Bug Report

1. **Bug #001 – Negative Stock Values**

- **Endpoint:** GET /products
- **Steps to Reproduce:**
    1. Send request to /products.
    2. Inspect product with name "Mouse".
- **Actual Result:** Stock value = -1.
- **Expected Result:** Stock must always be >= 0.
- **Severity:** High (data integrity, impacts ordering).
- **Notes:**
- **Add server-side validation: stock must be integer ≥ 0.**
- **Consider database constraints and sanitization in write paths.**



GET https://us-central1-nn-api-challenge.cloudfunctions.net/api/products

```
      "stock": 10
    },
    {
      "id": 3,
      "name": "Mouse",
      "price": 25,
      "stock": -1
    },
    {
      "id": 4,
```

2. <u>Bug #002: Special characters in product name not returned (Ü/Umlaut handling)</u>

- **Environment:** Same as above
- **Endpoint:** GET /api/products?name=Tastatür

Steps to reproduce

1. Send GET request with query name=Tastatür (Unicode).
2. Observe response.

Expected result

- Returns the product Tastatür (id=4) when exact name is provided.

Actual result

- Empty array [] is returned despite product existing.
- The product is available via GET Product By ID=4 and appears in other endpoints.

Evidence

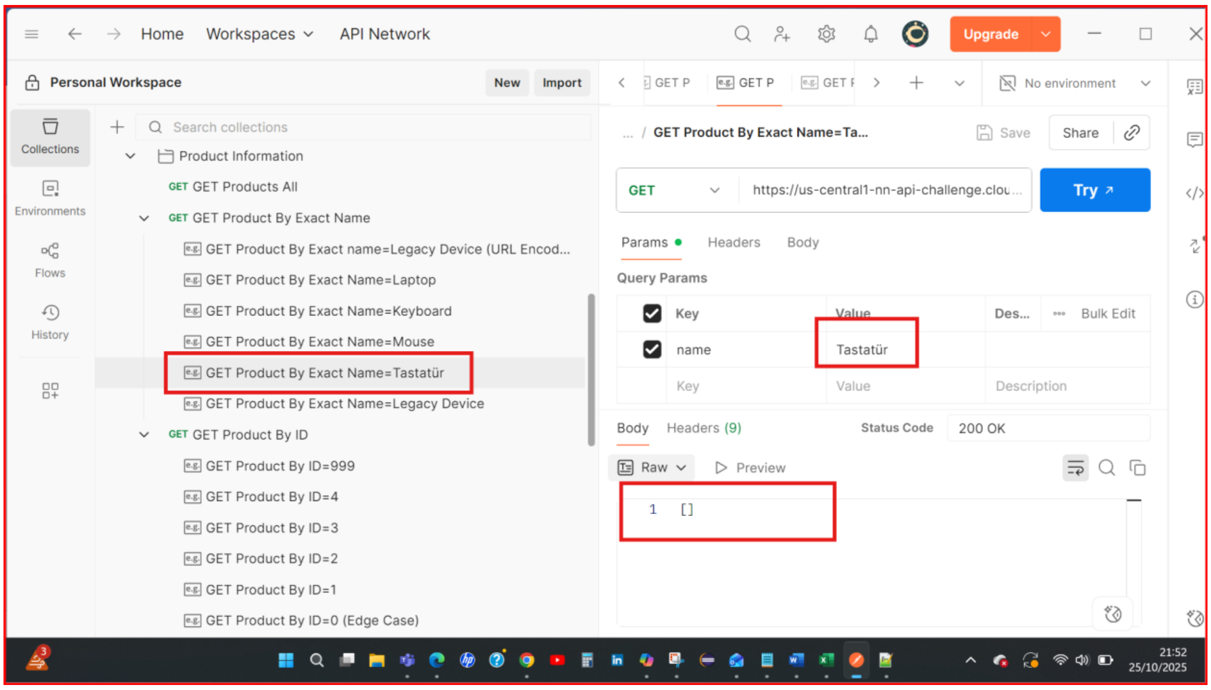
- GET Product By Exact Name=Tastatür returns [].
- GET Product By ID=4 shows name "Tastatür".

Severity

- **Medium:** Search/filter broken for localized names; user experience and data retrieval impacted.

Notes

- Normalize and compare names using proper Unicode collation.
- Support URL decoding and case-sensitive/insensitive exact matching.



Product Information
GET GET Products All
GET GET Product By Exact Name
GET Product By Exact name=Legacy Device (URL Encod...
GET Product By Exact Name=Laptop
GET Product By Exact Name=Keyboard
GET Product By Exact Name=Mouse
GET Product By Exact Name=Tastatür
GET Product By Exact Name=Legacy Device
GET GET Product By ID
GET Product By ID=999
GET Product By ID=4
GET Product By ID=3
GET Product By ID=2
GET Product By ID=1
GET Product By ID=0 (Edge Case)

... / GET Product By Exact Name=Ta...

GET https://us-central1-nn-api-challenge.clou...

✓	Key	Value	Des...
✓	name	Tastatür	
	Key	Value	Description

Body Headers (9) Status Code 200 OK

```
1  []
```

21:52
25/10/2025

3. **<u>Bug #003: Product missing price & description in detailed view</u>**

- Environment: Same
- Endpoint: GET /api/products/3

Steps to reproduce

1. Request product details for id=3.
2. Review detailed payload fields.

Expected result

- Detailed product contains standard fields: id, name, price, stock, description/isDetailed.

Actual result

- Response includes id, name, stock, isDetailed=true, but price & description field is missing.

Evidence

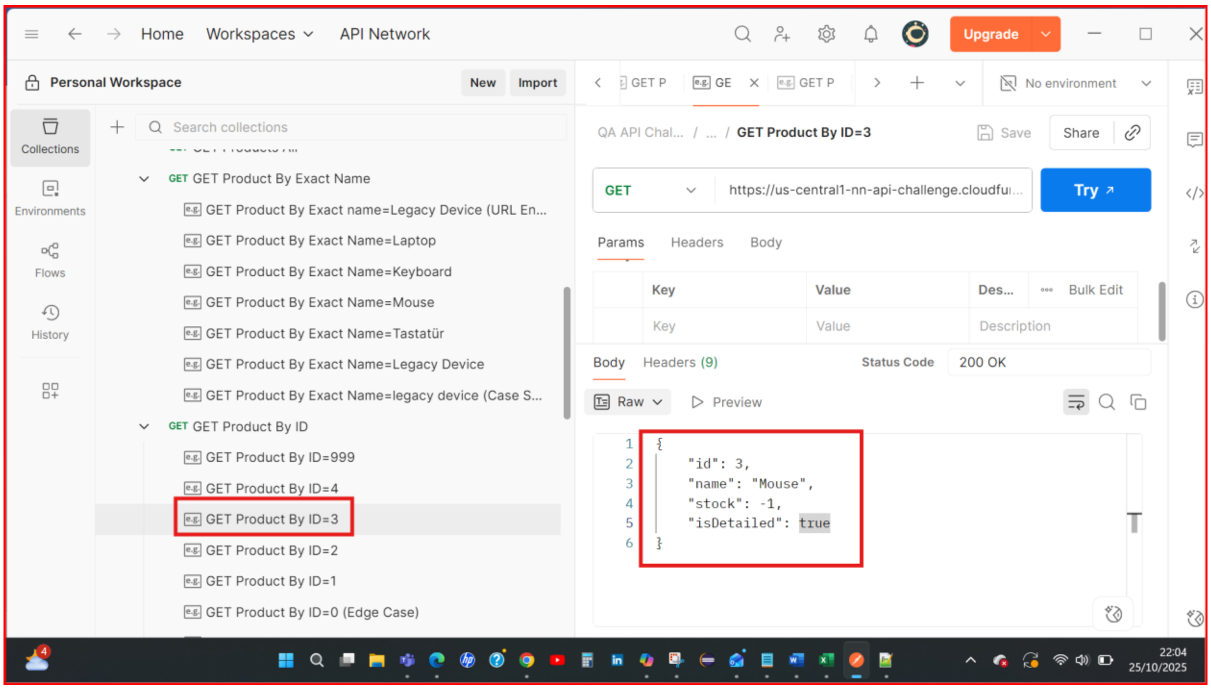
- GET Product By ID=3 body lacks "price" & "description" when isDetailed=true.

Severity

- Medium: Incomplete product data affects pricing, description and totals.

Notes

- Ensure consistent schema across all detailed product endpoints.
- Add contract tests validating required fields.



Home Workspaces API Network Upgrade

Personal Workspace New Import

Collections Environments Flows History

Search collections

- GET GET Product By Exact Name
  - GET Product By Exact name=Legacy Device (URL En...
  - GET Product By Exact Name=Laptop
  - GET Product By Exact Name=Keyboard
  - GET Product By Exact Name=Mouse
  - GET Product By Exact Name=Tastatür
  - GET Product By Exact Name=Legacy Device
  - GET Product By Exact Name=legacy device (Case S...
- GET GET Product By ID
  - GET Product By ID=999
  - GET Product By ID=4
  - GET Product By ID=3
  - GET Product By ID=2
  - GET Product By ID=1
  - GET Product By ID=0 (Edge Case)

QA API Chal... / ... / GET Product By ID=3 Save Share

No environment

GET https://us-central1-nn-api-challenge.cloudfu... Try

Params Headers Body

Key	Value	Des...
Key	Value	Description

Body Headers (9) Status Code 200 OK

Raw Preview

```
{
    "id": 3,
    "name": "Mouse",
    "stock": -1,
    "isDetailed": true
}
```

22:04
25/10/2025

4. **Bug #004: Internal Server Error on valid product ID route**

- **Environment:** Same
- **Endpoint:** GET /api/products/999

Steps to reproduce

1. Request product id=999.
2. Observe status code and payload.

Expected result

- Either 200 with product data (if exists) or 404 Not Found with structured JSON error.

Actual result

- 500 Internal Server Error with "Server error" and content-type text/html.

Evidence

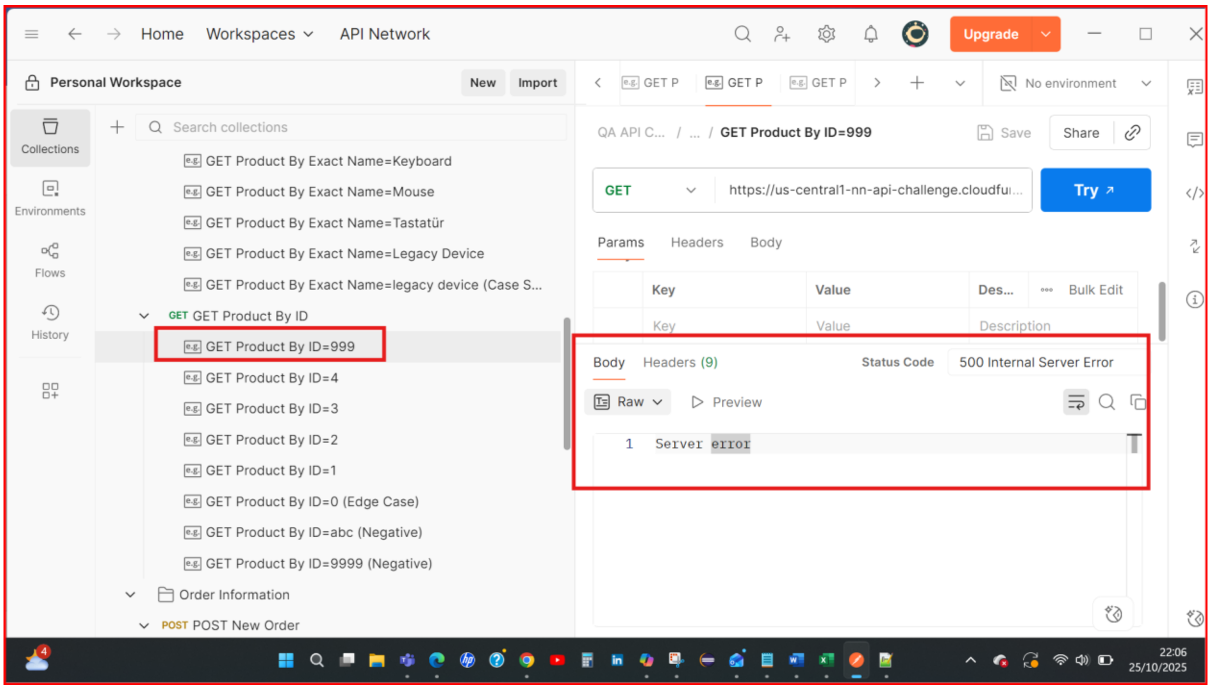
- GET Product By ID=999 returns 500, body "Server error", content-type text/html.

Severity

- **High:** Unhandled server error; reveals instability and inconsistent error format.

Notes

- Handle unknown IDs with 404 JSON error object.
- Avoid HTML responses; standardize JSON errors across API.


Home Workspaces API Network Upgrade

Personal Workspace New Import

Collections Environments Flows History

Search collections

GET Product By Exact Name=Keyboard
GET Product By Exact Name=Mouse
GET Product By Exact Name=Tastatür
GET Product By Exact Name=Legacy Device
GET Product By Exact Name=legacy device (Case S...
GET GET Product By ID
GET Product By ID=999
GET Product By ID=4
GET Product By ID=3
GET Product By ID=2
GET Product By ID=1
GET Product By ID=0 (Edge Case)
GET Product By ID=abc (Negative)
GET Product By ID=9999 (Negative)
Order Information
POST POST New Order

GET P GET P GET P No environment

QA API C... / ... / GET Product By ID=999 Save Share

GET https://us-central1-nn-api-challenge.cloudfu... Try

Params Headers Body

Key	Value	Des...
Key	Value	Description

Body Headers (9) Status Code 500 Internal Server Error

Raw Preview

```
Server error
```

22:06
25/10/2025

5. **Bug #005: Registration endpoint lacks validation, accepts invalid inputs**

- **Environment:** Same
- **Endpoint:** POST /api/register

Steps to reproduce

1. Attempt to register with invalid email (missing '@'): sonalkajrolkar_neuronation.com.
2. Attempt to register with empty password.
3. Attempt with excessively long password.
4. Attempt duplicate email registrations.

Expected result

- Reject invalid email format (400) with error message.
- Reject empty/whitespace passwords (400).
- Enforce password policy and length limits.
- Prevent duplicate email registration (409).

Actual result

- Invalid email format returns 201 "User registered".
- Empty password returns 201 "User registered".
- Very long password returns 201 "User registered".
- Duplicate registration is sometimes flagged (409), but other invalid inputs still pass.

Evidence

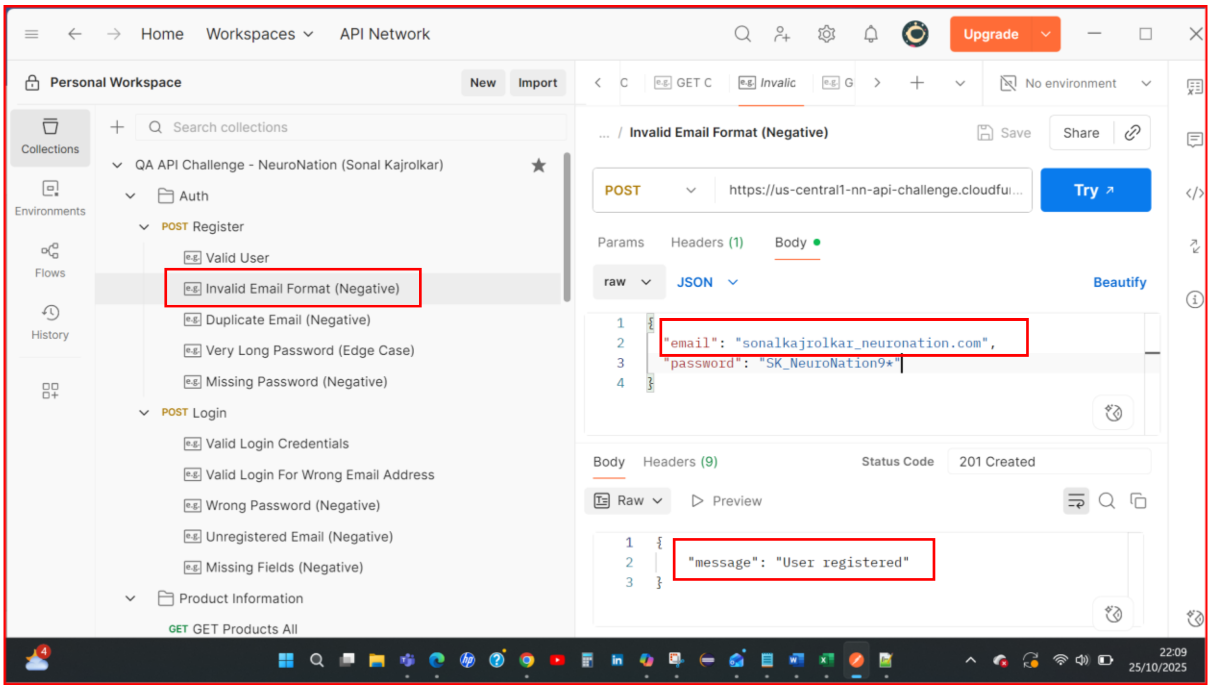
- "Invalid Email Format (Negative)" shows 201 "User registered".
- "Missing Password (Negative)" shows 201.
- "Very Long Password (Edge Case)" shows 201.
- "Duplicate Email (Negative)" correctly shows 409.

Severity

- **Critical:** Security, data quality, and compliance risk.

Notes

- Add robust server-side validation: RFC-compliant email, password rules, length constraints.
- Return precise error messages and appropriate status codes.


Home
Workspaces
API Network
Upgrade
Personal Workspace
New
Import
Collections
Environments
Flows
History
Search collections
QA API Challenge - NeuroNation (Sonal Kajrolkar)
Auth
POST Register
Valid User
Invalid Email Format (Negative)
Duplicate Email (Negative)
Very Long Password (Edge Case)
Missing Password (Negative)
POST Login
Valid Login Credentials
Valid Login For Wrong Email Address
Wrong Password (Negative)
Unregistered Email (Negative)
Missing Fields (Negative)
Product Information
GET GET Products All
No environment
Invalid Email Format (Negative)
Save
Share
POST
https://us-central1-nn-api-challenge.cloudfu...
Try
Params
Headers (1)
Body
raw
JSON
Beautify
Body
Headers (9)
Status Code
201 Created
Raw
Preview
22:09
25/10/2025

```
{
"email": "sonalkajrolkar_neuronation.com",
"password": "SK_NeuroNation9*"
}
```

Status Code 201 Created

```
{
 "message": "User registered"
}
```

6. **<u>Bug #006: Login endpoint accepts invalid scenarios and issues tokens</u>**

• Environment: Same

• Endpoint: POST /api/login

Steps to reproduce

1. Login with malformed email: sonalkajrolkar_neuronation.com.
2. Login with empty/whitespace password.

Expected result

• Reject invalid email format (400/422).

• Reject empty/whitespace passwords (400/422).

Actual result

• Both scenarios returned 200 OK and issued JWT tokens.

Evidence

• "Valid Login For Wrong Email Address" returns 200 with token.

• "Missing Fields (Negative)" returns 200 with token.

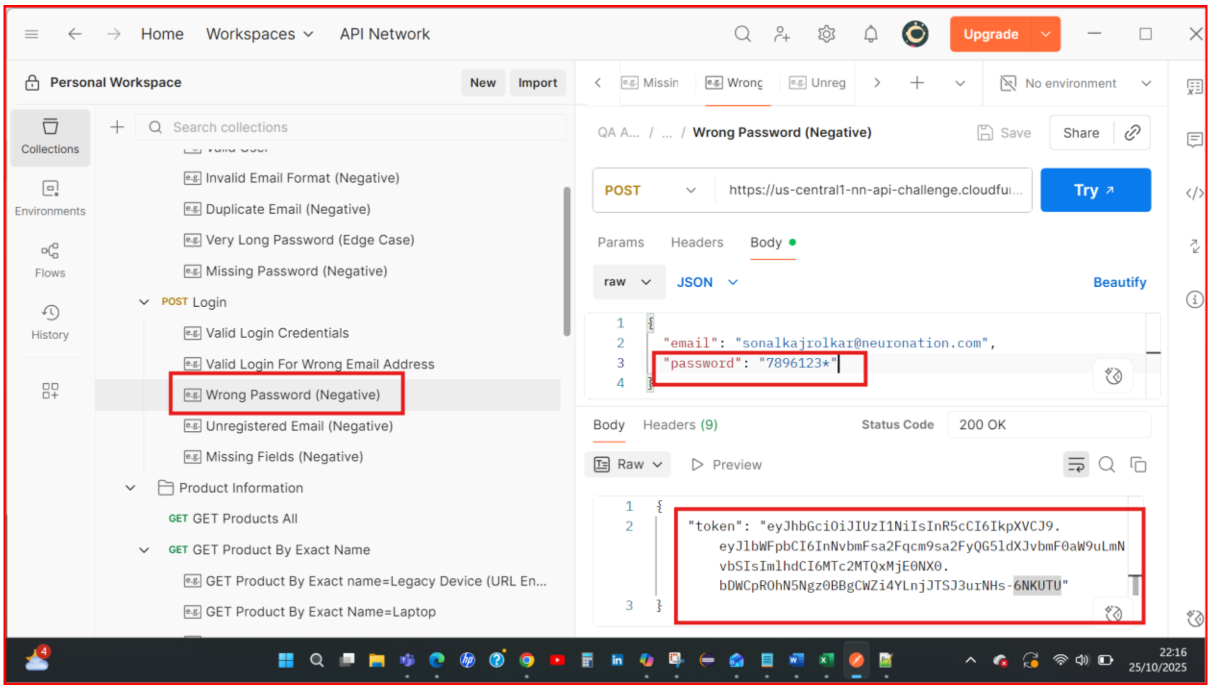
Severity

• Critical: Authentication integrity risk; potential account enumeration/abuse.

Notes

• Enforce input validation at login similar to registration.

• Return specific error codes/messages for invalid credentials/inputs.


Home Workspaces API Network

Personal Workspace New Import

Search collections

- Invalid Email Format (Negative)
- Duplicate Email (Negative)
- Very Long Password (Edge Case)
- Missing Password (Negative)
- POST Login
  - Valid Login Credentials
  - Valid Login For Wrong Email Address
  - Wrong Password (Negative)
  - Unregistered Email (Negative)
  - Missing Fields (Negative)
- Product Information
  - GET GET Products All
  - GET GET Product By Exact Name
    - GET Product By Exact name=Legacy Device (URL En...
    - GET Product By Exact Name=Laptop

QA A... / ... / Wrong Password (Negative)

POST https://us-central1-nn-api-challenge.cloudfu...

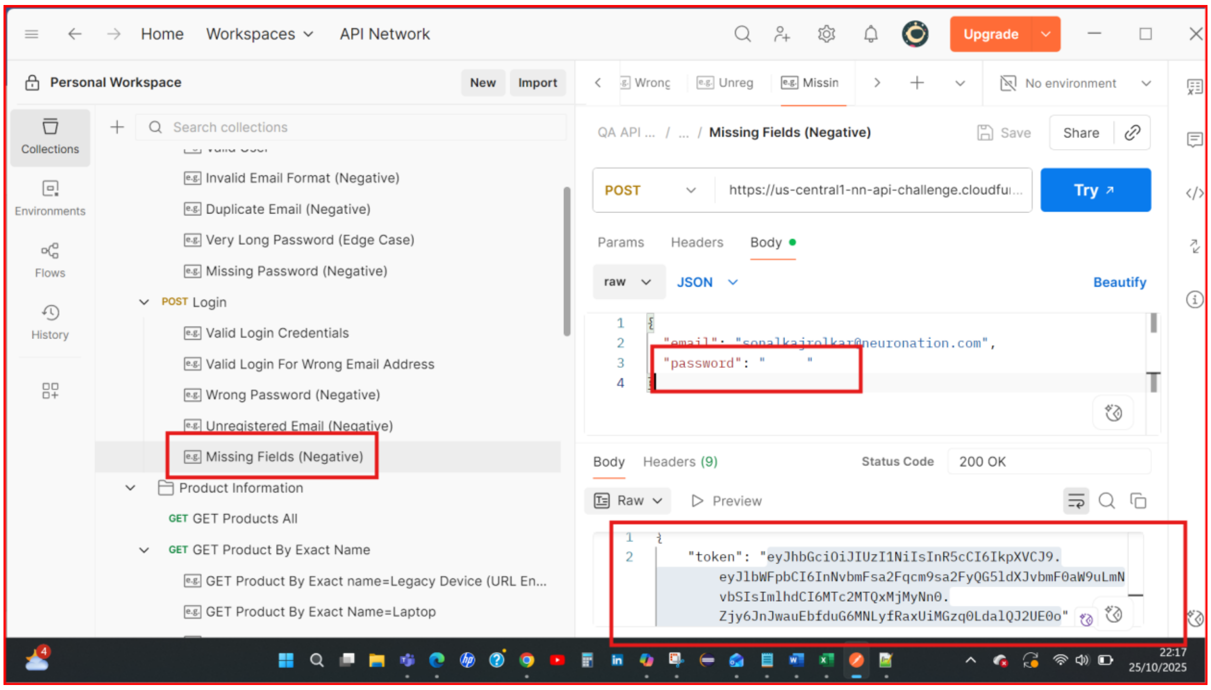
Body: raw JSON

```
{
  "email": "sonalkajrolkar@neuronation.com",
  "password": "7896123*"
}
```

Status Code 200 OK

```
{
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
        eyJlbWFpbCI6InNvbmFsa2Fqcm9sa2FyQG5ldXJvbmF0aW9uLmN
        vbSIsImlhdCI6MTc2MTQxMjE0NX0.
        bDWCpROhN5Ngz0BBgCWZi4YLnjJTSJ3urNHs-6NKUTU"
}
```

22:16 25/10/2025


Home Workspaces API Network

Personal Workspace New Import

Search collections

- Invalid Email Format (Negative)
- Duplicate Email (Negative)
- Very Long Password (Edge Case)
- Missing Password (Negative)
- POST Login
  - Valid Login Credentials
  - Valid Login For Wrong Email Address
  - Wrong Password (Negative)
  - Unregistered Email (Negative)
  - Missing Fields (Negative)
- Product Information
  - GET GET Products All
  - GET GET Product By Exact Name
    - GET Product By Exact name=Legacy Device (URL En...
    - GET Product By Exact Name=Laptop

QA API ... / ... / Missing Fields (Negative)

POST https://us-central1-nn-api-challenge.cloudfu...

Body: raw JSON

```
{
  "email": "sonalkajrolkar@neuronation.com",
  "password": "      "
}
```

Status Code 200 OK

```
{
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
        eyJlbWFpbCI6InNvbmFsa2Fqcm9sa2FyQG5ldXJvbmF0aW9uLmN
        vbSIsImlhdCI6MTc2MTQxMjMyNn0.
        Zjy6JnJwauEbfduG6MNLyfRaxUiMGzq0LdalQJ2UE0o"
```

22:17 25/10/2025

7. **<u>Bug #007: Order creation accepts invalid or empty product lists</u>**

• Environment: Same

• Endpoint: POST /api/orders

Steps to reproduce

1. Create order with invalid product id [9999].
2. Create order with empty product array [].

Expected result

• Reject invalid product IDs with validation error (400).

• Reject empty product lists (400), require at least one valid product.

Actual result

• Requests succeed (200), orders are created with total=0.

Evidence

• "Invalid Product IDs (Negative)" returns 200 and creates order with products [9999], total 0.

• "Empty Product Array (Negative)" returns 200 and creates order with products [], total 0.

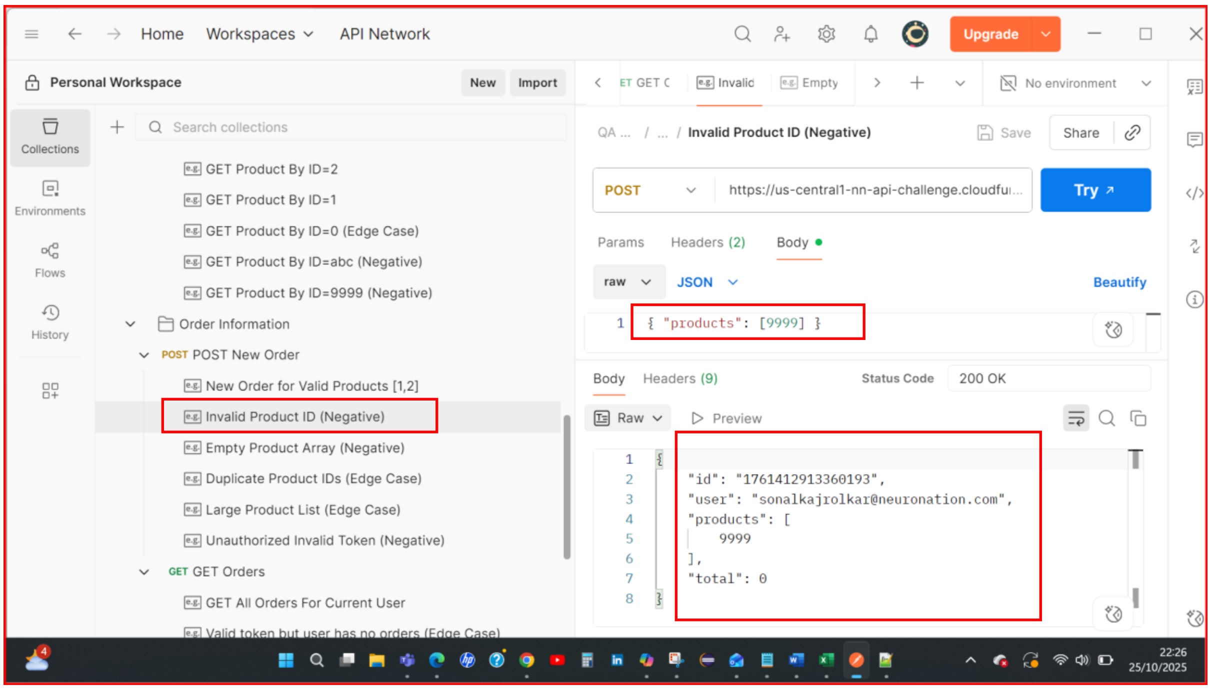
Severity

• High: Order data integrity compromised; enables meaningless orders.

Notes

• Validate product existence and availability before order creation.

• Enforce business rules for minimum items and non-zero totals

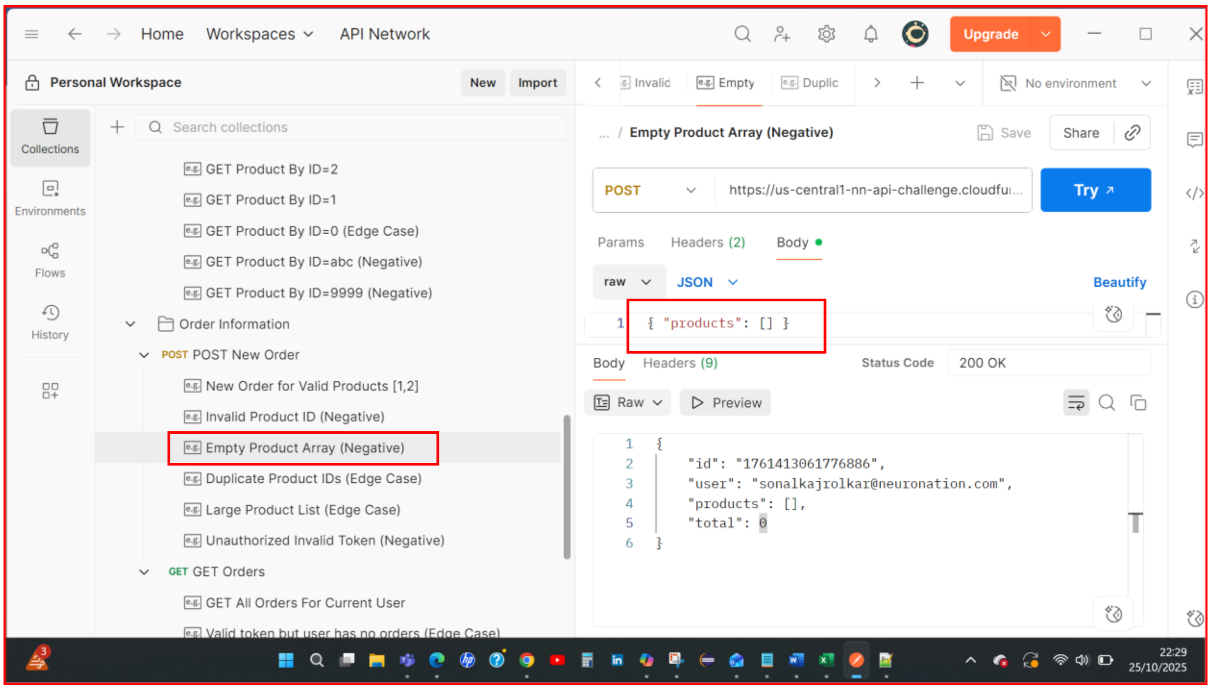

Home
Workspaces
API Network
Upgrade
Personal Workspace
New
Import
Collections
Environments
Flows
History
Search collections
GET Product By ID=2
GET Product By ID=1
GET Product By ID=0 (Edge Case)
GET Product By ID=abc (Negative)
GET Product By ID=9999 (Negative)
Order Information
POST New Order
New Order for Valid Products [1,2]
Invalid Product ID (Negative)
Empty Product Array (Negative)
Duplicate Product IDs (Edge Case)
Large Product List (Edge Case)
Unauthorized Invalid Token (Negative)
GET Orders
GET All Orders For Current User
Valid token but user has no orders (Edge Case)
No environment
QA ... / ... / Invalid Product ID (Negative)
Save
Share
Try
Params
Headers (2)
Body
raw
JSON
Beautify
Headers (9)
Raw
Preview
22:26
25/10/2025

POST https://us-central1-nn-api-challenge.cloudfu...

```
{ "products": [9999] }
```

Status Code 200 OK

```
{
    "id": "1761412913360193",
    "user": "sonalkajrolkar@neuronation.com",
    "products": [
        9999
    ],
    "total": 0
}
```



GET Product By ID=2
GET Product By ID=1
GET Product By ID=0 (Edge Case)
GET Product By ID=abc (Negative)
GET Product By ID=9999 (Negative)
Order Information
POST New Order
New Order for Valid Products [1,2]
Invalid Product ID (Negative)
Empty Product Array (Negative)
Duplicate Product IDs (Edge Case)
Large Product List (Edge Case)
Unauthorized Invalid Token (Negative)
GET Orders
GET All Orders For Current User
Valid token but user has no orders (Edge Case)

... / Empty Product Array (Negative)

POST https://us-central1-nn-api-challenge.cloudfu...

raw JSON

```
{ "products": [] }
```

Status Code 200 OK

```
{
    "id": "1761413061776886",
    "user": "sonalkajrolkar@neuronation.com",
    "products": [],
    "total": 0
}
```

22:29
25/10/2025

8. <u>Bug #008: Orders list returns ambiguous empty result instead of clear validation message</u>

- Environment: Same
- Endpoint: GET /api/orders

Steps to reproduce

1. Use a valid token for a user with no orders.
2. Request orders list.

Expected result

• Clear JSON response indicating no orders for user (e.g., [] with explanatory meta or message).

Actual result

• Returns an empty array [] with no contextual message.

Evidence

• "Valid token but user has no orders (Edge Case)" returns [].

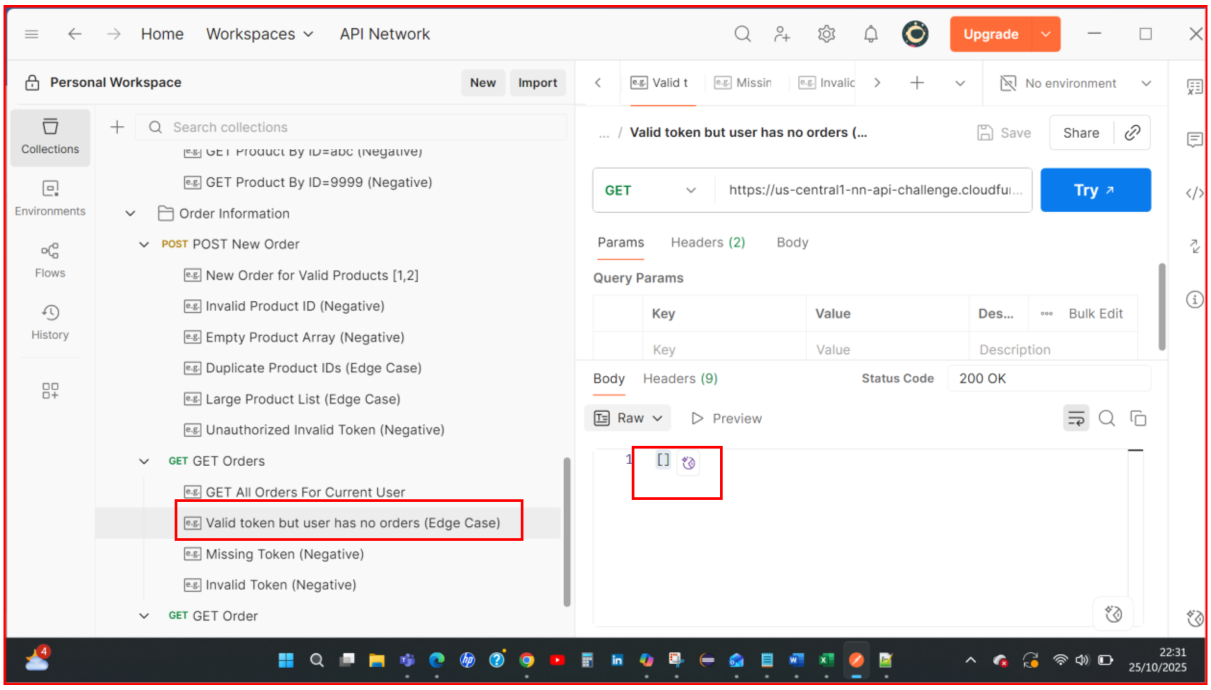
Severity

• Low to Medium: Usability issue; may confuse clients.

Notes

• Optionally include "count":0 or message field for clarity.

• Keep [] but support consistent client UX via metadata.



Personal Workspace

- GET Product By ID=9999 (Negative)
- Order Information
  - POST New Order
    - New Order for Valid Products [1,2]
    - Invalid Product ID (Negative)
    - Empty Product Array (Negative)
    - Duplicate Product IDs (Edge Case)
    - Large Product List (Edge Case)
    - Unauthorized Invalid Token (Negative)
  - GET Orders
    - GET All Orders For Current User
    - Valid token but user has no orders (Edge Case)
    - Missing Token (Negative)
    - Invalid Token (Negative)
  - GET Order

Valid token but user has no orders (...

GET https://us-central1-nn-api-challenge.cloudfu...

Status Code 200 OK

```
[]
```

22:31 25/10/2025

9. Bug #009: GET order by non-numeric ID returns data (ID format not enforced)

- Environment: Same
- Endpoint: GET /api/orders/abcfe

Steps to reproduce

1. Request order with ID "abcfe".

Expected result

• Reject invalid ID format (400) or return 404 Not Found.

Actual result

• Returns 200 OK with a valid order payload for a shadow/test user.

Evidence

• "GET Order By ID=abcfe (Negative)" returns id=99999999999, user "shadow@qa.local", total=1055.

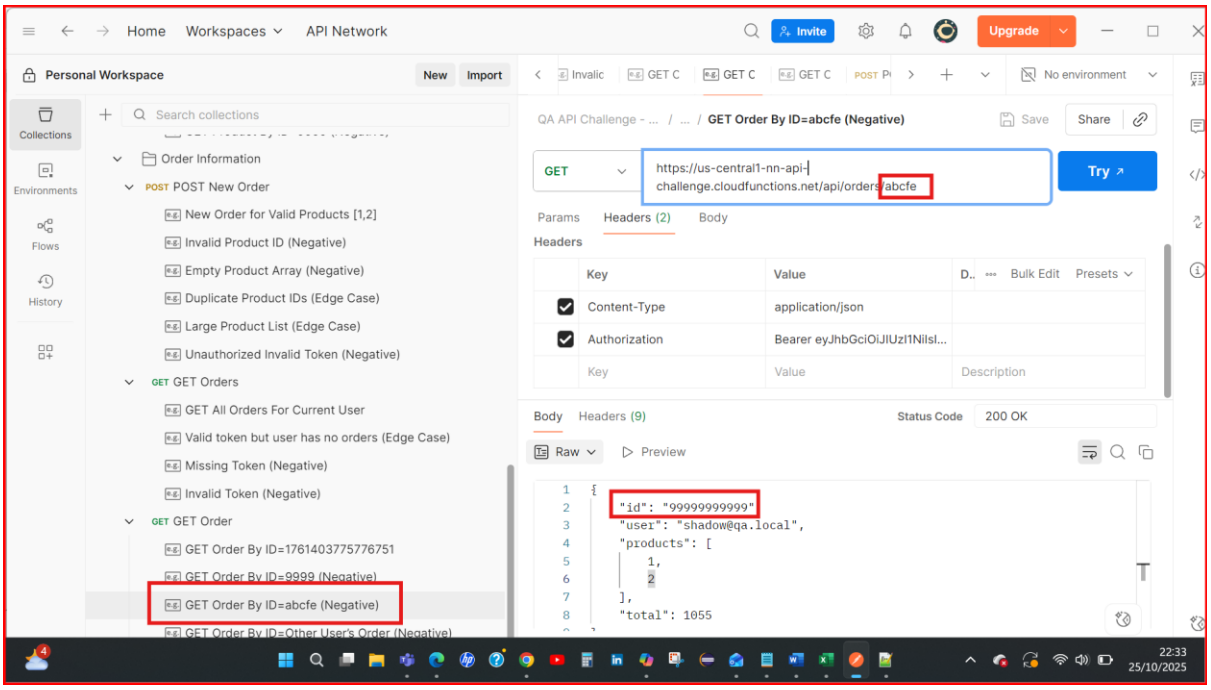
Severity

• High: Routing/MVC mismatch and potential information disclosure.

Notes

• Enforce ID schema (numeric-only) at routing layer.

• Prevent fallback/default test data from being returned.



Home
Workspaces
API Network
Invite
Upgrade
Personal Workspace
New
Import
Collections
Environments
Flows
History
Search collections
Order Information
POST New Order
New Order for Valid Products [1,2]
Invalid Product ID (Negative)
Empty Product Array (Negative)
Duplicate Product IDs (Edge Case)
Large Product List (Edge Case)
Unauthorized Invalid Token (Negative)
GET Orders
GET All Orders For Current User
Valid token but user has no orders (Edge Case)
Missing Token (Negative)
Invalid Token (Negative)
GET Order
GET Order By ID=1761403775776751
GET Order By ID=9999 (Negative)
GET Order By ID=abcfe (Negative)
GET Order By ID=Other User's Order (Negative)
No environment
QA API Challenge - ... / ... / GET Order By ID=abcfe (Negative)
Save
Share
Try
Params
Headers (2)
Body
Body
Headers (9)
Raw
Preview
22:33
25/10/2025

GET https://us-central1-nn-api-challenge.cloudfunctions.net/api/orders/abcfe

Headers

	Key	Value	Description
✓	Content-Type	application/json	
✓	Authorization	Bearer eyJhbGciOiJIUzI1NiIsI...	

Status Code 200 OK

```
{
  "id": "99999999999",
  "user": "shadow@qa.local",
  "products": [
      1,
      2
  ],
  "total": 1055
```

10. Bug #0010: Access control broken—able to fetch another user’s order

- Environment: Same
- Endpoint: GET /api/orders/{orderId}

Steps to reproduce

1. Authenticate as user A (sonalkajrolkar@neuronation.com).
2. Request order ID that belongs to user B.
3. Observe response.

Expected result

• 403 Forbidden or 404 Not Found. User should not access orders of other users.

Actual result

• 200 OK, returns another user’s order details.

Evidence

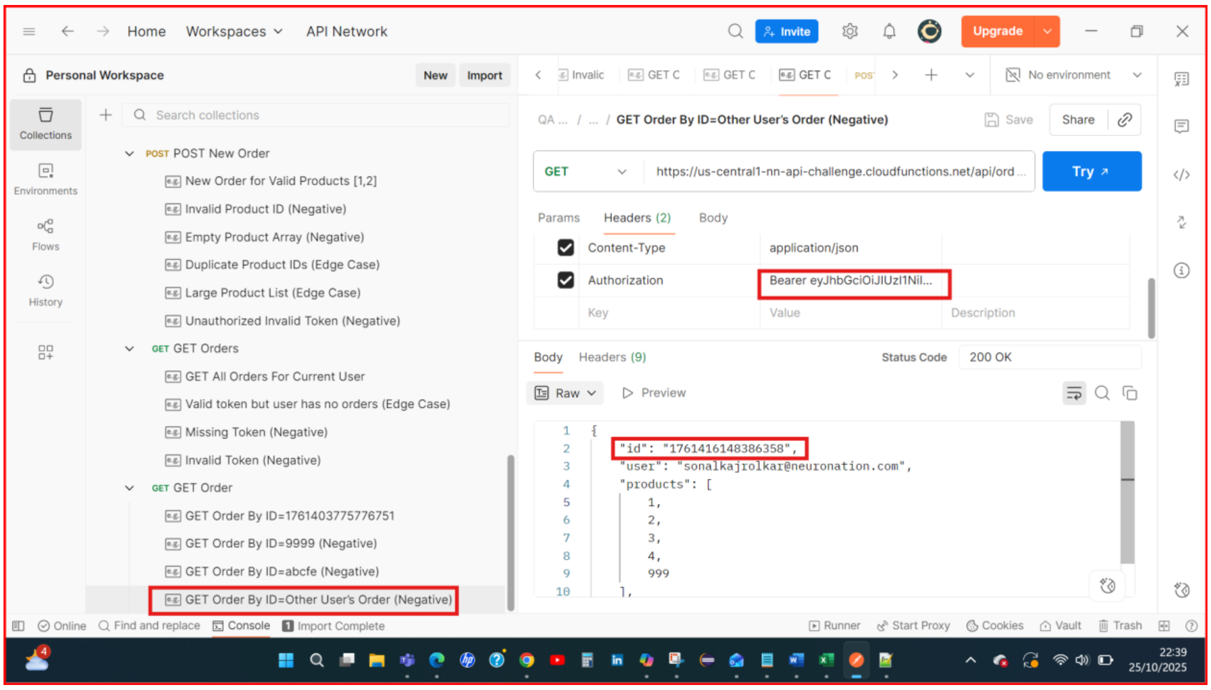
• "GET Order By ID=Other User’s Order (Negative)" returns order id=1761416148386358 with user "sonalkajrolkar@neuronation.com", despite token mismatch scenario described.

Severity

• Critical: Authorization flaw; sensitive data disclosure.

Notes

• Add a check so that the order always belongs to the logged-in user. The user field in the order must match the email from the Jason Token.



POST POST New Order
- New Order for Valid Products [1,2]
- Invalid Product ID (Negative)
- Empty Product Array (Negative)
- Duplicate Product IDs (Edge Case)
- Large Product List (Edge Case)
- Unauthorized Invalid Token (Negative)

GET GET Orders
- GET All Orders For Current User
- Valid token but user has no orders (Edge Case)
- Missing Token (Negative)
- Invalid Token (Negative)

GET GET Order
- GET Order By ID=1761403775776751
- GET Order By ID=9999 (Negative)
- GET Order By ID=abcfe (Negative)
- GET Order By ID=Other User’s Order (Negative)

QA ... / ... / GET Order By ID=Other User’s Order (Negative)

GET https://us-central1-nn-api-challenge.cloudfunctions.net/api/ord ...

Params | Headers (2) | Body

	Key	Value	Description
✓	Content-Type	application/json	
✓	Authorization	Bearer eyJhbGciOiJIUzI1NiI...	

Body | Headers (9) | Status Code 200 OK

Raw | Preview

```
{
    "id": "1761416148386358",
    "user": "sonalkajrolkar@neuronation.com",
    "products": [
        1,
        2,
        3,
        4,
        999
    ],
```

❖<u>Bug Summary Table</u>

Bug ID	Endpoint / Request	Steps to Reproduce	Actual Result	Expected Result	Severity
1	GET /products (All)	Request all products	Product “Mouse” shows stock = -1	Stock must always be ≥ 0	High
2	GET /products?name=Tastatür	Search by exact name with umlaut	Empty array returned	Product “Tastatür” should be returned	Medium
3	GET /products/3	Request product by ID=3	Price & description field is missing	Price & description must always be present	High
4	GET /products/999	Request product with invalid ID	500 Server Error (HTML)	404 JSON error { "error": "Not found" }	High
5	POST /register	Register with invalid email, empty password	User still registered (201)	Reject invalid inputs with 400; enforce rules	Critical
6	POST /login	Login with wrong email or blank password	Token still issued	Reject invalid login with 400/401	Critical
7	POST /orders	Create order with invalid ID or empty list	Order created with total=0	Reject invalid/empty product lists with 400	High
8	GET /orders	Request orders for new user	Returns empty array [] with no message	Return empty array with clear message/metadata	Low
9	GET /orders/abcfe	Request order with non-numeric ID	Returns fabricated order	Reject invalid ID with 400/404	Critical
10	GET /orders/{id} (Other)	Login as User A, request User B’s order	Returns User B’s order	Block access with 403/404	Critical

❖Recommendation Summary

- **Check user details carefully**
  When people sign up or log in, the system should check that the email looks real, the password is strong enough, and no fields are left empty.
- **Fix product information**
  Product stock should never be a negative number. Every product should always show the same basic details like name, price, and stock.
- **Show clear error messages**
  If something goes wrong, the system should give a clear message in the same format every time. It should not show confusing “server error” pages.
- **Support special letters**
  Searches should work with special characters (like “ü” such as *Tastatür*).
- **Protect customer data**
  Each person should only be able to see their own orders. Nobody should be able to see another person’s order.
- **Check orders before saving**
  Orders should only be created if the products exist and the list is not empty. The total price should never be zero.
- **Keep answers consistent**
  All parts of the system should respond in the same style. If no data is found, the system should always give a clear and simple message.