

Table of Contents

[Cover letter 3](#_Toc180623128)

[Tools used 4](#_Toc180623129)

[Team members 4](#_Toc180623130)

[Project main idea 4](#_Toc180623131)

[API Testing Using Postman 5](#_Toc180623132)

[Security checks 14](#_Toc180623133)

[Security test cases 15](#_Toc180623134)

[Manual test cases 16](#_Toc180623135)

[Collaboration and tasks 23](#_Toc180623136)

[Source code for java project and selenium project 23](#_Toc180623137)

# Cover letter

To whom it may concern,

I am writing to submit the Clinic Management System project, which has been designed and developed as part of our ongoing initiative to improve patient and clinic management processes. The system provides a streamlined solution to manage patient records, appointments, and clinic operations, focusing on efficiency and simplicity.

**Project Overview**

The Clinic Management System is designed to allow the clinic staff to easily add, search, and manage patient data. The key features include:

**Patient Registration and Management:** A simple interface for adding new patients to the system, ensuring unique patient identification with a robust search function to easily retrieve patient records.

**Data Persistence:** Patient data is stored and retrieved from a CSV file, ensuring that the information is not lost between sessions. The project is structured to easily transition to more advanced storage options, such as a relational database.

**Patient Record Deletion:** An intuitive system for searching and deleting patients by ID, keeping the clinic’s database clean and up-to-date.

**Console Interaction:** The current version allows interaction through the console for ease of testing and development, but it is designed to easily integrate with a graphical user interface (GUI) or web application in the future.

Next Steps and Future Development

The project, while functional in its current state, offers significant room for growth. It can be expanded to incorporate features like appointment scheduling, doctor management, and payment handling. Additionally, the integration of a database like MySQL or SQLite could enhance data management and security.

I have also included a detailed explanation of the code structure and functionality in the attached documentation to ensure a smooth handover to the next team of developers. I am confident that with these additional features and improvements, the system will serve as a comprehensive solution for clinic management.

Thank you for the opportunity to work on this project. I am looking forward to any feedback you may have and am happy to assist with further development or answer any questions about the current implementation.

Sincerely,

HAKM Team

# Tools used

Jira, Java, Eclipse, Postman, Excel for manual test cases, selenium

# Team members

1. Mohamed Hussien
2. Kerelos Emad Ibrahim
3. Ahmed Mohamed galal
4. Ahmed Hesham Mohamed Abd Elgawad
5. Hussein Bakr Ali Abdelrahman
6. Abdelhay Zaghloul Elkemary
7. Mahmoud Abbas Fawzy

# Project main idea

A system to manage clinics and bookings and record bookings in a report

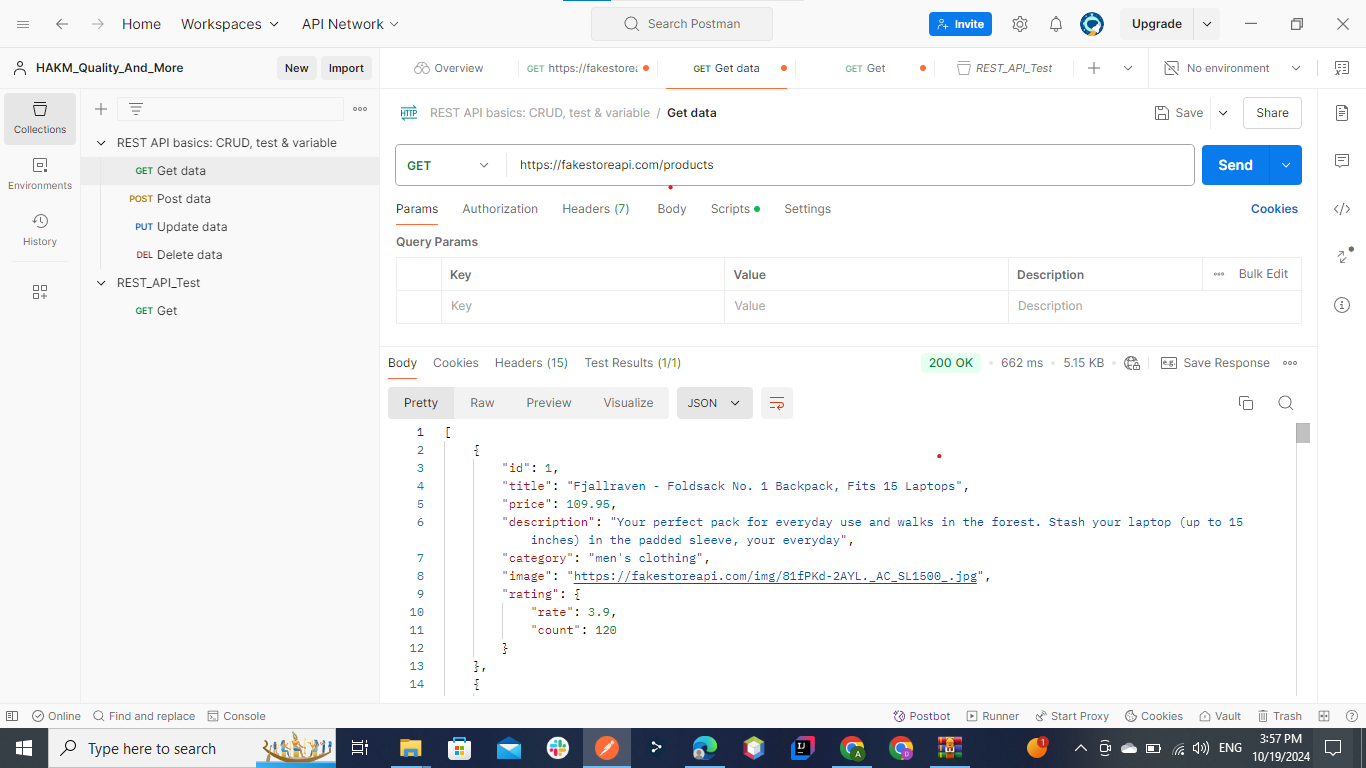
and then sent to the clinic or doctor with each patient, their condition, and details about the patient.

# API Testing Using Postman

**Team: HAKM Quality and more**

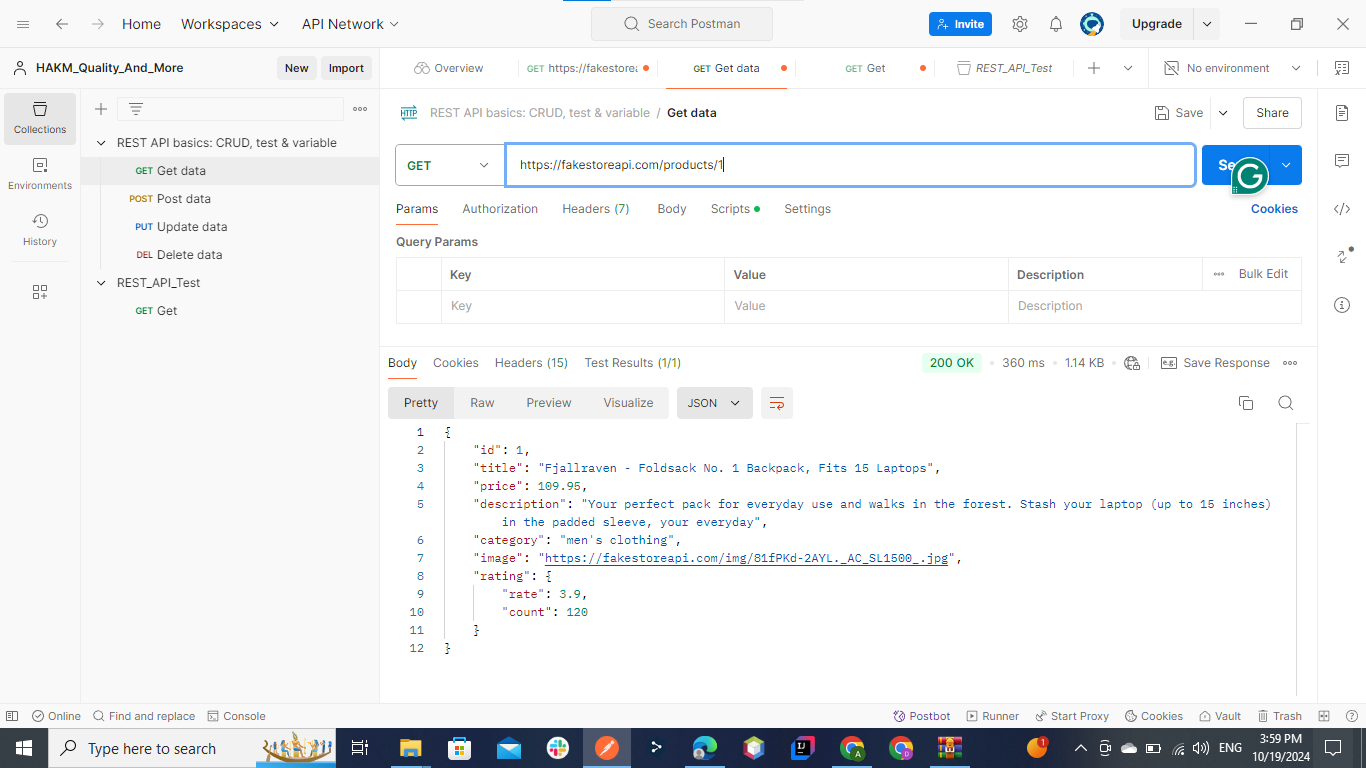
**Get request:**

* This shows a GET request made to the URL https://fakestoreapi.com/products.
* The response is a list of products, with one of them being displayed. The details include the product's id, title, price, description, category, and rating information (rate and count).
* The status of the request is also 200 OK, which confirms that the API returned the list of products successfully.

****

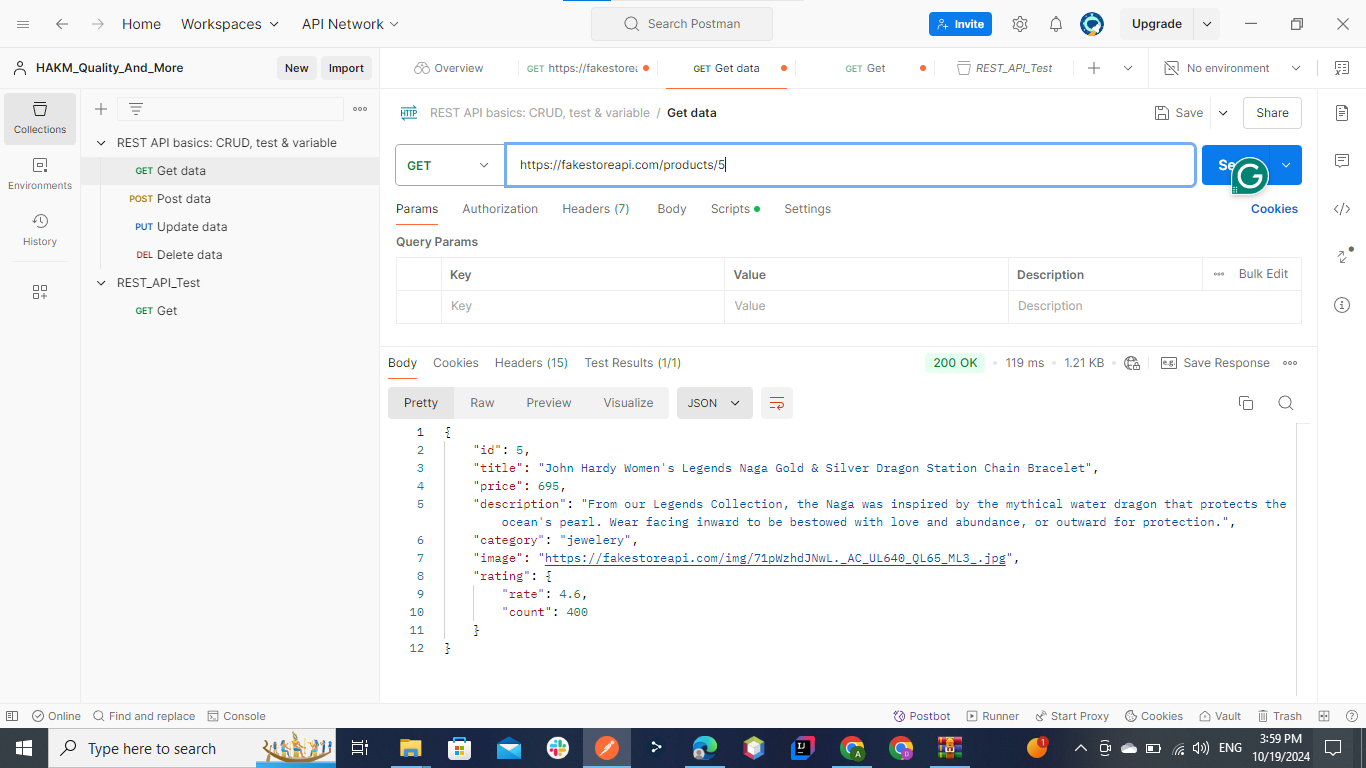
**Get request when id =1:**

* This shows a GET request made to the URL https://fakestoreapi.com/products/1.
* The response is a list of products, with one of them being displayed. The details include the product's id, title, price, description, category, and rating information (rate and count).
* The status of the request is also 200 OK, which confirms that the API returned the list of products successfully.

****

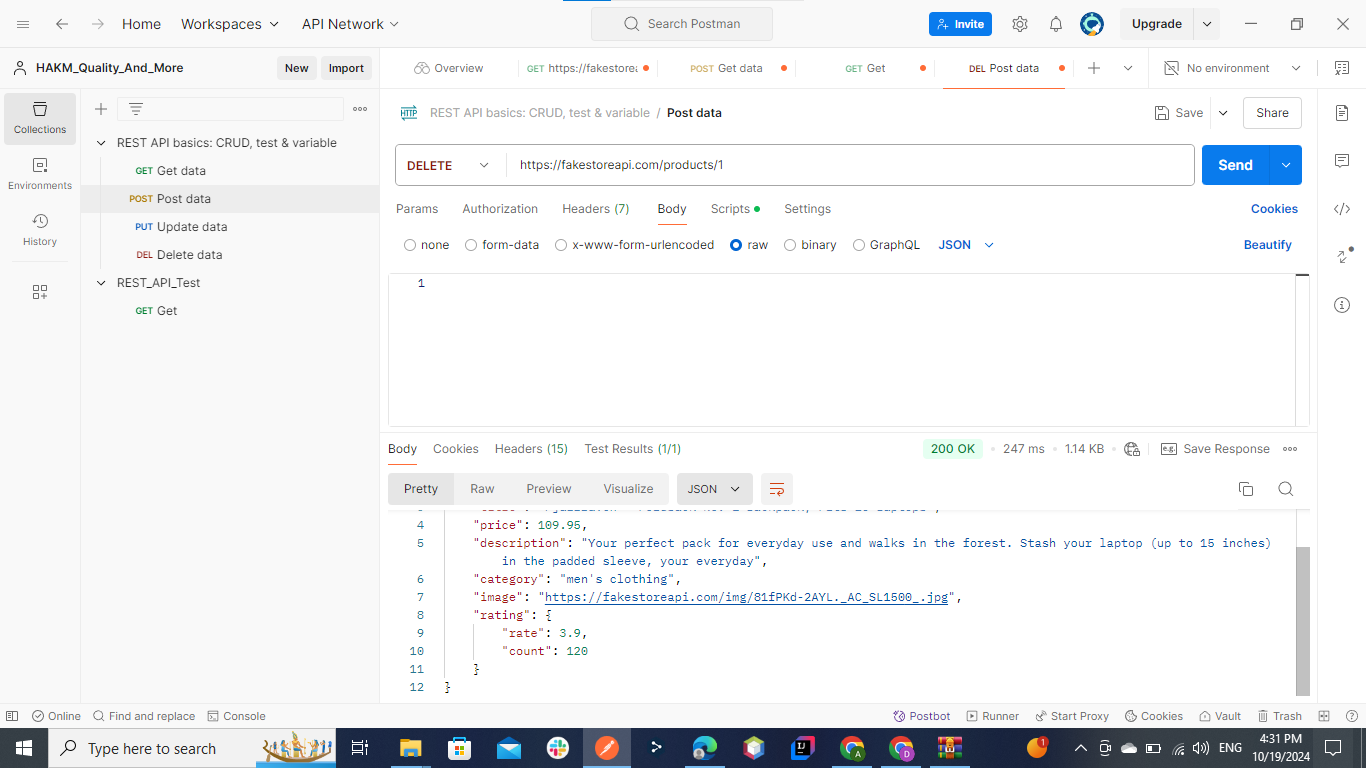
**Get request when id =5:**

* This shows a GET request made to the URL https://fakestoreapi.com/products/5.
* The response is a list of products, with one of them being displayed. The details include the product's id, title, price, description, category, and rating information (rate and count).
* The status of the request is also 200 OK, which confirms that the API returned the list of products successfully.

****

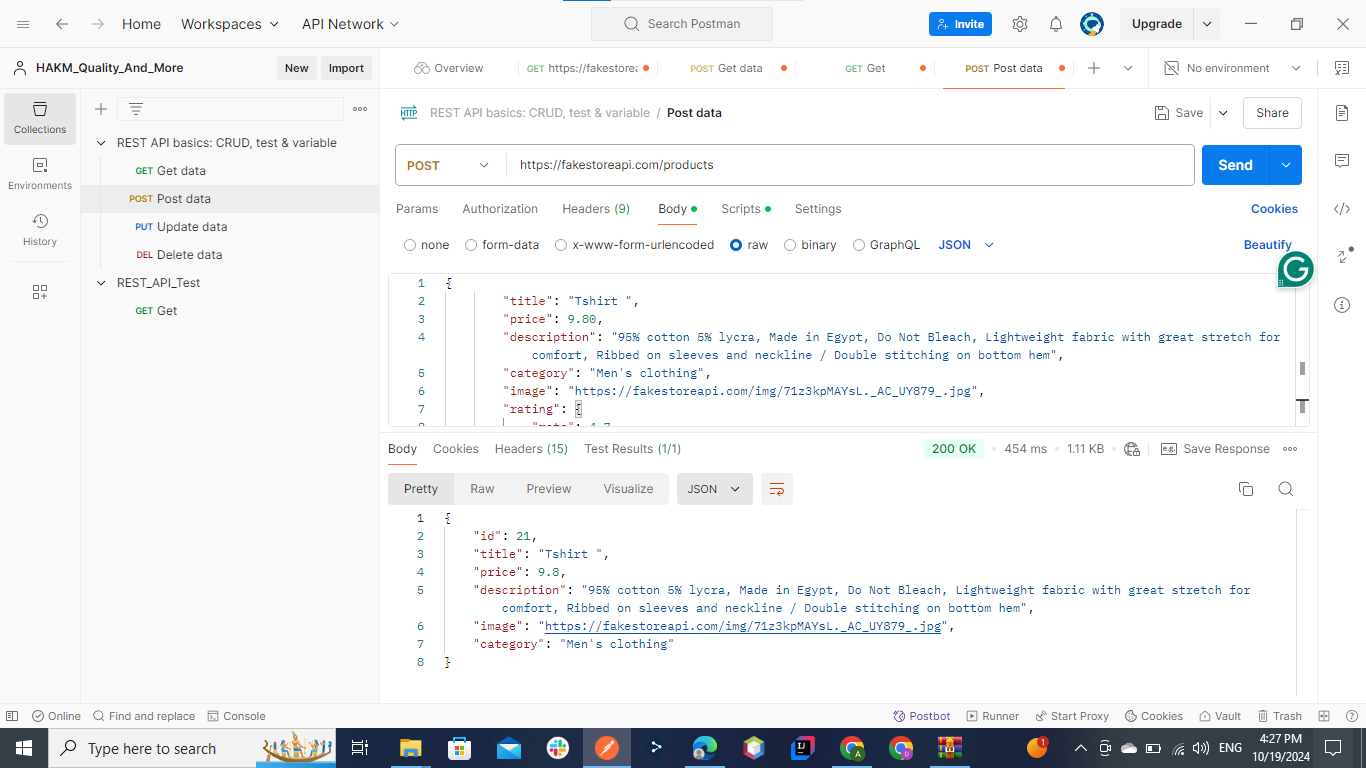
**DELETE request:**

* It shows a DELETE request being made to the URL https://fakestoreapi.com/products/1.
* In the body, we can see a product with details like title, price, description, category, and rating.
* The response status is 200 OK, which means the request was successful.
* This indicates that the API was able to successfully delete the product with ID 1.



**POST request:**

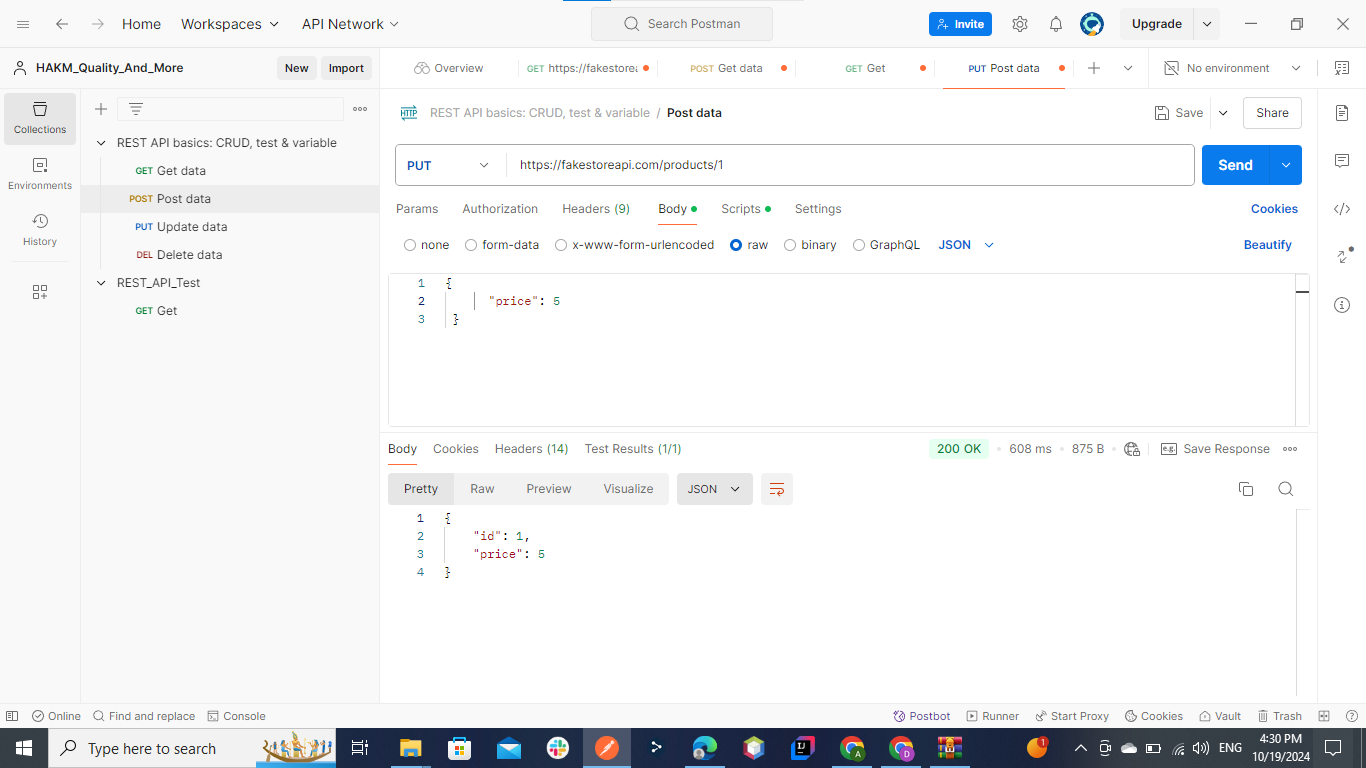
* It shows a POST request being made to the URL https://fakestoreapi.com/products.
* In the body, we can see a product with details like title, price, description, category, and rating.
* The response status is 200 OK, which means the request was successful.
* This indicates that the API was able to successfully POST the product.

****

**PUT request:**

* It shows a PUT request being made to the URL https://fakestoreapi.com/products/1.
* In the body, we can see the product price is exchanged to 5 as we need.
* The response status is 200 OK, which means the request was successful.
* This indicates that the API was able to successfully PUT the product

with ID 1.

****

**1. GET Request for All Products**

* **Test Case Name: Successful GET Request for All Products**
* **Description: Verify that a GET request to /products returns status code 200 and a list of products.**
* **Method: GET**
* **Endpoint: /products**
* **Headers:**
  + **Content-Type: application/json**
* **Assertions:**
  + **Status code is 200**
  + **Response time is less than 200ms**
  + **Response body is an array of products**
  + **Each product contains the fields: id, title, price, description, category, and rating**

**2. GET Request for Product with ID 1**

* **Test Case Name: Retrieve Product with ID 1**
* **Description: Verify that a GET request to /products/1 returns status code 200 and the correct product details.**
* **Method: GET**
* **Endpoint: /products/1**
* **Headers:**
  + **Content-Type: application/json**
* **Assertions:**
  + **Status code is 200**
  + **Response body contains the product with id = 1**
  + **Product details include id, title, price, description, category, and rating**

**3. GET Request for Product with ID 5**

* **Test Case Name: Retrieve Product with ID 5**
* **Description: Verify that a GET request to /products/5 returns status code 200 and the correct product details.**
* **Method: GET**
* **Endpoint: /products/5**
* **Headers:**
  + **Content-Type: application/json**
* **Assertions:**
  + **Status code is 200**
  + **Response body contains the product with id = 5**
  + **Product details include id, title, price, description, category, and rating**

**4. DELETE Request for Product with ID 1**

* **Test Case Name: Delete Product with ID 1**
* **Description: Verify that a DELETE request to /products/1 removes the product with ID 1 and returns a 200 status code.**
* **Method: DELETE**
* **Endpoint: /products/1**
* **Headers:**
  + **Content-Type: application/json**
* **Assertions:**
  + **Status code is 200**
  + **Response body confirms the product with id = 1 is deleted**
  + **A subsequent GET request to /products/1 returns a 404 Not Found**

**5. POST Request to Create a New Product**

* **Test Case Name: Create a New Product**
* **Description: Verify that a POST request to /products successfully creates a new product and returns status code 200.**
* **Method: POST**
* **Endpoint: /products**
* **Headers:**
  + **Content-Type: application/json**
* **Body:**

**{**

**"title": "New Product",**

**"price": 99.99,**

**"description": "This is a new product",**

**"category": "electronics",**

**"rating": {"rate": 4.7, "count": 50}**

**}**

* **Assertions:**
  + **Status code is 200**
  + **Response body contains the newly created product details**
  + **Product details include id, title, price, description, category, and rating**

# Security checks

**Security Recommendations for API:**

1. **Use HTTPS:**
   * Ensure all API communication is done over HTTPS to protect against attacks.
   * This ensures encryption in transit, so sensitive data is not exposed.
2. **Authentication and Authorization:**
   * Implement authentication mechanisms like OAuth2, JWT (JSON Web Tokens), or API keys to ensure only authorized users can access the API.
   * Role-based access control (RBAC) can be used to restrict certain actions to specific users or roles (e.g., admin, regular users).
   * Protect the API endpoints from unauthorized access by validating tokens or credentials.
3. **Input Validation:**
   * Validate all incoming parameters to protect against injection attacks such as SQL injection, NoSQL injection, or command injection.
   * Restrict the allowed data types and enforce character limits, patterns (e.g., for IDs, names), and data formats.
4. **CORS Policy:**
   * Restrict the API's CORS policy to only allow specific trusted domains to access the API.
   * Example: Only allow https://yourdomain.com to make requests to the API.
5. **Security Headers:**
   * Use HTTP security headers like Strict-Transport-Security, X-Content-Type-Options, Content-Security-Policy, and X-Frame-Options to enhance API security.

# Security test cases

**Security Test Cases:**

1. **Test for HTTPS Enforcement:**

Ensure the API is only accessible over HTTPS.

Test by making a request to http://fakestoreapi.com/products/1 and check if it gets redirected to https://.

1. **Test for Authentication Enforcement:**

Test if the API is properly secured with authentication.

Try accessing the API without an access token or credentials and expect a 401 Unauthorized error.

Attempt to use expired or invalid tokens and ensure access is denied.

# Manual test cases

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Title** | **Steps** | **Expected Results** |
| **TC-001** | **At Registration Page Enter Valid data "7 Characters" .** | **1- Open HAKM system 2- Enter the Name "Mai Ahmed" 3- Click on Submit Button** | **The placeholder is removed and  The Name is entered successfully** |
| **TC-002** | **At Registration Page Enter Valid data "25 Characters" .** | **1- Open HAKM system 2- Enter the Name " Almuntasir Billah Mohamed" 3- Click on Submit Button** | **The placeholder is removed and  The Name is entered successfully** |
| **TC-003** | **At Registration Page Enter invalid data "Spaces" .** | **1- Open HAKM system 2- Enter the Name " " 3- Click on Submit Button** | **The placeholder is removed and The Name is entered with wrong Msg "Enter Correct Your Name "** |
| **TC-004** | **At Registration Page Enter invalid data "Leave the field empty" .** | **1- Open HAKM system 2- Leave the Name "" 3- Click on Submit Button** | **The placeholder is removed and The Name is entered with wrong Msg "Your Name is Required "** |
| **TC-005** | **At Registration Page Enter invalid data "Special Character @”.** | **1- Open HAKM system 2- Leave the Name "Moh@med Ahmed" 3- Click on Submit Button** | **The placeholder is removed and The Name is entered with wrong Msg "Enter Correct Your Name "** |
| **TC-006** | **At Registration Page Enter invalid data "Email”.** | **1- Open HAKM system 2- Leave the Name "Mohamed12@gmail.com" 3- Click on Submit Button** | **The placeholder is removed and The Name is entered with wrong Msg "Enter Correct Your Name "** |
| **TC-007** | **At Registration Page Enter invalid data "Numbers" .** | **1- Open HAKM system 2- Leave the Name "1235869859699" 3- Click on Submit Button** | **The placeholder is removed and The Name is entered with wrong Msg "Enter Correct Your Name "** |
| **TC-008** | **At Registration Page Enter invalid data "paste a Name with space before the name" .** | **1- Open HAKM system 2- Leave the Name " Mai Ahmed" 3- Click on Submit Button** | **The placeholder is removed and The Name is entered with wrong Msg "Enter Correct Your Name "** |
| **TC-009** | **At Registration Page Enter invalid data "name with numbers after the name" .** | **1- Open HAKM system 2- Leave the Name "Mai Ahmed5" 3- Click on Submit Button** | **The placeholder is removed and The Name is entered with wrong Msg "Enter Correct Your Name "** |
| **TC-010** | **At Registration Page Enter invalid data "name with numbers Before the name" .** | **1- Open HAKM system 2- Leave the Name "65Mai Ahmed" 3- Click on Submit Button** | **The placeholder is removed and The Name is entered with wrong Msg "Enter Correct Your Name "** |
| **TC-011** | **At Registration Page Enter invalid data "Less than 7 characters" .** | **1- Open HAKM system 2- Leave the Name "Mi Ahmed" 3- Click on Submit Button** | **The placeholder is removed and The Name is entered with wrong Msg "Enter Correct Your Name "** |
| **TC-012** | **At Registration Page Enter invalid data "URL" .** | **1- Open HAKM system 2- Leave the Name "https://www.google.com" 3- Click on Submit Button** | **The placeholder is removed and The Name is entered with wrong Msg "Enter Correct Your Name "** |
| **TC-013** | **At Registration Page Enter invalid data "All Name in Lower Case" .** | **1- Open HAKM system 2- Leave the Name "mai ahmed" 3- Click on Submit Button** | **The placeholder is removed and The Name is entered with wrong Msg "Enter Correct Your Name "** |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Title** | **Steps** | **Expected Ruslts** |
| **TC-001** | **At Registration Page Enter Valid data "2 Numbers" .** | **1- Open HAKM system 2- Enter the Age "25" 3- Click on Submit Button** | **The placeholder is removed and  The Age is entered successfully** |
| **TC-002** | **At Registration Page Enter Valid data "1 Number" .** | **1- Open HAKM system 2- Enter the Age "5" 3- Click on Submit Button** | **The placeholder is removed and  The Age is entered successfully** |
| **TC-003** | **At Registration Page Enter invalid data "Spaces" .** | **1- Open HAKM system 2- Enter the Age " " 3- Click on Submit Button** | **The placeholder is removed and The Age is entered with wrong Msg "Enter Correct Your Age "** |
| **TC-004** | **At Registration Page Enter invalid data "Leave the field empty" .** | **1- Open HAKM system 2- Leave the Age "" 3- Click on Submit Button** | **The placeholder is removed and The Age is entered with wrong Msg "Your Age is Required"** |
| **TC-005** | **At Registration Page Enter invalid data "Special Character $" .** | **1- Open HAKM system 2- Enter the Age "$" 3- Click on Submit Button** | **The placeholder is removed and The Age is entered with wrong Msg "Enter Correct Your Age "** |
| **TC-006** | **At Registration Page Enter invalid data "Email" .** | **1- Open HAKM system 2- Enter the Age "ahmed12@gmail.com" 3- Click on Submit Button** | **The placeholder is removed and The Age is entered with wrong Msg "Enter Correct Your Age "** |
| **TC-007** | **At Registration Page Enter invalid data " Fraction Numbers" .** | **1- Open HAKM system 2- Enter the Age "25.5" 3- Click on Submit Button** | **The placeholder is removed and The Age is entered with wrong Msg "Enter Correct Your Age "** |
| **TC-008** | **At Registration Page Enter invalid data "paste a number with space before the number" .** | **1- Open HAKM system 2- Enter the Age " 25" 3- Click on Submit Button** | **The placeholder is removed and The Age is entered with wrong Msg "Enter Correct Your Age "** |
| **TC-009** | **At Registration Page Enter invalid data "name with numbers after the name" .** | **1- Open HAKM system 2- Enter the Age "ahmed25" 3- Click on Submit Button** | **The placeholder is removed and The Age is entered with wrong Msg "Enter Correct Your Age "** |
| **TC-010** | **At Registration Page Enter invalid data "0" .** | **1- Open HAKM system 2- Enter the Age "0" 3- Click on Submit Button** | **The placeholder is removed and The Age is entered with wrong Msg "Enter Correct Your Age "** |
| **TC-011** | **At Registration Page Enter invalid data "120" .** | **1- Open HAKM system 2- Enter the Age "120" 3- Click on Submit Button** | **The placeholder is removed and The Age is entered with wrong Msg "Enter Correct Your Age "** |
| **TC-012** | **At Registration Page Enter invalid data "URL" .** | **1- Open HAKM system 2- Enter the Age "https://www.google.com" 3- Click on Submit Button** | **The placeholder is removed and The Age is entered with wrong Msg "Enter Correct Your Age "** |

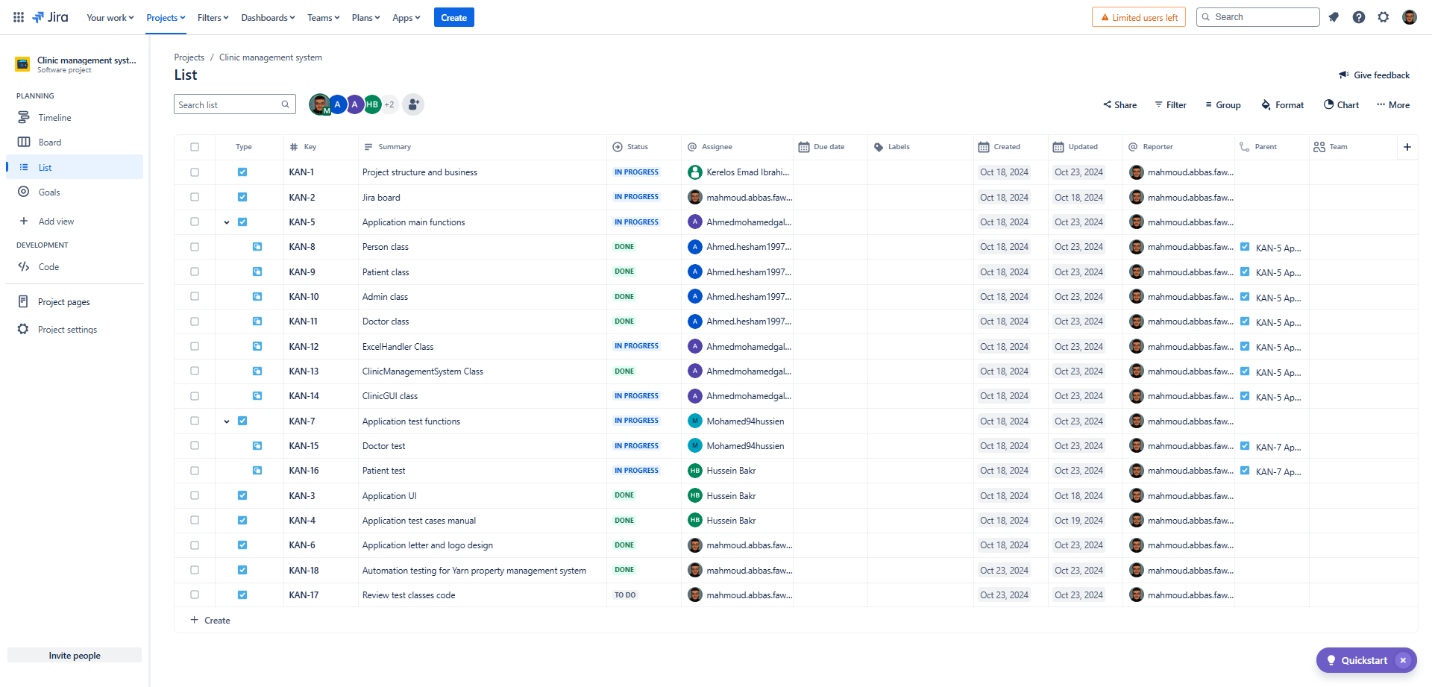
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Title** | **Steps** | **Expected Results** |
| **TC-001** | **At Registration Page Enter Valid data "Male" .** | **1- Open HAKM system 2- Click on Dropdown Menu Gender and choose (Male) 7-Click on Submit button** | **The placeholder is removed and The Gender Field is entered successfully** |
| **TC-002** | **At Registration Page Enter Valid data "Female" .** | **1- Open HAKM system 2- Click on Dropdown Menu Gender and choose (Female) 7-Click on Submit button** | **The placeholder is removed and The Gender Field is entered successfully** |
| **TC-003** | **At Registration Page Enter invalid data "Leave it empty" .** | **1- Open HAKM system 2- Leave age empty 7-Click on Submit button** | **The placeholder isn't removed and The Gender Field show error msg saying "Please, Gender is Required"** |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Title** | **Steps** | **Expected Ruslts** |
| **TC-001** | **At Registration Page Enter Valid data "14 characters" .** | **1- Open HAKM system 2- Enter the Disease "multiple sclerosis" 3- Click on Submit Button** | **The placeholder is removed and  The Disease is entered successfully** |
| **TC-002** | **At Registration Page Enter Valid data "3 characters".** | **1- Open HAKM system 2- Enter the Disease "flu" 3- Click on Submit Button** | **The placeholder is removed and  The Disease is entered successfully** |
| **TC-003** | **At Registration Page Enter invalid data "less than 3 characters " .** | **1- Open HAKM system 2- Enter the Disease "fl" 3- Click on Submit Button** | **The placeholder is removed and The Disease is entered with wrong Msg "Enter Correct Your Disease "** |
| **TC-004** | **At Registration Page Enter invalid data "Leave empty" .** | **1- Open HAKM system 2- Leave the Disease empty "" 3- Click on Submit Button** | **The placeholder is removed and The Disease is entered with wrong Msg "Enter Correct Your Disease "** |
| **TC-005** | **At Registration Page Enter invalid data "More than 14 " .** | **1- Open HAKM system 2- Enter the Disease "cold flue and headache" 3- Click on Submit Button** | **The placeholder is removed and The Disease is entered with wrong Msg "Enter Correct Your Disease "** |
| **TC-006** | **At Registration Page Enter invalid data "Spaces" .** | **1- Open HAKM system 2- Enter the Disease " " 3- Click on Submit Button** | **The placeholder is removed and The Disease is entered with wrong Msg "Enter Correct Your Disease "** |
| **TC-007** | **At Registration Page Enter invalid data "Special Character @" .** | **1- Open HAKM system 2- Enter the Disease "he@dche" 3- Click on Submit Button** | **The placeholder is removed and The Disease is entered with wrong Msg "Enter Correct Your Disease "** |
| **TC-008** | **At Registration Page Enter invalid data "URL" .** | **1- Open HAKM system 2- Enter the Disease "https://www.google.com" 3- Click on Submit Button** | **The placeholder is removed and The Disease is entered with wrong Msg "Enter Correct Your Disease "** |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Title** | **Steps** | **Expected Results** |
| **TC-001** | **At Registration Page Enter Valid data "Dr. smith" .** | **1- Open HAKM system 2- Click on Dropdown Menu Doctor and choose (Dr. Smith) 7-Click on Submit button** | **The placeholder is removed and The Doctor Field is entered successfully** |
| **TC-002** | **At Registration Page Enter Valid data "Dr. Rania" .** | **1- Open HAKM system 2- Click on Dropdown Menu Doctor and choose (Dr. Rania) 7-Click on Submit button** | **The placeholder is removed and The Doctor Field is entered successfully** |
| **TC-003** | **At Registration Page Enter invalid data "Leave it empty" .** | **1- Open HAKM system 2- Leave age empty 7-Click on Submit button** | **The placeholder isn't removed and The Doctor Field show error msg saying "Please, Doctor is Required"** |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Title** | **Steps** | **Expected Results** |
| **TC-001** | **At Registration Page Enter Valid data "9AM- 10AM" .** | **1- Open HAKM system 2- Click on Dropdown Menu Slot and choose (9AM- 10AM) 7-Click on Submit button** | **The placeholder is removed and The Slot Field is entered successfully** |
| **TC-002** | **At Registration Page Enter Valid data "9PM- 10PM" .** | **1- Open HAKM system 2- Click on Dropdown Menu Slot and choose (9PM- 10PM) 7-Click on Submit button** | **The placeholder is removed and The Slot Field is entered successfully** |
| **TC-003** | **At Registration Page Enter invalid data "Leave it empty" .** | **1- Open HAKM system 2- Leave the Dropdown Menu Slot  7-Click on Submit button** | **The placeholder isn't removed and The Slot Field show error msg saying "Please, Slot is Required"** |
| **TC-004** | **At Registration Page Enter invalid data "10PM-11PM" .** | **1- Open HAKM system 2- Click on Dropdown Menu Slot and choose (10PM-11PM) 7-Click on Submit button** | **The placeholder isn't removed and The Slot Field show error msg saying "Please, Slot is Required"** |
| **TC-005** | **At Registration Page Enter invalid data "6AM-12PM" .** | **1- Open HAKM system 2- Click on Dropdown Menu Slot and choose (6AM-12PM) 7-Click on Submit button** | **The placeholder isn't removed and The Slot Field show error msg saying "Please, Slot is Required"** |

# Collaboration and tasks



# Source code for java project and selenium project

1. <https://github.com/MahmoudAFawzy/DEPIProject>
2. <https://github.com/MahmoudAFawzy/TenantPortal>