

Test Case Development

This report details the test cases designed to validate the functionality, and reliability of the QA challenge. These test cases have been developed following best quality assurance practices to ensure comprehensive coverage of the defined requirements. Their purpose is to identify potential defects, ensure compliance with acceptance criteria, and provide a solid foundation for decision-making throughout the development cycle.

Frontend

Identifier: F-Login001

- Description: Validate credentials for a successful login. Correct **Username** and correct **Password**.
- Precondition: Access to **Login** page.
- Data:
 - **Username**: testuser
 - **Password**: password
- Steps:
 - Enter **Username**
 - Enter **Password**
 - Click **Login** button
- Expected result: Login is successful.

Identifier: F-Login002

- Description: Validate credentials with an incorrect **Username** and a correct **Password**.
- Precondition: Access to **Login** page.
- Data:
 - **Username**: badtestuser
 - **Password**: password
- Steps:
 - Enter **Username**
 - Enter **Password**
 - Click **Login** button
- Expected Result: Login is unsuccessful.

Identifier: F-Login003

- Description: Validate credentials with a correct **Username** and an incorrect **Password**.
- Precondition: Access to **Login** page.
- Data:
 - **Username**: testuser
 - **Password**: badpassword
- Steps:
 - Enter **Username**
 - Enter **Password**
 - Click **Login** button
- Expected Result: Login is unsuccessful.

Identifier: F-Login004

- Description: Validate credentials with an incorrect **Username** and an incorrect **Password**.
- Precondition: Access to **Login** page.
- Data:
 - **Username**: badtestuser
 - **Password**: badpassword
- Steps:
 - Enter **Username**
 - Enter **Password**
 - Click **Login** button
- Expected Result: Login is unsuccessful.

Identifier: F-Login005

- Description: Validate credentials with a correct **Username** and an empty **Password**.
- Precondition: Access to **Login** page.
- Data:
 - **Username**: testuser
 - **Password**:
- Steps:
 - Enter **Username**
 - Click **Login** button
- Expected Result: Login is unsuccessful.

Identifier: F-Login006

- Description: Validate credentials with an empty **Username** and a correct **Password**.
- Precondition: Access to **Login** page.
- Data:
 - **Username**:
 - **Password**: password
- Steps:
 - Enter **Password**
 - Click **Login** button
- Expected Result: Login is unsuccessful.

Identifier: F-Login007

- Description: Validate credentials with an empty **Username** and an empty **Password**.
- Precondition: Access to **Login** page.
- Data:
 - **Username**:
 - **Password**:
- Steps:
 - Click **Login** button
- Expected Result: Login is unsuccessful.

Identifier: F-Login008

- Description: Validate credentials with an empty space in the **Username** and a correct **Password**.
- Precondition: Access to **Login** page.
- Data:
 - **Username**: "testuser "
 - **Password**: password
- Steps:
 - Enter **Username**
 - Enter **Password**
 - Click **Login** button
- Expected Result: Login is unsuccessful.

Identifier: F-Login009

- Description: Validate credentials with a correct **Username** and an empty space in the **Password**.
- Precondition: Access to **Login** page.
- Data:
 - **Username**: testuser
 - **Password**: "password "
- Steps:
 - Enter **Username**
 - Enter **Password**
 - Click **Login** button
- Expected Result: Login is unsuccessful.

Identifier: F-Login010

- Description: Validate credentials with an empty space in the **Username** and an empty space in the **Password**.
- Precondition: Access to **Login** page.
- Data:
 - **Username**: "testuser "
 - **Password**: "password "
- Steps:
 - Enter **Username**
 - Enter **Password**
 - Click **Login** button
- Expected Result: Login is unsuccessful.

Identifier: F-Login011

- Description: Validate credentials with uppercase letters in the **Username** and a correct **Password**.
- Precondition: Access to **Login** page.
- Data:
 - **Username**: TESTUSER
 - **Password**: password

- Steps:
 - Enter **Username**
 - Enter **Password**
 - Click **Login** button
- Expected Result: Login is unsuccessful.

Identifier: F-Login012

- Description: Validate credentials with a correct **Username** and with uppercase letters in the **Password**.
- Precondition: Access to **Login** page.
- Data:
 - **Username**: testuser
 - **Password**: PASSWORD
- Steps:
 - Enter **Username**
 - Enter **Password**
 - Click **Login** button
- Expected Result: Login is unsuccessful.

Identifier: F-Login013

- Description: Validate credentials with uppercase letters in the **Username** and with uppercase letters in the **Password**.
- Precondition: Access to **Login** page.
- Data:
 - **Username**: TESTUSER
 - **Password**: PASSWORD
- Steps:
 - Enter **Username**
 - Enter **Password**
 - Click **Login** button
- Expected Result: Login is unsuccessful.

Identifier: F-Login014

- Description: Validate credentials with an incorrect **Username** and an empty **Password**.
- Precondition: Access to **Login** page.
- Data:
 - **Username**: badtestuser
 - **Password**:
- Steps:
 - Enter **Username**
 - Click **Login** button
- Expected Result: Login is unsuccessful.

Identifier: F-Login015

- Description: Validate credentials with an empty **Username** and an incorrect **Password**.
- Precondition: Access to **Login** page.
- Data:
 - **Username**:
 - **Password**: badpassword
- Steps:
 - Enter **Password**
 - Click **Login** button
- Expected Result: Login is unsuccessful.

Identifier: F- Login016

- Description: Validate credentials with special character letters in the **Username** and a correct **Password**.
- Precondition: Access to **Login** page.
- Data:
 - **Username**: %testuser*
 - **Password**: password
- Steps:
 - Enter **Username**
 - Enter **Password**
 - Click **Login** button
- Expected Result: Login is unsuccessful.

Identifier: F-Login017

- Description: Validate credentials with a correct **Username** and with special character letters in the **Password**.
- Precondition: Access to **Login** page.
- Data:
 - **Username**: testuser
 - **Password**: %password%
- Steps:
 - Enter **Username**
 - Enter **Password**
 - Click **Login** button
- Expected Result: Login is unsuccessful.

Identifier: F-Login018

- Description: Validate credentials with special character letters in the **Username** and with special character letters in the **Password**.
- Precondition: Access to **Login** page.
- Data:
 - **Username**: %testuser/
 - **Password**: %password%
- Steps:
 - Enter **Username**
 - Enter **Password**

- Click **Login** button
- Expected Result: Login is unsuccessful.

Identifier: F-Dashboard001

- Description: Validate correct access to the **Dashboard** page after a successful login.
- Precondition: The user has successfully logged in.
- Data:
 - **None**
- Steps:
 - Click on the **Dashboard** option in the menu list.
- Expected Result: The link redirects to the **Dashboard** page.

Identifier: F-Dashboard002

- Description: Validate correct redirection from the **Products** page to the **Dashboard** page, after a successful login
- Precondition: The user has successfully logged in. Access to **Products** page.
- Data:
 - **None**
- Steps:
 - Click on the **Dashboard** option in the menu list.
- Expected Result: The link redirects to the **Dashboard** page from the **Products** page.

Identifier: F-Dashboard003

- Description: Validate correct redirection from the **Orders** page to the **Dashboard** page, after a successful login
- Precondition: The user has successfully logged in. Access to **Orders** page.
- Data:
 - **None**
- Steps:
 - Click on the **Dashboard** option in the menu list.
- Expected Result: The link redirects to the **Dashboard** page from the **Orders** page.

Identifier: F-Product001

- Description: Validate correct access to the **Products** page after a successful login.
- Precondition: The user has successfully logged in.
- Data:
 - **None**
- Steps:
 - Click on the **Products** option in the menu list.
- Expected Result: The link redirects to the **Products** page.

Identifier: F-Product002

- Description: Validate correct redirection from the **Dashboard** page to the **Products** page, after a successful login
- Precondition: The user has successfully logged in. Access to **Dashboard** page.
- Data:
 - **None**
- Steps:

- Click on the **Products** option in the menu list.
- Expected Result: The link redirects to the **Products** page from the **Dashboard** page.

Identifier: F-Product003

- Description: Validate correct redirection from the **Orders** page to the **Products** page, after a successful login
- Precondition: The user has successfully logged in. Access to **Orders** page.
- Data:
 - **None**
- Steps:
 - Click on the **Products** option in the menu list.
- Expected Result: The link redirects to the **Products** page from the **Orders** page.

Identifier: F-Product004

- Description: Validate that products are listed and displayed correctly on the **Products** page.
- Precondition: Products must be created using **Postman** or **Swagger**.
- Data:
 - **id: 1; name: product1; price: 10**
 - **id: 2; name: product2; price: 20**
 - **id: 3; name: product3; price: 15**
- Steps:
 - Click on the **Products** option in the menu list.
- Expected Result: The link redirects correctly to the **Products** page, and the created products are listed correctly.

Identifier: F-Product005

- Description: Validate that products are listed and displayed correctly on the **Products** page, after the products are updated from the API.
- Precondition: Products must be created and updated using **Postman** or **Swagger**.
- Data:
 - **id: 1; name: product1; price: 10 > id: 1; name: product10; price: 20**
 - **id: 2; name: product2; price: 20 > id: 2; name: product20; price: 20**
 - **id: 3; name: product3; price: 15 > id: 3; name: product30; price: 20**
- Steps:
 - Click on the **Products** option in the menu list.
- Expected Result: The link redirects correctly to the **Products** page, and the updated products are listed correctly.

Identifier: F-Product006

- Description: Validate that products are listed and displayed correctly on the **Products** page, after the products are deleted from the API.
- Precondition: Products must be created and deleted using **Postman** or **Swagger**.
- Data:
 - **None**
- Steps:
 - Click on the **Products** option in the menu list.

- Expected Result: The link redirects correctly to the **Products** page, and the deleted products aren't shown in the list.

Identifier: F-Product007

- Description: Validate that products are listed and displayed correctly on the **Products** page, after the products are created, updated and deleted at the same time from the API.
- Precondition: Products must be created, updated and deleted using **Postman** or **Swagger**.
- Data:
 - **id: 1; name: product1; price: 10**
 - **id: 2; name: product2; price: 20**
 - **id: 3; name: product3; price: 15**
 - **id: 1; name: product1; price: 10** > Delete this product
 - **id: 2; name: product2; price: 20** > **id: 2; name: product20; price: 20**
- Steps:
 - Click on the **Products** option in the menu list.
- Expected Result: The link redirects correctly to the **Products** page, and the created and updated products are listed correctly. Also, the deleted products aren't shown in the list.

Identifier: F-Order001

- Description: Validate correct access to the **Orders** page after a successful login.
- Precondition: The user has successfully logged in.
- Data:
 - **None**
- Steps:
 - Click on the **Orders** option in the menu list.
- Expected Result: The link redirects to the **Orders** page.

Identifier: F-Orders002

- Description: Validate correct redirection from the **Dashboard** page to the **Orders** page, after a successful login
- Precondition: The user has successfully logged in. Access to **Dashboard** page.
- Data:
 - **None**
- Steps:
 - Click on the **Orders** option in the menu list.
- Expected Result: The link redirects to the **Orders** page from the **Dashboard** page.

Identifier: F-Orders003

- Description: Validate correct redirection from the **Products** page to the **Orders** page, after a successful login
- Precondition: The user has successfully logged in. Access to **Products** page.
- Data:
 - **None**
- Steps:
 - Click on the **Orders** option in the menu list.

- Expected Result: The link redirects to the **Orders** page from the **Products** page.

Identifier: F-Order004

- Description: Validate that orders are created and displayed correctly on the **Orders** page.
- Precondition: Orders must be created using **Postman** or **Swagger**.
- Data:
 - **id: 1; productName: product1; quantity: 1; status: Complete**
 - **id: 2; productName: product2; quantity: 5; status: Pending**
 - **id: 3; productName: product3; quantity: 10; status: Cancelled**
- Steps:
 - Click on the **Orders** option in the menu list.
- Expected Result: The link redirects correctly to the **Orders** page, and the created orders are listed correctly.

Identifier: F-Order005

- Description: Validate that orders are updated and displayed correctly on the **Orders** page.
- Precondition: Orders must be created and updated using **Postman** or **Swagger**.
- Data:
 - **id: 1; productName: product1; quantity: 1; status: Complete > id: 1; productName: product1; quantity: 1; status: Cancelled**
 - **id: 2; productName: product2; quantity: 5; status: Pending > id: 2; productName: product2; quantity: 5; status: Complete**
 - **id: 3; productName: product3; quantity: 10; status: Cancelled > id: 3; productName: product3; quantity: 10; status: In progress**
- Steps:
 - Click on the **Orders** option in the menu list.
- Expected Result: The link redirects correctly to the **Orders** page, and the updated orders are listed correctly.

Identifier: F-Order006

- Description: Validate that orders are listed and displayed correctly on the **Orders** page, after the orders are created and updated at the same time from the API.
- Precondition: Orders must be created, updated and deleted using **Postman** or **Swagger**.
- Data:
 - **id: 1; productName: product1; quantity: 1; status: Complete**
 - **id: 2; productName: product2; quantity: 2; status: Pending > id: 2; productName: product2; quantity: 2; status: Complete**
- Steps:
 - Click on the **Orders** option in the menu list.
- Expected Result: The link redirects correctly to the **Orders** page, and the created and updated orders are listed correctly.

Identifier: F-Accessibility001

- Description: Validate the correct functionality of the website across different browsers.
- Precondition: Google Chrome, Microsoft Edge, and Opera must be installed.
- Data:
 - **None**
- Steps:
 - Access the challenge website link using **Google Chrome**.
 - Access the challenge website link using **Microsoft Edge**.
 - Access the challenge website link using **Opera**.
- Expected Result: Access to the website is successful in all browsers.

Identifier: F-Performance001

- Description: Verify the performance of the **Login** page (mobile and desktop) using the **Lighthouse** tool in **DevTools**.
- Precondition: **DevTools** must have the **Lighthouse** tool available.
- Data:
 - **None**
- Steps:
 - Access the **Login** page.
 - Open **DevTools** and navigate to the **Lighthouse** tab.
 - Select the following options:
 - Mode: Navigation
 - Device: Desktop or Mobile
 - Categories: All categories
 - Click the "*Analyze page load*" button.
 - Wait for the results to be generated.
- Expected Result: Lighthouse successfully analyzes the page and provides performance results for both mobile and desktop.

Identifier: F-Performance002

- Description: Verify the performance of the **Dashboard** page (mobile and desktop) using the **Lighthouse** tool in **DevTools**.
- Precondition: **DevTools** must have the **Lighthouse** tool available.
- Data:
 - **None**
- Steps:
 - Access the **Dashboard** page.
 - Open DevTools and navigate to the **Lighthouse** tab.
 - Select the following options:
 - Mode: Navigation
 - Device: Desktop or Mobile
 - Categories: All categories
 - Click the "*Analyze page load*" button.
 - Wait for the results to be generated.
- Expected Result: **Lighthouse** successfully analyzes the page and provides performance results for both mobile and desktop.

Identifier: F-Performance003

- Description: Verify the performance of the **Products** page (mobile and desktop) using the **Lighthouse** tool in **DevTools**.
- Precondition: **DevTools** must have the **Lighthouse** tool available.
- Data:
 - **None**
- Steps:
 - Access the **Products** page.
 - Open **DevTools** and navigate to the **Lighthouse** tab.
 - Select the following options:
 - Mode: Navigation
 - Device: Desktop or Mobile
 - Categories: All categories
 - Click the "*Analyze page load*" button.
 - Wait for the results to be generated.
- Expected Result: Lighthouse successfully analyzes the page and provides performance results for both mobile and desktop.

Identifier: F-Performance004

- Description: Verify the performance of the **Orders** page (mobile and desktop) using the **Lighthouse** tool in **DevTools**.
- Precondition: **DevTools** must have the **Lighthouse** tool available.
- Data:
 - **None**
- Steps:
 - Access the **Orders** page.
 - Open **DevTools** and navigate to the **Lighthouse** tab.
 - Select the following options:
 - Mode: Navigation
 - Device: Desktop or Mobile
 - Categories: All categories
 - Click the "*Analyze page load*" button.
 - Wait for the results to be generated.
- Expected Result: Lighthouse successfully analyzes the page and provides performance results for both mobile and desktop.

Backend

Identifier: B-Login001

- Description: Validate credentials for a successful login using **Postman**.
- Precondition: **Username** and **Password** variables are set to send the request via **Postman**.
- Data:
 - **Username**: testuser
 - **Password**: password
- Steps:
 - Open **Postman** and create a POST request.
 - Set the **Username** and **Password** in the request body.
 - Send the POST request to the login API endpoint.

- Expected Result: The API responds with a **200 OK** status code and validates the credentials successfully.

Identifier: B-Login002

- Description: Validate credentials for an unsuccessful login using **Postman**. With an incorrect **Username**.
- Precondition: **Username** and **Password** variables are set to send the request via **Postman**.
- Data:
 - **Username**: badtestuser
 - **Password**: password
- Steps:
 - Open **Postman** and create a POST request.
 - Set the **Username** and **Password** in the request body.
 - Send the POST request to the login API endpoint.
- Expected Result: The API responds with a **401 unauthorized** status code and validates the login isn't successfully.

Identifier: B-Login003

- Description: Validate credentials for an unsuccessful login using **Postman**. With an incorrect **Password**.
- Precondition: **Username** and **Password** variables are set to send the request via **Postman**.
- Data:
 - **Username**: testuser
 - **Password**: badpassword
- Steps:
 - Open **Postman** and create a POST request.
 - Set the **Username** and **Password** in the request body.
 - Send the POST request to the login API endpoint.
- Expected Result: The API responds with a **401 unauthorized** status code and validates the login isn't successfully.

Identifier: B-Login004

- Description: Validate credentials for an unsuccessful login using **Postman**. With an incorrect **Username**, and an incorrect **Password**.
- Precondition: **Username** and **Password** variables are set to send the request via **Postman**.
- Data:
 - **Username**: badtestuser
 - **Password**: badpassword
- Steps:
 - Open **Postman** and create a POST request.
 - Set the **Username** and **Password** in the request body.
 - Send the POST request to the login API endpoint.
- Expected Result: The API responds with a **401 unauthorized** status code and validates the login isn't successfully.

Identifier: B-Login005

- Description: Validate credentials for an unsuccessful login using **Postman**. With the **Username** field empty.
- Precondition: **Username** and **Password** variables are set to send the request via **Postman**.
- Data:
 - **Username**:
 - **Password**: password
- Steps:
 - Open **Postman** and create a POST request.
 - Set the **Username** and **Password** in the request body.
 - Send the POST request to the login API endpoint.
- Expected Result: The API responds with a **401 unauthorized** status code and validates the login isn't successfully.

Identifier: B-Login006

- Description: Validate credentials for an unsuccessful login using **Postman**. With the **Password** field empty.
- Precondition: **Username** and **Password** variables are set to send the request via **Postman**.
- Data:
 - **Username**: testuser
 - **Password**:
- Steps:
 - Open **Postman** and create a POST request.
 - Set the **Username** and **Password** in the request body.
 - Send the POST request to the login API endpoint.
- Expected Result: The API responds with a **401 unauthorized** status code and validates the login isn't successfully.

Identifier: B-Login007

- Description: Validate credentials for an unsuccessful login using **Postman**. With the **Username** and **Password** fields empty.
- Precondition: **Username** and **Password** variables are set to send the request via **Postman**.
- Data:
 - **Username**:
 - **Password**:
- Steps:
 - Open **Postman** and create a POST request.
 - Set the **Username** and **Password** in the request body.
 - Send the POST request to the login API endpoint.
- Expected Result: The API responds with a **401 unauthorized** status code and validates the login isn't successfully.

Identifier: B-Login008

- Description: Validate credentials for an unsuccessful login using **Postman**. With an empty space in the **Username** and a correct **Password**.
- Precondition: **Username** and **Password** variables are set to send the request via **Postman**.
- Data:
 - **Username**: "testuser "
 - **Password**: password
- Steps:
 - Open **Postman** and create a POST request.
 - Set the **Username** and **Password** in the request body.
 - Send the POST request to the login API endpoint.
- Expected Result: The API responds with a **401 unauthorized** status code and validates the login isn't successfully.

Identifier: B-Login009

- Description: Validate credentials for an unsuccessful login using **Postman**. With a correct **Username** and an empty space in the **Password**.
- Precondition: **Username** and **Password** variables are set to send the request via **Postman**.
- Data:
 - **Username**: testuser
 - **Password**: "password "
- Steps:
 - Open **Postman** and create a POST request.
 - Set the **Username** and **Password** in the request body.
 - Send the POST request to the login API endpoint.
- Expected Result: The API responds with a **401 unauthorized** status code and validates the login isn't successfully.

Identifier: B-Login010

- Description: Validate credentials for an unsuccessful login using **Postman**. With an empty space in the **Username** and an empty space in the **Password**.
- Precondition: **Username** and **Password** variables are set to send the request via **Postman**.
- Data:
 - **Username**: "testuser "
 - **Password**: "password "
- Steps:
 - Open **Postman** and create a POST request.
 - Set the **Username** and **Password** in the request body.
 - Send the POST request to the login API endpoint.
- Expected Result: The API responds with a **401 unauthorized** status code and validates the login isn't successfully.

Identifier: B-Login011

- Description: Validate credentials for an unsuccessful login using **Postman**. With uppercase letters in the **Username** and a correct **Password**.
- Precondition: **Username** and **Password** variables are set to send the request via **Postman**.
- Data:
 - **Username**: TESTUSER
 - **Password**: password
- Steps:
 - Open **Postman** and create a POST request.
 - Set the **Username** and **Password** in the request body.
 - Send the POST request to the login API endpoint.
- Expected Result: The API responds with a **401 unauthorized** status code and validates the login isn't successfully.

Identifier: B-Login012

- Description: Validate credentials for an unsuccessful login using **Postman**. With a correct **Username** and with uppercase letters in the **Password**.
- Precondition: **Username** and **Password** variables are set to send the request via **Postman**.
- Data:
 - **Username**: testuser
 - **Password**: PASSWORD
- Steps:
 - Open **Postman** and create a POST request.
 - Set the **Username** and **Password** in the request body.
 - Send the POST request to the login API endpoint.
- Expected Result: The API responds with a **401 unauthorized** status code and validates the login isn't successfully.

Identifier: B-Login013

- Description: Validate credentials for an unsuccessful login using **Postman**. With uppercase letters in the **Username** and with uppercase letters in the **Password**.
- Precondition: **Username** and **Password** variables are set to send the request via **Postman**.
- Data:
 - **Username**: TESTUSER
 - **Password**: PASSWORD
- Steps:
 - Open **Postman** and create a POST request.
 - Set the **Username** and **Password** in the request body.
 - Send the POST request to the login API endpoint.
- Expected Result: The API responds with a **401 unauthorized** status code and validates the login isn't successfully.

Identifier: B-Login014

- Description: Validate credentials for an unsuccessful login using **Postman**. With an incorrect **Username** and an empty **Password**.
- Precondition: **Username** and **Password** variables are set to send the request via **Postman**.
- Data:
 - **Username**: badtestuser
 - **Password**:
- Steps:
 - Open **Postman** and create a POST request.
 - Set the **Username** and **Password** in the request body.
 - Send the POST request to the login API endpoint.
- Expected Result: The API responds with a **401 unauthorized** status code and validates the login isn't successfully.

Identifier: B-Login015

- Description: Validate credentials for an unsuccessful login using **Postman**. With an empty **Username** and an incorrect **Password**.
- Precondition: **Username** and **Password** variables are set to send the request via **Postman**.
- Data:
 - **Username**:
 - **Password**: badpassword
- Steps:
 - Open **Postman** and create a POST request.
 - Set the **Username** and **Password** in the request body.
 - Send the POST request to the login API endpoint.
- Expected Result: The API responds with a **401 unauthorized** status code and validates the login isn't successfully.

Identifier: B-Login016

- Description: Validate credentials for an unsuccessful login using **Postman**. With special character letters in the **Username** and a correct **Password**.
- Precondition: **Username** and **Password** variables are set to send the request via **Postman**.
- Data:
 - **Username**: %testuser*
 - **Password**: password
- Steps:
 - Open **Postman** and create a POST request.
 - Set the **Username** and **Password** in the request body.
 - Send the POST request to the login API endpoint.
- Expected Result: The API responds with a **401 unauthorized** status code and validates the login isn't successfully.

Identifier: B-Login017

- Description: Validate credentials for an unsuccessful login using **Postman**. With a correct **Username** and with special character letters in the **Password**.
- Precondition: **Username** and **Password** variables are set to send the request via **Postman**.
- Data:
 - **Username**: testuser
 - **Password**: %password%
- Steps:
 - Open **Postman** and create a POST request.
 - Set the **Username** and **Password** in the request body.
 - Send the POST request to the login API endpoint.
- Expected Result: The API responds with a **401 unauthorized** status code and validates the login isn't successfully.

Identifier: B-Login018

- Description: Validate credentials for an unsuccessful login using **Postman**. With special character letters in the **Username** and with special character letters in the **Password**.
- Precondition: **Username** and **Password** variables are set to send the request via **Postman**.
- Data:
 - **Username**: %testuser/
 - **Password**: %password%
- Steps:
 - Open **Postman** and create a POST request.
 - Set the **Username** and **Password** in the request body.
 - Send the POST request to the login API endpoint.
- Expected Result: The API responds with a **401 unauthorized** status code and validates the login isn't successfully.

Identifier: B-Products001

- Description: Validate the correct functionality of the **Products** page using **Postman**. Verify that a product can be created with valid data.
- Precondition: Successful login, and variables for **id**, **name**, and **price** are configured to send the request via **Postman**.
- Data:
 - **id**: 1
 - **name**: product1
 - **price**: 10
- Steps:
 - Open **Postman** and create a POST request.
 - Set the variables **id**, **name**, and **price** in the request body.
 - Send the POST request to the API endpoint to create the product.
- Expected Result: The API responds with a **201 Created** status code.

Identifier: B-Products002

- Description: Validate the correct functionality of the **Products** page using **Postman**. Verify that a product can't be created with empty **id**.
- Precondition: Successful login, and variables for **id**, **name**, and **price** are configured to send the request via **Postman**.
- Data:
 - **id**:
 - **name**: product1
 - **price**: 10
- Steps:
 - Open **Postman** and create a POST request.
 - Set the variables **id**, **name**, and **price** in the request body.
 - Send the POST request to the API endpoint to create the product.
- Expected Result: The API responds with a **400 Bad request** status code.

Identifier: B-Products003

- Description: Validate the correct functionality of the **Products** page using **Postman**. Verify that a product can't be created with empty **name**.
- Precondition: Successful login, and variables for **id**, **name**, and **price** are configured to send the request via **Postman**.
- Data:
 - **id**: 1
 - **name**:
 - **price**: 10
- Steps:
 - Open **Postman** and create a POST request.
 - Set the variables **id**, **name**, and **price** in the request body.
 - Send the POST request to the API endpoint to create the product.
- Expected Result: The API responds with a **400 Bad request** status code.

Identifier: B-Products004

- Description: Validate the correct functionality of the **Products** page using **Postman**. Verify that a product can't be created with empty **price**.
- Precondition: Successful login, and variables for **id**, **name**, and **price** are configured to send the request via **Postman**.
- Data:
 - **id**: 2
 - **name**: product2
 - **price**:
- Steps:
 - Open **Postman** and create a POST request.
 - Set the variables **id**, **name**, and **price** in the request body.
 - Send the POST request to the API endpoint to create the product.
- Expected Result: The API responds with a **400 Bad request** status code.

Identifier: B-Products005

- Description: Validate the correct functionality of the **Products** page using **Postman**. Verify that the product list can be consulted.
- Precondition: Successful login, and variables for **id**, **name**, and **price** are configured to send the request via **Postman**. And a set of products were created successfully.
- Data:
 - **None**
- Steps:
 - Open **Postman** and create a GET request.
 - Set the request to /api/Product
- Expected Result: The API responds with a **200 OK** status code.

Identifier: B-Products006

- Description: Validate the correct functionality of the **Products** page using **Postman**. Verify that the product can be consulted by a specific **id**.
- Precondition: Successful login, and variables for **id**, **name**, and **price** are configured to send the request via **Postman**. And a set of products were created successfully.
- Data:
 - **id: 1**
- Steps:
 - Open **Postman** and create a GET request.
 - Set the request to /api/Product with a specific id.
- Expected Result: The API responds with a **200 OK** status code.

Identifier: B-Products007

- Description: Validate the correct functionality of the **Products** page using **Postman**. Verify that the product can be consulted by nonexistent **id**.
- Precondition: Successful login, and variables for **id**, **name**, and **price** are configured to send the request via **Postman**. And a set of products were created successfully.
- Data:
 - **id: 100001**
- Steps:
 - Open **Postman** and create a GET request.
 - Set the request to /api/Product with a specific nonexistent id.
- Expected Result: The API responds with a **404 Error: Not found** status code.

Identifier: B-Products008

- Description: Validate the correct functionality of the **Products** page using **Postman**. Verify that the product can be updated by **id**. Update **name** and **price**.
- Precondition: Successful login, and variables for **id**, **name**, and **price** are configured to send the request via **Postman**. And a set of products were created successfully.
- Data:
 - **id: 1**
 - **name: product1 > name: product2**
 - **price: 10 > price: 20**
- Steps:
 - Open **Postman** and create a PUT request.
 - Set the request to /api/Product with a specific id.

- Expected Result: The API responds with a **204 No Content** status code.

Identifier: B-Products009

- Description: Validate the correct functionality of the **Products** page using **Postman**. Verify that the product can be updated by **id**. Update **name**.
- Precondition: Successful login, and variables for **id**, **name**, and **price** are configured to send the request via **Postman**. And a set of products were created successfully.
- Data:
 - **id**: 1
 - **name**: product1 > **name**: product2
 - **price**: 10
- Steps:
 - Open **Postman** and create a PUT request.
 - Set the request to **/api/Product** with a specific id.
- Expected Result: The API responds with a **204 No Content** status code.

Identifier: B-Products010

- Description: Validate the correct functionality of the **Products** page using **Postman**. Verify that the product can be updated by **id**. Update **price**.
- Precondition: Successful login, and variables for **id**, **name**, and **price** are configured to send the request via **Postman**. And a set of products were created successfully.
- Data:
 - **id**: 1
 - **name**: product1
 - **price**: 10 > **price**: 20
- Steps:
 - Open **Postman** and create a PUT request.
 - Set the request to **/api/Product** with a specific id.
- Expected Result: The API responds with a **204 No Content** status code.

Identifier: B-Products011

- Description: Validate the correct functionality of the **Products** page using **Postman**. Verify that the product can be updated by **id**. Update the product with an empty **price**.
- Precondition: Successful login, and variables for **id**, **name**, and **price** are configured to send the request via **Postman**. And a set of products were created successfully.
- Data:
 - **id**: 1
 - **name**: product1
 - **price**: 10 > **price**:
- Steps:
 - Open **Postman** and create a PUT request.
 - Set the request to **/api/Product** with a specific id.
- Expected Result: The API responds with a **400 Bad request** status code.

Identifier: B-Products012

- Description: Validate the correct functionality of the **Products** page using **Postman**. Verify that the product can be updated by **id**. Update the product with an empty **name**.
- Precondition: Successful login, and variables for **id**, **name**, and **price** are configured to send the request via **Postman**. And a set of products were created successfully.
- Data:
 - **id**: 1
 - **name**: product1 > **name**:
 - **price**: 10
- Steps:
 - Open **Postman** and create a PUT request.
 - Set the request to /api/Product with a specific id.
- Expected Result: The API responds with a **400 Bad request** status code.

Identifier: B-Products013

- Description: Validate the correct functionality of the **Products** page using **Postman**. Verify that a product can be deleted by **id**. With valid **id**.
- Precondition: Successful login, and variables for **id**, **name**, and **price** are configured to send the request via **Postman**. And a set of products were created successfully.
- Data:
 - **id**: 1
- Steps:
 - Open **Postman** and create a DELETE request.
 - Set the request to /api/Product with a specific id.
- Expected Result: The API responds with a **204 No Content** status code.

Identifier: B-Products014

- Description: Validate the correct functionality of the **Products** page using **Postman**. Verify that a product can't be deleted by **id**. With a nonexistent **id**.
- Precondition: Successful login, and variables for **id**, **name**, and **price** are configured to send the request via **Postman**. And a set of products were created successfully.
- Data:
 - **id**: 100001
- Steps:
 - Open **Postman** and create a DELETE request.
 - Set the request to /api/Product with a specific id.
- Expected Result: The API responds with a **404 Error: Not found** status code.

Identifier: B-Orders001

- Description: Validate the correct functionality of the **Orders** page using **Postman**. Verify that the order list can be consulted.
- Precondition: Successful login, and variables for **id**, **productName**, **quantity** and **status** are configured to send the request via **Postman**. And a set of orders were created successfully.
- Data:
 - **None**
- Steps:

- Open **Postman** and create a GET request.
 - Set the request to `/api/Order`
- Expected Result: The API responds with a **200 OK** status code.

Identifier: B-Orders002

- Description: Validate the correct functionality of the **Orders** page using **Postman**. Verify that the order can be consulted by a specific **id**.
- Precondition: Successful login, and variables for **id**, **productName**, **quantity** and **status** are configured to send the request via **Postman**. And a set of orders were created successfully.
- Data:
 - **id**: 1
- Steps:
 - Open **Postman** and create a GET request.
 - Set the request to `/api/Order`
- Expected Result: The API responds with a **200 OK** status code.

Identifier: B-Orders003

- Description: Validate the correct functionality of the **Products** page using **Postman**. Verify that the order can be consulted by a nonexistent **id**.
- Precondition: Successful login, and variables for **id**, **productName**, **quantity** and **status** are configured to send the request via **Postman**. And a set of orders were created successfully.
- Data:
 - **id**: 100001
- Steps:
 - Open **Postman** and create a GET request.
 - Set the request to `/api/Order`
- Expected Result: The API responds with a **404 Error: Not found** status code.

Identifier: B-Orders004

- Description: Validate the correct functionality of the **Orders** page using **Postman**. Verify that the order can be created with valid data.
- Precondition: Successful login, and variables for **id**, **productName**, **quantity** and **status** are configured to send the request via **Postman**.
- Data:
 - **id**: 1
 - **productName**: product1
 - **quantity**: 1
 - **status**: Complete
- Steps:
 - Open **Postman** and create a POST request.
 - Set the variables **id**, **productName**, **quantity** and **status** in the request body.
 - Send the POST request to the API endpoint to create the order.
- Expected Result: The API responds with a **201 Created** status code.

Identifier: B-Orders005

- Description: Validate the correct functionality of the **Orders** page using **Postman**. Verify that the order can't be created with empty **id**.
- Precondition: Successful login, and variables for **id**, **productName**, **quantity** and **status** are configured to send the request via **Postman**.
- Data:
 - **id**:
 - **productName**: product1
 - **quantity**: 1
 - **status**: Complete
- Steps:
 - Open **Postman** and create a POST request.
 - Set the variables **id**, **productName**, **quantity** and **status** in the request body.
 - Send the POST request to the API endpoint to create the order.
- Expected Result: The API responds with a **400 Bad request** status code.

Identifier: B-Orders006

- Description: Validate the correct functionality of the **Orders** page using **Postman**. Verify that the order can't be created with empty **productName**.
- Precondition: Successful login, and variables for **id**, **productName**, **quantity** and **status** are configured to send the request via **Postman**.
- Data:
 - **id**: 1
 - **productName**:
 - **quantity**: 1
 - **status**: Complete
- Steps:
 - Open **Postman** and create a POST request.
 - Set the variables **id**, **productName**, **quantity** and **status** in the request body.
 - Send the POST request to the API endpoint to create the order.
- Expected Result: The API responds with a **400 Bad request** status code.

Identifier: B-Orders007

- Description: Validate the correct functionality of the **Orders** page using **Postman**. Verify that the order can't be created with empty **quantity**.
- Precondition: Successful login, and variables for **id**, **productName**, **quantity** and **status** are configured to send the request via **Postman**.
- Data:
 - **id**: 1
 - **productName**: product1
 - **quantity**:
 - **status**: Complete
- Steps:
 - Open **Postman** and create a POST request.
 - Set the variables **id**, **productName**, **quantity** and **status** in the request body.
 - Send the POST request to the API endpoint to create the order.
- Expected Result: The API responds with a **400 Bad request** status code.

Identifier: B-Orders008

- Description: Validate the correct functionality of the **Orders** page using **Postman**. Verify that the order can be updated by **id**. Update **status**.
- Precondition: Successful login, and variables for **id**, **productName**, **quantity** and **status** are configured to send the request via **Postman**. And a set of orders were created successfully.
- Data:
 - **id**: 1
 - **productName**: product1
 - **quantity**: 1
 - **status**: Complete > Refunded
- Steps:
 - Open **Postman** and create a PUT request.
 - Set the request to **/api/Order** with a specific **id**.
- Expected Result: The API responds with a **204 No Content** status code.

Identifier: B-Orders009

- Description: Validate the correct functionality of the **Orders** page using **Postman**. Verify that the product can be updated by **id**. Update the product with an empty **id**.
- Precondition: Successful login, and variables for **id**, **productName**, **quantity** and **status** are configured to send the request via **Postman**. And a set of orders were created successfully.
- Data:
 - **id**: 1 > **id**:
 - **productName**: product1
 - **quantity**: 1
 - **status**: Complete
- Steps:
 - Open **Postman** and create a PUT request.
 - Set the request to **/api/Order** with a specific **id**.
- Expected Result: The API responds with a **400 Bad request** status code.

Identifier: B-Orders010

- Description: Validate the correct functionality of the **Orders** page using **Postman**. Verify that the product can be updated by **id**. Update the product with an empty **productName**.
- Precondition: Successful login, and variables for **id**, **productName**, **quantity** and **status** are configured to send the request via **Postman**. And a set of orders were created successfully.
- Data:
 - **id**: 1
 - **productName**: product1 > **productName**:
 - **quantity**: 1
 - **status**: Complete
- Steps:
 - Open **Postman** and create a PUT request.
 - Set the request to **/api/Order** with a specific **id**.
- Expected Result: The API responds with a **400 Bad request** status code.

Identifier: B-Orders011

- Description: Validate the correct functionality of the **Orders** page using **Postman**. Verify that the product can be updated by **id**. Update the product with an empty **quantity**.
- Precondition: Successful login, and variables for **id**, **productName**, **quantity** and **status** are configured to send the request via **Postman**. And a set of orders were created successfully.
- Data:
 - **id**: 1
 - **productName**: product 1
 - **quantity**: 1 > **quantity**:
 - **status**: Complete
- Steps:
 - Open **Postman** and create a PUT request.
 - Set the request to /api/Order with a specific id.
- Expected Result: The API responds with a **400 Bad request** status code.

Identifier: B-Orders012

- Description: Validate the correct functionality of the **Orders** page using **Postman**. Verify that the product can be updated by **id**. Update the product with an empty **status**.
- Precondition: Successful login, and variables for **id**, **productName**, **quantity** and **status** are configured to send the request via **Postman**. And a set of orders were created successfully.
- Data:
 - **id**: 1
 - **productName**: product 1
 - **quantity**: 1
 - **status**: Complete > **status**:
- Steps:
 - Open **Postman** and create a PUT request.
 - Set the request to /api/Order with a specific id.
- Expected Result: The API responds with a **400 Bad request** status code.

Identifier: B-Orders013

- Description: Validate the correct functionality of the **Orders** page using **Postman**. Verify that an order can be deleted by **id**. With valid **id**.
- Precondition: Successful login, and variables for **id**, **productName**, **quantity** and **status** are configured to send the request via **Postman**. And a set of orders were created successfully.
- Data:
 - **id**: 1
- Steps:
 - Open **Postman** and create a DELETE request.
 - Set the request to /api/Order with a specific id.
- Expected Result: The API responds with a **204 No Content** status code.

Identifier: B-Orders014

- Description: Validate the correct functionality of the **Orders** page using **Postman**. Verify that an order can't be deleted by **id**. With a nonexistent **id**.
- Precondition: Successful login, and variables for **id**, **productName**, **quantity** and **status** are configured to send the request via **Postman**. And a set of orders were created successfully.
- Data:
 - **id**: 100001
- Steps:
 - Open **Postman** and create a DELETE request.
 - Set the request to **/api/Order** with a specific **id**.
- Expected Result: The API responds with a **404 Error: Not found** status code.

Identifier: B-Orders015

- Description: Validate the correct functionality of the **Orders** page using **Postman**. Verify that an order can't be deleted by **id**. With a nonexistent **id**.
- Precondition: Successful login, and variables for **id**, **productName**, **quantity** and **status** are configured to send the request via **Postman**. And a set of orders were created successfully.
- Data:
 - **id**: 100001
- Steps:
 - Open **Postman** and create a DELETE request.
 - Set the request to **/api/Order** with a specific **id**.
- Expected Result: The API responds with a **404 Error: Not found** status code.