PART 2 QA CHALLENGE: Andrés Vergara

TEST CASE DEVELOPMENT

Backend

1. User Authentication

Test cases:

1.1. Verify user login with valid credentials

- **Objective:** Ensure users can log in successfully with correct credentials.
- **Steps:**
    1. Submit a valid **username and password**.
    2. Verify the response contains a valid authentication token.
    3. Check that the user is redirected to the dashboard or homepage.
- **Expected Result:** Successful login and user redirected correctly.

1.2. Verify error message for invalid credentials

- **Objective: Ensure proper error message is returned when login credentials are invalid.**
- **Steps:**
    1. **Submit an invalid username and/or password.**
    2. **Verify the response includes an appropriate error message (e.g., "Invalid username or password").**
- **Expected Result:** Clear error message indicating invalid credentials

1.3. Test token generation and expiration

- **Objective: Verify token generation and expiration mechanisms work as expected.**
- **Steps:**
    1. Log in with valid credentials.
    2. Verify token generation.
    3. Wait for the specified token expiration time.
- **Expected Result:** Token is generated correctly and expires as expected, blocking access after expiration.

1.4. Verify login with multiple concurrent sessions

- **Objective: Ensure that a user can be logged in from multiple devices or sessions simultaneously without issues.**
- **Steps:**
    1. Log in with valid credentials on one device.
    2. Log in with the same credentials on a different device or browser.
    3. Verify that both sessions remain active and independent.

Expected Result: The user should be able to maintain multiple concurrent sessions across different devices or browsers.

1.5. Verify account lockout after multiple failed login attempts

- **Objective: Ensure that the system locks the account after a set number of failed login attempts.**
- **Steps:**
    1. Attempt to log in with invalid credentials multiple times (e.g., 5 failed attempts).
    2. Verify that the account is temporarily locked and a proper message is displayed

Expected Result: The account is locked after the specified number of failed login attempts, with an appropriate error message

1.6. Verify login with empty username

- **Objective:** Ensure that the system rejects a login attempt when the username field is empty.
- **Steps:**
    1. Leave the username field empty.
    2. Enter a valid password.
    3. Attempt to log in.
- **Expected Result:** The system should display an error message indicating that the username field cannot be empty ("Username is required").

1.7. Verify login with incorrect password

Objective: Ensure that the system rejects a login attempt when the password is incorrect.

Steps:

1. Enter a valid username.
2. Enter an incorrect password.
3. Attempt to log in.

Expected Result: The system should display an error message indicating that the password is incorrect ("Incorrect password").

1.8. Verify login with incorrect username

- **Objective:** Ensure that the system rejects a login attempt when the username is incorrect.
- **Steps:**
    1. Enter an incorrect username.
    2. Enter the correct password.

PART 2 QA CHALLENGE: Andrés Vergara

3. Attempt to log in.

- **Expected Result:** The system should display an error message indicating that the username is incorrect ("Username not found")

1.9. Verify login with both username and password empty

- **Objective:** Ensure that the system rejects a login attempt when both the username and password fields are empty.
- **Steps:**
  1. Leave both the username and password fields empty.
  2. Attempt to log in.
- **Expected Result:** The system should display an error message indicating that both fields are required ("Username and password are required").

1.10. Verify login with leading or trailing spaces in username

- **Objective:** Ensure that the system handles leading or trailing spaces in the username correctly.
- **Steps:**
  1. Enter a valid username with leading or trailing spaces (e.g., " user " or " user").
  2. Enter a valid password.
  3. Attempt to log in.
- **Expected Result:** The system should trim the spaces and log in successfully if the credentials are correct.

1.11. Verify login with leading or trailing spaces in password

- **Objective:** Ensure that the system handles leading or trailing spaces in the password correctly.
- **Steps:**
  1. Enter a valid username.
  2. Enter a valid password with leading or trailing spaces (" p assword " or " password").
  3. Attempt to log in.
- **Expected Result:** The system should trim the spaces and log in successfully if the credentials are correct.

1.12. Verify login with special characters in username and password

- **Objective:** Ensure that the system supports special characters in both usernames and passwords.
- **Steps:**
  1. Enter a username and password containing special characters (!@#$%^&*())
  2. Log in using these credentials.
- **Expected Result:** The user should be able to log in successfully if the credentials are correct.

PART 2 QA CHALLENGE: Andrés Vergara

CHICKS GOLD

2. Product Management

2.1. Create, read, update, and delete (CRUD) operations for products

- **Objective: Validate CRUD operations for managing product data.**
- **Steps:**
    1. Create a new product with valid data.
    2. Retrieve the product details.
    3. Update the product information.
    4. Delete the product.

  Expected Result: Each CRUD operation performs successfully, with data being correctly created, retrieved, updated, and deleted.

2.2. Verify error handling for invalid product data

- **Objective: Ensure the system handles invalid product data properly.**
- **Steps:**
    1. Attempt to create or update a product with invalid data (e.g., missing required fields, incorrect format).
    2. Verify the response includes an appropriate error message.
- **Expected Result: Clear error messages indicating which fields are invalid and why**

2.3. Verify product listing display

- **Objective: Ensure that all products are displayed correctly in the listing/dashboard.**
- **Steps:**
    1. Create multiple products.
    2. Access the product listing/dashboard**.**
- **Expected Result: All created products should be displayed with correct details**

2.4. Verify product update with valid data

- **Objective: Ensure that an existing product is successfully updated with valid data.**
- **Steps:**
    1. Select a product from the listing/dashboard.
    2. Edit its details (e.g., update price or stock).
    3. Save the changes.
- **Expected Result: The updated details should be reflected in the listing/dashboard.**

2.5. Verify product deletion functionality

- **Objective: Ensure that products can be successfully deleted.**

PART 2 QA CHALLENGE: Andrés Vergara

- **Steps:**
    1. Select a product from the listing/dashboard.
    2. Perform the delete action.
    3. Confirm the deletion.
- **Expected Result: The product should be removed from the listing/dashboard, and the system should display a confirmation message.**

2.6. Test user dashboard data display and refresh

- **Objective: Ensure that the user dashboard displays the correct data and refreshes properly.**
- **Steps:**
    1. Log in and access the user dashboard.
    2. Verify that the displayed data (recent orders, account details) is correct.
    3. Check for data refresh upon interaction.
- **Expected Result: Accurate data is displayed, and the dashboard refreshes when necessary.**

2.7. Verify successful order creation with valid details

- **Objective: Ensure an order is created successfully when all required details are valid.**
- **Steps:**
    1. Add valid product(s) to the cart.
    2. Provide valid user and payment information.
    3. Submit the order.
- **Expected Result: The system confirms order creation and generates a unique order ID. The order appears in the user's order history.**

2.8. Verify Correct Display of Orders in Order History

- **Objective: Ensure all orders are displayed accurately in the user's order history, including details like order ID, product names, quantities, prices, status, and timestamps.**
- **Steps:**
    1. Create multiple orders with varying details (e.g., different products, quantities, and statuses).
    2. Navigate to the user's order history page.
    3. Verify that each order is displayed with the correct details
    4. Check displayed data
- **Expected Result: All orders are displayed accurately, with no missing or incorrect details.**

Performance between modules - FRONTEND

3.0. Measure Login Page Load Time

- **Objective: Verify that the login page loads within acceptable time limits.**
- **Steps:**

PART 2 QA CHALLENGE: Andrés Vergara

1. Measure the time taken to fully load the login page under normal network conditions.
2. Repeat the test under slow (3G) and high-latency networks.

- **Expected Result: The login page should load within 2 seconds on high-speed networks and within 5 seconds on slow networks.**

3.1. Test Product Listing Page Load Time

- **Objective: Ensure the product listing page loads efficiently under normal conditions.**
- **Steps:**
    1. Navigate to the product listing page.
    2. Measure the time taken to load the page fully, including images, product details, and filters.
- **Expected Result: The product listing page should load within 3 seconds.**

3.2. Verify Performance of Time Dashboard Updates

- **Objective: Assess the dashboard's ability to handle real-time data updates without performance degradation.**
- **Steps:**
    1. Log in to the dashboard.
    2. Simulate real-time updates to the data displayed on the dashboard..
- **Expected Result: Real-time updates should reflect within 1 second without UI freezing or lag.**

3.3. Measure Redirection Speed from Login to Dashboard

- **Objective: Verify the time taken to redirect users from the login page to the dashboard upon successful authentication.**
- **Steps:**
    1. Log in with valid credentials.
    2. Measure the time taken to navigate from the login page to the fully rendered dashboard.
- **Expected Result:**

    High-speed: Redirection within 1 second.

    Moderate: Redirection within 2 seconds.

    Slow: Redirection within 5 seconds.

PART 2 QA CHALLENGE: Andrés Vergara

3.4. Evaluate Redirection Speed from Dashboard to Product Listing

- **Objective:** Ensure quick navigation from the dashboard to the product listing page.
- **Steps:**
    1. Log in and access the dashboard.
    2. Click on the "Products" section to navigate to the product listing page.
- **Expected Result:**

    High-speed: Redirection within 1 second.

    Moderate: Redirection within 2 seconds.

    Slow: Redirection within 4 seconds.

3.5. Test Redirection Speed from Product Listing to Order Creation

- **Objective:** Verify smooth redirection from the product listing page to the order creation workflow.
- **Steps:**
    1. Navigate to the product listing page.
    2. Add products to the cart and click "Checkout."
    3. Test under varying network conditions.
- **Expected Result:**
    - High-speed: Redirection within 1 second.
    - Moderate: Redirection within 2 seconds.
    - Slow: Redirection within 5 seconds.

3.6. Evaluate Redirection Speed Between Orders in the Dashboard

- **Objective:** Verify navigation performance when switching between different order views.
- **Steps:**
    1. Log in and access the dashboard.
    2. Switch between tabs or sections.
    3. Measure the time taken for redirection and data loading.
- **Expected Result:**
    - High-speed: Redirection and data loading within 1 second.
    - Moderate: Redirection and data loading within 3 seconds.
    - Slow: Redirection and data loading within 5 seconds.

3.7. Test Redirection from Dashboard to Login

- **Objective:** Ensure efficient redirection when logging out and returning to the login page.
- **Steps:**
    1. Log in to the dashboard.
    2. Measure the time taken to redirect to the login page.
    3. Verify no delay under varying network speeds.
- **Expected Result:**

PART 2 QA CHALLENGE: Andrés Vergara

- High-speed: Redirection within 1 second.
- Moderate: Redirection within 2 seconds.
- Slow: Redirection within 4 seconds

3.8. Validate Data Refresh Efficiency on Dashboard

- **Objective:** Ensure dashboard data refreshes efficiently when triggered manually.
- **Steps:**
  1. Load the dashboard.
  2. Trigger a manual data refresh.
  3. Measure the time taken for data to update.
- **Expected Result:**
  - Data refresh completed within 1 second.

3.9. Verify Login Page Load Time

- **Objective:** Ensure the login page loads within acceptable time limits.
- **Steps:**
  1. Navigate to the login page.
  2. Measure the page load time under different network conditions
- **Expected Result:**
  - High-speed: Load time ≤ 1 second.
  - Medium-speed: Load time ≤ 2 seconds.
  - Low-speed: Load time ≤ 4 seconds.

3.10.Check UI Responsiveness for Different Screen Sizes

- **Objective:** Verify UI responsiveness on devices with different resolutions.
- **Steps:**
  1. Test the application on desktop and mobile devices.
  2. Ensure layouts adjust dynamically without performance degradation.
- **Expected Result:**
  - UI adjusts instantly without delays or misaligned components.

PART 2 QA CHALLENGE: Andrés Vergara