**Assignment Task 1: Web Automation**

1. **Describe your approach to automating the login functionality of a web application.**

* Locating login page element such as username and password fields, login button or error messages.
* Write positive and negative test cases for login functionality.
* Setting up automation framework like choosing automation tool (selenium or cypress), user Page Object Model (POM) to maintain a clean and clear structure, using Pytest for test execution and reporting and maybe use Excel for data-driven testing (DDT)

1. **List your test cases for this functionality**

**Positive test cases:**

**TC\_01 - Login with valid credentials**

* Steps: Click login after entering a working username and password.
* Output: The user is taken to the dashboard.

**TC\_02 - Login with case-insensitive username (if applicable)**

* Steps: Type your username in different scenarios (such as TestUser and testuser).
* Output: If case insensitivity is permitted, the user should successfully log in.

**TC\_03 - Verify "Remember Me" functionality**

* **Steps:** Check the "Remember Me" box, log in, close and reopen the browser.
* **Expected Result:** User remains logged in.

**TC\_04 - Verify session timeout after inactivity**

* **Steps:** Log in and remain idle for a specified time (e.g., 15 minutes).
* **Expected Result:** User session should expire, and they should be redirected to the login page.

**Negative test cases:**

**TC\_05 - Login with invalid username**

* **Steps:** Enter an incorrect username and valid password, click login.
* **Expected Result:** Error message "Invalid username or password" should appear.

**TC\_06 - Login with invalid password**

* **Steps:** Enter a valid username but an incorrect password.
* **Expected Result:** User should not be logged in, and an error message should appear.

**TC\_07 - Login with empty username and password**

* **Steps:** Leave both fields blank and click login.
* **Expected Result:** Validation message should appear.

**TC\_08 - Login with empty username but a valid password**

* **Steps:** Leave the username field empty and enter a valid password.
* **Expected Result:** A username validation message should be displayed.

**TC\_09 - Login with valid username but empty password**

* **Steps:** Enter a valid username but leave the password field blank.
* **Expected Result:** A password error message should appear.

**TC\_10 - Login with multiple incorrect attempts (Account Lockout)**

* **Steps:** Enter wrong credentials multiple times (e.g., 5 attempts).
* **Expected Result:** The account should be temporarily locked.

**Assignment Task 2**

**How do you automate error message validation for required fields in a form?**

 **Identify Required Fields & Error Messages**

* Locate the form elements (input fields like username and password and submit button).
* Identify the error message locators

 **Create Page Object Model (POM) Class for the Form**

* Define locators for required fields and their corresponding error messages.
* Implement methods to interact with the form that will return the validation message

 **Write a Test Script**

* Leave required fields empty and submit the form.
* Capture the displayed error messages and compare them with expected values.

 **Use Assertions for Validation**

* Validate that error messages appear as expected using assertions.

**GitHub link for task 1 and task 2 combined:**

<https://github.com/gauravshakyaa/login-and-validation-field>

**Collection link for API testing:**  
<https://www.postman.com/winter-desert-163049/workspace/https-www-videogamedb-uk-swagger-ui/collection/25674824-15838d25-8fd3-49df-81db-e523caa5be7f?action=share&source=copy-link&creator=25674824>