

Phase I

Based on the following user story, answer the questions.

User Story

Description: As a user, I want to be able to log into the application.

Preconditions: User has an account.

Acceptance criteria:

1. When I put good login and password I should login to platform
2. When I put bad login and password I shouldn't login to platform

Question 1: Are you going to ask any additional, more precise questions during the refinement / requirement definition stage?

Yes, some of which would be:

- *A request to provide a link to the designs repo for the log-in page.*
- *What are the messages for invalid login scenarios?*
- *Can a user be simultaneously logged in with the same credentials in a different browser?*
- *Does the login session have a timeout functionality?*
- *Does the login functionality have a limited number of retries?*

Question 2: If you have received the answers to all your questions from the Product Owner / Manager / Client, create test cases to the above-mentioned user story using Gherkin.

Feature: Login functionality of desktop website [pearson.com](https://www.pearson.com) - US store

Scenario: Verify if the font, text color, and color coding of the Login page is as per the standard. (positive UI)

UI - Positive

Given: the user is on the [Sign In page](#)

Then: The *Sign In* page is displayed according to designs [insert link here].

Scenario: Verify if a user cannot login with an invalid email and a valid password.

Functional - Negative

Given: the user is on the *Sign In* page

When: the user intentionally uses an unregistered email address

And: the user uses a valid password

And: the user clicks on the *Sign in* button

Then: the message “Sorry, the credentials you have entered do not match. Please try again.” is displayed

And: the user is **not** signed into the portal

Scenario: Verify if a user cannot login with a valid email and an invalid password.

Functional - Negative

Given: the user is on the *Sign In* page

When: the user uses a registered and valid email address

And: the user uses an invalid password

And: the user clicks on the *Sign in* button

Then: the message “Sorry, the credentials you have entered do not match. Please try again.” is displayed

And: the user is **not** signed into the portal

Scenario: Verify the login page for both email and password fields, when the fields are blank.

Functional - Negative

Given: the user is on the *Sign In* page

When: the user intentionally leaves the Email and Password fields empty

And: the user clicks on the *Sign in* button

Then: the messages “Please enter a username” and “Please enter a password” are displayed for each respective field

And: the user is **not** signed into the portal

Scenario: Verify if the data in the password field is visible as bullet signs.

Security & UI - Positive

Given: the user is on the <i>Sign In page</i> When: the user inputs a password	Then: the data in the password field is visible as bullet signs
---	--

Scenario: <i>Verify if a user will be able to login with a valid username and password.</i>	<i>Functional - Positive</i>
Given: the user is on the <i>Sign In page</i> When: the user uses a registered and valid email address And: the user uses a valid password And: the user clicks on the <i>Sign in</i> button	Then: the user is successfully signed into the portal

Scenario: <i>Verify if the login page allows to log in simultaneously with the same credentials in different browsers.</i>	<i>Security - Positive</i>
Given: the user uses two different browsers And: the user is on the <i>Sign In page</i> on both browsers When: the user uses a registered and valid email address on both browsers And: the user uses a valid password on both browsers And: the user clicks on the <i>Sign in</i> button on both browsers	Then: the user is successfully signed into the portal on both browsers

Scenario: <i>Verify the timeout functionality of the login session.</i>	<i>Security - Positive</i>
Given: the user is signed in When: the user does not perform any action for 5 minutes	Then: the user is automatically logged out after 5 minutes of inactivity

Scenario: <i>Verify that once logged out, pressing the 'Back button' of the browser does not re-login the user.</i>	<i>Security - Positive</i>
Given: the user is signed in When: the user logs out And: the user clicks the 'Back' button on the browser	Then: the user is redirected to the Sign In page And: the user is not signed in.

note: The above test examples are only a sample of a test suite for the log-in functionality

Phase II

Task 1: Solutions to questions from the previous phase and the two tasks from this phase place in the code repository as a file (*.feature, *.doc, *.txt, *.pdf).

I attached this .pdf you're reading now to the git repo ;)

Task 2: Using TypeScript or JavaScript create test scenarios for the search functionality on *pearson.com* website and implement them using selenium webdriver.

Requirements:

- Search input should be visible on the main page.
- First page with the search result should have at most 10 elements.
- Clicking on the Next button should display the next page with 10 more results.
- The third searched element should redirect to the proper article.

Use selenium webdriver, and known design patterns. The tests must be executable in any environment. Implemented code must be delivered on an outside repository, for example GitHub.

I've never set up my own project before so this was a new fun experience for me. I followed some JS-Selenium tutorials to set it up and I started testing the pearson.com website.

The environment setup took me about 1-2 hours with a lot of troubleshooting and learning a few new things. I did most of the test in about 2 hours and then I got stuck at the last step hitting a wall finding the .pdf that's opened in a new tab. Finally after I figured it out and all tests passed, I cleaned it up a bit and uploaded it to this repo.

The test plan has the following tests:

- Load page
- Accept cookies
- Find the search box by id
- Enter keywords and click enter
- Close the browser after running tests
- Wait the total of search results and count them
- Scroll to the Next button and click it
- Scroll to the third article on page and open it
- Verify article open