

19-Feb-2023

project submission date

**POSTGRADUATE STUDIES „Software tester”**

**WSB University in Poznań**

MINI PROJECT - AUTOMATION OF TESTS FOR WEB APPLICATION USING ROBOT FRAMEWORK

Project goal: Automation of test cases for the Bookstore Application web application

Checked by:

mgr inż. Michał Szajkowski

Prepared by:

Anna Górecka

The name of the tested application: **Book Store Application**  
(<https://demoqa.com/books>)

**Environment:**

Computer OS: Windows 11

Chrome: 110.0.5481.104 (64-bit)

Pycharm: 17.0.5

Python: 3.11.1

Robot Framework: 6.0.2

## Test cases:

**ID:**001

**Test name:** Validate the password when a new user is registered

### Prerequisites:

1. The application is running.
2. User isn't logged in.
3. Password requirement: Passwords must have at least one non alphanumeric character, one digit ('0'-'9'), one uppercase ('A'-'Z'), one lowercase ('a'-'z'), one special character and Password must be eight characters or longer.

Lp.	Test step	Expected result
1.	Check if page is loaded	Address <a href="https://demoqa.com/books">https://demoqa.com/books</a> is visible in the browser bar and the <i>Book Store</i> title is visible on the page.
2.	Click on the <i>Login</i> button	After clicking the <i>Login</i> button the page with the <i>Login</i> title is visible.
3.	Click on the <i>New User</i> button	After clicking the <i>New User</i> button the page with the <i>Register</i> title is visible.
4.	Populate <i>First Name</i> , <i>Last Name</i> , <i>UserName</i> and <i>Password</i> in the <i>Register</i> page.	<i>First Name</i> , <i>Last Name</i> , <i>UserName</i> and <i>Password</i> were populated in the <i>Register</i> page.
5.	Manually click on <i>I'm not a robot</i> checkbox. Perform additional verification as per pop-up, if required.	CAPTCHA verification completed successfully.
6.	Click on the <i>Register</i> button.	Error message is shown at the bottom of the <i>Register</i> page: <i>Passwords must have at least one non alphanumeric character, one digit ('0'-'9'), one uppercase ('A'-'Z'), one lowercase ('a'-'z'), one special character and Password must be eight characters or longer.</i>

Test is considered Successful when application prevents user from registering when password does not meet the requirements.

**ID:002**

**Test name:** Logging in the registered user

**Prerequisites:**

1. The application is running.
2. Registered user exists in the application; UserName = aabb; Password = M3cybefidy!.
3. User isn't logged in.

Lp.	Test step	Expected result
1.	Check if page is loaded	Address <a href="https://demoqa.com/books">https://demoqa.com/books</a> is visible in the browser bar and the <i>Book Store</i> title is visible on the page.
2.	Click on the <i>Login</i> button	After clicking the <i>Login</i> button the page with the <i>Login</i> title is visible.
3.	Populate <i>UserName</i> and <i>Password</i> in the <i>Register</i> page.	<i>UserName</i> and <i>Password</i> were populated in the <i>Register</i> page.
4.	Click on the <i>Login</i> button	After clicking the <i>Login</i> button the <i>Book Store</i> page is open with the user logged in. <i>User Name</i> is visible on the page.

Test is considered Successful when the user is logged into the application.

**ID:003**

**Test name:** Adding book to the registered user collection. Verification that the user cannot add two times the same book to the collection.

**Prerequisites:**

1. The application is running.
2. Registered user exists in the application; UserName = aabb; Password = M3cybefidy!.
3. User isn't logged in.
4. User doesn't have a *Git Pocket Guide* book in his collection.

Lp.	Test step	Expected result
1.	Check if page is loaded	Address <a href="https://demoqa.com/books">https://demoqa.com/books</a> is visible in the browser bar and the <i>Book Store</i> title is visible on the page.
2.	Click on the <i>Login</i> button	After clicking the <i>Login</i> button the page with the <i>Login</i> title is visible.
3.	Populate <i>UserName</i> and <i>Password</i> in the <i>Register</i> page.	<i>UserName</i> and <i>Password</i> were populated in the <i>Register</i> page.
4.	Click on the <i>Login</i> button	After clicking the <i>Login</i> button the <i>Book Store</i> page is open with the user logged in. <i>User Name</i> is visible on the page.
5.	Click on the <i>Git Pocket Guide</i> link.	After clicking the <i>Git Pocket Guide</i> link the <i>Book Store</i> page is open with the information about the <i>Git Pocket Guide</i> book.
6.	Click on the <i>Add To Your Collection</i> button.	Pop up with a text <i>Book added to your collection</i> is visible to the user.
7.	Click <i>OK</i> on the pop up. Click second time on the <i>Add To Your Collection</i> button.	Pop up with a text <i>Book already present in the your collection!</i> is visible to the user.

Test is considered Successful when the user is able to add the book to the collection and when is not allowed to add the second time the same book to the collection.

**ID:004**

**Test name:** Verification of book sorting functionality.

**Prerequisites:**

1. The application is running.

Lp.	Test step	Expected result
1.	Check if page is loaded	Address <a href="https://demoqa.com/books">https://demoqa.com/books</a> is visible in the browser bar and the <i>Book Store</i> title is visible on the page.
2.	Click on column header with <i>Title</i> name	Books titles are sorted in ascending order by name.

Test is considered Successful when books titles are sorted in ascending order by name.

## Attachment 1.

Code for test case ID:001

```
*** Settings ***

Library           SeleniumLibrary

Library           Dialogs           #Enable manual step during automated test execution

Test Setup       Open Webpage

Test Teardown    Close Webpage

Suite Setup      Set Selenium Speed    0.5    #Changing the speed of test execution

*** Variables ***

${login_button}    xpath://*[@id="login"]

${new_user_button}    xpath://*[@id="newUser"]

${first_name_input}    xpath://*[@id="firstname"]

${last_name_input}    xpath://*[@id="lastname"]

${user_name_input}    xpath://*[@id="userName"]

${password_input}    xpath://*[@id="password"]

${register}    xpath://*[@id="register"]
```

```

${error}                                xpath://*[@id="name"]

${expected_error_text}    Passwords must have at least one non alphanumeric character, one digit ('0'-'9'), one
uppercase ('A'-'Z'), one lowercase ('a'-'z'), one special character and Password must be eight characters or longer.

${login_info}              Login

${register_info}           Register

```

\*\*\* Test Cases \*\*\*

Validate the password when a new user is registered

Check If Page Is Loaded

Go To Login Page

Go To New User Page

Fill In First Name

Fill In Last Name

Fill In User Name

Fill In Password

Check I Am Not A Robot

Click Register

Check Expected Result



\*\*\* Keywords \*\*\*

Open Webpage

Open Browser                      *https://demoqa.com/books*                      *chrome*

Maximize Browser Window

Check If Page Is Loaded

Wait Until Page Contains      *Book Store*

Check If Page Is Loaded With Argument \${title}

Wait Until Page Contains      \${title}

Go To Login Page

Click Button                      \${login\_button}

Go To New User Page

Check If Page Is Loaded With Argument \${login\_info}

Click Button                      \${new\_user\_button}

Fill In First Name

Click Element      `{register_info}`

Click Element      `{first_name_input}`

Input Text      `{first_name_input}`      *Name*

Fill In Last Name

Click Element      `{last_name_input}`

Input Text      `{last_name_input}`      *Last*

Fill In User Name

Click Element      `{user_name_input}`

Input Text      `{user_name_input}`      *User*

Fill In Password

Click Element      `{password_input}`

Input Text      `{password_input}`      *123*

Check I am not a robot

Pause Execution    *Enter captcha*

Sleep    *2s*

Click Register

Click Element    *\${register}*

Sleep    *2s*

Check Expected Result

*\${output}* = Get WebElement    *\${error}*

Log To Console    *\${output.text}*

Should Match    *\${output.text}*    *\${expected\_error\_text}*

Close Webpage

Close Browser

## Attachment 1.A

Report from executed test case ID:001.

Test completed successfully.

## Id001 Report

Generated 20230219 21:34:52 UTC+01:00  
50 seconds ago

LOG

### Summary Information

Status: All tests passed

Start Time: 20230219 21:34:08.856

End Time: 20230219 21:34:52.928

Elapsed Time: 00:00:44.072

Log File: [log.html](#)

### Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	1	1	0	0	00:00:44	<div></div>

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
No Tags						<div></div>

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Id001	1	1	0	0	00:00:44	<div></div>

### Test Details

All Tags Suites Search

Status: 1 test total, 1 passed, 0 failed, 0 skipped

Total Time: 00:00:43.954

Name	Documentation	Tags	Status	Message	Elapsed	Start / End
Id001: Validate the password when a new user is registered			PASS		00:00:43.954	20230219 21:34:08.972 20230219 21:34:52.926

## Attachment 2.

Code for test cases ID:002 and ID:003

```
*** Settings ***

Library           SeleniumLibrary

Test Setup        Open Webpage
```

Test Teardown      Close Webpage

Suite Setup          Set Selenium Speed      0.5      #Changing the speed of test execution

\*\*\* Variables \*\*\*

<code>\${login_button}</code>	<code>xpath://*[@id="login"]</code>
<code>\${user_name}</code>	<code>xpath://*[@id="userName"]</code>
<code>\${password}</code>	<code>xpath://*[@id="password"]</code>
<code>\${user_name_to_populate}</code>	<code>aabb</code>
<code>\${password_to_populate}</code>	<code>M3cybefidy!</code>
<code>\${user_name_value}</code>	<code>xpath://*[@id="userName-value"]</code>
<code>\${book1}</code>	<code>xpath://*[@id="see-book-Git Pocket Guide"]</code>
<code>\${add_to_your_collection_button}</code>	<code>xpath:(//*[@class="text-right fullButton"]/*[@id="addNewRecordButton"])</code>
<code>\${message_with_second_book}</code>	<code>Book already present in the your collection!</code>
<code>\${message_with_add_book}</code>	<code>Book added to your collection.</code>
<code>\${login_info}</code>	<code>Login</code>
<code>\${book_title}</code>	<code>Git Pocket Guide</code>

```
*** Test Cases ***
```

```
Logging in the registered user
```

```
    Check If Page Is Loaded
```

```
    Click Login Button
```

```
    Check If Page Is Loaded With Argument ${login_info}
```

```
    Fill In UserName
```

```
    Fill In Password
```

```
    Click Login Button
```

```
    Check Expected Result
```

```
Add Book Test
```

```
    Check If Page Is Loaded
```

```
    Click Login Button
```

```
    Fill In UserName
```

```
    Fill In Password
```

```
    Click Login Button
```

```
    Click On Book
```

Add Book

Add Book Second Time

\*\*\* Keywords \*\*\*

Open Webpage

Open Browser                    *https://demoqa.com/books*                    *chrome*

Maximize Browser Window

Click Login Button

Click Button        *\${login\_button}*

Check If Page Is Loaded

Wait Until Page Contains        *Book Store*

Check If Page Is Loaded With Argument *\${title}*

Wait Until Page Contains        *\${title}*

Fill In UserName

```
Check If Page Is Loaded With Argument ${login_info}
```

```
Click Element    ${user_name}
```

```
Input Text      ${user_name}    ${user_name_to_populate}
```

#### Fill In Password

```
Click Element    ${password}
```

```
Input Text      ${password}    ${password_to_populate}
```

#### Check Expected Result

```
Check If Page Is Loaded With Argument ${book_title}
```

```
${output} = Get WebElement    ${user_name_value}
```

```
Log To Console    ${output.text}
```

```
Should Match      ${output.text}    ${user_name_to_populate}
```

#### Click On Book

```
Check If Page Is Loaded With Argument ${book_title}
```

```
Click Element    ${book1}
```



#### Add Book

Check If Page Is Loaded With Argument `${book_title}`

execute javascript `window.scrollTo(0,500)`

Click Button `${add_to_your_collection_button}`

`${message}=` Handle Alert `action=accept`

Log To Console `${message}`

Should Match `${message_with_add_book}` `${message}`

#### Add Book Second Time

Check If Page Is Loaded With Argument `${book_title}`

Click Button `${add_to_your_collection_button}`

`${message}=` Handle Alert `action=accept`

Log To Console `${message}`

Should Match `${message_with_second_book}` `${message}`

#### Close Webpage

Close Browser

## Attachment 2.A

Report from executed test cases ID:002 and ID:003

Both tests were completed successfully.

### Id002 Report

Generated  
20230219 21:44:33 UTC+01:00  
28 seconds ago

LOG

#### Summary Information

**Status:** All tests passed  
**Start Time:** 20230219 21:43:42.208  
**End Time:** 20230219 21:44:33.875  
**Elapsed Time:** 00:00:51.667  
**Log File:** [log.html](#)

#### Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	2	2	0	0	00:00:52	<div></div>

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
No Tags						<div></div>

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Id002	2	2	0	0	00:00:52	<div></div>

#### Test Details

All Tags Suites Search

**Status:** 2 tests total, 2 passed, 0 failed, 0 skipped  
**Total Time:** 00:00:51.524

Name	Documentation	Tags	Status	Message	Elapsed	Start / End
Id002.Add Book Test			PASS		00:00:30.334	20230219 21:44:03.539 20230219 21:44:33.873
Id002.Logging in the registered user			PASS		00:00:21.190	20230219 21:43:42.347 20230219 21:44:03.537

## Attachment 3.

Code for test case ID:004

```
*** Settings ***

Library                SeleniumLibrary

Library                Collections    #Enabling list creation and working on the lists.

Test Setup             Open Webpage

Test Teardown          Close Webpage

Suite Setup            Set Selenium Speed    0.5    #Changing the speed of test execution


*** Variables ***

${title}               xpath: (//*[@class="rt-resizable-header-content"]) [2]

${table_title_element}  xpath: //*[@class="rt-td"]/*[@class="action-buttons"]


*** Test Cases ***

Verification of book sorting functionality

    Check If Page Is Loaded

    Click Title Column

    Check Order
```

```
*** Keywords ***
```

```
Open Webpage
```

```
    Open Browser          https://demoqa.com/books      chrome
```

```
    Maximize Browser Window
```

```
Check If Page Is Loaded
```

```
    Wait Until Page Contains  Book Store
```

```
Click Title Column
```

```
    Click Element    ${title}
```

```
Check Order
```

```
    @{list_web_elements} =      Get WebElements    ${table_title_element}
```

```
    ${list} = Create List
```

```
FOR      {book}  IN  @{list_web_elements}
```

```
    Append To List    ${list}    {book.text}
```

```
END
```

```
${sorted_list} = Copy List  ${list}

Sort List  ${sorted_list}

Log To Console      ${sorted_list}

Lists Should Be Equal      ${sorted_list}      ${list}

Close Webpage

Close Browser
```

Attachment 3.A

Report from executed test case ID:004.

Test failed.

**Defect ID:** 004-001 Sorting functionality not working properly for column *Title* on *Book Store* page.

**Environment:** Windows 11; Chrome 110.0.5481.104 (64-bit)

**Detected in:** test case ID:004; step 2

Test step	Expected result	Actual result
Click on column header with <i>Title</i> name	Books titles are sorted in ascending order by name.	Books titles <b>are not</b> sorted in ascending order by name.

Defect category: Application error

Id003 Report

Generated  
20230219 21:48:10 UTC+01:00  
7 minutes 43 seconds ago

LOG

Summary Information

Status:1 test failed

Start Time:20230219 21:47:56.607

End Time:20230219 21:48:10.302

Elapsed Time:00:00:13.695

Log File:[log.html](#)

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	1	0	1	0	00:00:14	<div></div>

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
No Tags						<div></div>

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Id003	1	0	1	0	00:00:14	<div></div>

Test Details

AllTagsSuitesSearch

Status:1 test total, 0 passed, 1 failed, 0 skipped

Total Time:00:00:13.579

Name	Documentation	Tags	Status	Message	Elapsed	Start / End
Id003: Verification of book sorting functionality			FAIL	Lists are different. Index 0: Designing Evolvable Web APIs with ASP.NET != Git Pocket Guide Index 1: Eloquent JavaScript, Second Edition != Learning JavaScript Design Patterns Index 2: Git Pocket Guide != Designing Evolvable Web APIs with ASP.NET Index 3: Learning JavaScript Design Patterns != Speaking JavaScript Index 4: Programming JavaScript Applications != You Don't Know JS Index 5: Speaking JavaScript != Programming JavaScript Applications Index 6: Understanding ECMAScript 6 != Eloquent JavaScript, Second Edition Index 7: You Don't Know JS != Understanding ECMAScript 6	00:00:13.579	20230219 21:47:56.720 20230219 21:48:10.299