Title: Alert handling

Steps:

1. Click on the “Alerts (JavaScript)” link in the “Alert” list element.
2. Click on the “Show alert box” button, in the displayed alert click on the “Ok” button.

Expected results:

1. Page opened.
2. Alert displayed and had text ”I am an alert box!”.

Title: Confirm handling

Steps:

1. Click on the “Alerts (JavaScript)” link in the “Alert” list element.
2. Click on the “Show confirm box” button, in the displayed alert click on the “Cancel” button.

Expected results:

1. Page opened.
2. The text “False You clicked Cancel, confirm returned false.” appears under the button.

Title: Prompt handling

Steps:

1. Click on the “Alerts (JavaScript)” link in the “Alert” list element.
2. Click on the “Show prompt box” button, In the displayed prompt type random text and click on the “Ok” button.

Expected results:

1. Page opened.
2. The text “{Your text} You clicked OK. 'prompt' returned {Your text}” appears under the button.

Title: Handling and interaction with fake alerts

Steps:

1. Click on the “Fake Alerts” link in the “Alert” list element.
2. Click on the “Show alert box” button.
3. In the displayed modal dialog click on the “Ok” button.

Expected results:

1. Page opened.
2. Fake alert displayed.
3. Fake alert closed.

Title: Handling and interaction with modal dialog

Steps:

1. Click on the “Fake Alerts” link in the “Alert” list element.
2. Click on the “Show modal dialog” button.
3. In the displayed modal dialog click on background.

Expected results:

1. Page opened.
2. Modal dialog displayed.
3. Modal dialog closed.

Title: Work with ’ iframe’ elements

Steps:

1. Click on the “iFrames test page” link in the “Frames” list element.
2. Inside “iFrame Example List” frame get id of element with text “iFrame List Item 40”.

Expected results:

1. Page opened.
2. Element id equal to “`iframe40`”.

Title: Work with links inside ‘iframe’ elements

Steps:

1. Click on the “iFrames test page” link in the “Frames” list element.
2. Inside the “iFrame Example Header” frame click on “Index” link.

Expected results:

1. Page opened.
2. Main page opened.

Title: JavaScript events triggering

Steps:

1. Click on the “Events (JavaScript)” link inside the “JavaScript” list element.
2. Focus on the “On blur” button, then blur.
3. Click on the “On click” button.
4. Right click on the “OnContextMenu” button.
5. Double click on the “OnDoubleClick” button.
6. Focus on the “OnFocus” button.
7. Focus on the “OnKeyDown” button, key down any key.
8. Focus on the “OnKeyUp” button, key down then key up any button.
9. Focus on the “OnKeyPress” button, press any key.

10. Mouse over on the “OnMouseOver“ button.
11. Mouse over on the “OnMouseLeave” button, then mouse out.
12. Mouse down on the “OnMouseDown” button.

Expected results:

1. Page opened.
2. Under the button appears the text ”Event triggered”.
3. Under the button appears the text ”Event triggered”.
4. Under the button appears the text ”Event triggered”.
5. Under the button appears the text ”Event triggered”.
6. Under the button appears the text ”Event triggered”.
7. Under the button appears the text ”Event triggered”.
8. Under the button appears the text ”Event triggered”.
9. Under the button appears the text ”Event triggered”.
10. Under the button appears the text ”Event triggered”.
11. Under the button appears the text ”Event triggered”.
12. Under the button appears the text ”Event triggered”.

Title: Handling elements inside other elements with ‘hover’ pseudo class

Steps:

1. Click on the “Hover Test Page” link inside “CSS Pseudo Classes” list element.
2. Trigger hover on the “Hover Para” button.

Expected results:

1. Page opened.
2. Under the button appears a text element.

Title: Click on link inside element which use ‘hover ’ pseudo class

Steps:

1. Click on the “Hover Test Page” link inside “CSS Pseudo Classes” list element.
2. Trigger hover on the “Hover Div” button.
3. Click on the “Can you click me?” link.

Expected results:

1. Page opened.
2. Under the button appears text and link.
3. New page opened.

Title: Clicking on buttons that appears some time after clicking on another button

Steps:

1. Click on the “Dynamic Button Challenge 01” inside ‘’Synchronisation” list element.
2. Click on the “Start” button.
3. Click on the “One” button.

4. Click on the “Two” button.
5. Click on the “Three” button.

Expected results:

1. Page opened.
2. A new button appears.
3. Under the buttons appears the text “Wait…”, after a few seconds a new button appears.
4. Under the buttons appears the text “Wait…”, after a few seconds a new button appears.
5. Text above the buttons changed from “Click all 4 buttons” to “All buttons clicked”.

Title: Clicking on buttons that become enabled some time after clicking on another button

Steps:

1. Click on the “Dynamic Button Challenge 02” inside ‘’Synchronization” list element.
2. Click on the “Start” button.
3. Click on the “One” button.
4. Click on the “Two” button.
5. Click on the “Three” button.

Expected results:

1. Page opened.
2. After a few seconds, button “One” became enabled.
3. Under the buttons appears the text “Wait…”, after a few seconds, button “Two” became enabled.
4. Under the buttons appears the text “Wait…”, after a few seconds, button “Three” became enabled.
5. Text above the buttons changed from “Click Buttons in order” to “All buttons clicked”.

Title: Sending client-server form with an HTTP request

Steps:

1. Click on “Client Server Form Input Validation” inside “Micro apps” list element.
2. Send POST request to ‘https://testpages.herokuapp.com/validate/input-validation’ with form.
3. Open returned page.

Expected results:

1. Page opened.
2. Returned response with body.
3. The page contains the data that was sent in the previous step.

Title: Cookies handling

Steps:

1. Click on “Cookies Controlling Page Access” inside “Cookies” list element.
2. Add new cookie with key ‘loggedin’ and value ‘Admin’.
3. Reload page.

Expected results:

1. Page opened. Page doesn’t contain cookies.
2. —
3. Page reloaded, new cookie added, page changed to admin view.

Title: Handling download file

Steps:

1. Click on “File Download“ inside “Files” list element.
2. Click on “Direct Link Download” button, wait until the download finishes.

Expected results:

1. Page opened.
2. File downloaded, contains “This is a text file.” text in it and is 142 KB in size.

Title: Interactions with HTML 5 forms

Test Data:

Steps:

1. Click on “HTML5 Element Form Test Page” inside “Forms & Windows” list element.
2. Fill form using test data.
3. Click on “Submit” button.

Expected results:

1. Page opened.
2. —
3. Results page contains form data from previous step.

Title: Sending file

Steps:

1. Click on “File Upload Example Page” inside “Files” list element.
2. Choose random file in the file input, click on the “Upload button”

Expected results:

1. Page opened.
2. New page opened, under “You uploaded this file:” is the name of file.

15.Title: Using “User agent” to redirect on the mobile version of sites

Steps:

1. Click on “User Agent Redirect Page” inside “User Agents” list element.

2. Visit the same page with a mobile user agent.

Expected results:

1. Page opened.
2. Opened page which contains “mobile” inside page Header and Url.

Title: Interception specific request and modification request body

Steps:

1. Click on “XHTTP Messages Processing” inside “Synchronization” list element.
2. Intercept GET request to ‘https://testpages.herokuapp.com/styled/sync/messageset01.json’ and change is body to array with one element: object with property ‘message’ and value ‘hello it\'s Johnny!‘.
3. Wait until the request to be sent and handled by the server.

Expected results:

1. Page opened.
2. —
3. Request body equal to the data set in the previous step.

Title: Interception specific request and modification request headers

Steps:

1. Click on “XHTTP Messages Processing” inside “Synchronization” list element.
2. Intercept GET request to ‘https://testpages.herokuapp.com/styled/sync/messageset03.json’ and add to his header property ‘new-custom-header’ with value ‘added new header’
3. Wait until the request to be sent and handled by the server.

Expected results:

1. Page opened.
2. —
3. Request headers contain the property set in the previous set.

Title: Interception specific request and modification request headers

Steps:

1. Click on “XHTTP Messages Processing” inside “Synchronization” list element.
2. Add to all GET request to ‘https://testpages.herokuapp.com/styled/sync/*’ new header property ‘general-header’ with value ‘this is general header’
3. Wait until two requests are sent and handled by the server.

Expected results:

1. Page opened.
2. —
3. Requests headers contain the property set in the previous set.