


```markdown

**\*\* Test Environment:\*\***

- Browser: [Google Chrome Versión 120.0.6099.199 (Build oficial) (x86\_64)]
-  Operating System: [ OS Catalina 10.15.7]


**\*\*  Preconditions:\*\***

The web page "https://the-internet.herokuapp.com/" is accessible.

**\*\*  QA Tester:\*\***

Santiago\_J\_Montes\_de\_Oca

**\*\*Test Case ID:\*\* TC\_2AddRemoveElements\_001**

**\*\*Test Case Title:\*\* Verify Adding and Removing Elements **

**\*\*  Test Objective:\*\***


To ensure that the "Add/Remove Elements" functionality allows users to successfully add and remove elements on the page.

**\*\*  Test Data:\*\* None**

**\*\*Test Steps:\*\***



1. Open the web page and click on [Add/Remove Elements](#) button/section.

**\*\*  Expected Result:\*\***

 The page is successfully loaded.


2. Click on the "Add Element" button.

**\*\*  Expected Result:\*\***

  An element is added on the page.

3. Click the "Delete" button.


**\*\*  Expected Result:\*\***


  The added element is successfully removed.

**\*\* Test Data Variations:\*\* None**

**\*\*Postconditions:\*\***

The page should remain in a stable state, with the added element removed upon clicking "Delete."

**\*\*  Date:\*\* 15/01/23**

**\*\*  Test Verdict:\*\***

**Pass**



**\*\*📝 Notes and Comments:\*\***

- Ensure that the page response is prompt.
- Check for any error messages or unexpected behaviors.

**\*\*📎 Attachments:\*\***

[No attachments]

...

---

```markdown

****Test Case ID:** TC_3BasicAuth_001**

****Test Case Title:** Verify Basic Authentication Functionality 🔒**

****🎯 Test Objective:****

To ensure that the "Basic Auth" functionality allows users to authenticate successfully using basic authentication credentials.

****📊 Test Data:****

- Username: [valid_username]
- Password: [valid_password]

****Test Steps:****

1. ****Step Description:**** Accessing the Basic Auth section.

- Open the web page and click on the "Basic Auth" link.

****👁 Expected Result:****

- ✅ The browser's authentication dialog is prompted.

2. ****Step Description:**** Entering valid credentials.

- Enter the valid username and password in the authentication dialog.

****👁 Expected Result:****

- ✅ The user is successfully authenticated, and the protected content is accessible.

**** Test Data Variations:****

- ***Variation 1: Incorrect Username***

- Username: [invalid_username]
- Password: [valid_password]

- ****👁 Expected Result:****

- ❌ Authentication fails, and access is denied.

- ***Variation 2: Incorrect Password***

- Username: [valid_username]
- Password: [invalid_password]




- ****👁 Expected Result:****

- ❌ Authentication fails, and access is denied.

****Postconditions:****

The protected content should be accessible after successful authentication.

****🚩 Test Verdict:****

[Pass / Fail / Not Executed]   

****📝 Notes and Comments:****

- 🚀 Ensure that the authentication process is secure and prompt.
- 🤖 Check for any error messages or unexpected behaviors.

****📎 Attachments:****

[No attachments]

****👤 Test Case Author:****



Santiago_J_Montes_de_Oca

****📅 Date:****[insertar fecha]

``

``markdown

****Test Case ID:**** TC_4BrokenImages_001

****Test Case Title:**** Verify Handling of Broken Images  

****🎯 Test Objective:****

To confirm that the "Broken Images" section displays appropriate behavior when encountering broken images.

**** Test Environment:****

- Browser: [Google Chrome Versión 120.0.6099.199 (Build oficial) (x86_64)]
- 🖥️ Operating System: [OS Catalina 10.15.7]

****🌐 Preconditions:****

The web page "https://the-internet.herokuapp.com/" is accessible.

****📊 Test Data:**** None

****Test Steps:****

1. **** Step Description:**** Opening website section.

- Open the web page and click on the "Broken Images" link.

****👁️ Expected Result:****

 The "Broken Images" section is successfully loaded.

2. ****Step Description:**** Verify the presence of images.

- Check if there are images displayed in the section.

****👁️ Expected Result:****

 Images are present and displayed on the page.

3. **Step Description:** Introduce broken image URLs.

- Modify the URLs of some images to simulate broken images.

Expected Result:

- ✓ The images are now broken or not displayed.

4. **Step Description:** Confirm proper handling of broken images.

- Verify that the page gracefully handles broken images, providing suitable placeholders or alternative text.

Expected Result:

- ✓ The page should handle broken images without causing layout issues.

Test Data Variations: None

Postconditions:

The page should remain in a stable state, with appropriate handling of broken images.

Test Verdict:

[Pass / Fail / Not Executed] ✓ ✗ 🚧

Notes and Comments:

- 🚀 Ensure that the page response is prompt.
- 🤖 Check for any error messages or unexpected behaviors.

Attachments:

[No attachments]

Test Case Author:

Santiago_J_Montes_de_Oca

Date: [Insert Date]

...

```markdown

**Test Case ID:** TC\_5ChallengingDOM\_001

**Test Case Title:** Verify Challenging DOM Elements 🧩

**Test Objective:**

To ensure that the "Challenging DOM" section functions correctly and displays elements as expected.

**Test Environment:**









- Browser: [Google Chrome Versión 120.0.6099.199 (Build oficial) (x86\_64)]
- 🖥️ Operating System: [ OS Catalina 10.15.7]

**Preconditions:**

The web page "https://the-internet.herokuapp.com/" is accessible.

**\*\* Test Data:\*\*** None

**\*\*Test Steps:\*\***

- \*\* Step Description:\*\*** Opening challenging DOM section.
  - Open the web page and navigate to the "Challenging DOM" section.**\*\* Expected Result:\*\***
  -  The "Challenging DOM" section is successfully loaded.
- \*\*Step Description:\*\*** Verify the presence of challenging elements.
  - Ensure that challenging elements (e.g., buttons, checkboxes) are visible on the page.**\*\* Expected Result:\*\***
  -  Challenging elements are present and visible.
- \*\*Step Description:\*\*** Interact with challenging elements.
  - Interact with each challenging element (e.g., click buttons, select checkboxes).**\*\* Expected Result:\*\***
  -  Challenging elements respond as expected.
- \*\*Step Description:\*\*** Verify dynamic content.
  - Check for the presence of dynamic content on the page.**\*\* Expected Result:\*\***
  -  Dynamic content is present and updates appropriately.

**\*\* Test Data Variations:\*\*** None



**\*\*Postconditions:\*\***

The "Challenging DOM" page should remain in a stable state, with elements responding as expected.

**\*\* Test Verdict:\*\***

[Pass / Fail / Not Executed]   

**\*\* Notes and Comments:\*\***


-  Ensure that the page response is prompt.
-  Check for any error messages or unexpected behaviors.

**\*\* Attachments:\*\***

[No attachments]

**\*\* Test Case Author:\*\***

Santiago\_J\_Montes\_de\_Oca



**\*\* Date:\*\*** [Fecha]

...

---

```markdown


****Test Case ID:** TC_6Checkboxes_001**

****Test Case Title:** Verify Checking and Unchecking Checkboxes**  

****🎯 Test Objective:****

To ensure that the "Checkboxes" functionality allows users to successfully check and uncheck checkboxes on the page.

****Test Environment:****

- Browser: [Google Chrome Versión 120.0.6099.199 (Build oficial) (x86_64)]
-  Operating System: [OS Catalina 10.15.7]

****🌐 Preconditions:****

The web page "https://the-internet.herokuapp.com/" is accessible.


****📊 Test Data:** None**

****Test Steps:****

1. ****Step Description:**** Opening website section.

- Open the web page and click on the "Checkboxes" button/section.


****👁 Expected Result:****

 The page is successfully loaded.

2. ****Step Description:**** Check the first checkbox.

- Click on the first checkbox.

****👁 Expected Result:****

  The first checkbox is successfully checked.

3. ****Step Description:**** Uncheck the second checkbox.

- Click on the second checkbox.

****👁 Expected Result:****


  The second checkbox is successfully unchecked.

****Test Data Variations:** None**



****Postconditions:****

The page should remain in a stable state, with checkboxes reflecting their checked or unchecked status.

****🏁 Test Verdict:****

[Pass / Fail / Not Executed]   

****📝 Notes and Comments:****

-  Ensure that checkbox actions are responsive.
-  Check for any error messages or unexpected behaviors.

****📎 Attachments:****

[No attachments]

****👤 Test Case Author:****

Santiago_J_Montes_de_Oca

****📅 Date:**** [insertar fecha] 📅

...

```markdown

**\*\*Test Case ID:\*\* TC\_7ContextMenu\_001**

**\*\*Test Case Title:\*\* Verify Context Menu Functionality 🍲**

**\*\*🎯 Test Objective:\*\***

To ensure that the "Context Menu" functionality allows users to interact with the context menu on the page.

**\*\* Test Environment:\*\***

- Browser: [Google Chrome Versión 120.0.6099.199 (Build oficial) (x86\_64)]
- 🖥️ Operating System: [OS Catalina 10.15.7]

**\*\*🌐 Preconditions:\*\***

The web page "<https://the-internet.herokuapp.com/>" is accessible.

**\*\*📊 Test Data:\*\* None**

**\*\*Test Steps:\*\***

1. **\*\* Step Description:\*\*** Opening the Context Menu section.
  - Open the web page and click on the Context Menu button/section.

**\*\*👁️ Expected Result:\*\***

✅ The page is successfully loaded.

2. **\*\*Step Description:\*\*** Right-click on the context menu area.
  - Right-click on the designated context menu area.

**\*\*👁️ Expected Result:\*\***

✅ A context menu is triggered and displayed.

3. **\*\* Step Description:\*\*** Verify options in the context menu.
  - Check for the presence of standard options in the context menu (e.g., Copy, Paste, etc.).

**\*\*👁️ Expected Result:\*\***

✅ The context menu contains expected options.

**\*\* Test Data Variations:\*\* None**

**\*\*Postconditions:\*\***

The page should remain in a stable state, with the context menu options functioning as expected.

**\*\*🏁 Test Verdict:\*\***

[Pass / Fail / Not Executed]   

**\*\*📝 Notes and Comments:\*\***


- 🚀 Ensure that the context menu response is prompt.
- 🤖 Check for any error messages or unexpected behaviors.

**\*\*📎 Attachments:\*\***

[No attachments]

**\*\*👤 Test Case Author:\*\***

Santiago\_J\_Montes\_de\_Oca

**\*\*📅 Date:\*\*** [Insertar fecha] 

```

```markdown


**\*\*Test Case ID:\*\*** TC\_8DigestAuthentication\_001

**\*\*Test Case Title:\*\*** Verify Digest Authentication 

**\*\*🎯 Test Objective:\*\***

To ensure that the "Digest Authentication" functionality works correctly, allowing users to authenticate successfully.

**\*\* Test Environment:\*\***

- Browser: [Google Chrome Versión 120.0.6099.199 (Build oficial) (x86\_64)]
-  Operating System: [ OS Catalina 10.15.7]

**\*\*🌐 Preconditions:\*\***

The web page "https://the-internet.herokuapp.com/" is accessible.

**\*\*📊 Test Data:\*\***

- Username: [Provided username for authentication]
- Password: [Provided password for authentication]

**\*\*Test Steps:\*\***

1. **\*\* Step Description:\*\*** Opening Digest Authentication section.
  - Open the web page and click on the "Digest Authentication" link.

**\*\*👁 Expected Result:\*\***

-  The login prompt for Digest Authentication is displayed.

2. **\*\*Step Description:\*\*** Enter valid credentials.



- Enter the provided username and password in the authentication prompt.

**\*\*👁 Expected Result:\*\***

✅ Credentials are accepted, and access is granted.

3. **\*\* Step Description:\*\*** Verify successful login.

- Ensure that after entering valid credentials, the user is successfully authenticated.

**\*\*👁 Expected Result:\*\***

✅ User gains access to the Digest Authentication protected content.


**\*\* Test Data Variations:\*\***

- Test the scenario with an invalid username or password to ensure proper authentication failure.

**\*\*Postconditions:\*\***

The page should remain in a stable state, with the user authenticated and granted access to the protected content.

**\*\*🚩 Test Verdict:\*\***

[Pass / Fail / Not Executed]   

**\*\*📝 Notes and Comments:\*\***

- 🚀 Ensure that the authentication process is prompt.
- 🤖 Check for any error messages or unexpected behaviors.

**\*\*📎 Attachments:\*\***

[No attachments]

**\*\*👤 Test Case Author:\*\***

Santiago\_J\_Montes\_de\_Oca

**\*\*📅 Date:\*\*** [Insert Date]

...

---

```markdown


****Test Case ID:**** TC_9DisappearingElements_001

****Test Case Title:**** Verify Adding and Removing Elements 

****🎯 Test Objective:****


To ensure that the "Disappearing Elements" functionality allows users to successfully add and remove elements on the page.

**** Test Environment:****

- Browser: [Google Chrome Versión 120.0.6099.199 (Build oficial) (x86_64)]
-  Operating System: [OS Catalina 10.15.7]

****🌐 Preconditions:****

The web page "https://the-internet.herokuapp.com/" is accessible.


**** Test Data:**** None

****Test Steps:****

1. **** Step Description:**** Opening website section.

- Open the web page and click on the "Disappearing Elements" button/section.

**** Expected Result:****

-  The page is successfully loaded.

2. ****Step Description:**** Verify the presence of initial elements.

- Check for the presence of existing elements on the page.

**** Expected Result:****

-  The initial elements are visible.

3. ****Step Description:**** Click the "Add Element" button.


- Click on the "Add Element" button.


**** Expected Result:****

-   An additional element is added on the page.

4. **** Step Description:**** Verify the presence of the added element.

- Check for the presence of the newly added element.


**** Expected Result:****

-  The added element is visible.

5. **** Step Description:**** Click the "Refresh" button.


- Click on the "Refresh" button to reload the page.



**** Expected Result:****

-  The page is refreshed without errors.

6. **** Step Description:**** Verify the disappearance of the added element after refresh.

- Check if the previously added element is no longer visible.

**** Expected Result:****

-   The added element is not visible after refreshing.

**** Test Data Variations:**** None


****Postconditions:****

The page should remain in a stable state, with the added element no longer visible after the refresh.

**** Test Verdict:****

[Pass / Fail / Not Executed]   

**** Notes and Comments:****

-  Ensure that the page response is prompt.

- 🤖 Check for any error messages or unexpected behaviors.

📎 Attachments:

[No attachments]

👤 Test Case Author:

Santiago_J_Montes_de_Oca

📅 Date: [Current Date]

...
