|  |  |
| --- | --- |
| **Project:** Module 4.2 Assignment |  |
| **Course:** CSD380-A339 DevOps (2257-DD) |  |
| **Description:** Write at least five (5) test cases against the Todo application. The test plan must address the actual requirements in the business requirements document. |  |
| **Date:** 2025/06/19 |  |
| **Name:** Lea Trueworthy |  |

To complete the test case plan, fill out the information for Project, Course, Description and Date in the header below.

**TABLE OF CONTENTS**

[**Confirm users can access the site via a standard URL** 2](#_Toc201219132)

[**Creating a New Todo Task Item** 3](#_Toc201219133)

[**Edit a Todo item** 4](#_Toc201219134)

[**Deleting a Todo Task Item** 5](#_Toc201219135)

[**Display custom 404 error page for invalid URLs** 6](#_Toc201219136)

For each test, the developer should: provide a test description, a test objective, the developer’s name and date tested. For each step, fill out the actions to be taken and describe the expected results, and indicate yes or no for each step depending upon the result. The peer tester should provide their name, date tested, indicate yes or no for each step depending upon the result, and a screenshot (thumbnail) of the result.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 1 | **Confirm users can access the site via a standard URL** | | | |
|  | **Test Objective:** Verify that the website is accessible through the correct URL | **Developer:** Lea Trueworthy  **Date tested:** 2025/06/19 | **Peer tester:**  **Date tested:** | |
| **Step** | **Action** | **Expected results** | **Developer pass/fail** | **Tester pass/fail**  **+ Screenshot** |
| 1 | Navigate to <https://buwebdev.github.io/todo/> | Page loads without errors | yes |  |
| 2 | Refresh the browser window | Page reloads and displays correctly | yes |  |
| 3 | Open the site in Chrome | Page loads correctly | yes |  |
| 4 | Resize the browser window | Table adjusts and reflows content responsively | yes |  |
| 5 | Open the site in Firefox (or a different browser) | Page loads correctly | yes |  |
| **Comments** | The site loads across major browsers and responds well to different screen widths, maintaining a clean and usable layout. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 2 | **Creating a New Todo Task Item** | | | |
|  | **Test Objective:** Confirm that users can create a new Todo item | **Developer:** Lea Trueworthy  **Date tested:** 2025/06/19 | **Peer tester:**  **Date tested:** | |
| **Step** | **Action** | **Expected results** | **Developer pass/fail** | **Tester pass/fail**  **+ Screenshot** |
| 1 | Navigate to the main page | Input field and “Add Task” button are visible | yes |  |
| 2 | Enter "Hello Darkness" into the input field | Text appears in field | yes |  |
| 3 | Click the “Add Task” button | "Hello Darkness" is added to the table | yes |  |
| 4 | Leave the input field blank | “Add Task” button remains inactive | yes |  |
| 5 | Enter only spaces | “Add Task” button remains inactive | yes |  |
| **Comments** | The form correctly accepts valid input and blocks empty or space-only submissions. Functionality behaves as expected and prevents invalid data from being added to the list. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 3 | **Edit a Todo item** | | | |
|  | **Test Objective:** Verify that a user can successfully modify an existing task item | **Developer:** Lea Trueworthy  **Date tested:** 2025/06/19 | **Peer tester:**  **Date tested:** | |
| **Step** | **Action** | **Expected results** | **Developer pass/fail** | **Tester pass/fail**  **+ Screenshot** |
| 1 | Add a task called “Travel” | “Travel” appears in the table | yes |  |
| 2 | Click the pencil icon next to “Travel” | Item becomes editable | yes |  |
| 3 | Change the text to “Travel Plans” | Field updates with new text | yes |  |
| 4 | Click “Save” | “Travel Plans” appears in the list | yes |  |
| 5 | Edit the item again, delete all text, and click “Save” | System should prevent saving and display an error message | no |  |
| 6 | Edit another item, enter only spaces, and click “Save” | System should prevent saving and display an error message | no |  |
| **Comments** | While basic editing works, there is no validation to prevent saving blank or space-only entries. The system fails to enforce expected behavior. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 4 | **Deleting a Todo Task Item** | | | |
|  | **Test Objective:** Verify that a user can successfully remove a task item from the Todo list | **Developer:** Lea Trueworthy  **Date tested:** 2025/06/19 | **Peer tester:**  **Date tested:** | |
| **Step** | **Action** | **Expected results** | **Developer pass/fail** | **Tester pass/fail**  **+ Screenshot** |
| 1 | Locate a preloaded item (e.g., "Attend 2PM meeting") | Item is visible in the list | yes |  |
| 2 | Click the trash can icon next to that item | Item disappears from the list | yes |  |
| 3 | Refresh the browser | Preloaded item should remain deleted | no |  |
| 4 | Add a new item (e.g., “Test Deletion”) and click “Add Task” | New item appears in the table | yes |  |
| 5 | Delete the new item | “Test Deletion” disappears from the list | yes |  |
| 6 | Refresh the browser | New item should still be deleted | yes |  |
| **Comments** | Deletion appears to work visually for both preloaded and newly added items. However, only deletions of newly added items persist. Preloaded items reappear after a refresh, which may be intentional for demonstration purposes but may not meet typical expectations for deletion behavior. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 5 | **Display custom 404 error page for invalid URLs** | | | |
|  | **Test Objective:** Confirm that invalid URLs redirect to a custom 404 page | **Developer:** Lea Trueworthy  **Date tested:** 2025/06/19 | **Peer tester:**  **Date tested:** | |
| **Step** | **Action** | **Expected results** | **Developer pass/fail** | **Tester pass/fail**  **+ Screenshot** |
| 1 | Visit <https://buwebdev.github.io/todo/fakepage> | Redirects to /todo/not-found | yes |  |
| 2 | Observe the message on the page | Text displays: “not-found works!” | yes |  |
| 3 | Try variations like /todo/random404 | All redirect to the same custom not-found page | yes |  |
| 4 | Look for branding or navigation | Page has minimal content (just text) | yes |  |
| **Comments** | The 404 handling works as intended. Users are redirected to a custom route that confirms the invalid path was caught. | | | |