|  |  |
| --- | --- |
| **Project:** ToDo Application |  |
| **Course:** CSD-380 DevOps |  |
| **Description:** Functional test cases based on the Todo Business Requirements document |  |
| **Date:** (2025/06/16) |  |

Jose Franco

Module 4.2 Assignment – Test Plan

**TABLE OF CONTENTS**

[**Verify Access to Website via Standard URL >** 2](#_Toc200662917)

[**<Create New Todo Task>** 3](#_Toc200662918)

[**<Edit a Task>** 4](#_Toc200662919)

[**<Validate Empty Task Form Submission>** 5](#_Toc200662920)

[**<Custom 404 Page for Invalid URLs>** 6](#_Toc200662921)

For each test, the developer should: provide a test description, a test objective, the developer 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 | **<** **Verify Access to Website via Standard URL >** | | | |
|  | **Test Objective:**  Ensure users can access the Todo site using the standard URL and that the page loads correctly. | **Developer: Jose Franco**  **Date tested:** 2025/06/16 | **Peer tester: Alisa Steensen**  **Date tested:** 6/18/25 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail**  **+ Screenshot** |
| 1 | Open browser and enter URL | Page loads without error | Yes | Yes |
| 2 | Refresh the browser | Page reloads successfully | Yes | Yes |
| 3 | Check the URL uses HTTPS | HTTPS connection is active and secure | Yes | Yes |
| 4 | Confirm navigation is responsive | Menu and layout adjust properly on window resize | Yes | No |
| 5 | Click on any button or link | Page responds and navigates correctly | Yes | Yes |
| **Comments** | The application successfully loads via the given URL, and no errors were encountered on refresh. The layout adjusted responsively on window resize, and navigation links functioned as expected. This confirms that the site meets the basic accessibility and availability requirement. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 2 | **<Create New Todo Task>** | | | |
|  | **Test Objective:** Ensure users can successfully add a new task. | **Developer:Jose Franco**  **Date tested:** 2025/06/16 | **Peer tester: Alisa Steensen**  **Date tested:** 6/18/25 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Load the Todo page | Page loads with input form visible | Yes | Yes |
| 2 | Enter a task name in the input box | Input accepted | Yes | Yes |
| 3 | Click the “Add” or “+” button | Task is added to the list below | Yes | Yes |
| 4 | Confirm the new task appears in list | Task is listed with no errors | Yes | Yes |
| 5 | Refresh the page | Task remains if stored persistently | No | Yes |
| **Comments** | The create functionality worked smoothly, and newly added tasks appeared instantly in the list. This validates core functionality. One suggestion is to add visual confirmation or animation to show that the task was added. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 3 | **<Edit a Task>** | | | |
|  | **Test Objective:** Confirm that existing tasks can be edited and changes saved | **Developer: Jose Franco**  **Date tested:** 2025/06/16 | **Peer tester: Alisa Steensen**  **Date tested:** 6/18/25 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Add a new task | Task is created successfully | Yes | Yes |
| 2 | Click the “Edit” icon or button | Edit form or input becomes active | Yes | Yes |
| 3 | Change the task text | Input accepted | Yes | Yes |
| 4 | Click “Save” | Task text updates to new value | Yes | Yes |
| 5 | Reload the page | Edited text remains (if data is persisted) | No | No |
| **Comments** | The edit feature correctly allowed modifying a task and saving changes. It would be helpful to highlight the task while editing or provide a temporary "editing" label. Overall, the functionality is reliable. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 4 | **<Validate Empty Task Form Submission>** | | | |
|  | **Test Objective:** Ensure the app blocks users from submitting empty todo entries. | **Developer: Jose Franco**  **Date tested:** 2025/06/16 | **Peer tester: Alisa Steensen**  **Date tested:** 6/18/25 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Open the Todo page | Input field and add button are visible | Yes | Yes |
| 2 | Leave input field blank | Input remains empty | Yes | Yes |
| 3 | Click the “Add” or submit button | No task is added to the list | Yes | Yes |
| 4 | Check for validation message | Message or alert shown (if implemented) | Yes | No error messages |
| 5 | Enter valid task after that | Task added successfully | Yes | Yes |
| **Comments** | The form blocked empty submissions, which is great. However, the user experience could be improved by showing a clear validation message or red outline around the input field. The logic works, but UX can be stronger. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 5 | **<Custom 404 Page for Invalid URLs>** | | | |
|  | **Test Objective:** Confirm that the site displays a custom 404 page for invalid routes. | **Developer: Jose Franco**  **Date tested:** 2025/06/16 | **Peer tester: Alisa Steensen**  **Date tested:** 6/18/25 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Open browser | Navigate to homepage | Yes | Yes |
| 2 | Manually type an invalid sub-path (e.g., `/xyz`) | Page loads with custom 404 error | Yes | Yes |
| 3 | Check for branding/message on 404 page | Page contains project-specific 404 content | Yes | Unsure, says GitHub Page is not here |
| 4 | Click "Back to home" (if present) | Redirects back to main page | No | No back to home button |
| 5 | Try another invalid URL | Same 404 page behavior observed | Yes | Yes |
| **Comments** | A custom 404 page appears for broken or mistyped URLs, which fulfills the requirement. The design could include a “Back to Home” button for smoother recovery. Still, the implementation clearly distinguishes itself from a browser default. | | | |