|  |  |
| --- | --- |
| **Project:** Todo Application | **Developer:** Tazmin Somerville |
| **Course:** CSD 380 - DevOps | Tester: Robert Minkler |
| **Description:** Test Plan for verifying core functionalities of the Todo Application |  |
| **Date:** 2025/04/09 |  |

**TABLE OF CONTENTS**

[**Verify Landing Page Loads (Requirement #10 + Prototype Check)** 2](#_Toc195205394)

[**Create a New Todo Task Item (Requirement #2)** 5](#_Toc195205395)

[**Delete a Todo Task Item (Requirement #4)** 8](#_Toc195205396)

[**Edit a Todo Task Item (Requirement #3)** 11](#_Toc195205397)

[**Invalid URL Handling (Requirement #5: Custom 404 Page)** 14](#_Toc195205398)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 1 | **Verify Landing Page Loads (Requirement #10 + Prototype Check)** | | | |
|  | **Test Objective:** Ensure that users can access the Todo website through its standard URL, and that the landing page loads correctly without errors. | **Developer:** Tazmin Somerville  **Date tested:** 2025/04/09 | **Peer tester: Robert Minkler**  **Date tested:** 2025/04/10 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail**  **+ Screenshot** |
| 1 | Visit <https://buwebdev.github.io/todo/> | Landing page loads without errors | <yes/no> - Yes | Yes |
| 2 | Refresh the browser window | The page reloads correctly without errors | <yes/no> - Yes | Yes |
| 3 | Inspect responsiveness on desktop browser by resizing window | Layout adjusts without breaking | <yes/no> - Yes | Yes |
| 4 | Open the landing page on a mobile browser | Layout is responsive and readable | <yes/no> - Yes | Yes |
| 5 | Use keyboard navigation (tab, enter) on input and buttons | Elements are accessible via keyboard | <yes/no> - Yes | Yes |
| **Comments** | The landing page loaded quickly without any visible errors or delays. Refreshing the browser also reloaded the page smoothly, maintaining consistent performance. There were no display issues observed across the tested views.  Tester Comments:  The page loaded quickly on all platforms when visiting the link. The website was responsive and adjusted appropriately when changing window size and when loading on a mobile device. When using the tab key to navigate the page, it was easy to see what element was active. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 2 | **Create a New Todo Task Item (Requirement #2)** | | | |
|  | **Test Objective:** Verify that users can add a new todo item via the input field on the landing page. | **Developer:** Tazmin Somerville  **Date tested:** 2025/04/09 | **Peer tester: Robert Minkler**  **Date tested:** 2025/04/10 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Visit <https://buwebdev.github.io/todo/> | Landing page loads with visible input field | <yes/no> - Yes | Yes  A computer screen shot of a task form  AI-generated content may be incorrect. |
| 2 | |  | | --- | |  |  |  | | --- | | Click into the input field and type “New Task” | | The input is accepted without errors | <yes/no> - Yes | Yes |
| 3 | Click the **Add Task** button | The new task is added to the list below | <yes/no> - Yes | Yes |
| 4 | Confirm the new task item displays with an edit icon and delete icon | Item is added correctly with full functionality | <yes/no> - Yes | Yes |
| 5 | Refresh the browser window | Task is cleared after refresh; persistence not required by this test case | <yes/no> - Yes | Yes |
| **Comments** | The task was added successfully to the list using the input field and **Add Task** button. The interface was responsive and intuitive. However, refreshing the browser causes the task to disappear, indicating that persistence (e.g., local storage or backend saving) is not implemented in this version of the app.  Tester Comments:  The task was added as expected with edit and delete icons. One test that we did not perform was to confirm these actions work on a mobile device. Input fields and buttons may react differently on a mobile device with touch controls. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 3 | **Delete a Todo Task Item (Requirement #4)** | | | |
|  | **Test Objective:** Verify that users can delete an existing task from the todo list. | **Developer:** Tazmin Somerville  **Date tested:** 2025/04/09 | **Peer tester: Robert Minkler**  **Date tested:** 2025/04/10 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Visit <https://buwebdev.github.io/todo/> | Page loads with visible input and existing tasks | <yes/no> - Yes | Yes |
| 2 | Add a new task named “Delete Test Task” | Task is added to the list below | <yes/no> - Yes | Yes |
| 3 | Locate the “Delete” (trash can) icon next to the new task | Delete icons are visible and clickable | <yes/no> - Yes | Yes  A screenshot of a computer  AI-generated content may be incorrect. |
| 4 | Click the “Delete” icon | Task is immediately removed from the list | <yes/no> - Yes | Yes |
| 5 | Confirm the task does not reappear on refresh | Task is gone permanently after refresh | <yes/no> - Yes | Yes  A screenshot of a computer  AI-generated content may be incorrect. |
| **Comments** | Deleting tasks works as expected. The delete icon was responsive and removed the task immediately upon interaction. After refreshing the page, the task remained deleted, confirming it does not persist unexpectedly. This aligns with the expected behavior for a temporary session-based list.  Tester Comments:  Clicking the trash can icon next to a task removes it from the list immediately. Further testing should be performed to test these buttons on mobile devices and deleting pre-existing tasks.  Refreshing the page does not confirm that deleting the task worked properly because the task list is not persistent. Deleted preexisting tasks will reappear. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 4 | **Edit a Todo Task Item (Requirement #3)** | | | |
|  | **Test Objective:** Verify that users can edit an existing task on the list using the edit icon and update the task text. | **Developer:** Tazmin Somerville  **Date tested:** 2025/04/09 | **Peer tester: Robert Minkler**  **Date tested:** 2025/04/10 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Visit <https://buwebdev.github.io/todo/> | Landing page loads with visible tasks | <yes/no> - Yes | YesA screenshot of a computer  AI-generated content may be incorrect. |
| 2 | Add a task labeled “Edit Test Task” | Task is added to the list | <yes/no> - Yes | Yes |
| 3 | Click the “Edit” (pencil) icon next to the new task | Task text becomes editable | <yes/no> - Yes | Yes – see comments |
| 4 | Change the task text to 'This is the Edit' and click the Save button | Task is updated in the list with new text | <yes/no> - Yes | Yes |
| 5 | Refresh the page | Edited task disappears (no persistence) | <yes/no> - Yes | Yes |
| **Comments** | The editing feature functioned smoothly. When the pencil icon is clicked, a pop-up allows editing, and the update was applied immediately on saving with button. However, like with task creation, the edit did not persist after a page refresh, which is consistent with the app’s current design. The interface was clear and intuitive throughout the process.  Tester Comments:  Step 3: The task text did not directly become editable as described, but a popup window with the editable task appeared on screen. While I passed this test, the expected results are unclear, and the test should be revised for clarity.  Task editing worked as expected with the previous task replaced with a new task.  The requirement to not save a blank entry when editing was not tested. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 5 | **Invalid URL Handling (Requirement #5: Custom 404 Page)** | | | |
|  | **Test Objective:** Verify that users are routed to a basic 404-style message when attempting to access a non-existent URL. | **Developer:** Tazmin Somerville  **Date tested:** 2025/04/09 | **Peer tester: Robert Minkler**  **Date tested:** 2025/04/10 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Navigate to an invalid URL (e.g., <https://buwebdev.github.io/todo/invalid> | User is redirected to /not-found route, which displays a 404-style message | <yes/no> - Yes | Yes |
| 2 | Observe the message on the /not-found page | Text "not-found works!" is displayed | <yes/no> - Yes | Yes |
| 3 | Confirm the page lacks advanced styling or navigation | The page is plain, with no back or home link | <yes/no> - Yes | YesA screenshot of a computer  AI-generated content may be incorrect. |
| 4 | Use the browser back button | User returns to the homepage (/todo) successfully | <yes/no> - Yes | No |
| **Comments** | The app redirects to a dedicated /not-found route when an invalid URL is entered. While the message “not-found works!” confirms the route is functioning, the page is extremely basic and lacks styling or navigation aids. There's no visible back or home button, but browser back functionality works as expected. While minimal, the behavior satisfies the requirement of a working custom 404 route in an SPA context.  Tester Comments:  When an incorrect URL is entered the custom 404 page is displayed at <https://buwebdev.github.io/todo/not-found>  The error message is unclear to users even though the expected functionality works. “not-found works!” is not an effective message for users.  Step 4 is dependent on what was viewed prior to the incorrect url. If the 404 page was the first page loaded the back button is disabled. Step 4 is testing browser functionality vs application functions and should be reconsidered. | | | |