|  |  |
| --- | --- |
| **Project:** Test Plan |  |
| **Course:** **CSD380** |  |
| **Description:** Create 5 Test Cases and Verify the Functionality of the Todo Application <https://buwebdev.github.io/todo/> |  |
| **Date:** 2024/11/15 |  |

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

.

**TABLE OF CONTENTS**

[**Tests 1** 2](#_Toc132718195)

[**Tests 2** 3](#_Toc132718195)

[**Tests 3** 4](#_Toc132718195)

[**Tests 4** 5](#_Toc132718195)

[**Tests 5** 6](#_Toc132718195)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 1 | **Access Todo Application via URL** | | | |
|  | **Test Objective: Verify that users can access the Todo application through a standard URL** | **Developer: Joseph Huffer**  **Date tested:** 2024/11/16 | **Peer tester: Hlee Xiong**  **Date tested:** <2024/11/dd> | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Open a Web Browser | Browser launches and opens successfully | Yes - Pass | YesA screenshot of a computer  Description automatically generated |
| 2 | Enter the following URL in the address bar: <https://buwebdev.github.io/todo/>. And press ‘ENTER’ to navigate to the page | The Todo app’s main page loads without any errors. | Yes - Pass | Yes  A screenshot of a computer  Description automatically generated |
| 3 | Confirm that the page loads | Page Loads | Yes - Pass | Yes  A screenshot of a computer  Description automatically generated |
| 4 | Perform a refresh on the browser page | Page Refreshes and displays identical data. | Yes - Pass | Yes A screenshot of a computer  Description automatically generated |
| 5 | Confirm that the page reloads correctly | Page loads correctly. | Yes - Pass | Yes A screenshot of a computer  Description automatically generated |
| **Comments** | Dev Notes: This test successfully verifies that the todo application is accessible via the provided URL. The page loads without errors, and the data remains consistent after refreshing the browser. Confirming compatibility with different browsers could be used to expand the results and use case of tests like this, just as a note.  Tester Notes: After conducting initial validation, the todo application successfully loads and operates at the provided URL. Testing confirms proper page rendering without error messages and demonstrates consistent data persistence across browser refresh cycles. The core functionality meets all acceptance criteria, with todo items maintaining their state reliably. Based on the development team's suggestions, expanding test coverage to include cross-browser compatibility would be beneficial. Specifically, systematic testing across Chrome, Firefox, Safari, and Edge should be implemented to ensure consistent behavior across all major platforms. Additional considerations for future test iterations should include multiple tab behavior, network interruption handling, and local storage limits. While the current implementation passes all base requirements, documenting specific browser versions and incorporating performance metrics would strengthen the overall test documentation. No blocking issues were identified during this test phase, and the application is ready for expanded compatibility testing as recommended. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 2 | **Create a New Todo Task** | | | |
|  | **Test Objective:** Test the creation of a new Todo item | **Developer: Joe Huffer**  **Date tested:** 2024/11/16 | **Peer tester: Hlee Xiong**  **Date tested:** 2024/11/17 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail**  **+ Screenshot** |
| 1 | Navigate to the todo app landing page | Page loads successfully | Yes - Pass | Yes A screenshot of a computer  Description automatically generated |
| 2 | Type a new task into the ‘Task’ input field. | The task will populate corresponding to user input | Yes - Pass | YesA screenshot of a computer  Description automatically generated |
| 3 | Click the ‘Add Task’ button | The new task appears at the bottom of the list of tasks. | Yes - Pass | YesA screenshot of a computer  Description automatically generated |
| 4 | Leave the input field blank and attempt to click “Add Task” | The system should gray out and prevent the item from being added. | Yes - Pass | YesA screenshot of a computer  Description automatically generated |
| **Comments** | Dev Notes: This test is designed to provide the following details: what happens when a new item is added, and what happens if no item is provided and the “Add Task” is still attempted. With these steps, and these results, we see that you can navigate to the page, open the app, add a new item successfully (and the behavior that takes place after clicking ‘Add Task’), to further that, we also attempt to add an item with no input provided, which does not work and no new item is created. Also, we do not need to acknowledge browser since I assume these tests will be ran in sequence.  Tester Notes: The scope of this testing focuses on the task addition functionality within the application, specifically examining both successful task creation and error prevention scenarios. The testing procedure begins with verifying successful task addition. To execute this test, navigate to the task management page and attempt to add a new task by entering valid task details and clicking the "Add Task" button. The expected outcome is that a new task will be created and displayed in the task list, with all entered details accurately represented. The UI should update accordingly to reflect this addition. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 3 | **Edit an Existing Todo Task** | | | |
|  | **Test Objective: Test editing functionality for an existing Todo item** | **Developer: Joe Huffer**  **Date tested:** 2024/11/16 | **Peer tester: Hlee Xiong**  **Date tested:** 2024/11/17 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | While still on the Todo App Landing Page, view the list towards the bottom of the application page detailing current “Todo List”. | All current tasks are displayed on the screen. | Yes - Pass | Yes  A screenshot of a computer  Description automatically generated |
| 2 | Click the ‘Pen’ icon located to the right of the item you select to modify (select any) | This task will appear in an editable input form. | Yes - Pass | YesA screenshot of a computer  Description automatically generated |
| 3 | Modify the task description and click “Save” | Task description updates successfully in the task list and displays the changes to the field. | Yes - Pass | Yes A screenshot of a computer  Description automatically generated |
| 4 | On the same or a different item, select the edit (“pen”) icon, and enter no new input, then click save. | The system should allow for no edit to be made, and the original information to remain in that field (and allow all actions described) | Yes - Pass | Fail |
| **Comments** | Dev Notes: The goal of this task is to modify elements that have been added. By selecting the edit icon (“pen”) icon, the item in the app becomes modifiable, the property corresponds to whatever input typed, and once ‘save’ is clicked, the item updated. We also depict that if you select an item, add no new input, the user can still save that value successfully without error (resulting in no change).  Tester Notes: testing process should focus on verifying that the edit functionality works as intended. Begin by selecting the edit icon (“pen”) on an item within the app and ensure that the item becomes modifiable. Input various changes to the properties and confirm that the updates are applied correctly after clicking 'save.' Test edge cases by leaving the input field blank and saving to verify that the original value remains unchanged without causing any errors. Additionally, confirm that attempting to save without modifying the input does not result in unexpected behavior, ensuring that the item remains unaltered. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 4 | **Delete a Todo Task** | | | |
|  | **Test Objective: Test the deletion of a task from the Todo list** | **Developer: Joe Huffer**  **Date tested:** 2024/11/16 | **Peer tester: Hlee Xiong**  **Date tested:** 2024/11/17 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | While still on the Todo App Landing Page, view the list towards the bottom of the application page detailing current “Todo List”. | All current tasks are displayed on the screen. | Yes - Pass | YesA screenshot of a computer  Description automatically generated |
| 2 | Next to an item, click the ‘trashcan’ or delete button. | The selected task will be removed from the list immediately | Yes - Pass | Yes  A screenshot of a computer  Description automatically generated |
| 3 | Confirm deletion | Make sure task no longer populates the list. | Yes - Pass | Yes  A screenshot of a computer  Description automatically generated |
| **Comments** | Confirming that the deletion function works and that items can be terminated at will. This is a fairly simple test/outcome, so for the sake of substance, a potential enhancement could be the addition of a “Are you sure?” option to confirm deletion, and/or an “undo” feature since deletion is simply a one-click task (and potentially clicked on accident). | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 5 | **Invalid URL, Encountering custom 404 Page** | | | |
|  | **Test Objective:** | **Developer: Joe Huffer**  **Date tested:** 2024/11/16 | **Peer tester: Hlee Xiong**  **Date tested:** 2024/11/17 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Open a new tab, or a new instance of web browser. | A new tab or new web browsing application instance will open. | Yes - Pass | Pass |
| 2 | Enter an invalid URL in the browser (example if you need one: <https://buwebdev.github.io/todo/nositehere>) | You will not be directed to the main Todo application site. You will receive an error web page detailing that the site could not be reached. | Yes - Pass | PassA screenshot of a computer  Description automatically generated |
| **Comments** | Less steps than in test 4, so there isn’t a lot to comment on. Our goal is to activate within the application a page detailing that any suffix after the URL portion: todo/ will result in a specific error page. With an invalid url, we receive the web page: <https://buwebdev.github.io/todo/not-found>, which states: “not-found works!”. | | | |