|  |  |
| --- | --- |
| **Project:** Todo Application |  |
| **Course:** CSD 380 |  |
| **Description:** Functional Test Plan for Todo Application |  |
| **Date: 2025/06/23** |  |

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

.

**TABLE OF CONTENTS**

[**Access Application via URL & View Task List** 2](#_Toc201487032)

[**Create New Task & Prevent Empty Submission** 2](#_Toc201487033)

[**Edit Task & Prevent Empty Submission** 3](#_Toc201487034)

[**Delete Task** 3](#_Toc201487035)

[**Custom 404 Error Page** 4](#_Toc201487036)

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 | **Access Application via URL & View Task List** | | | |
|  | **Test Objective:** Verify users can access the application via the standard URL and view the task list | **Developer: Jess Long-Heinicke**  **Date tested:** 2025/06/23 | **Peer tester:**  **Date tested:** <yyyy/mm/dd> | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail**  **+ Screenshot** |
| 1 | Navigate to <https://buwebdev.github.io/todo/> | Page loads without errors (console/UI) | Yes | <yes/no> |
| 2 | Verify page title | Title displays “Todo.” | Yes | <yes/no> |
| 3 | Check for task list | Existing tasks are visible (or placeholder if empty) | Yes | <yes/no> |
| 4 | Refresh the browser | Page reloads with no errors; task list persists | Yes | <yes/no> |
| 5 | Resize window to mobile view | Layout adapts to mobile (<1024px). | Yes | <yes/no> |
| **Comments** | Ensure no JavaScript erors in console. List must render tasks or show clear empty state. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 2 | **Create New Task & Prevent Empty Submission** | | | |
|  | **Test Objective: Verify users can create tasks and cannot submit empty forms** | **Developer: Jess Long-Heinicke**  **Date tested:** 2025/06/23 | **Peer tester:**  **Date tested:** <yyyy/mm/dd> | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Enter “Buy groceries” in the input field | Text appears in the input | Yes | <yes/no> |
| 2 | Click “Add Item” | Task “Buy groceries” appears in the list | Yes | <yes/no> |
| 3 | Leave input blank, click “Add Item” | Form rejects submission; no task added. | Yes | <yes/no> |
| 4 | Press Enter with empty input | Form rejects submission; no task added | Yes | <yes/no> |
| 5 | Refresh page | “Buy groceries” persists in the list | No | <yes/no> |
| **Comments** | Empty submission must trigger validation (error message/disabled button. New tasks must save to localStorage | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 3 | **Edit Task & Prevent Empty Submission** | | | |
|  | **Test Objective: Verify users can edit tasks and cannot save empty updates** | **Developer: Jess Long-Heinicke**  **Date tested:** 2025/06/23 | **Peer tester:**  **Date tested:** <yyyy/mm/dd> | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Click “Edit” next to “Buy groceries” | Task switches to edit mode (text field) | Yes | <yes/no> |
| 2 | Change text to “buy organic groceries”. Click “Save” | Task updates in the list | Yes | <yes/no> |
| 3 | Edit the task again, Delete all text, Click “Save” | Update rejected; original text remains | No | <yes/no> |
| 4 | Edit task, Clear text, Press Enter | Update rejected; task unchanged | No | <yes/no> |
| 5 | Refresh page | “Buy organic groceries” persists | No | <yes/no> |
| **Comments** | Empty edits should revert to original text. Ensure save icon is disabled during invalid submissions. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 4 | **Delete Task** | | | |
|  | **Test Objective: Verify users can delete tasks** | **Developer: Jess Long-Heinicke**  **Date tested:** 2025/06/23 | **Peer tester:**  **Date tested:** <yyyy/mm/dd> | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Click “Delete” (trash) next to “Buy organic groceries” | Task disappears from the list | Yes | <yes/no> |
| 2 | Refresh the page | Task remains deleted; does not reappear | Yes | <yes/no> |
| 3 | Add a new task, delete it | Task is removed immediately | Yes | <yes/no> |
| 4 | Check browser console | No errors during deletion | Yes | <yes/no> |
| 5 | Verify localStorage | Deleted task removed from storage | Yes | <yes/no> |
| **Comments** | Deletion must update localStorage. Confirm visual feedback for user experience. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 5 | **Custom 404 Error Page** | | | |
|  | **Test Objective: Verify custom 404 page for invalid URLs** | **Developer: Jess Long-Heinicke**  **Date tested:** 2025/06/23 | **Peer tester:**  **Date tested:** <yyyy/mm/dd> | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Navigate to <https://buwebdev.github.io/todo/invalid-path> | Custom 404 page loads (not browser default) | Yes | <yes/no> |
| 2 | Verify error message | Page shows “not-found works!” | Yes | <yes/no> |
| 3 | Check for homepage link | Link to <https://buwebdev.github.io/todo/> is present | No | <yes/no> |
| 4 | Click homepage link | Redirects to Todo landing page | No | <yes/no> |
| 5 | Open browser dev tools -> Network tab | HTTP status code is 404 | yes | <yes/no> |
| **Comments** | Custom 404 page must be user-friendly with clear navigation. Confirm no broken assets. | | | |