# Brian Preston

CSD-380 DevOps

Date: 2025/06/19

# **Test Case Template Project: Todo App**

**Test Plan for *Todo App* Business Requirements**

## **Test 1**

View list of todo task items

Test Objective: “Verify” todo list displays properly

Developer: Brian Preston

tested: 2025/06/19

Peer tester: Anton DeCesare

Date tested: 6/20/2025

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Action | Expected results | Developer pass/fail | Tester pass/fail |
| 1 | Open browser and visit https://buwebdev.github.io/todo/ | List of todo items appears | yes/no | Yes |
| 2 | Verify that task items are visible in the list | Task items display correctly | yes/no | Yes |
| 3 | Add a new task and refresh page | New task appears after refresh | yes/no | No |
| 4 | Delete a task and refresh page | Deleted task is removed from list after refresh | yes/no | No |

Comments: This test evaluates whether the application properly persists task list data between sessions or page reloads. While the initial display of tasks works as expected and newly added items appear in the list, failure to retain changes after a page refresh reveals a significant limitation. Specifically, the inability to persist added or deleted tasks suggests that the application is not saving state to a backend or local storage. This undermines user confidence, as it gives the appearance of functionality without ensuring data integrity. To improve, the app should implement persistent storage (e.g., localStorage or a database) so that all modifications are reliably maintained across sessions.

## **Test 2**

Handle invalid URL (404 page)

Test Objective: Verify custom 404 error page displays for invalid URLs

Developer: Brian Preston

tested: 2025/06/19

Peer tester: Anton DeCesare

Date tested: 6/20/2025

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Action | Expected results | Developer pass/fail | Tester pass/fail |
| 1 | Open browser and visit https://buwebdev.github.io/todo/ | Homepage loads correctly | yes/no | Yes |
| 2 | Manually change URL to https://buwebdev.github.io/todo/badurl | Custom 404 page is displayed | yes/no | Yes |
| 3 | Verify that 404 page contains navigation elements (ex: site logo) | Navigation elements appear | yes/no | No |
| 4 | Click on the site logo or link on 404 page | Site returns to homepage | yes/no | No |

Comments: This test verifies the behavior of the application when users encounter invalid URLs. While the custom 404 page loads as expected—demonstrating that the site has basic error handling in place—the lack of navigation elements like a site logo or return link diminishes the user experience. Without a clear path back to the homepage, users may become disoriented or abandon the session entirely. To improve usability, the 404 page should include branding and an actionable link (e.g., “Return Home”) to guide users back into the main application flow. Adding these elements would enhance professionalism and maintain engagement even during navigation errors.

## **Test 3**

Prevent empty edit submission

Test Objective: Verify blank edit submission is blocked

Developer: Brian Preston

Date tested: 2025/06/19

Peer tester: Anton DeCesare

Date tested: 6/20/2025

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Action | Expected results | Developer pass/fail | Tester pass/fail |
| 1 | Select an existing task | Task is selected | yes/no | Yes |
| 2 | Click “Edit” button | Edit form appears | yes/no | Yes |
| 3 | Clear all text from the input field | Field is empty | yes/no | Yes |
| 4 | Click “Save” button | Submission is blocked or error message is displayed | yes/no | No |
| 5 | Check that task is unchanged in list | Original task text remains unchanged | yes/no | No |

Comments: This test evaluates whether the application prevents users from saving an empty task when editing an existing entry. Although the form allows selection and clearing of the input, it fails to block the submission or display an error message, resulting in a blank or corrupted task replacing the original. This behavior weakens the application's data integrity and user experience. Ideally, the app should replicate the validation used during task creation—blocking empty submissions and providing visual feedback (e.g., red highlighting or inline errors). Implementing consistent validation across both create and edit workflows would improve reliability and usability.

## **Test 4**

Mobile view

Test Objective: Verify layout displays correctly on mobile devices

Developer: Brian Preston

Date tested: 2025/06/19

Peer tester:

Date tested:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Action | Expected results | Developer pass/fail | Tester pass/fail |
| 1 | Open site in desktop browser | Homepage loads correctly | yes/no | Yes |
| 2 | Open Chrome Dev Tools and enable mobile emulation | Mobile view enabled | yes/no | Yes |
| 3 | Refresh the page | Page reloads in mobile layout | yes/no | Yes |
| 4 | Verify that todo list and controls adjust to mobile screen size | Mobile-friendly layout is visible | yes/no | Yes |
| 5 | Test tap on “Add New Task” button | Button works properly in mobile view | yes/no | Yes |

Comments: This test confirms that the application adapts effectively to mobile device conditions. When emulated in Chrome Dev Tools, the layout successfully responds to screen size changes, ensuring that the todo list and controls remain readable and usable. Key interactive elements, such as the “Add New Task” button, also respond properly to tap actions, which demonstrates good mobile accessibility and responsive design practices. These results indicate that the application's CSS and layout structure are mobile-friendly. Continued testing on physical mobile devices would further confirm usability, but overall, the mobile experience meets expectations for a responsive web app.

## Test 5

Tablet view

Test Objective: Verify layout displays correctly on tablet devices

Developer: Brian Preston

Date tested: 2025/06/19

Peer tester:

Date tested:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Action | Expected results | Developer pass/fail | Tester pass/fail |
| 1 | Open site in desktop browser | Homepage loads correctly | yes/no | Yes |
| 2 | Open Chrome Dev Tools and enable tablet emulation | Tablet view enabled | yes/no | Yes |
| 3 | Refresh the page | Page reloads in tablet layout | yes/no | Yes |
| 4 | Verify that todo list and controls adjust to tablet screen size | Tablet-friendly layout is visible | yes/no | Yes |
| 5 | Test tap on “Edit” or “Delete” button | Buttons work properly in tablet view | yes/no | Yes |

Comments: This test verifies the application's behavior and responsiveness when accessed on a tablet-sized display. Using Chrome Dev Tools' tablet emulation, the layout adjusted appropriately to the larger touch interface while maintaining proper readability and spacing. All core interactive elements, including the “Edit” and “Delete” buttons, functioned correctly with tap input, confirming strong support for touch interactions. The interface appears optimized for medium-sized screens, preserving both usability and design integrity. These results suggest that the application was thoughtfully built with responsive design principles, ensuring a smooth experience across devices beyond desktop and mobile.