|  |  |
| --- | --- |
| **Project:** | **Assignment 4.2** |
| **Course:** | **CSD 380** |
| **Description:** | **Test plan for todo application** |
| **Date:** | **2025/06/17** |

.

**TABLE OF CONTENTS**

[**Create a new todo task item** 2](#_Toc201062479)

[**Delete a todo task item** 2](#_Toc201062480)

[**For invalid URLs, provide users with a custom 404 error page** 3](#_Toc201062481)

[**Prevent users from submitting a task form without entering a todo item** 4](#_Toc201062482)

[**Prevent users from submitting the edit form without entering a todo item** 5](#_Toc201062483)

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 3 | **For invalid URLs, provide users with a custom 404 error page** | | | |
|  | **Test Objective: Verifies that an invalid URL displays a custom 404 error page** | **Developer: Megan Wheeler**  **Date tested:** 2025/06/17 | **Peer tester: Anton DeCesare**  **Date tested:** 6/19/2025 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Open Web browser and go to <https://buwebdev.github.io/todo/> | Todo application loads successfully with an empty task list or existing tasks displayed | yes | Yes |
| 2 | Modify the URL to include an invalid path (e.g., <https://buwebdev.github.io/todo/invalid-page>) | Custom 404 error page is displayed instead of a generic browser 404 | yes | Yes |
| 3 | Try accessing a non-existent subdirectory (e.g., <https://buwebdev.github.io/todo/admin/nonexistent>) | Custom 404 error page is displayed | no | Yes |
| 4 | Try accessing a file that doesn't exist (e.g., <https://buwebdev.github.io/todo/random.html>) | Custom 404 error page is displayed | no | No |
| 5 | Check the content and styling of the displayed 404 page | Page shows custom branding, helpful message, and navigation back to main application | no | No |
| **Comments** | This test evaluates the application's handling of invalid URLs by checking whether a custom 404 error page is served instead of the browser's generic fallback. It confirms that the app responds appropriately to incorrect paths, nonexistent subdirectories, and missing files. The initial steps confirm that the custom 404 page is correctly triggered, but failures in Steps 3–5 indicate inconsistent implementation or lack of routing fallback for deeper URL structures. Additionally, the lack of clear branding or helpful navigation in the 404 page suggests a missed opportunity to guide users back to the main application. Enhancing the 404 page with visual consistency and actionable links would improve the user experience and help mitigate navigation errors. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 4 | **Prevent users from submitting a task form without entering a todo item** | | | |
|  | **Test Objective: Verifies that users cannot submit a task form without a todo item** | **Developer: Megan Wheeler**  **Date tested:** 2025/06/17 | **Peer tester: Anton DeCesare**  **Date tested:** 06/19/2025 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Open Web browser and go to <https://buwebdev.github.io/todo/> | Todo application loads successfully with an empty task list or existing tasks displayed | yes | Yes |
| 2 | Leave the input field empty and press Enter or click submit button | Form submission is prevented, no new empty item is added to the list | yes | Yes |
| 3 | Enter only spaces or tabs in the input field and attempt to submit | Form submission is prevented, whitespace-only items are not added | no | Yes |
| 4 | Enter only special characters (e.g., "!!!") and attempt to submit | Form either accepts valid special characters or prevents submission based on validation rules | no | Yes |
| 5 | Check if any error message or visual feedback is provided when submission is prevented | User receives clear feedback about why submission failed (if applicable) | no | Yes |
| **Comments** | This test confirms that the application handles all invalid task submissions effectively. When users attempt to submit the form with an empty field, only whitespace, or special characters, the system prevents the submission and provides immediate visual feedback by highlighting the “Task” label in red. This feedback mechanism is both intuitive and user-friendly, clearly signaling input issues without disrupting the experience. The consistent enforcement of input validation across all test scenarios ensures high data quality and a polished user interface. This reflects thoughtful design and reliable front-end validation logic. | | | |