Mark White

CSD380

Assigment#4.2

**Test Plan**

**Test Case 1: Mobile Device View**

**Requirement(s):** #8  
**Description:** Verify that the website displays correctly on a mobile device  
**Steps:**

1. Open the website using a mobile device to simulate a mobile viewport
2. ://buwebdev.github.io/todo/.
3. Observe the layout, font sizes, and element arrangement.
4. Attempt to interact with UI elements (add, edit, delete tasks).

**Expected Result:**  
The layout should be mobile-responsive, content should fit the screen, be readable, and interactive elements should function as expected.

Jon Davis 6/22/2025: A screenshot of a task form

AI-generated content may be incorrect.

**Test Case 2: Create a New Todo Task**

**Requirement(s):** #2 and #6  
**Description:** Verify a user can create a new todo task and that the form prevents empty submissions  
**Steps:**

1. Navigate to the task entry form.
2. Leave the input field blank and try to submit.
3. Confirm an error message or prevention of submission occurs.
4. Enter a valid todo item.
5. Submit the form.

**Expected Result:**  
Empty submissions are prevented. When a valid task is entered, it is added to the task list and displayed on the page.

Jon Davis 6/22/2025: A screenshot of a phone

AI-generated content may be incorrect.

AND

A screenshot of a to do list

AI-generated content may be incorrect.

**Test Case 3: View and Delete a Todo Task**

**Requirement(s):** #1 and #4  
**Description:** Ensure a user can view tasks and delete one successfully  
**Steps:**

1. Add a sample task.
2. Confirm the task appears in the list.
3. Click the delete/trash icon next to that task.
4. Confirm the task is removed from the list.

**Expected Result:**  
Tasks are visible after being added. Upon deletion, the selected task is removed without affecting others.

Jon Davis 6/22/2025: A screenshot of a to do list

AI-generated content may be incorrect.

ANDA screenshot of a list

AI-generated content may be incorrect.

**Test Case 4: Edit an Existing Todo Task**

**Requirement(s):** #3 and#7  
**Description:** Confirm a user can edit a task and cannot submit an empty edit  
**Steps:**

1. Add a new task.
2. Click the edit icon or button on that task.
3. Clear the input field and attempt to submit.
4. Confirm that the edit is not accepted, and an error or validation message appears.
5. Enter a correct task and submit.

**Expected Result:**  
Empty edits are blocked. Valid edits update the task successfully.

Jon Davis 6/22/2025 A white background with black dots

AI-generated content may be incorrect.

AND

A white background with black text

AI-generated content may be incorrect.

App fails to catch blank item, could use either an if else statement or a try catch block to catch these errors.

**Test Case 5: Invalid URL – Custom 404 Page**

**Requirement(s):** #5  
**Description:** Verify that visiting an invalid URL shows a custom 404 error page  
**Steps:**

1. In the browser, go to https://buwebdev.github.io/todo/invalid-page.
2. Observe the page response.

**Expected Result:**  
A custom 404 error page is shown to the user instead of a generic browser error page.

Jon Davis 6/22/2025: A screenshot of a computer error page

AI-generated content may be incorrect.