| **Project:** Test Plan |  |
| --- | --- |
| **Course:** CSD 380 |  |
| **Description:** Drafting a test plan based on the business requirements presented in the “Todo Business Requirements” document |  |
| **Date:** 2025/04/13 |  |

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

**TABLE OF CONTENTS**

[**Access Todo Website** 2](#_31zax9z1qkjy)

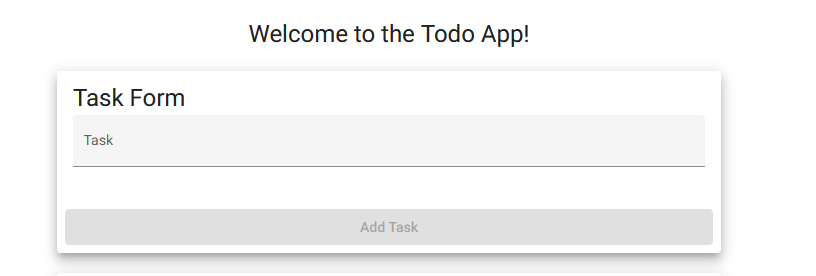
[**Add a New Todo Item** 2](#_tqukxzdf50x9)

[**Edit Todo Item** 3](#_wsgregu5of8c)

[**Delete a Todo Item** 3](#_a1ldsgr19sr5)

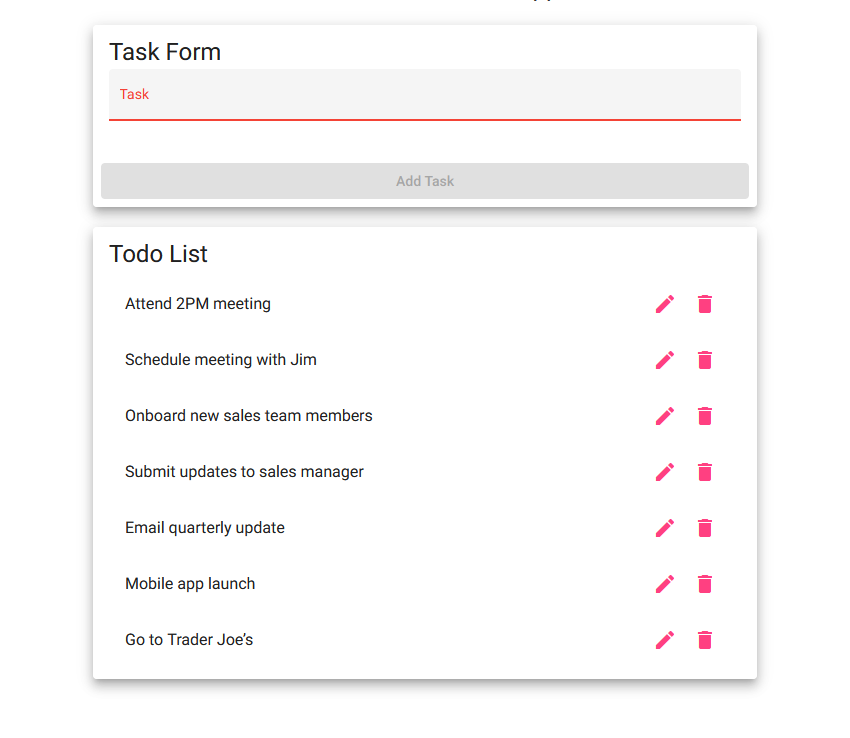
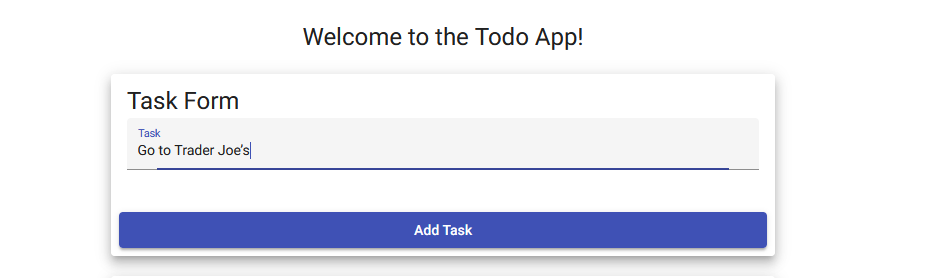
[**Prevent Submitting Blank Todo Item** 4](#_qxjzzhjtedks)

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.

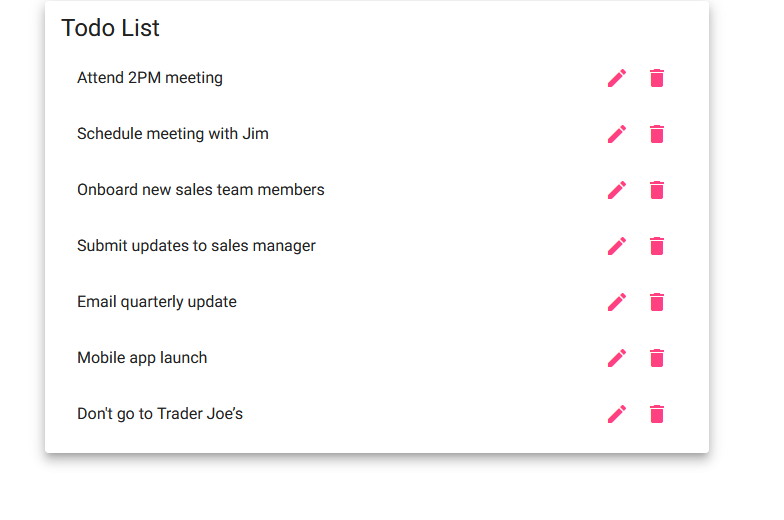
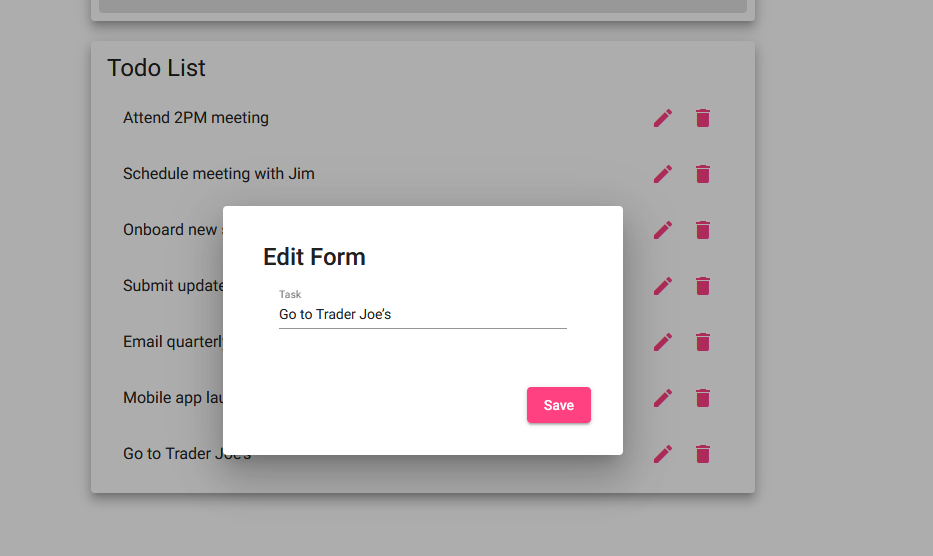


| Test 1 | **Access Todo Website** | | | |
| --- | --- | --- | --- | --- |
|  | **Test Objective:** Verify the Todo loading page loads successfully through a web browser | **Developer: Mikaela**  **Date tested:** 2025/04/10 | **Peer tester: William Renard**  **Date tested:** 2025/04/13 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail**  **+ Screenshot** |
| 1 | Open a browser | Browser opens | <yes/no> | yes |
| 2 | Navigate to https://buwebdev.github.io/todo/ | Link takes you to a loading page | <yes/no> | yes |
| 3 | Observe the page loading | Page loads without throwing any errors | <yes/no> | yes |
| **Comments** | Navigating to that url loads just fine. Web page loads and then shows the default todo list with its list items. | | | |

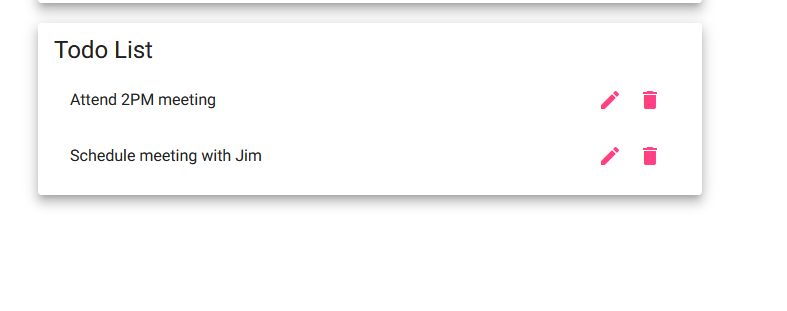
| Test 2 | **Add a New Todo Item** | | | |
| --- | --- | --- | --- | --- |
|  | **Test Objective:** verify a user can add a new todo item | **Developer: Mikaela**  **Date tested:** 2025/04/10 | **Peer tester: WIlliam Renard**  **Date tested:** 2025/04/13 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Navigate to the Todo app | Todo app opens | <yes/no> | yes |
| 2 | In the text input field, type “Go to Trader Joe’s” | Text field allows user input | <yes/no> | yes |
| 3 | Click “Add Task” | Add task button is present and users can click it | <yes/no> | yes |
| 4 | Verify the task was added to the list | User input gets added to the bottom of the list | <yes/no> | yes |
| **Comments** | Todo list opens when navigating to url. Text field allows user input, the specified user input works as expected. Add Task button is present, functional, and clicking it adds tthe task to the bottom of the list. | | | |



| Test 3 | **Edit Todo Item** | | | |
| --- | --- | --- | --- | --- |
|  | **Test Objective:** Verify a user can mark a todo item as completed | **Developer: Mikaela**  **Date tested:** 2025/04/10 | **Peer tester: William Renard**  **Date tested:** 2025/04/13 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Add a new item (“Go to Trader Joe’s”) | User input gets added to the bottom of the list | <yes/no> | yes |
| 2 | Click on the pencil icon next to the task | Pencil button is clickable and takes the user to the editing function | <yes/no> | yes |
| 3 | Make changes to the text | Text can be changed | <yes/no> | yes |
| 4 | Click save | Save button works and allows the user to save changes made to the task | <yes/no> | yes |
| 5 | Verify the task was updated on the list | The updated task is visible with the corrections in the list | <yes/no> | yes |
| **Comments** | Pencil button is clickable, input field comes up allowing the text to be changed. Changes can be saved and the task does get updated with the new text. | | | |



| Test 4 | **Delete a Todo Item** | | | |
| --- | --- | --- | --- | --- |
|  | **Test Objective:** Verify a user can delete an item from the todo list | **Developer: Mikaela**  **Date tested:** 2025/04/10 | **Peer tester:**  **Date tested:** 2025/04/13 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Click the garbage can icon next to a task | Garbage can icon is clickable | <yes/no> | yes |
| 2 | Ensure item is removed from list | Task is removed after clicking the garbage can icon | <yes/no> | yes |
| **Comments** | Garbage can icon is clickable. Clicking the icon does in fact remove the item from the list no matter the position within the list. | | | |



| Test 5 | **Prevent Submitting Blank Todo Item** | | | |
| --- | --- | --- | --- | --- |
|  | **Test Objective:** Verify that users cannot submit a blank todo item | **Developer: Mikaela**  **Date tested:** 2025/04/10 | **Peer tester: William Renard**  **Date tested:** 2025/04/13 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Navigate to the website | Ensure website loads with the task form | <yes/no> | yes |
| 2 | Leave the input field empty | Ensure that the text field is blank | <yes/no> | yes |
| 3 | Ensure “Add Task” button is inoperable unless text is added to the field | The “add task” button should not be clickable unless there is text present in the text field | <yes/no> | yes |
| **Comments** | Add Task button is indeed inoperable unless text is added. If the input field is blank like it is by default there is no way to add a task, even adding spaces or adding characters and then deleting them until it is blank again will not allow you to click the button. | | | |

