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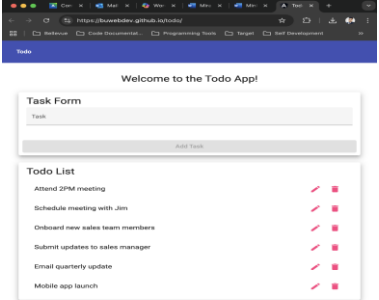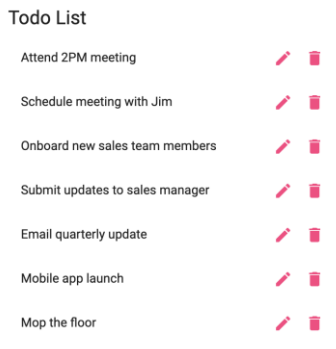
To complete the test case plan, fill out the information for Project, Course, Description and Date in the header below.

Project: TODO	
Course: CSD 380	
Description: Write Three Test Cases	
Date: 2025/04/06	

.

TABLE OF CONTENTS

Test Case Template

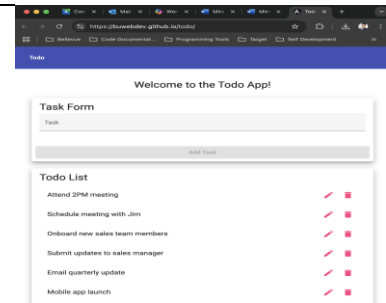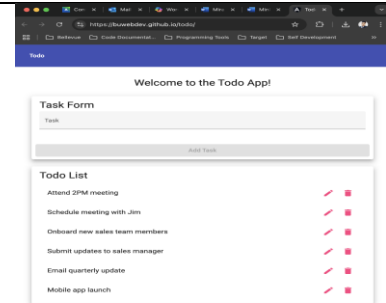| Test 1	**Create New TODO Tasks**		
	Test Objective: Verifies tasks can be added to the list. Requirements tested: 2, 6, 10.	**Developer:** Robert Minkler **Date tested:** 2025/04/06	**Peer tester: Noel Miranda Date tested:** 2025/04/08
Step	**Action**	**Expected results:**	**Developer pass/fail**	**Tester pass/fail + Screenshot**
1	Visit https://buwebdev.github.io/todo/	Confirm the application loads	Yes	pass
2	Type "Mop the floor" into the task field then click Add Task.	Confirm "Mop the floor" was added to the task list.	Yes	pass

Test Case Template

3	Ensure the task field is empty. Click the Add Task button.	Confirm that the button is disabled, and no new tasks were created by clicking on the disabled button.	Yes	pass	
4	Type "Clean the dishes" into the task field then click Add Task.	Confirm "Clean the dishes" was added to the task list and "Mop the floor" remains.	Yes	Pass	
Comments	Even though these steps pass, I believe this test case should focus on one business requirement at a time. The reason for the suggestion is because while following the actions stated, they do not fully test the business requirements and give false pass outcomes. As an example, for step 3, it does pass when following the step, but the business requirement #6 states it should prevent users from submitting a task form without entering a todo item. If I were to only include 4 white spaces and click add task, then the form was submitted without an actual todo item.				

Test Case Template

Test 2	**Provide Custom 404 Error Page**			
	Test Objective: Verifies that a custom 404 message is shown when a page not found error is thrown. Requirements tested: 5	**Developer:** Robert Minkler **Date tested:** 2025/04/06	**Peer tester:** Noel Miranda **Date tested:** 2025/04/08	
Step	**Action**	**Expected results:**	**Developer pass/fail**	**Tester pass/fail**
1	Visit https://buwebdev.github.io/todo/lost	Confirm that a custom 404 page is displayed to the client displaying "not-found works!"	Yes	not-found works! Pass
Comments	This test case passes the expected results and meets the business requirement. However, the custom 404-page message could be more aesthetically pleasing by being in the center of the page and adding some color to it. Also, adding a small comment apologizing for the error and/or adding an error image could be a nice touch.			

Test Case Template

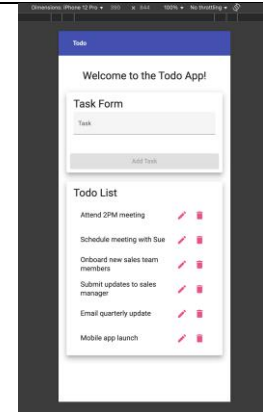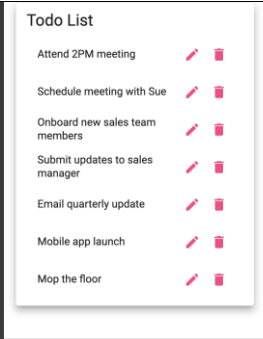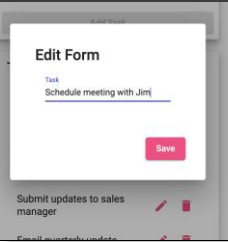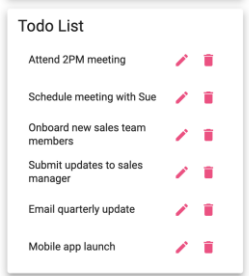| Test 3	**Edit TODO Task Items**			
	Test Objective: Verifies that task items can be edited, but not to an empty task. Requirements tested: 1, 3, 7, 10	**Developer:** Robert Minkler **Date tested:** 2025/04/06	**Peer tester:** Noel Miranda **Date tested:** 2025/04/08	
Step	**Action**	**Expected results:**	**Developer pass/fail**	**Tester pass/fail**
1	Visit https://buwebdev.github.io/todo/	Confirm the application loads	Yes	Pass
2	View the list of TODO items	Confirm the application displays a list of TODO items.	Yes	Pass

Test Case Template

3	Click the edit icon to the right of the "Schedule meeting with Jim" task.	An edit dialogue should open with "Schedule meeting with Jim"	Yes	Pass	
4	Delete the text in the edit dialogue box then click Save.	The save button should be disabled once the text field is empty, requiring a todo item entered in the edit form before saving.	No – It is possible to save an edit with no text in the todo field. Requirement 7 failed.	Fail	
5	Change the message to "Schedule meeting with Sue" then click Save.	The edit dialogue box should close. Confirm that "Schedule meeting with Jim" has been replaced with "Schedule meeting with Sue".	Yes – Must skip step 4 until issue is resolved.	Pass	
Comments	Step 1 (business requirement 10) was already tested in test case one, therefore it is not necessary because it might not be the most efficient approach in terms of resource utilization. Also, this test case is testing way too many business requirements without intensively testing all the possibilities of each functionality. Based on the steps listed, everything passed except for step 4, which allows for empty fields to be submitted through the edit form. Adding a check for empty/whitespace in the edit form will help.				

Test Case Template

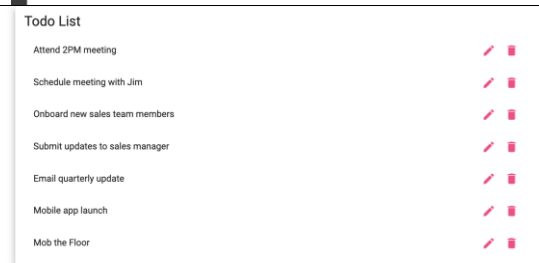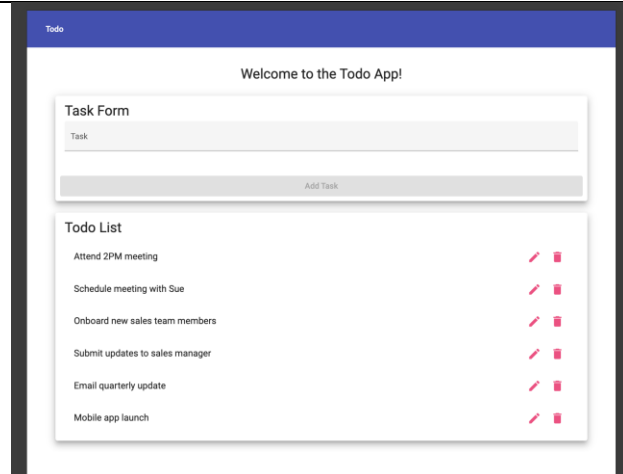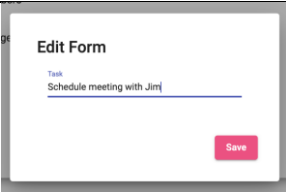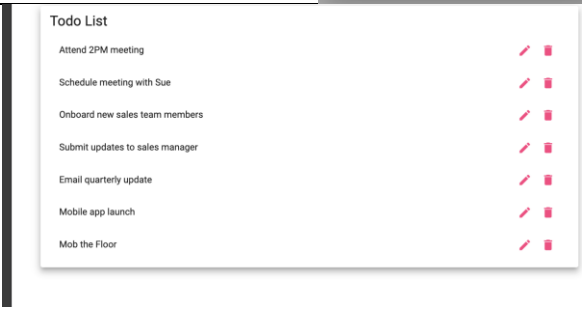| Test 4	**Test Mobile View**			
	Test Objective: Verifies that the application works on a mobile device. Confirm controls and dialogue boxes appear properly on screen and function as they should on a touch screen device. Requirements tested: 2, 3, 8, 10	**Developer:** Robert Minkler **Date tested:** 2025/04/07	**Peer tester:** Noel Miranda **Date tested:** 2025/04/08	
Step	**Action**	**Expected results:**	**Developer pass/fail**	**Tester pass/fail**
1	Visit https://buwebdev.github.io/todo/ on a mobile device.	Confirm the application loads and is formatted to fit onto the screen of a mobile device.	Yes	Pass

Test Case Template

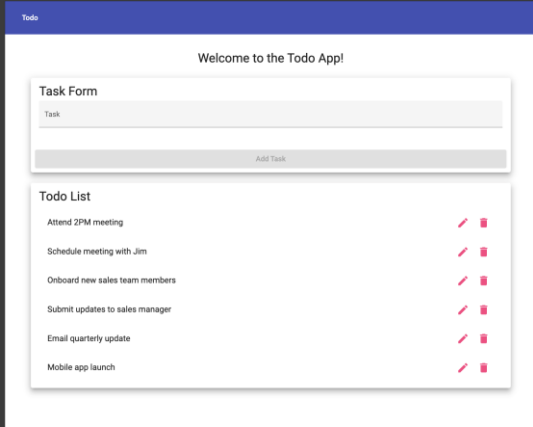| 2	Type "Mop the floor" into the task field then click Add Task.	Confirm "Mop the floor" was added to the task list.	Yes	Pass	
3	Click the edit icon to the right of the "Schedule meeting with Jim" task.	An edit dialogue should open with "Schedule meeting with Jim"	Yes	Pass	
4	Change the message to "Schedule meeting with Sue" then click Save.	The edit dialogue box should close. Confirm that "Schedule meeting with Jim" has been replaced with "Schedule meeting with Sue".	Yes	Pass	
Comments	The Todo application is successfully accessible on mobile phones based on this test case. However, the same issues persist from the browser tests. As the tester, I was able to test it through multiple phone dimensions due to the browser's developer tools.				

Test Case Template

Test 5	**Test Tablet View**			
	Test Objective: Verifies that the application works on a tablet device. Requirements tested: 2, 3, 9, 10	**Developer:** Robert Minkler **Date tested:** 2025/04/07	**Peer tester: Noel Miranda** **Date tested:** 2025/04/08	
Step	**Action**	**Expected results:**	**Developer pass/fail**	**Tester pass/fail**
1	Visit https://buwebdev.github.io/todo/ on a tablet.	Confirm the application loads and is formatted to fit onto the screen of a tablet.	Yes	Pass
2	Type "Mop the floor" into the task field then click Add Task.	Confirm "Mop the floor" was added to the task list.	Yes	Pass

Test Case Template

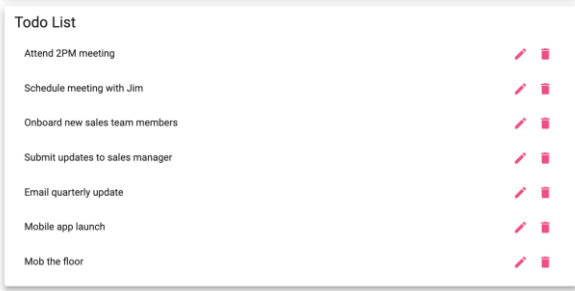| 3	Click the edit icon to the right of the "Schedule meeting with Jim" task.	An edit dialogue should open with "Schedule meeting with Jim"	Yes	Pass	
4	Change the message to "Schedule meeting with Sue" then click Save.	The edit dialogue box should close. Confirm that "Schedule meeting with Jim" has been replaced with "Schedule meeting with Sue".	Yes	Pass	
Comments	The Todo application is successfully accessible on tablets based on this test case. However, the same issues persist from the browser and mobile phone tests. As the tester, I was able to test it through multiple tablet dimensions due to the browser's developer tools. The only one that displayed it too stretched out was the iPad Pro. The interface may need some adjustments to fit more pleasingly on wider table dimensions.				

Test Case Template

Test 6	**Test Delete Task**			
	Test Objective: Verifies that tasks can be deleted. Requirements tested: 2, 4, 10	**Developer:** Robert Minkler **Date tested:** 2025/04/07	**Peer tester: Noel Miranda** **Date tested:** 2025/04/08	
Step	**Action**	**Expected results:**	**Developer pass/fail**	**Tester pass/fail**
1	Visit https://buwebdev.github.io/todo/ on a tablet.	Confirm the application loads.	Yes	Pass
2	Type "Mop the floor" into the task field then click Add Task.	Confirm "Mop the floor" was added to the task list.	Yes	Pass

Test Case Template

3	Click the trash icon next to each task item.	Verify that the task item is removed when the delete button is clicked. Confirm items delete until there are no items in the list.	Yes	Pass	Todo List
Comments	Once again, there are many repetitive tests of business requirements in this test case. Also, is the tablet the main target audience of this application or why is the delete feature required to be tested on the tablet only as stated by step 1? Other than that, this test case does prove the application is able to delete task items effectively. However, I would suggest prompting the user to confirm if they are sure they want to delete a Todo item before proceeding. This could lead to unintentional mistakes, thus impacting the user experience.				