

Test Case Assignment

Project: Todo Application	
Course: CSD-380 Web Development	
Description: Functional tests for the core features of the todo application. These test verify that users can interact with the todo list to create, mark, delete tasks and ensure page access.	
Date: 09/08/2024	

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

TABLE OF CONTENTS

The table of contents is empty because you aren't using the paragraph styles set to appear in it.

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 Case Assignment

Test 1	Accessing the Website			
	Test Objective: Ensure that users can access the Todo website without errors.	Developer: Date tested: <yyyy/mm/dd>	Peer tester: Date tested: <yyyy/mm/dd>	
Step	Action	Expected results:	Developer pass/fail	Tester pass/fail + Screenshot
1	Visit https://buwebdev.github.io/todo/	The website loads without error.	<yes/no>	<yes/no>
2	Refresh the browser	The page reloads correctly	<yes/no>	<yes/no>
3	<Enter an action>	<Enter the expected results>	<yes/no>	<yes/no>
4	<Enter an action>	<Enter the expected results>	<yes/no>	<yes/no>
5	<Enter an action>	<Enter the expected results>	<yes/no>	<yes/no>
Comments	Website access worked smoothly with no errors during testing.Refreshing the page confirmed the site reloads without issues. Simple functionality but essential for user experience.			
Test 2	Adding a new Todo Item			
	Test Objective: Confirm that users can add new tasks to the todo list.	Developer: Date tested: <yyyy/mm/dd>	Peer tester: Date tested: <yyyy/mm/dd>	
Step	Action	Expected results:	Developer pass/fail	Tester pass/fail
1	Click on the input box and type a task (e.g., "Walk the dog")	Task is added to the input box	<yes/no>	<yes/no>
2	Press "Add" or hit Enter	Task appears in the todo list below	<yes/no>	<yes/no>
3	<Enter an action>	<Enter the expected results>	<yes/no>	<yes/no>
4	<Enter an action>	<Enter the expected results>	<yes/no>	<yes/no>
5	<Enter an action>	<Enter the expected results>	<yes/no>	<yes/no>
Comments	Comments should be substantive; this means there should be at least 2-3 well-structured sentences with constructive feedback.			

Test Case Assignment

Test 3	Marking a Task as completed			
	Test Objective: Verify that users can mark tasks as completed.	Developer: Date tested: <yyyy/mm/dd>	Peer tester: Date tested: <yyyy/mm/dd>	
Step	Action	Expected results:	Developer pass/fail	Tester pass/fail
1	Select an existing task and click the checkbox next to it	The task is marked as completed (crossed out or grayed out)	<yes/no>	<yes/no>
2	<Enter an action>	<Enter the expected results>	<yes/no>	<yes/no>
3	<Enter an action>	<Enter the expected results>	<yes/no>	<yes/no>
4	<Enter an action>	<Enter the expected results>	<yes/no>	<yes/no>
5	<Enter an action>	<Enter the expected results>	<yes/no>	<yes/no>
Comments	Comments should be substantive; this means there should be at least 2-3 well-structured sentences with constructive feedback.			

Test 4	Deleting a task			
	Test Objective: Ensure that users can delete tasks from the list	Developer: Date tested: <yyyy/mm/dd>	Peer tester: Date tested: <yyyy/mm/dd>	
Step	Action	Expected results:	Developer pass/fail	Tester pass/fail
1	Select a task and click the delete (trash) icon	The task is removed from the list	<yes/no>	<yes/no>
2	<Enter an action>	<Enter the expected results>	<yes/no>	<yes/no>
3	<Enter an action>	<Enter the expected results>	<yes/no>	<yes/no>
4	<Enter an action>	<Enter the expected results>	<yes/no>	<yes/no>
5	<Enter an action>	<Enter the expected results>	<yes/no>	<yes/no>
Comments	Comments should be substantive; this means there should be at least 2-3 well-structured sentences with constructive feedback.			

Test Case Assignment

Test 5	Task Persistence After Refresh			
	Test Objective: Check the tasks remain in the list after refreshing the page.	Developer: Date tested: <yyyy/mm/dd>	Peer tester: Date tested: <yyyy/mm/dd>	
Step	Action	Expected results:	Developer pass/fail	Tester pass/fail
1	Add two tasks to the list (e.g., "Buy groceries" and "Check emails")	Tasks are added successfully	<yes/no>	<yes/no>
2	Refresh the page	The tasks remain in the list after	<yes/no>	<yes/no>
3	<Enter an action>	<Enter the expected results>	<yes/no>	<yes/no>
4	<Enter an action>	<Enter the expected results>	<yes/no>	<yes/no>
5	<Enter an action>	<Enter the expected results>	<yes/no>	<yes/no>
Comments	Comments should be substantive; this means there should be at least 2-3 well-structured sentences with constructive feedback.			