

Test Case Template

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

Project: Module 4.2 Assignment	
Course: CSD380-M320 DevOps	
Description: Test Plan	
Date: 2025/04/10	

TABLE OF CONTENTS

Create New Todo Item	3
View Todo Task List	4
Edit Todo Item	5
Delete Todo Item	6
404 Page Handling	8


Test Case Template

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 Template

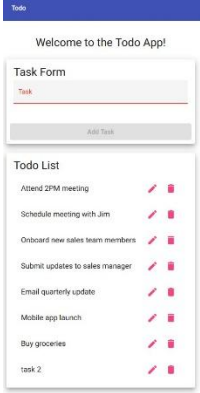
Test 1	Create New Todo Item			
	Test Objective: Verifies a user can create a new todo task item	Developer: Seth Glover Date tested: : 2025/04/10	Peer tester: Tazmin Somerville Date tested: 2025/04/13	
Step	Action	Expected results:	Developer pass/fail	Tester pass/fail + Screenshot
1	Navigate to https://buwebdev.github.io/todo/	Landing page loads successfully	yes	yes
2	Enter "Buy groceries" into the new task input field	Input field accepts text	yes	yes
3	Click the "Add" button	Task is added to the	yes	yes

Test Case Template

		todo list		
4	View the newly added task	"Buy groceries" is displayed in the task list	yes	yes 
5	Refresh the page	The task remains in the list (if persistent) or clears if temporary	Yes (temporary)	yes (temporary)
Comments	The create functionality works smoothly. Task input is intuitive and updates in real time. We may want to consider adding a confirmation message or brief animation upon successful addition.			
Comments (peer)	Confirmed & tested			





Test 2	View Todo Task List		
	Test Objective:	Developer: Date tested: 2025/04/10	Peer tester: Tazmin Somerville Date tested: 2025/04/13

Test Case Template

Step	Action	Expected results:	Developer pass/fail	Tester pass/fail
1	Navigate to https://buwebdev.github.io/todo/	Todo list interface loads	yes	yes
2	Observe the display area	Task list (if any) appears on screen	yes	yes
3	Add multiple tasks	All added tasks appear in list format	yes	yes
4	Scroll or resize browser	Task list layout adjusts responsively	yes	yes 
5	View on multiple browsers (Chrome, Firefox)	List displays consistently across platforms	yes	yes
Comments	Task list rendering is responsive and reliable. Minor styling improvements could further enhance cross-device readability, especially on smaller mobile screens.			
Comments(peer)	renders fine on mobile & different browsers			



Test 3	Edit Todo Item		
	Test Objective:	Developer: Date tested: 2025/04/10	Peer tester: Tazmin Somerville Date tested: 2025/04/13

Test Case Template


Step	Action	Expected results:	Developer pass/fail	Tester pass/fail
1	Select a task and click "Edit"	Edit form appears	yes	yes
2	Clear the input field	Input becomes empty	yes	yes
3	Attempt to submit empty form	System prevents submission	no	no
4	Enter "Buy snacks" and submit	Task updates to new value	yes	yes Buy snacks   Onboard new sales team members  
5	Confirm updated task is displayed	New text is visible in the list	yes	yes
Comments	Edit validation does not block empty submissions. The user feedback could be improved with inline error messaging or highlighting. Overall, edit workflow is user-friendly and effective.			
Comments (peer)				

Test 4	Delete Todo Item		
	Test Objective:	Developer: Date tested: 2025/04/10	Peer tester: Tazmin Somerville Date tested: 2025/04/13

Test Case Template

Step	Action	Expected results:	Developer pass/fail	Tester pass/fail
1	Add task "Test Deletion"	Task appears in list	yes	yes
2	Click the Trash Bin next to the task	Task is removed from the list	yes	yes Test Deletion  
3	Refresh the page	Deleted task does not reappear	yes	yes
4	Try deleting a non-existent task (e.g., quickly deleted)	No error, clean UI, unable to double delete	yes	yes
5	Confirm UI updates without reload	Task disappears smoothly	yes	yes
Comments	The delete functionality is efficient and reacts instantly. Adding a brief confirmation popup or undo option might prevent accidental deletions.			
Comments (peer)	Confirmed delete functionality			

Test Case Template

Test 5	404 Page Handling			
	Test Objective:	Developer : Date tested: 2025/04/10	Peer tester: Tazmin Somerville Date tested: 2025/04/13	
Step	Action	Expected results:	Developer pass/fail	Tester pass/fail
1	Navigate to https://buwebdev.github.io/todo/invalidpage	404 page is displayed	yes	Yes
2	Verify custom design	Branded or styled 404 error is shown	yes	yes 
3	Check for helpful navigation (e.g., "Back to Home" link)	User can return to main page	no	no
4	Use browser back button	Returns to last valid page	yes	yes
5	Inspect page in dev tools	Custom HTML/CSS for 404 visible	yes	yes
Comments	The 404 handling is not perfect. While there is a friendly message there is not a link back to the home page which would improve user experience. Consider adding a back to home page button on the 404 page to help users.			
Comments (peer)	Recommend improvement like Developer			

Test Case Template