Todo Application Functional Test Plan

Project: Todo Application	**Developer:** Tazmin Somerville
Course: CSD 380 - DevOps	
Description: Test Plan for verifying core functionalities of the Todo Application	
Date: 2025/04/09	

TABLE OF CONTENTS

Todo Application Functional Test Plan

Test 1	**Verify Landing Page Loads (Requirement #10 + Prototype Check)**			
	Test Objective: Ensure that users can access the Todo website through its standard URL, and that the landing page loads correctly without errors.	**Developer:** Tazmin Somerville **Date tested:** 2025/04/09	**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/	Landing page loads without errors	<yes/no> - Yes	<yes/no>
2	Refresh the browser window	The page reloads correctly without errors	<yes/no> - Yes	<yes/no>
3	Inspect responsiveness on desktop browser by resizing window	Layout adjusts without breaking	<yes/no> - Yes	<yes/no>
4	Open the landing page on a mobile browser	Layout is responsive and readable	<yes/no> - Yes	<yes/no>
5	Use keyboard navigation (tab, enter) on input and buttons	Elements are accessible via keyboard	<yes/no> - Yes	<yes/no>
Comments	The landing page loaded quickly without any visible errors or delays. Refreshing the browser also reloaded the page smoothly, maintaining consistent performance. There were no display issues observed across the tested views.			

Test 2	**Create a New Todo Task Item (Requirement #2)**			
	Test Objective: Verify that users can add a new todo item via the input field on the landing page.	**Developer:** Tazmin Somerville **Date tested:** 2025/04/09	**Peer tester:** **Date tested:** <yyyy/mm/dd>	
Step	**Action**	**Expected results:**	**Developer pass/fail**	**Tester pass/fail**

Todo Application Functional Test Plan

1	Visit https://buwebdev.github.io/todo/	Landing page loads with visible input field	<yes/no> - Yes	<yes/no>
2	Click into the input field and type “New Task”	The input is accepted without errors	<yes/no> - Yes	<yes/no>
3	Click the **Add Task** button	The new task is added to the list below	<yes/no> - Yes	<yes/no>
4	Confirm the new task item displays with an edit icon and delete icon	Item is added correctly with full functionality	<yes/no> - Yes	<yes/no>
5	Refresh the browser window	Task is cleared after refresh; persistence not required by this test case	<yes/no> - Yes	<yes/no>
Comments	The task was added successfully to the list using the input field and **Add Task** button. The interface was responsive and intuitive. However, refreshing the browser causes the task to disappear, indicating that persistence (e.g., local storage or backend saving) is not implemented in this version of the app.			

Test 3	**Delete a Todo Task Item (Requirement #4)**			
	Test Objective: Verify that users can delete an existing task from the todo list.	**Developer:** Tazmin Somerville **Date tested:** 2025/04/09	**Peer tester:** **Date tested:** <yyyy/mm/dd>	
Step	**Action**	**Expected results:**	**Developer pass/fail**	**Tester pass/fail**
1	Visit https://buwebdev.github.io/todo/	Page loads with visible input and existing tasks	<yes/no> - Yes	<yes/no>
2	Add a new task named “Delete Test Task”	Task is added to the list below	<yes/no> - Yes	<yes/no>
3	Locate the “Delete” (trash can) icon next to the new task	Delete icons are visible and clickable	<yes/no> - Yes	<yes/no>
4	Click the “Delete” icon	Task is immediately removed from the list	<yes/no> - Yes	<yes/no>
5	Confirm the task does not reappear on refresh	Task is gone permanently after refresh	<yes/no> - Yes	<yes/no>

Todo Application Functional Test Plan

Comments	Deleting tasks works as expected. The delete icon was responsive and removed the task immediately upon interaction. After refreshing the page, the task remained deleted, confirming it does not persist unexpectedly. This aligns with the expected behavior for a temporary session-based list.

Test 4	**Edit a Todo Task Item (Requirement #3)**			
	Test Objective: Verify that users can edit an existing task on the list using the edit icon and update the task text.	**Developer:** Tazmin Somerville **Date tested:** 2025/04/09	**Peer tester:** **Date tested:** <yyyy/mm/dd>	
Step	**Action**	**Expected results:**	**Developer pass/fail**	**Tester pass/fail**
1	Visit https://buwebdev.github.io/todo/	Landing page loads with visible tasks	<yes/no> - Yes	<yes/no>
2	Add a task labeled “Edit Test Task”	Task is added to the list	<yes/no> - Yes	<yes/no>
3	Click the “Edit” (pencil) icon next to the new task	Task text becomes editable	<yes/no> - Yes	<yes/no>
4	Change the task text to 'This is the Edit' and click the Save button	Task is updated in the list with new text	<yes/no> - Yes	<yes/no>
5	Refresh the page	Edited task disappears (no persistence)	<yes/no> - Yes	<yes/no>
Comments	The editing feature functioned smoothly. When the pencil icon is clicked, a pop-up allows editing, and the update was applied immediately on saving with button. However, like with task creation, the edit did not persist after a page refresh, which is consistent with the app’s current design. The interface was clear and intuitive throughout the process.			

Test 5	**Invalid URL Handling (Requirement #5: Custom 404 Page)**			
	Test Objective: Verify that users are routed to a basic 404-style message when attempting to access a non-existent URL.	**Developer:** Tazmin Somerville **Date tested:** 2025/04/09	**Peer tester:** **Date tested:** <yyyy/mm/dd>	
Step	**Action**	**Expected results:**	**Developer pass/fail**	**Tester pass/fail**

Todo Application Functional Test Plan

1	Navigate to an invalid URL (e.g., https://buwebdev.github.io/todo/invalid	User is redirected to /not-found route, which displays a 404-style message	<yes/no> - Yes	<yes/no>
2	Observe the message on the /not-found page	Text "not-found works!" is displayed	<yes/no> - Yes	<yes/no>
3	Confirm the page lacks advanced styling or navigation	The page is plain, with no back or home link	<yes/no> - Yes	<yes/no>
4	Use the browser back button	User returns to the homepage (/todo) successfully	<yes/no> - Yes	<yes/no>
Comments	The app redirects to a dedicated /not-found route when an invalid URL is entered. While the message “not-found works!” confirms the route is functioning, the page is extremely basic and lacks styling or navigation aids. There's no visible back or home button, but browser back functionality works as expected. While minimal, the behavior satisfies the requirement of a working custom 404 route in an SPA context.			