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, Jessica Hall	
Course: CSD380	
Date: 01/29/2025	

TABLE OF CONTENTS

For each test, the <u>developer</u> 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 <u>peer tester</u> 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: Website Accessibility

Test Objective: Verify that users can access the Todo application via the provided URL.

Test 1	**\<DESCRIPTION\>**				
	Test Objective: Verifies input data meets requirements	**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	\<Enter an action\>	\<Enter the expected results\>		\<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.				

Step	Action	Expected Results	Developer	Date Tested	Peer Tester	Tester Pass/Fail
1	Open a web browser.	The browser opens.	yes	01/29/2025		
2	Navigate to https://buwebdev.github.io/todo/	The website loads successfully without errors.	yes	01/29/2025		
3	Refresh the browser.	The page reloads without errors.	yes	01/29/2025		

Comments:

Test Case Template

This test makes sure that the specified functionality works as intended.

Test 2: Creating a New Todo Task

Test Objective: Verify that users can create a new todo item.

Step	Action	Expected Results	Developer	Date Tested	Peer Tester	Tester Pass/Fail
1	Navigate to the Todo application.	The page loads successfully.	yes	01/29/2025		
2	Locate the input field for adding a new todo item.	The input field is visible and enabled.	yes	01/29/2025		
3	Enter 'Buy groceries' in the input field.	The text appears in the input field.	yes	01/29/2025		
4	Click the 'Add' button.	The task appears in the list.	yes	01/29/2025		

Comments:

This test checks that the specified functionality works as intended.

Test 3: Editing an Existing Todo Task

Test Objective: Verify that users can modify an existing todo item.

Step	Action	Expected Results	Developer	Date Tested	Peer Tester	Tester Pass/Fail
1	Navigate to the Todo application.	The page loads successfully.	yes	01/29/2025		
2	Click the 'Edit' button for an existing todo item.	The input field allows editing.	yes	01/29/2025		

| 3 | Change the task text to 'Buy bread instead'. | The modified text appears in the input field. | yes | 01/29/2025 | | |
| 4 | Click the 'Save' button. | The updated task appears in the list. | yes | 01/29/2025 | | |

Comments:

This test ensures that the specified functionality works as intended.

Test 4: Deleting a Todo Task

Test Objective: Verify that users can delete a todo item.

Step	Action	Expected Results	Developer	Date Tested	Peer Tester	Tester Pass/Fail
1	Navigate to the Todo application.	The page loads successfully.	yes	01/29/2025		
2	Click the 'Delete' button for an existing todo item.	The task disappears from the list.	yes	01/29/2025		

This test ensures that the specified functionality works as intended.

Test 5: Handling Empty Input Submission

Test Objective: Ensure users cannot submit an empty todo item.

Step	Action	Expected Results	Developer	Date Tested	Peer Tester	Tester Pass/Fail
1	Navigate to the Todo application.	The page loads successfully.	yes	01/29/2025		

2	Click the 'Add' button without entering any text.	An error message is displayed, and no task is added.	no	01/29/2025		
3	Click the 'Edit' button for an existing task and delete its text.	The input field allows deletion.	yes	01/29/2025		
4	Click the 'Save' button.	An error message is displayed, preventing submission.	no	01/29/2025		

Comments: This test checks that the specified functionality works as intended.

Test 6: Responsive Design Check – Mobile/Tablet Views

Test Objective: Ensure the Todo application properly adjusts to mobile and tablet devices.

Step	Action	Expected Results	Developer	Date Tested	Peer Tester	Tester Pass/Fail
1	Open the Todo application on a desktop browser.	The application loads successfully.	yes	01/29/2025		
2	Resize the browser window to mobile dimensions.	The layout adjusts correctly for mobile view.	yes	01/29/2025		
3	Resize the browser window to tablet dimensions.	The layout adjusts correctly for tablet view.	yes	01/29/2025		
4	Use a mobile device to access the Todo application.	The mobile view loads correctly.	yes	01/29/2025		

Test Case Template

| 5 | Use a tablet device to access the Todo application. | The tablet view loads correctly. | yes | 01/29/2025 | | |

Comments: This test ensures that the specified functionality works as intended.

Test 7: Invalid URL Handling - 404 Page Check

Test Objective: Verify that navigating to an invalid URL displays the custom 404 error page.

Step	Action	Expected Results	Developer	Date Tested	Peer Tester	Tester Pass/Fail
1	Open a web browser.	The browser opens.	yes	01/29/2025		
2	Navigate to https://buwebdev.github.io/todo/nopage	The custom 404 error page appears.	yes	01/29/2025		
3	Verify that the error page includes a message indicating 'Page Not Found'.	The message is displayed.	yes	01/29/2025		
4	Check for a link redirecting to the homepage.	A 'Return to Homepage' link is present.	no	01/29/2025		
5	Click the homepage link.	The user is redirected to the main Todo application page.	no	01/29/2025		

Comments: This test checksthat the specified functionality works as intended.