

## *To Do List Website*

Assuming the website supports the creation of multiple to-do lists, each list offers comprehensive functionality including the ability to add, update and delete item effectively implementing full CRUD operations.

The user interface is designed to be intuitive and user-friendly, featuring input fields for adding or editing items, buttons for saving changes or deleting items, and checkboxes next to each item to allow users to mark tasks as completed.

I utilized the website “ <https://lambdatest.github.io/sample-todo-app/> ” to execute my testing scripts, as it closely aligns with the assumed functionalities required for the test scenario. While the application supports adding and checking off items, it does not explicitly include a delete function. However, it simulates deletion by marking tasks as completed when the associated checkbox is selected, effectively treating the item as removed from the active list.

### *Objectives:*

In alignment with the seven fundamental principles of software testing, the objectives of the testing plan or strategy are as follows:

1. **Verify core functionalities** to ensure the application behaves as expected under various conditions.
2. **Ensure UI consistency and usability**, confirming that the interface is intuitive, accessible, and visually coherent across different interactions.
3. **Identify and report defects** to enhance overall software quality by detecting deviations from expected behavior and documenting them for resolution.

### Scope:

The testing approach includes **functional testing** of CRUD operations to validate the application's ability to create, read, update, and delete items accurately. Additionally, **non-functional testing** is performed to assess UI responsiveness across different screen sizes and devices. **Boundary value testing** is also conducted to verify the application's behavior with edge-case inputs, ensuring robustness and input validation.

We can also add other things in test plans such as:

<b>Features not to be tested (out of scope)</b>	<b>Test Environment</b>
<b>Entry and Exit Criteria</b>	<b>Test Levels</b>

### Test Cases:

Each test case includes the following components:

<b>Test Case ID</b>	<b>Test Case Description</b>	<b>Steps to Execute</b>
<b>Expected Result</b>	<b>Actual Result</b>	<b>Status (Pass/Fail)</b>

And we can also add other things such as:

<b>Pre-Conditions</b>	<b>Post Conditions</b>
<b>Priority</b>	<b>Date and People work on each case</b>

## 1) Functional Testing:

Test Case ID	Description	Steps to Execute	Expected Result	Actual Result	Status
FN-1	Add a new task	1. Open the application. 2. Enter a task in the input field. 3. Click the <b>"Add"</b> button.	Task appears in the list.	-	Pass / Fail
FN-2	Mark task as complete	1. Add a new task. 2. Click the checkbox next to the task.	Task is marked as "Done".	-	Pass / Fail
FN-3	Delete a task	1. Add a new task. 2. Click the <b>"Delete"</b> button (if available) or mark it as completed depending on UI behavior.	Task is removed from the list.	-	Pass / Fail

## 2) UI Testing:

Test Case ID	Description	Steps to Execute	Expected Result	Actual Result	Status
UI-1	Verify input field placeholder	1. Open the application.	Placeholder text displays: <div>Enter a new task...</div>	-	Pass / Fail
UI-2	Check button responsiveness	1. Open the application. 2. Click <b>Add</b> , <b>Edit</b> , or <b>Delete</b> button.	Immediate visual feedback indicating user action.	-	Pass / Fail
UI-3	Verify checkbox visibility/UX	1. Open the application. 2. Add a new task. 3. Locate the checkbox next to it.	Checkbox is clearly visible, clickable, and styled properly.	-	Pass / Fail

## 3) Boundary Testing:

Test Case ID	Description	Steps to Execute	Expected Result	Actual Result	Status
BD-1	Empty task submission	1. Open the application. 2. Click <b>"Add"</b> without entering any text.	Error message: <div>Task cannot be empty!</div>	-	Pass / Fail
BD-2	Duplicate task entry	1. Add a task. 2. Add the same task again.	Error or warning: <div>Task already exists!</div>	-	Pass / Fail
BD-3	Input text length limit	1. Enter a long task text (beyond allowed limit). 2. Click <b>"Add"</b> .	Input is either truncated or rejected with a warning.	-	Pass / Fail

## **Bud Report:**

Each bug listed in that report includes the following components:

<b>Bug ID &amp; Bug Title</b>	<b>Bug Description</b>	<b>Steps to Reproduce</b>
<b>Expected Result</b>	<b>Actual Result</b>	<b>Priority</b>
<b>Severity (Low/Medium/High)</b>		<b>Screenshot (if any)</b>

Bug ID	Title	Severity	Steps to Reproduce	Expected Result	Actual Result
Bug-1	Adding empty task does not show error	Minor	1. Open the application. 2. Click " <b>Add</b> " without entering text.	Error message should be displayed.	Nothing happens.
Bug-2	Delete button does not remove task	Major	1. Add a task. 2. Click " <b>Delete</b> " next to the task.	Task should be removed from the list.	Task remains in the list.