|  |  |
| --- | --- |
| **Project:** Todo application |  |
| **Course:** DevOps CSD-380 |  |
| **Description:** Todo list application |  |
| **Date:** 2024/11/13 |  |

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

.

Contents

[**Accessing and verifying the Todo Application** 2](#_Toc182416037)

[**Adding and Displaying a New Todo Item** 3](#_Toc182416038)

[**Deleting a Todo Item** 3](#_Toc182416039)

[**Marking a Todo Item as Completed** 4](#_Toc182416040)

[**Filtering Todo Items by Status** 5](#_Toc182416041)

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 1 | **Accessing and verifying the Todo Application** | | | |
|  | **Ensure that the application loads correctly and displays the expected initial content.** | **Developer:**  **Date tested:** 2024/11/14 | **Peer tester:** Melissa Lawrence  **Date tested:** 2024/11/16 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail**  **+ Screenshot** |
| 1 | Open a web browser and enter the URL for the Todo application https://buwebdev.github.io/todo/ | The Todo application loads successfully without any errors, displaying the landing page | yes | Pass |
| 2 | Observe the initial content and layout upon loading | The landing page should display an input field for adding new todo items and an empty list where todo items will appear. | yes | Pass See above Screen Shot |
| 3 | Refresh the browser window to check if the page reloads correctly. | The page should reload without any errors and retain all functional elements like the input field and list area without any changes. | yes | Pass |
| **Comments** | The application's load time and initial display are crucial for first impressions. It's important to ensure that there are no script errors or layout issues upon initial load. This test verifies basic accessibility and operational status of the application. Refreshing tests the stability of the application under repeat load conditions. ML- Page loads without error. Upon refresh no changes were observed. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 2 | **Adding and Displaying a New Todo Item** | | | |
|  | **Test Objective: Verify that users can add a new todo item and see it displayed in the list.** | **Developer:**  **Date tested:** 2024/11/14 | **Peer tester:** Melissa  **Date tested:** 2024/11/16 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Enter "Complete assignment" in the input field | The text "Complete assignment" appears in the input field. | yes | Pass |
| 2 | Click the "Add" button | "Complete assignment" is added to the list below the input field. | yes | Pass |
| 3 | Check the list for the presence of the new item. | The list contains "Complete assignment" as an active todo item. | yes | Pass See above screen shot |
| **Comments** | This test verifies if the input field is interactive and checks the application’s ability to update the content dynamically. Adding an item test both the frontend functionality and the backend's ability to handle new data inputs. This ensures that user inputs are correctly captured and displayed. ML- Text box and ‘Add task’ button are clickable. Text box allows for text input. Button action adds text to bottom of list. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 3 | **Deleting a Todo Item** | | | |
|  | **Test Objective: Confirm that users can remove a todo item from the list.** | **Developer:**  **Date tested:** 2024/11/14 | **Peer tester:** Melissa Lawrence  **Date tested:** 2024/11/16 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Add a new item named "Delete this test item." | "Delete this test item" appears in the todo list. | yes | Pass |
| 2 | Click the delete button next to "Delete this test item." | "Delete this test item" is removed from the list immediately. | yes | Pass |
| 3 | Verify the list is updated correctly. | The list no longer shows "Delete this test item"; updates reflect the item's removal. | yes | Pass  See above screen shot |
| **Comments** | Deleting functionality is critical for managing items effectively. This test checks the application’s responsiveness to user actions for removing items and updating accordingly. It's essential to ensure that the delete action does not affect other items, maintaining data integrity.  ML – Trash Can Icon is clickable. The button removed the items in the correct line of the list. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 4 | **Marking a Todo Item as Completed** | | | |
|  | **Test Objective: Test the functionality of marking a todo item as completed.** | **Developer:**  **Date tested:** 2024/11/14 | **Peer tester:** Melissa Lawrence  **Date tested:** 2024/11/16 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Add a new item named "Finish project." | "Finish project" is added to the todo list. | yes | Pass |
| 2 | Click the checkbox next to "Finish project" to mark it as completed. | The item shows a visual change (e.g., strikethrough, color change) to indicate it is completed. | yes | Fail |
| 3 | Verify the list reflects the completed status. | "Finish project" remains in the list but is visually distinct as a completed task | yes | Fail |
| **Comments** | This test assesses the application's ability to handle state changes in todo items. The visual feedback on marking an item completed is important for user satisfaction and usability. It tests both the user interface and the underlying data handling mechanisms.  ML – Tester does not see a checkbox next to listed items. There are only three items in each line, the text of the ‘To Do List’, the pencil/edit icon, and the trashcan/delete icon. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 5 | **Filtering Todo Items by Status** | | | |
|  | **Test Objective: Ensure the filter functionality works correctly for displaying todos based on their status.** | **Developer:**  **Date tested:** 2024/11/14 | **Peer tester:** Melissa Lawrence  **Date tested:** 2024/11/16 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Ensure there are multiple items, some marked as completed. | The list displays both active and completed items. | yes | Fail |
| 2 | Use the filter options to select "Active." | Only active (not completed) items are displayed. | yes | Fail |
| 3 | Switch the filter to "Completed." | Only completed items are displayed. | yes | Fail |
| **Comments** | Filter functionality enhances the user experience by allowing users to view subsets of data. This test ensures that the filter controls are effective and accurate. It checks the application’s capability to segregate data based on predefined criteria, which is vital for users dealing with many items.  ML- Items in List cannot be marked as completed. Tester does not see a filter option. | | | |