|  |  |
| --- | --- |
| **Project:** Todo Business Requirements | Sarah Ewing |
| **Course:** DevOps - CSD380 | Module 4.2 Assignment – Test Plan |
| **Description:** Todo is a business application focused on the monitoring of task items. Users of this system can create, update, and delete items from their “Todo List.” Views have been created to support most major devices and web browsers. |  |
| **Date:** 2024-11-13 |  |

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

.

**TABLE OF CONTENTS**

[Create a New To Do Task Item 2](#_Toc1366690128)

[Deleting a To Do List Item 2](#_Toc2133129637)

[Prevent Submission of Empty Task Form 3](#_Toc475398336)

[Editing a To Do Task Item 3](#_Toc1808011540)

[Custom 404 Error Page for Invalid URLs 4](#_Toc455704953)

For each test, the developer should: provide a test description, a test objective, the developer's 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 | **Create a New To Do Task Item** | | | |
|  | **Test Objective:** Verify that users can create a new to do task item | **Developer:** Sarah Ewing  **Date tested:** 2024-11-13 | **Peer tester: Margaret Shimerdla**  **Date tested:** 11/14/2024 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail**  **+ Screenshot** |
| 1 | Visit https://buwebdev.github.io/todo/ | The page loads without errors | yes | Yes |
| 2 | Click on the text box to enter a task | Cursor appears in text box | yes | Yes |
| 3 | Fill in the text of the To Do item | The text box accepts input | yes | Yes |
| 4 | Click the ‘Add Task’ button | The task is submitted, and the form is cleared | yes | Yes |
| 5 | Confirm that the new task appears in the list of to do items | The newly added task is visible in the to do list | yes | Yes |
| **Comments** | Creating a to-do task can be very in depth. Following this action list step by step is very easy. At the end of each step it showed that the action had passed according to what was to be done. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 2 | **Deleting a To Do List Item** | | | |
|  | **Test Objective:** Verify that users can delete a to do list item | **Developer:** Sarah Ewing  **Date tested:** 2024-11-13 | **Peer tester: Margaret Shimerdla**  **Date tested:** 11/14/2024 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Visit https://buwebdev.github.io/todo/ | The page loads without errors | yes | Yes |
| 2 | Click on the Trash Can icon to the right of one of the tasks | The task to the left of the Trash Can icon disappears | yes | Yes |
| 3 | Verify that the item no longer appears in the list of to do tasks | The deleted task is removed from the list | yes | Yes |
| **Comments** | This test shows how to delete a task on the list. Getting to the page is pretty straightforward as long as you have the correct URL. Then just deciding which task to delete and pressing the trashcan button. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 3 | **Prevent Submission of Empty Task Form** | | | |
|  | **Test Objective:** Verify that users cannot submit a task form without entering a to do item | **Developer:** Sarah Ewing  **Date tested:** 2024-11-13 | **Peer tester: Margaret Shimerdla**  **Date tested:** 11/14/2024 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Visit https://buwebdev.github.io/todo/ | The page loads without errors | yes | Yes |
| 2 | Click on the text box to enter a task | Cursor appears in text box | yes | Yes |
| 3 | Leave the task empty | The task remains empty | yes | Yes |
| 4 | Click away from the text box | The word ‘Task’ and the line under the text box turns red indicating the task is a required field | yes | Yes |
| 5 | Click the ‘Add Task’ button | The button will be greyed out and unable to be pressed | yes | Yes |
| **Comments** | This test shows that a task has to be entered in order for somebody to add a task. If the task line is blank then the add task button will be greyed out. When trying to click the add task button is not possible if blank. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 4 | **Editing a To Do Task Item** | | | |
|  | **Test Objective:** Verify that users can edit an existing to do task item | **Developer:** Sarah Ewing  **Date tested:** 2024-11-13 | **Peer tester: Margaret Shimerdla**  **Date tested:** 11/14/2024 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Visit https://buwebdev.github.io/todo/ | The page loads without errors | yes | Yes |
| 2 | Select the Pencil icon to the right of one of the to do list items | The To do list item appears in an ‘Edit Form’ box | yes | Yes |
| 3 | Edit the To Do list item | The text box allows the text to be edited | yes | Yes |
| 4 | Click the ‘Save’ button | The task is saved and the ‘Edit Form’ box disappears | yes | Yes |
| 5 | Confirm that the edited task appears in the list of to do items | The edited task is visible in the to do list | yes | Yes |
| **Comments** | This was a great task. It was very specific on how to edit a task that has already been entered. The actions that were written on this list were very specific . | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 5 | **Custom 404 Error Page for Invalid URLs** | | | |
|  | **Test Objective:** Verify that users are provided with a custom 404 error page for invalid URLs | **Developer:** Sarah Ewing  **Date tested:** 2024-11-13 | **Peer tester: Margaret Shimerdla**  **Date tested:** 11/14/2024 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Visit https://buwebdev.github.io/todo/badurl | The page loads with custom message | yes | Yes |
| 2 | Confirm that the 404 error page provides a message indicating the page is not found | The page displays the message: ‘not-found works!’ | yes | Yes |
| 3 | Visit https://buwebdev.github.io/todo/anotherbadurl | The page loads with custom message | yes | Yes |
| 4 | Confirm that the 404 error page provides a message indicating the page is not found | The page displays the message: ‘not-found works!’ | yes | Yes |
| **Comments** | The error message URL works for both the first and second incorrect URL. This shows that if you enter an incorrect URL then you will be sent an error message. Checking if the URL is incorrect and will give error is important. But checking if URL is correct is important too. | | | |