|  |  |
| --- | --- |
| **Project:** | To Do App Test Cases |
| **Course:** | **CSD380-H306 DevOps (2255-DD)** |
| **Description:** | Creating Test plans for a To Do App site. |
| **Date:** | 2025/02/02 |

**TABLE OF CONTENTS**

[**View a list of todo task items** 2](#_Toc189410256)

[**Edit a Todo task item** 3](#_Toc189410257)

[**Invalid URL’s provide a custom 404 error page** 3](#_Toc189410258)

[**Prevent users from submitting the edit form without entering a Todo item** 4](#_Toc189410259)

[**Provide users a with a view for mobile devices** 4](#_Toc189410260)

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 | **View a list of todo task items** | | | |
|  | **Test Objective: Verify that you can view the Todo task items.** | **Developer:**  **Date tested:** 2025/02/02 | **Peer tester:** Jacob Darling  **Date tested:** 2025/02/02 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail**  **+ Screenshot** |
| 1 | Navigate to https://buwebdev.github.io/todo | Web page should load | yes | yes (See Figure A.1) |
| 2 | Check the formatting of Todo list |  | yes | yes |
| **Comments** | Website should load without any issues and show the site above. Overall everything worked and the page formatting is approachable and easy to comprehend. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 2 | **Edit a Todo task item** | | | |
|  | **Test Objective: Check that the edit function works.** | **Developer:**  **Date tested:** 2025/02/02 | **Peer tester:** Jacob Darling  **Date tested:** 2025/02/02 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Navigate to https://buwebdev.github.io/todo | Web page should load | Yes | yes (See Figure A.1) |
| 2 | Select edit icon next to the delete icon. | New edit form window opens. | Yes | Yes (See Figure B.1) |
| 3 | Confirm text in the edit box matches the line item you clicked. | Prior to changes the text should match the task text. | Yes | yes |
| 4 | Change the desired task and click save. | Changes saved and taken back to the Todo list main page | Yes | yes |
| 5 | Confirm that the edited text is correct. | Task text should match the edited text. | Yes | Yes (See Figure B.2) |
| **Comments** | Adding some form of hover text that shows what each icon means would be good for users who may not be familiar with that icon. Making it so pressing enter while on the edit page saves the changes and submits it would make for a more accessible program. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 3 | **Invalid URL’s provide a custom 404 error page** | | | |
|  | **Test Objective: Check that invalid URL’s display is a custom page.** | **Developer:**  **Date tested:** 2025/02/02 | **Peer tester:** Jacob Darling  **Date tested:** 2025/02/02 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Navigate to https://buwebdev.github.io/todo | Web page should load | Yes | yes (See Figure A.1) |
| 2 | Utilize the URL bar to navigate to a page after Todo. Example link would be: https://buwebdev.github.io/todo/okay |  | Yes | yes |
| 3 | Confirm new URL showing is correct. | Expected URL: https://buwebdev/todo/not-found | Yes | yes (See Figure C.1) |
| **Comments** | A custom 404 page should show that a page does not work so additional information should be added. I do like having a custom 404 page though as it can show that the server is active and website is accessible but the something did not load the desired page quite right. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 4 | **Prevent users from submitting the edit form without entering a Todo item** | | | |
|  | **Test Objective:** | **Developer:**  **Date tested:** 2025/02/02 | **Peer tester:** Jacob Darling  **Date tested:** 2025/02/02 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Navigate to https://buwebdev.github.io/todo | Web page should load. | Yes | yes (See Figure A.1) |
| 2 | Delete any items in the Todo form. | Should have an empty Todo list. | Yes | yes |
| 3 | Attempt to click edit without any items in the Todo list. | There should be no option to edit since there are no items in the list. | Yes | yes (See Figure D.1) |
| **Comments** | This works as intended removing the edit and delete button when an item is removed from the list preventing a user from clicking edit. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 5 | **Provide users a with a view for mobile devices** | | | |
|  | **Test Objective:** | **Developer:**  **Date tested:** 2025/02/02 | **Peer tester:** Jacob Darling  **Date tested:** 2025/02/02 | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Navigate to https://buwebdev.github.io/todo | Web page should load. | <yes/no> | yes (See Figure E.1) |
| 2 | Check for errors within the formatting | No overlapping parts of code or issues with viewing tasks. | <yes/no> | yes |
| 3 | Ensure that the page is navigable or mobile device. | Tapping to slide down the list and view different items, edit button, delete button, and add button should all work when tapped on. | <yes/no> | yes (See Figure E.2&E.3) |
| **Comments** | Site uses a responsive approach to development allowing it to be rescaled to many different sizes and formats. The site is well formatted for use on a mobile device as it is a single column allowing the column to take up the entire width without conflict. | | | |

**Screenshots:**

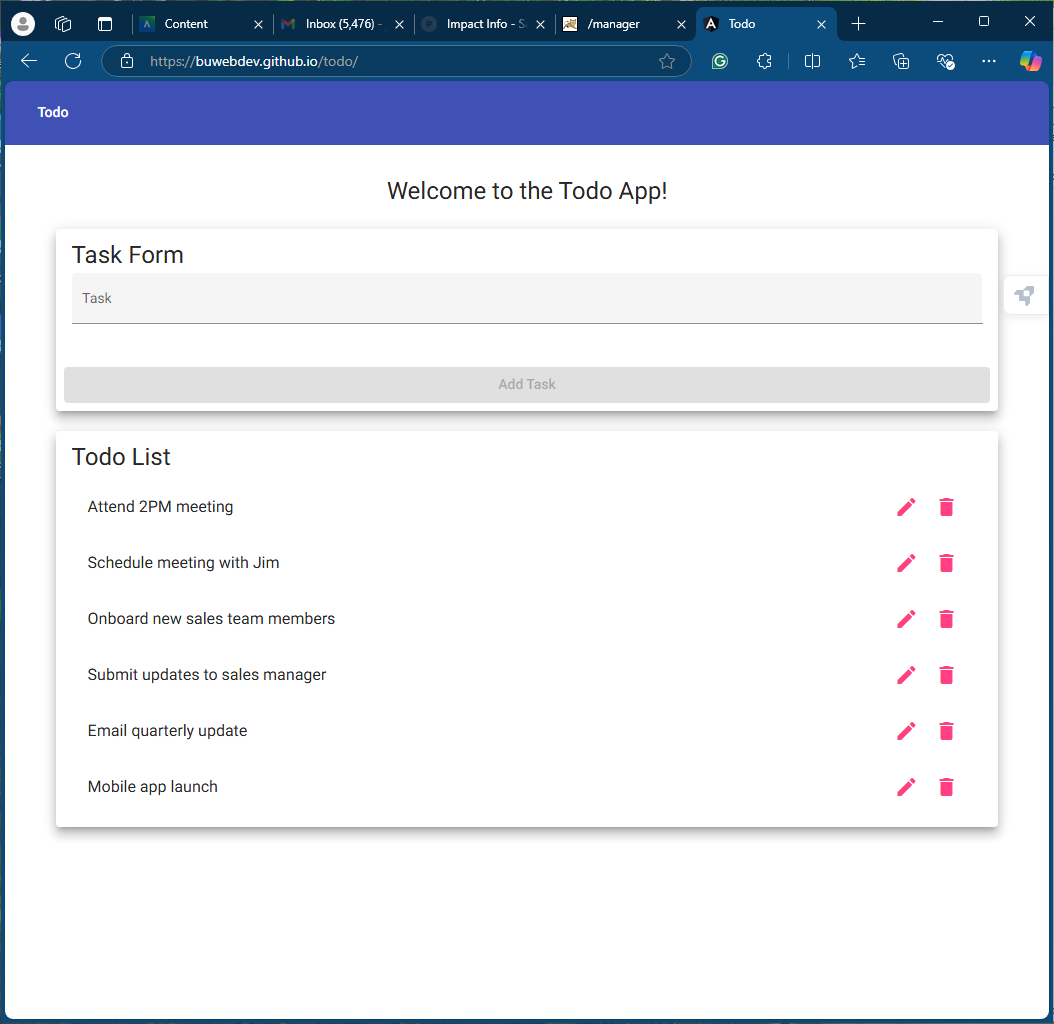
Figure A.1

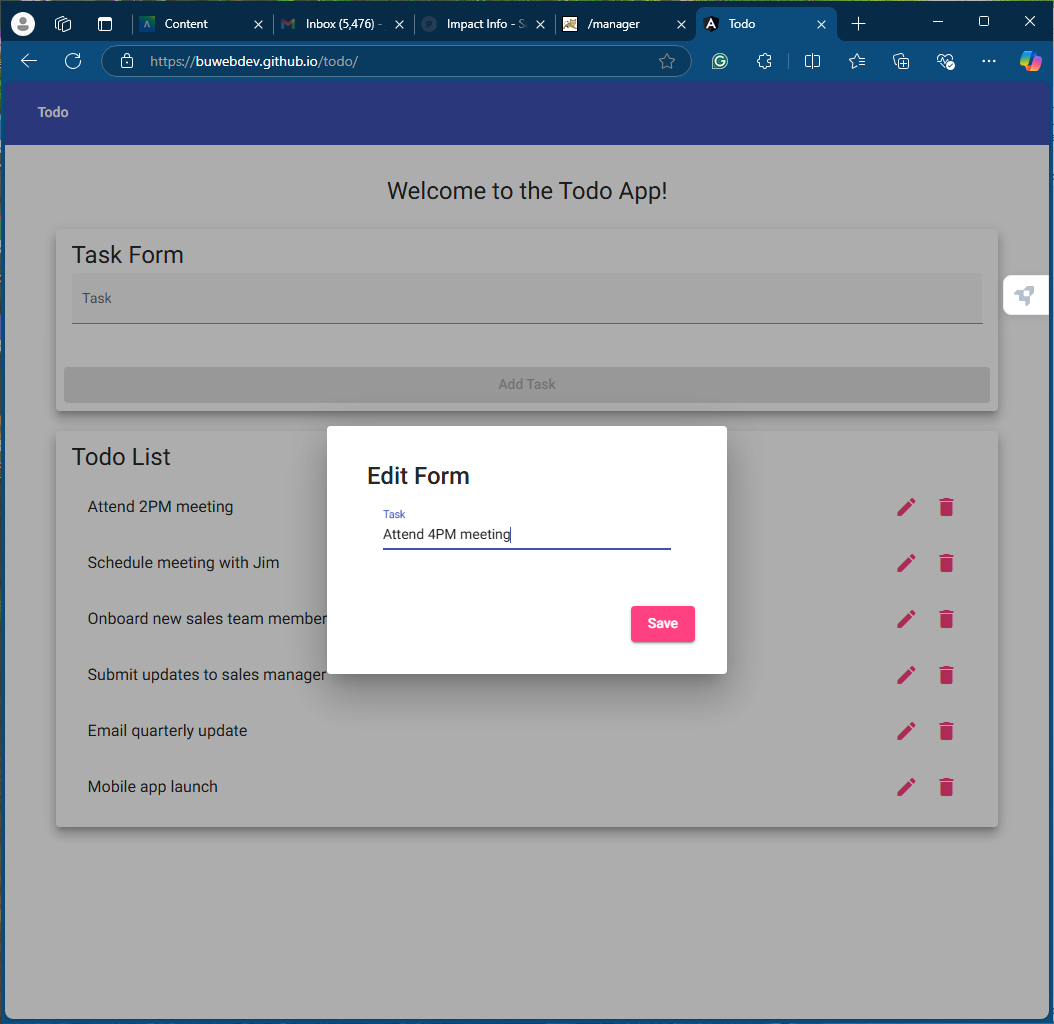
Figure B.1

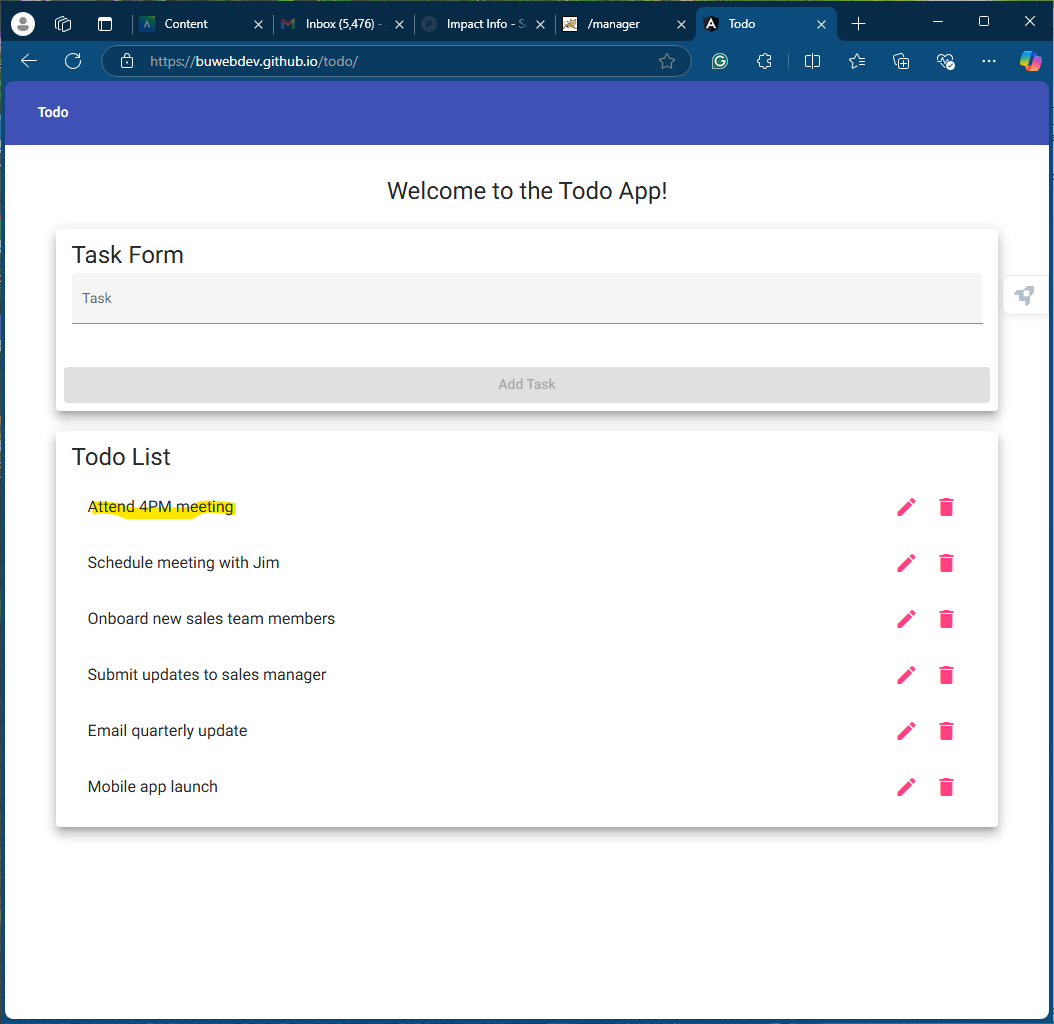
Figure B.2

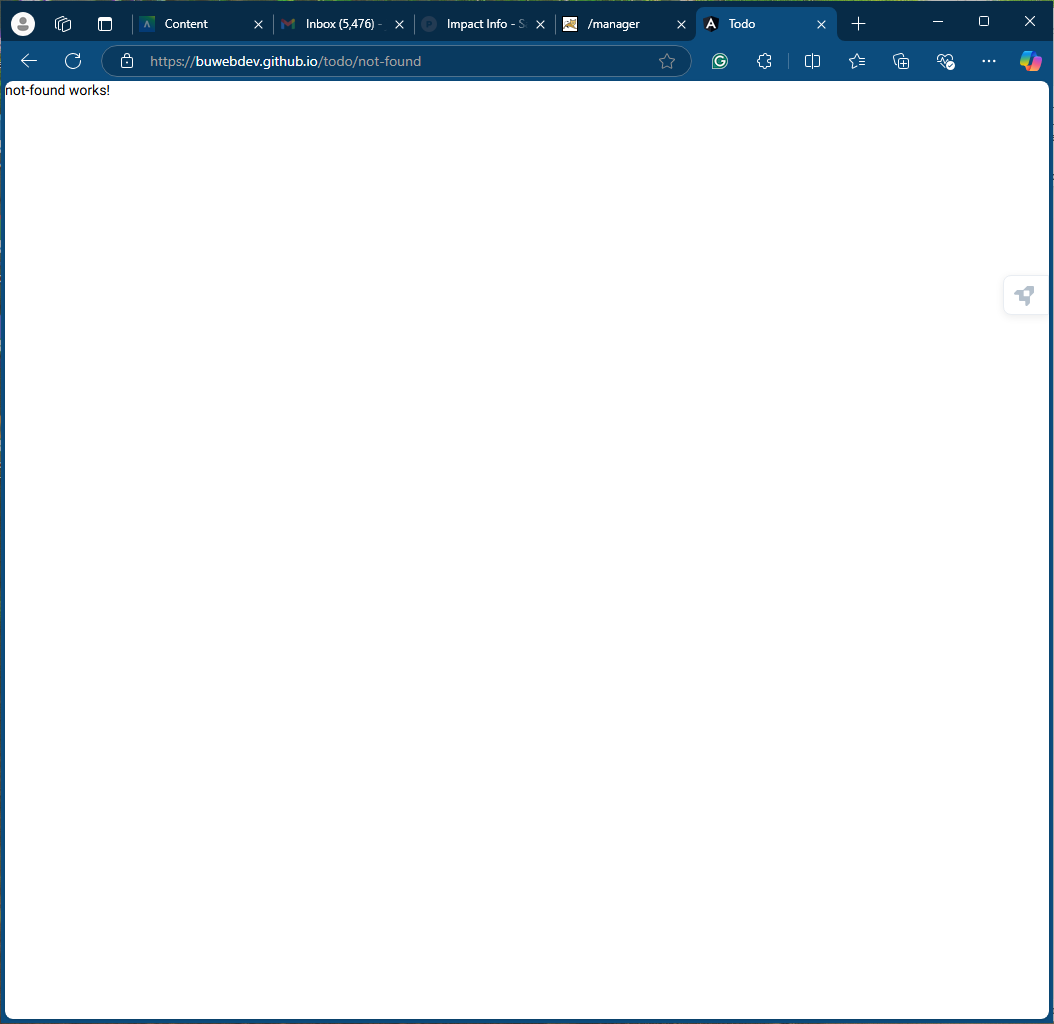
Figure C.1

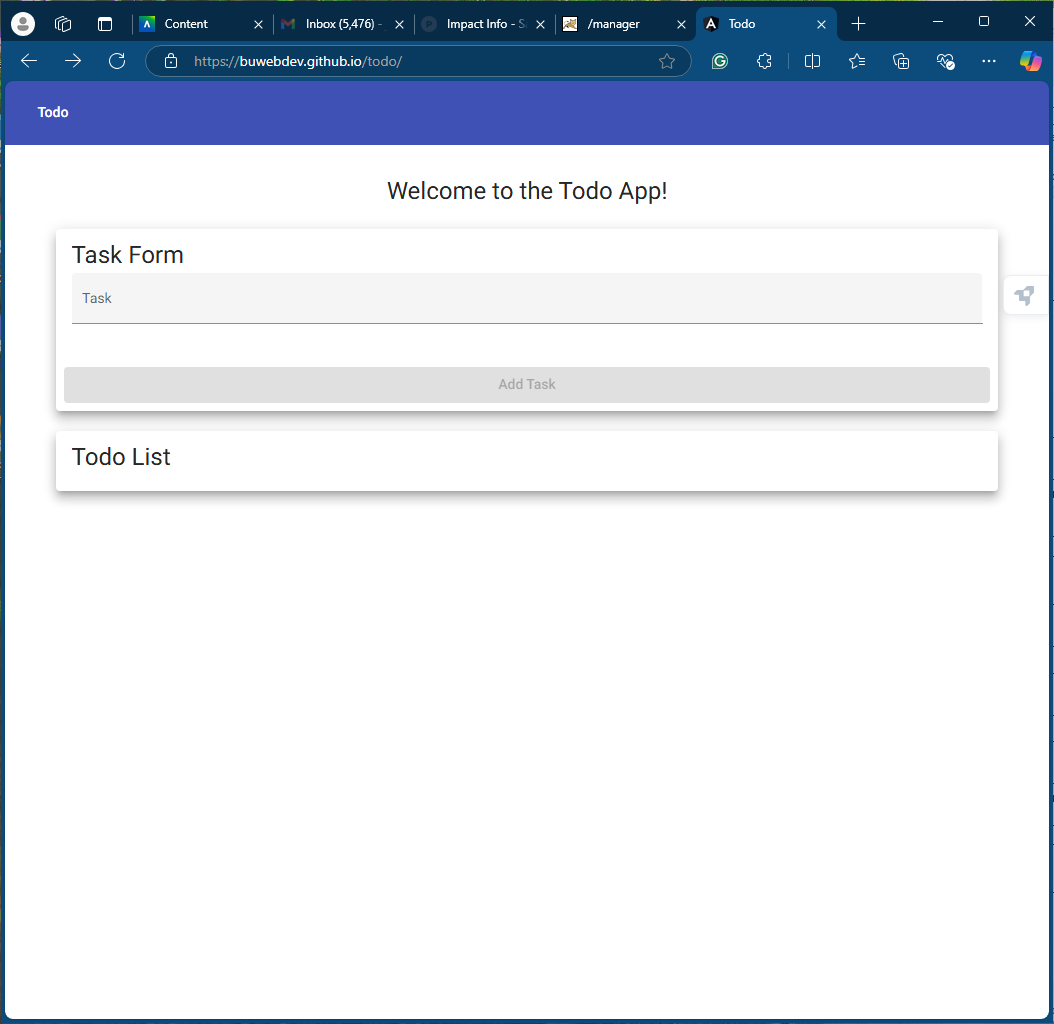
Figure D.1

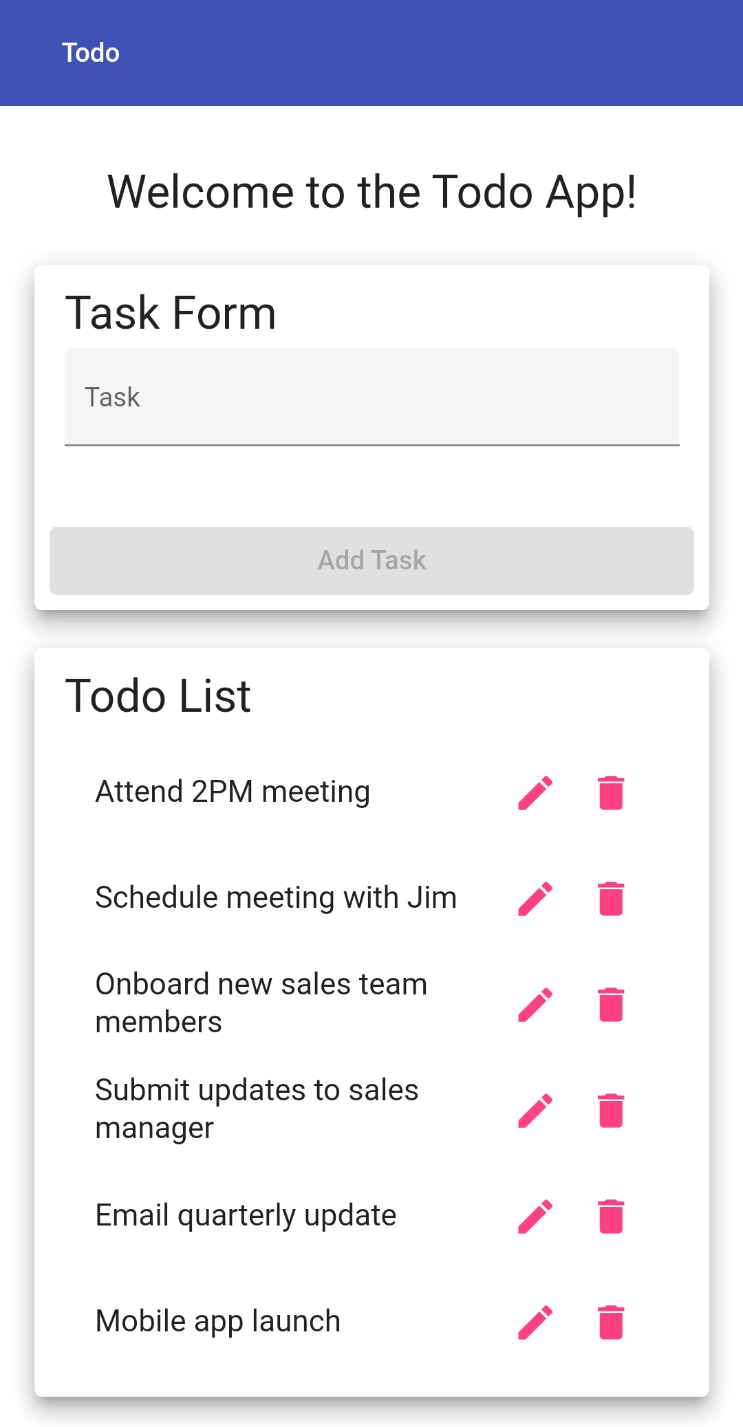
Figure E.1

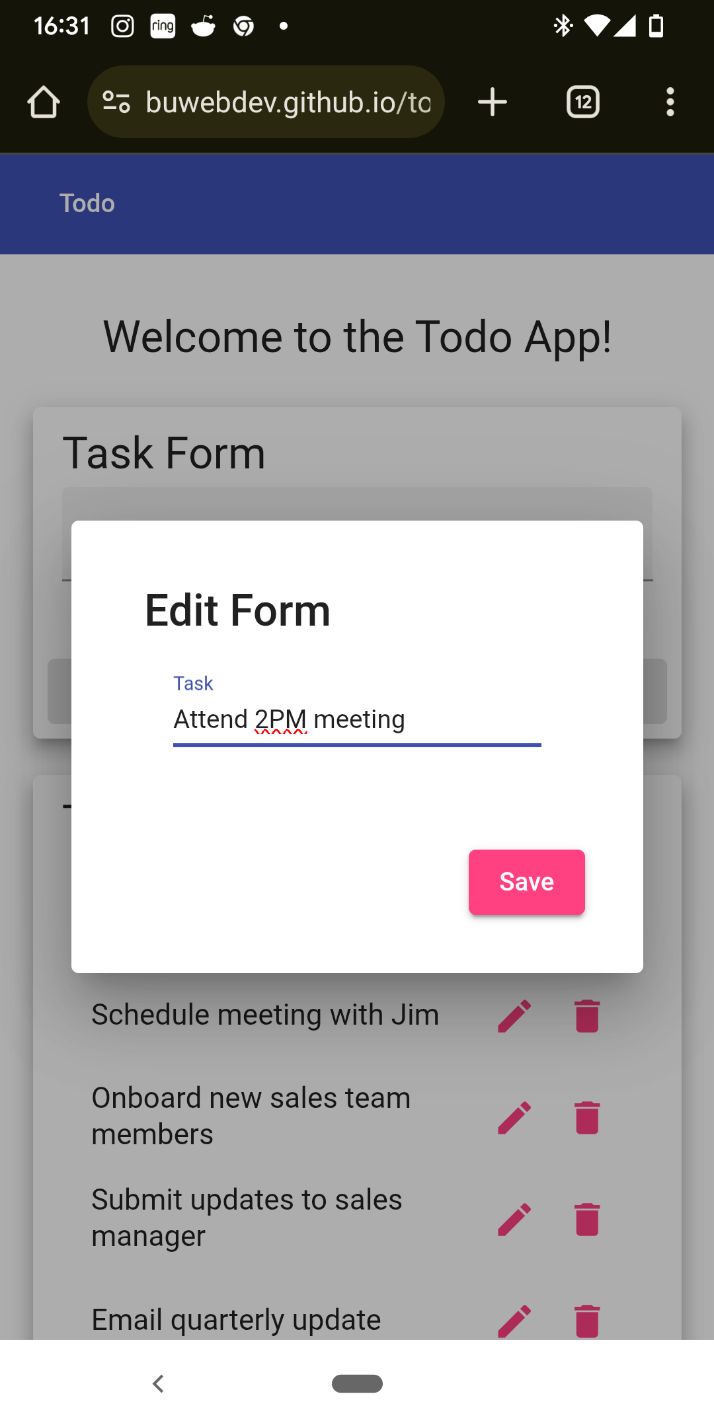
Figure E.2

Figure E.3