

## Criterion B: Design Overview

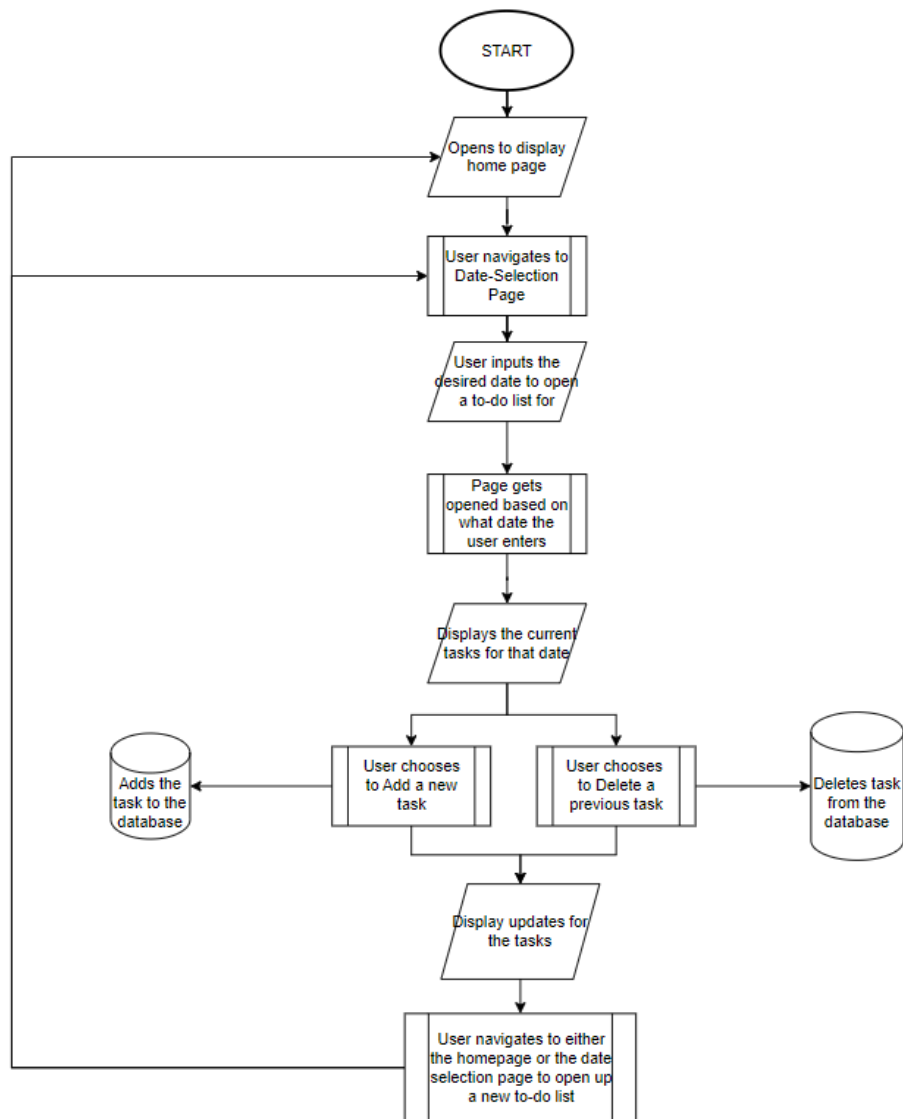


Figure 1: A flowchart that shows the overall flow of the program and how the various pages link together.

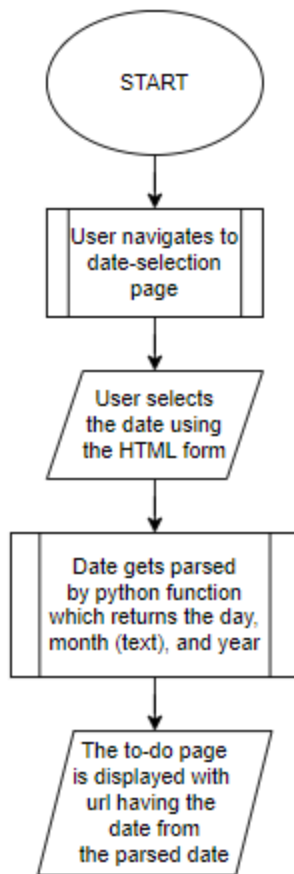


Figure 2: A flowchart that shows how a new page with the correct date is obtained.

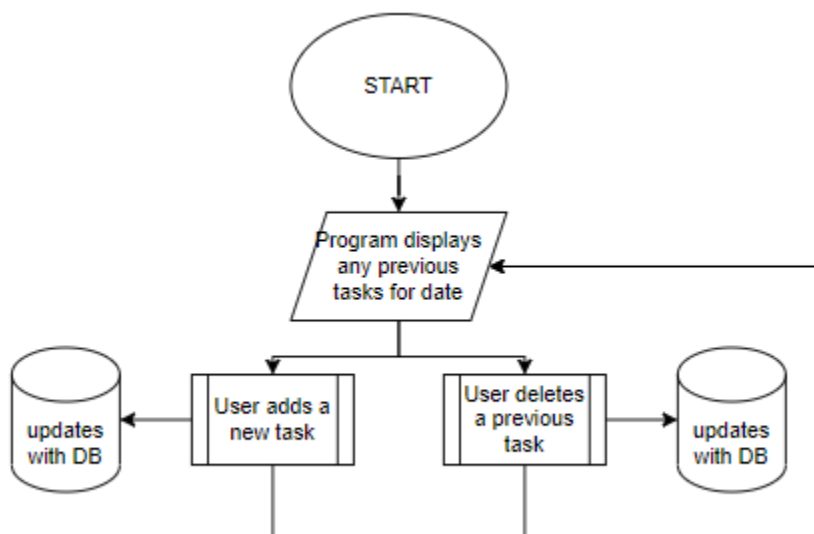


Figure 3: A flowchart of the process of entering or deleting a task.

tasks
<ul style="list-style-type: none"> <li>+ task_name: Int (Not Null)</li> <li>+ task_day: Int (Not Null)</li> <li>+ task_month: Text (Not Null)</li> <li>+ task_year: Int (Not Null)</li> </ul>

Figure 4: A UML table that shows the attributes of each task. All attributes are not null.

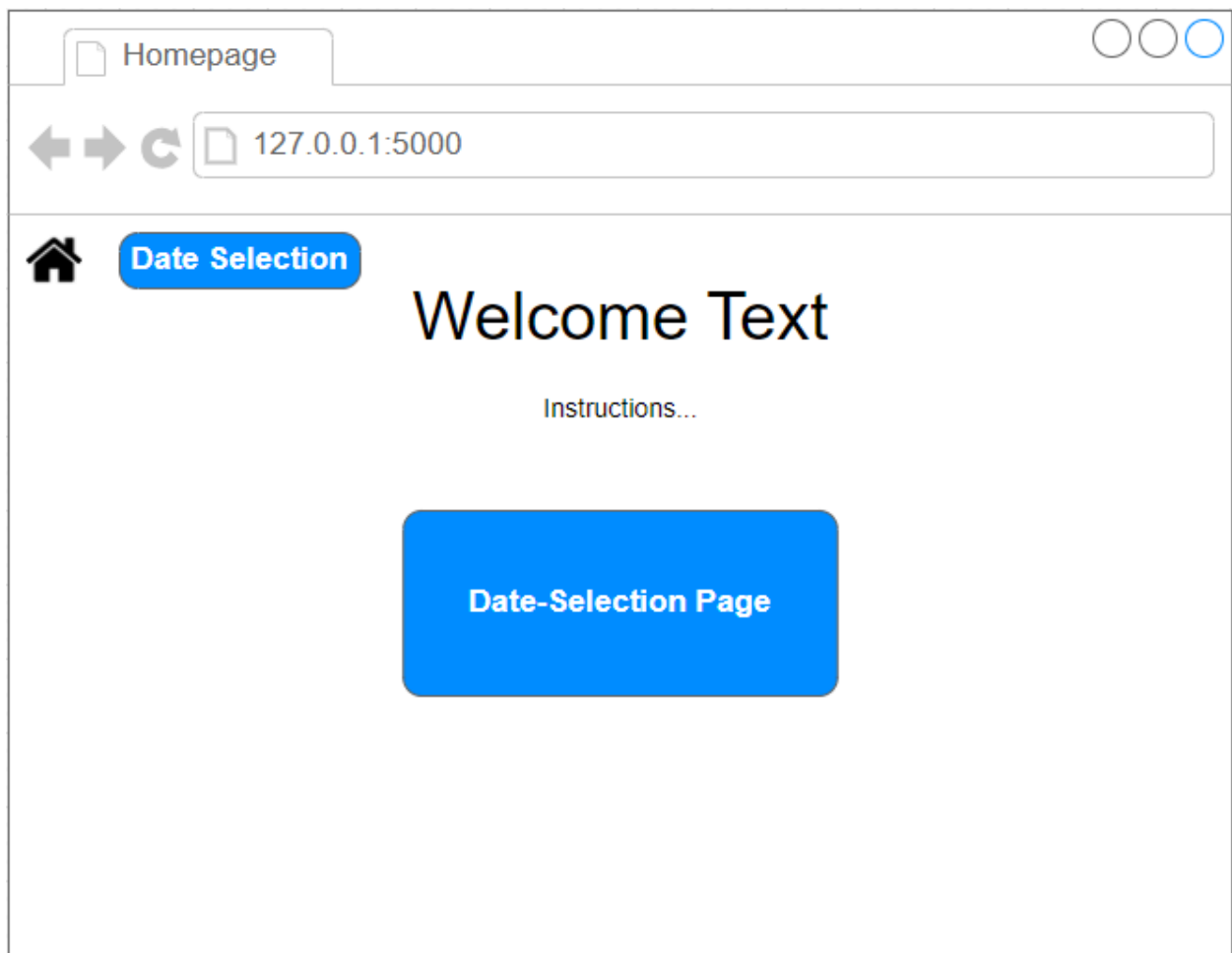


Figure 5.1: A wireframe of what the homepage will look like. It has a navigation bar, and a button in the center to direct the user to the date-selection page.

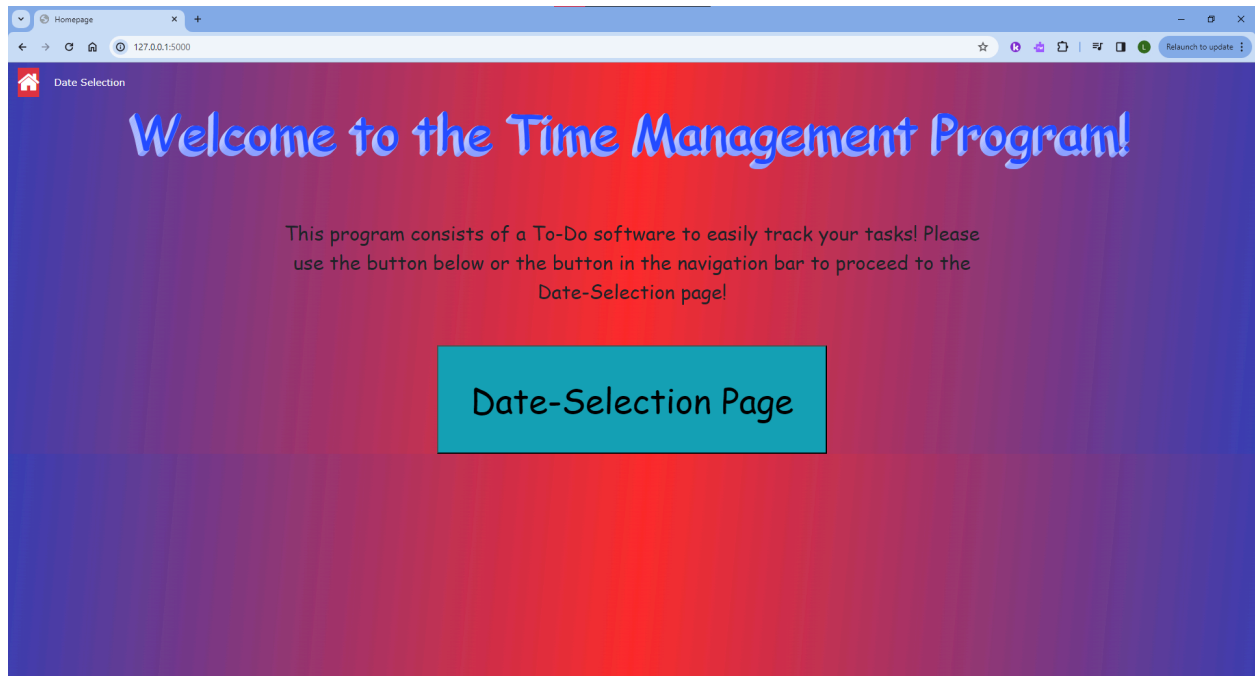


Figure 5.2: A screenshot from the actual program after completing the project, showcasing the homepage.

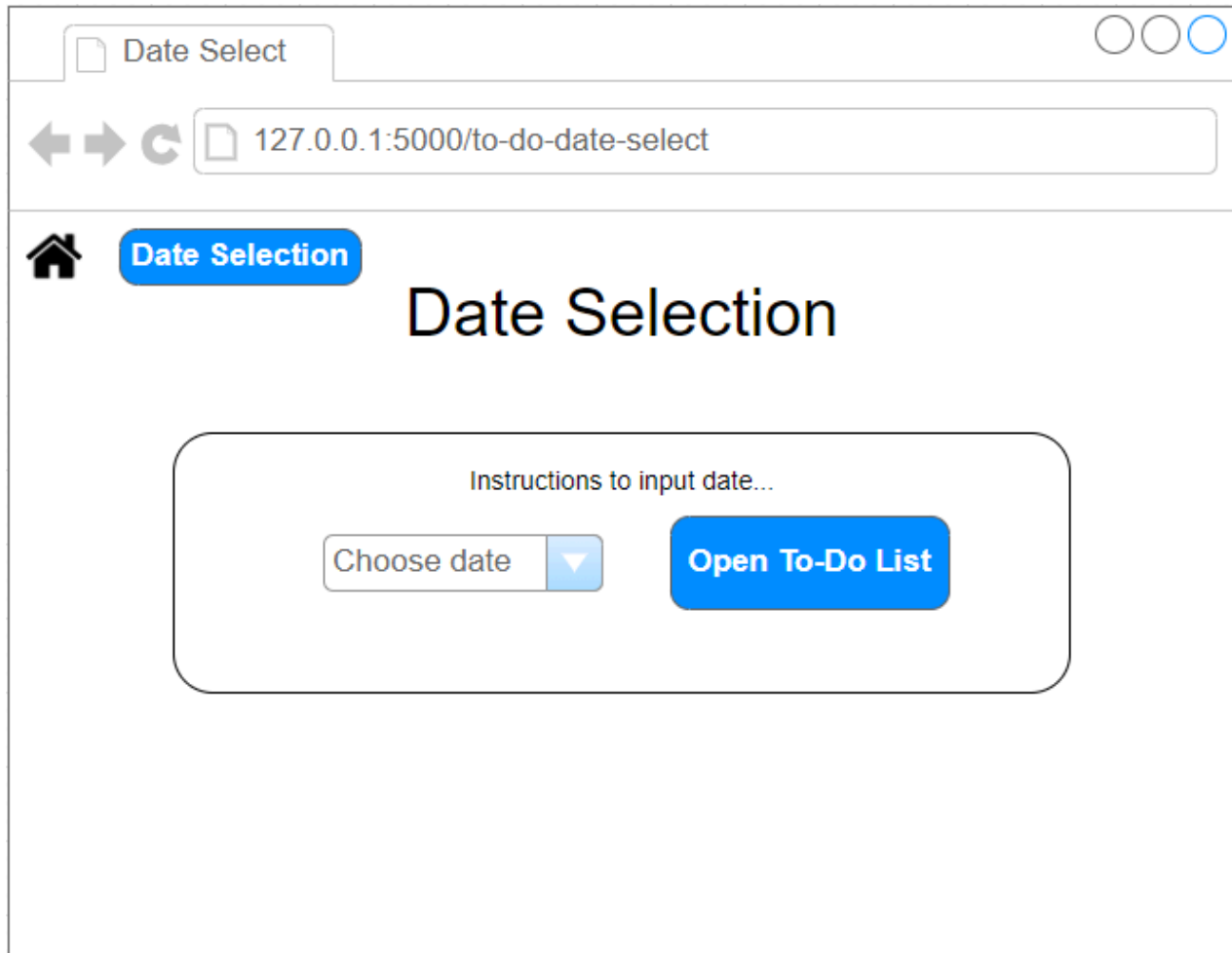


Figure 6.1: A wireframe that shows the date-selection page. It has the same navigation bar, the url says date-selection, and it has a container for the user to select the date.

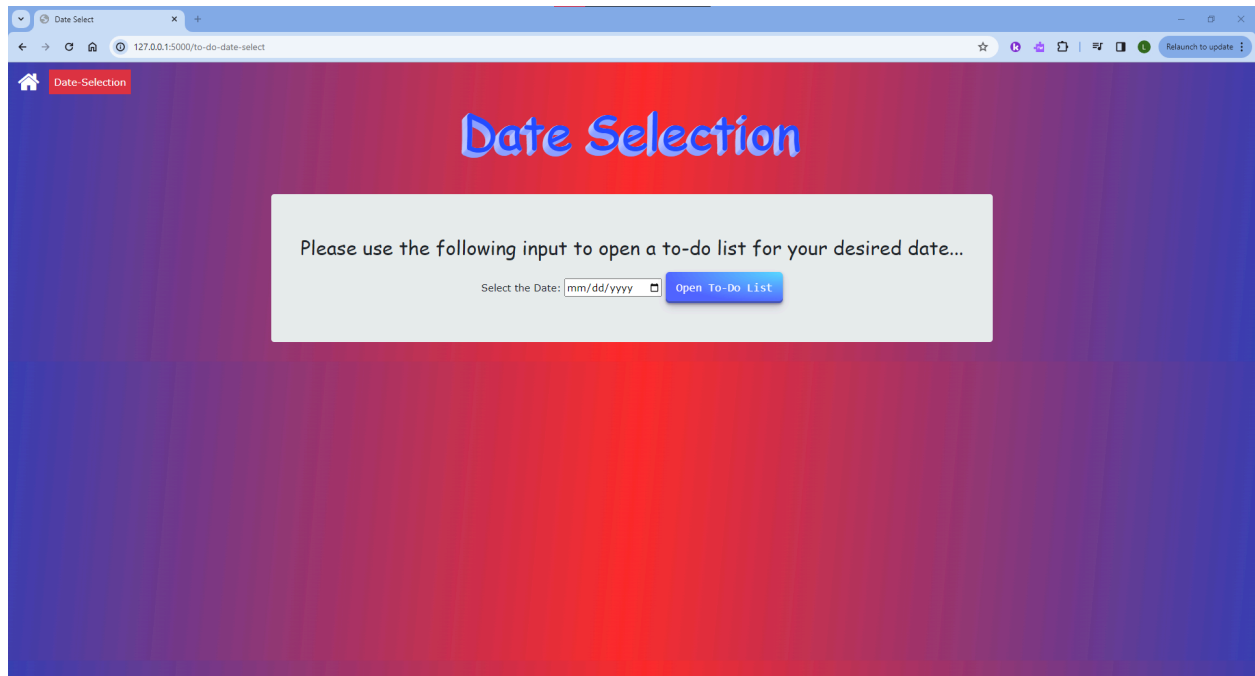


Figure 6.2: A screenshot from the actual program that shows the date-selection page. It has the navbar and the container for the user to select a date.

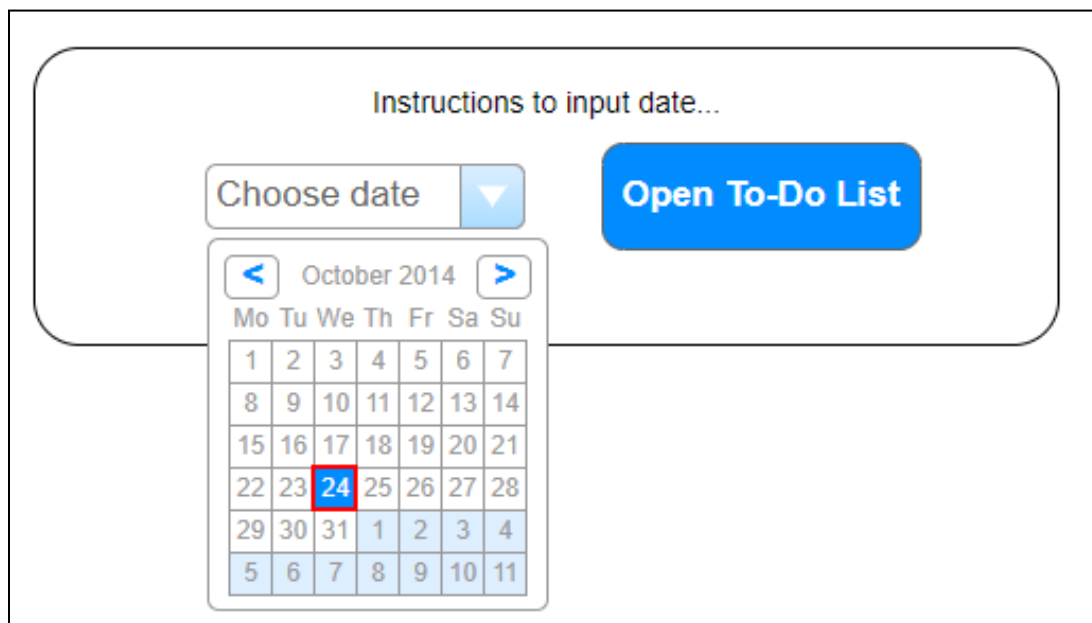


Figure 7.1: A wireframe showing how the date input interface will work. There will be a calendar popup that opens allowing the user to select their date.

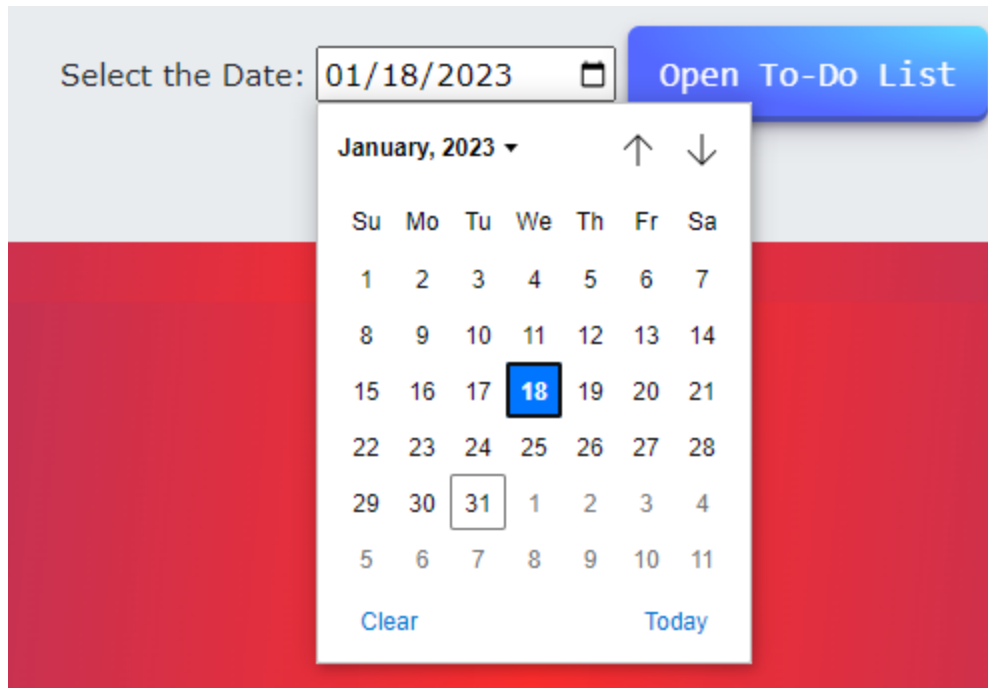


Figure 7.2: A screenshot of the date selection popup from the actual program.

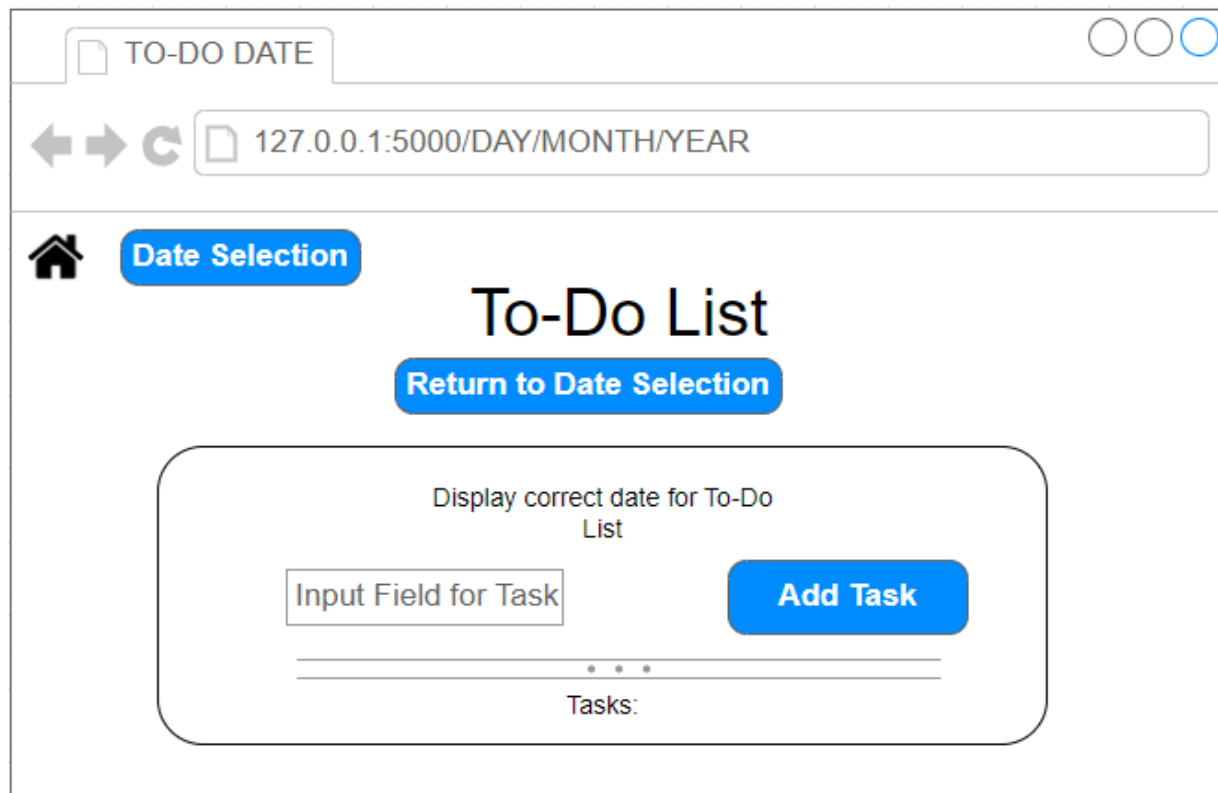


Figure 8.1: A wireframe that shows the to-do list page for a given date. The date will be displayed on the url as well as the tab label. There will be a container to add tasks. It would also display the tasks underneath.

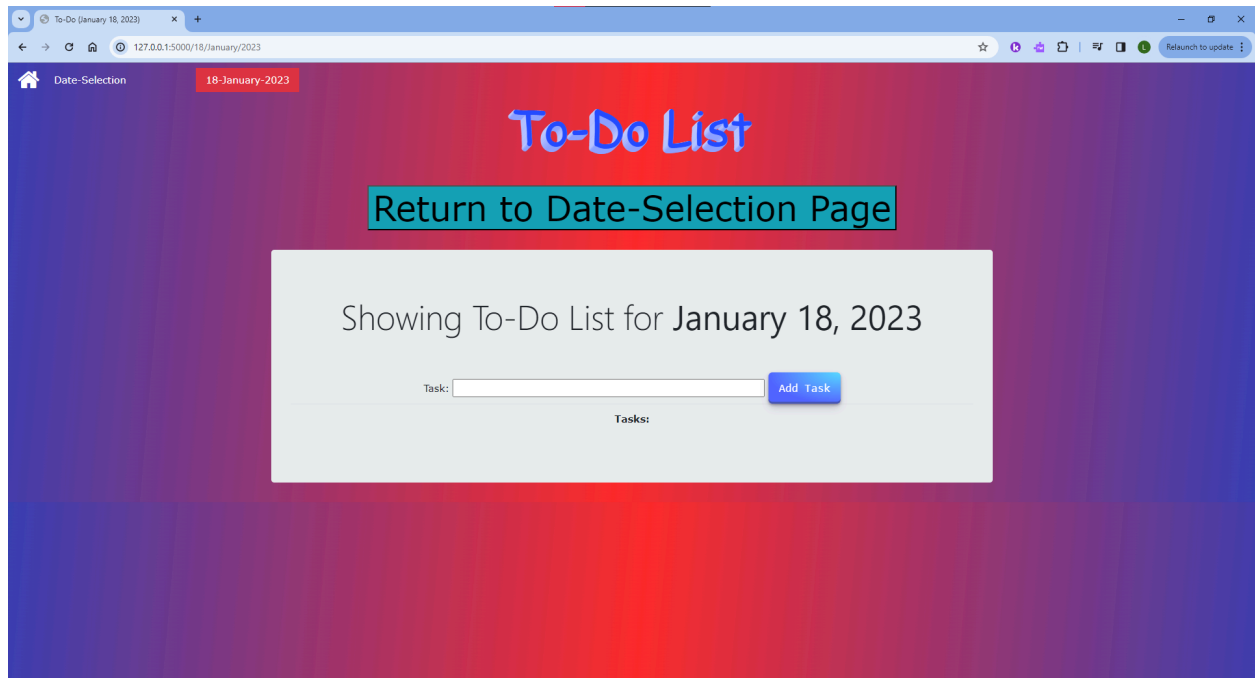


Figure 8.2: A screenshot of the actual program showcasing the to-do list page. The url and the tab label have the correct date and there is an option for the user to add a task.



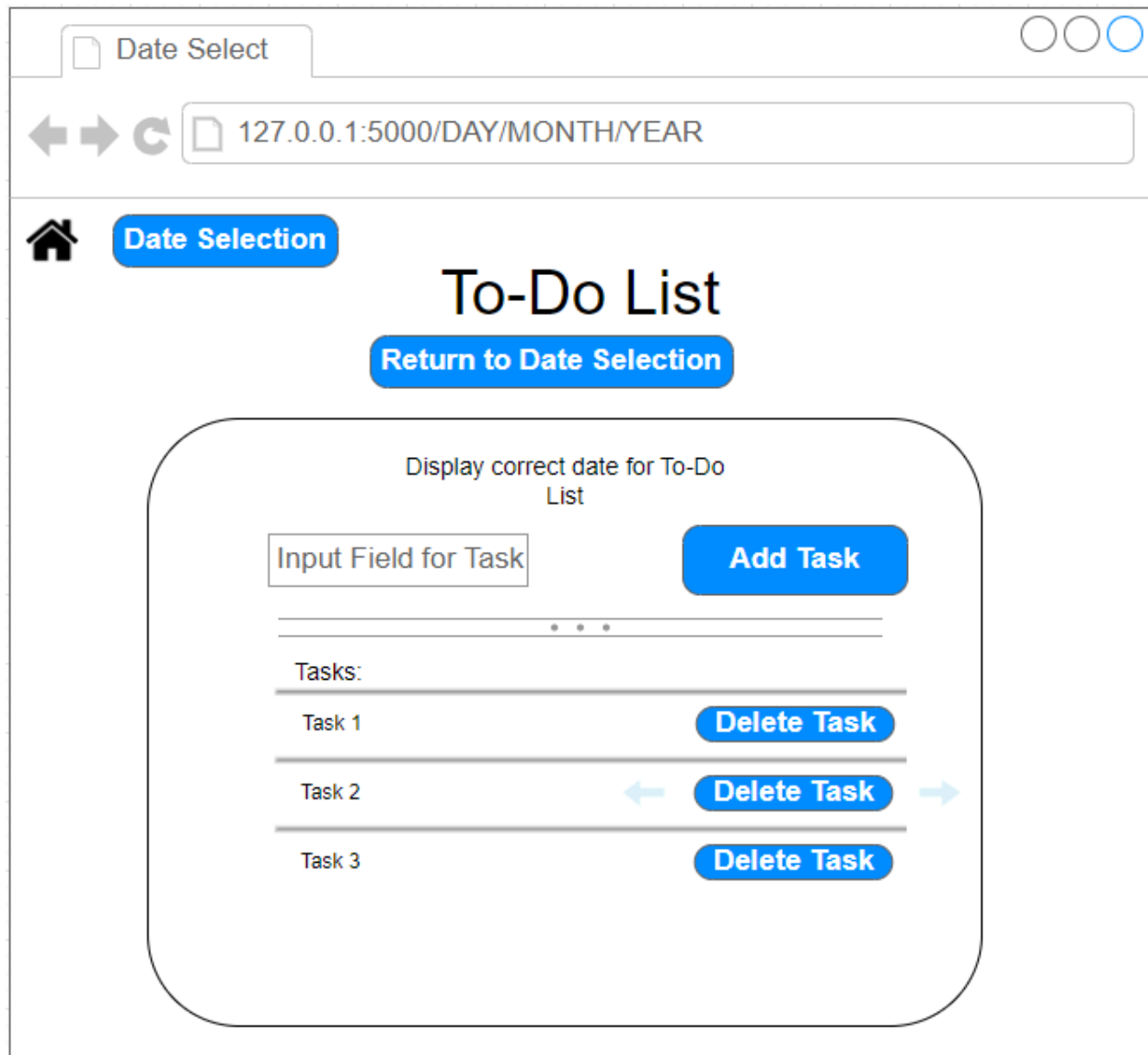


Figure 9.1: A wireframe that shows the to-do list page with a few tasks. Each task displays the task name as well as a button to delete the task.

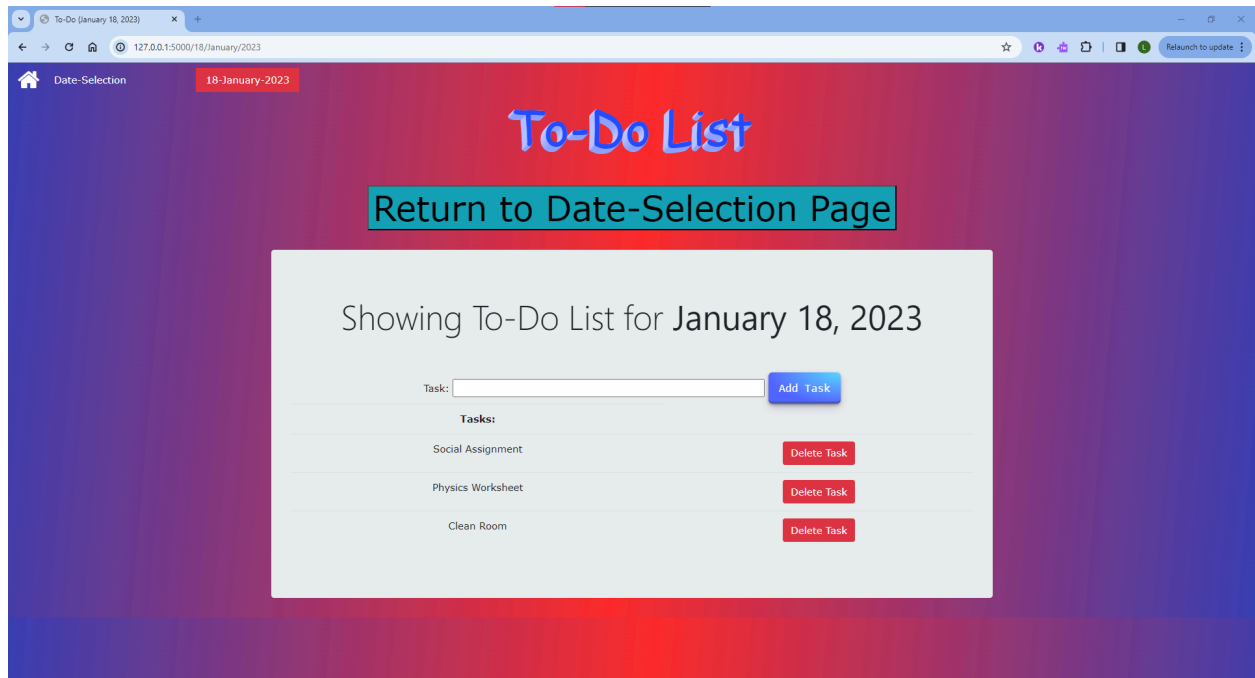


Figure 9.2: A screenshot from the actual program showing a few tasks in the to-do list page. Each task has a button next to it to delete the task.

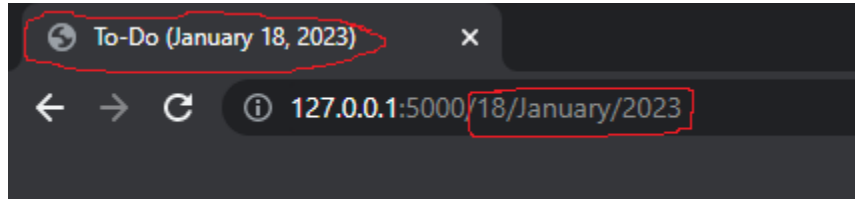


Figure 10: This is a screenshot of the tab which shows that the page url and the page title are of the same date as the to-do list.

## **Test Plan**

Action Number	Action to be Tested	Test Method
1	Program opens and renders the homepage	Run the program and see if the homepage is the first thing that is rendered
2	User can cycle through the homepage and the date-selection page using the navbar	Open the date-selection page using the navbar and return to the home page using the navbar  Open the date-selection page using the button on the home page and return to the homepage using the navbar
3	User can select which date they would like to open a to-do list for	Select a random date as well as clicking the "Today" option on the date selector
4	The user can add a new task for the day	Add 3-4 new distinct tasks
5	The user can delete tasks in a certain day	Delete 1-2 of the tasks
6	The user is able to open another day's to-do list without having to re-run the program	Return back to the date-selection page and open a to-do list for another day
7	The user is able to reopen the to-do list for a certain day and all previous tasks are still present	Return back to the date-selection page and reopen the to-do list for the original day and see if the distinct tasks that were added before are still present.

Word Count: 342