JSL11 CODESPACE THABO LIKOTSI PORTFOLIO PIECE

bysi\0/;top:-4px\0/;left:-6px\0/;rie DEBUGGING AGILE BOARD PROJECT(TO-DO APP)

pointer:display:block:text-de

SU SPE SPE #CCC).gort1 .gom(-moz-EM

or #ecc; display: block; position: absolu



INTRODUCTION

- In this presentation I will detail my solution to debugging and completing the Agile Board Task Management project.
- The goal of the project was to test my debugging skills and to import helper functions, import our initial JSON data, display boards, and to-do tasks, and implement various functionalities such as adding tasks and toggling a sidebar.

LIST OF BUGS ENCOUNTERED

List of Bugs:

In the fetchanddisplaytasks function the one bug I encountered was the fact that the ternary operator was not implemented correctly so I replace; with: and this cause the function to correctly implement the conditional statement.

In the displayBoards function the bug is the fact that the click event listener is not correctly implemented since the addEventlistener DOM method is not used along with a callback function.

Bug's Impact

The Bug halted the function from running properly and as such the tasks did not correctly appear on the UI.

The impact of this bug lies in the fact that we won't be able to filter our to-do tasks since we won't have a 'Launch Career' boards or a 'RoadMap' board and the for loop will ensure that we can switch between the two boards with a button with the boardname as the textcontent of the button.

Changed the assignment — to a comparison ———.

LIST OF BUGS ENCOUNTERED

List of Bugs:

The longest and most complex function filterAndDisplayTasksByBoardName the first bug is the fact that we have to use strict equality operator '===' instead of assignment '='.

The second bug is also related to changing the = to a === when we apply the filter method on our filteredTaskArray obtained from fixing our first bug

Bug's Impact

The impact of the bug is that the callback function won't compare the elements and filter the tasks properly.

The bug will negatively impact how the app separates the to-do items and what board they belong to either Launch Career or Roadmap.If not fixed we won't have our tasks appear on the correct board.

LIST OF BUGS ENCOUNTERED

List of Bugs:

The last bug in the longest and most complex function filterAndDisplayTasksByBoardNam e is the fact that the event listener is not correctly implemented since click() is used instead of 'click' it is incorrect because click is not a method but a type of event listener.

Bug's Impact

The impact of the bug lies in the fact that it will prevent users from being able to edit and update their tasks since the callback function for this eventlistener opens the edit task modal allowing users of the app to make changes.

Changed the assignment — to a comparison ———.

LIST OF BUGS ENCOUNTERED

List of Bugs:

In the styleActiveBoard function the first bug lies in the fact that CamelCase is not used for the forEach method instead they wrote this method as foreach.

The second is DOM-related since we want to add and remove classes the issue here is the fact that the .classList property is not used along with add or remove giving us btn.classList.add('active')

Bug's Impact

The impact of the bug lies in the fact that it will prevent the code from correctly looping through the board button which are either launch career or roadmap and this will also prevent them from being styled properly in CSS.

This bug will impact our styling of the boards launch career or roadmap since these classes are being added so that they can styled in CSS

LIST OF BUGS ENCOUNTERED

List of Bugs:

In the toggleModal function the ternary operator is not correctly implemented since I had to change => to: After this change is made the ternary operator to set the display style either block or none of the modal correctly.

Bug's Impact

The impact of the bug lies in the fact that it prevents the modal from being toggled when the user clicks on a task to edit it, as the logic of the ternary operator is not correctly implemented."

- I had to complete code for five incomplete functions these functions are addTask, toggleSidebar,toggleTheme,openEditT askModal and lastly the saveTaskChanges function.
- 2. For the first function the addTask I had to add the task object. Then I used helper functiond such as toggleModal and createNew Task to successfully add new tasks to UI and hiding the modal when new task is added

To complete this addTask function I ensured that the to-do item is push or added to our JSON initialData and then it is added to the local storage so that when user refreshes the information is still available.

1. The second function is the toggleSidebar function.and I completed the function by using an if statement to toggle the display property of the sidebar either to a flex display or none and this is also done for the showSideBarBtn which toggles the sidebar using a click event listener.

The third function is the toggleTheme function and this function toggles the light or dark them for this app. This is accomplished by using a ternary operator along with the classList.contains() method which returns a Boolean value that we need for the logic of the ternary operator.

- 1. The fourth function is the editTaskModal function and I completed it by updating the inputs of the modal by assigning them to the properties of our task which is a parameter for this function.
- 2. To finish the function off I added two event listeners for the saveTaskchanges and the deleteTaskbutton and I used helper function saveTaskChanges and deleteTask functions and we also use the toggleModal function to toggle the ediTtask modal.

We also use the helper function refreshUI() to refresh our UI after editing the tasks.

- 1. The last function is the saveTaskChanges function and I completed the function by using the DOM to reference the inputs for title and description as well as the status.
- 2. The next step I took was storing these in an object and using the .value property to access the values. Lastly, I used the patchTask function which I exported to update the task on the UI. I closed off by using the toggleModal function to remove the editTask modal and I also refreshed the UI

CONCLUSION

This project has been a valuable learning experience for me, highlighting the importance of teamwork, the skill of debugging using the console, and the use of local storage as a form of database system. Additionally, it has emphasized that software development and coding demand a great deal of patience and a keen attention to detail.



THANK YOU

Thabo Likotsi

CodeSpace Student
likotsithabo065@gmail.com