Interactive Calendar Due Date

Augustus Peine

This interactive calendar is a dynamic web application that allows users to manage their tasks and due dates effectively. Users can add, edit, describe, and delete tasks on any day on the calendar. They can also view upcoming and past tasks in the dashboard. There is a dynamic navigation system for dates, and users can learn more on the about page. This project shows interactive features of JavaScript through event handling, DOM manipulation, data storage with arrays, etc.

Features

Calendar Page: A calendar page with a cell for each day is present at the base page of the website. These cells are all clickable for tasks to be added. Each task is then shown to the user on top of that day. When there are more than two tasks for a day, a +x button is added to showcase the rest of the tasks. This is clickable to see those tasks and make changes. These tasks are also color-coded, with red being past due and green being coming due. When selecting each month and year, the calendar will update instantly, along with when making changes to the tasks within the cells.

Dashboard Page: A dashboard page is available with a click on the navigation bar at the top. All tasks are displayed on this page and are available to be clicked and edited. They are separated by past and coming due dates based on the current day. This is done dynamically and is color-coordinated. This page is all updated dynamically and can be searched through by the name of the task, description, or date. A short message appears about how to add tasks if none have been added yet.

About Page: An about page is accessible with a click on the navigation bar and provides a short description of the website.

Task Modals: Two modals are used to ensure the user must interact with the pop-up box. The first is when adding or editing a task. The user must input a title for the task and an optional description. The other modal appears when users click on the +x tasks button. They can select any tasks from that day, which brings up the previously mentioned modal.

Functionality

Event Listeners: Event listeners are used a multitude of times to trigger updates. These event listeners generally consisted of clicks and the Enter key. This allows for user interaction with the pages.

DOM Manipulation: All of the calendar cells, tasks, and dashboard elements are generated dynamically. Along with this, any changes made to the tasks or calendar months and years are updated instantly. The past and coming color scheme and dashboard organization are also dynamically handled, along with the search feature of the dashboard.

Loops & Logic: Loops are used to fill in the calendar dates and dashboard elements, arrays are used to hold the tasks and days, and conditional statements are used in a wide variety of functions.

Reusable Functions: All functions were implemented to be reusable elsewhere in the program. These include rendering the calendar, dashboard board, and both the modals.

Data Handling: The data is stored in the local user storage to allow for memory. The tasks are stored in an array, which allows for dynamic updates based on the user's choice.

Styling: The layout incorporates a light blue theme along with a color palette of red and green for the past and coming task dates.

JavaScript Enhancements

The user can experience a reactive and live website through the use of JavaScript. In this website, all changes are made instantaneously, and this is reflected when the user interacts with it through the changing of tasks, dates, or pages. JavaScript also allows for dynamic updates of the HTML. The dashboard is completely manipulated in JavaScript when there are no tasks. This is not possible without the use of JavaScript. On top of this, the data is stored in the local browser, which is accessible through the JS. This allows for memory of previously input tasks. The tasks are also searched in real time in the dashboard. All of these things indicated how JavaScript can enhance the user experience and ensure the website is more fluid and enjoyable.

Challenges & Solutions

Throughout the design of this calendar, I faced many challenges. I first thought the design and implementation would be seamless and easy. This was not the case. I have spent many hours trying to debug the site and get it the way I envisioned it in my head. On top of this, the idea and website seem simple, but the underlying code is much more complicated. I have to rely on the help of forums online to learn new things, such as how to get the current date, how to store data in the browser, etc. I had to look up many things to get this project how I wanted it, which took extensive effort and time. On top of this, I have always struggled with CSS, so it was hard for me to get all the elements to look good and have the look I desired. There are many specific challenges and bugs I had to work out, such as not allowing more than two tasks to pop up for each day cell, but I was eventually able to find a solution through the use of the internet and trial and error through all issues.

Future Enhancements

Some ideas I have for future improvements would be adding task categorization. These could maybe be color-coded to allow for even more accurate task tracking. Along with this, it would be helpful to have an option to have recurring tasks, such as cleaning the room every Sunday. I would also like to add drag and drop features to the tasks so that each task does not have to be manually added and deleted to its respective day. Also, using a backend database would be helpful to hold all data so that it can be accessed through multiple devices. All of these ideas would serve as an improvement to this project and make its dynamic capabilities even stronger and more fluid.

Conclusion

This project highlights the use cases and helpfulness of JavaScript in its interaction with a website. It showcases DOM, manipulation, event handling, arrays, logic, functions, loops, etc., to allow for a fluid and seamless experience for the user. I had lots of fun designing it and thoroughly enjoyed the class!