**T.R**

**USKUDAR UNIVERSITY**



**SUMMER PRACTICE 1 PROJECT**

**NAME: MAHMOUD MOHAMED MAHMOUD FOUAD AHMED**

**ID: 210209994**

<https://mahmood7117.github.io/task.managment.app/>

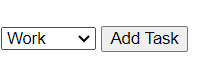
<https://github.com/mahmood7117/task.managment.app>

***TASK MANAGEMENT APP***

* I have made a project for my summer practice1 with DR. Belaynesh Biadgilign Chekol’s permission since I had no chance to find an internship opportunity.
* I chose to build a ***task management app*** since I felt it will be a suitable beginning for me to test my skills on the *html, CSS and Java Script* and learn more and find my weak points and work on it,
* I have attached the final result of my *Task Management app. I’ve started structuring my Task Management App* using *HTML* then I’ve started styling the app using *CSS* and at the end I started adding some functionalities to my app so the user can use all the features I’ve structured and designed before.

**How Task Management APP Works**

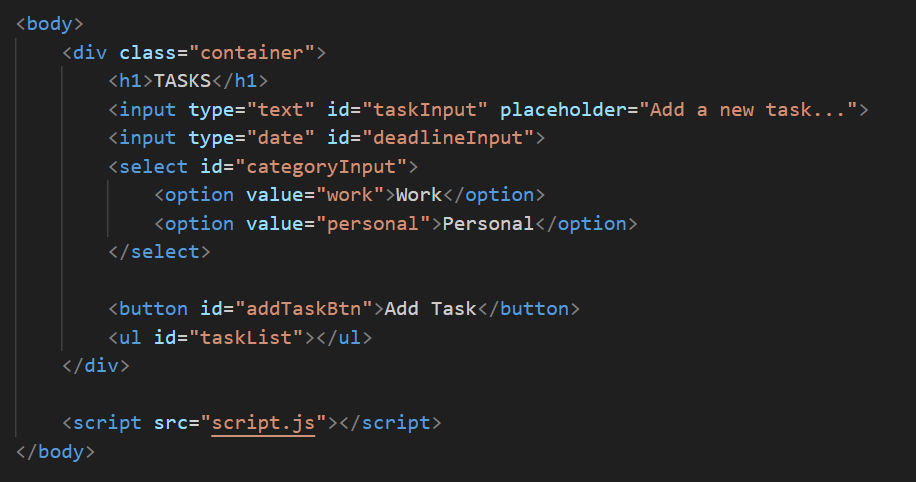
1. User will have to **Add a New Task** through the input field I have created with *HTML.*
2. Then the user can *set a deadline* for his task.
3. Then he can *categorize* his task type whether its *(work or personal)* task.
4. The user can finally add his task by pressing the *submit* button.
5. If a deadline is set, the app will schedule a reminder to alert the user 1 day before the task is due.
6. The user also can remove the task by pressing the *Remove* button I’ve created with *Java script.*

AFTER CSS


WITH CSS

HTML CODE

**1st : Structuring the app *(HTML)*:**

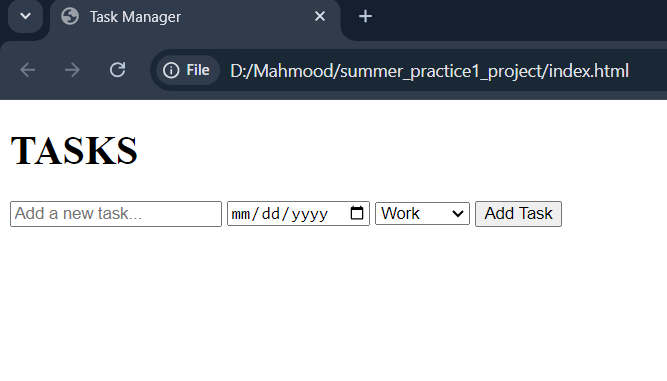
****HTML CODE:

***HTML* Code Requirments and key elements:**

The HTML Code, includes input fields for tasks, deadlines, and categories.

1. The **<input>** tags for entering new tasks and deadlines.
2. A **<select>** dropdown for choosing the task category.
3. A **button** to trigger the "Add Task" functionality.
4. A list **(<ul>)** where the tasks will be displayed.

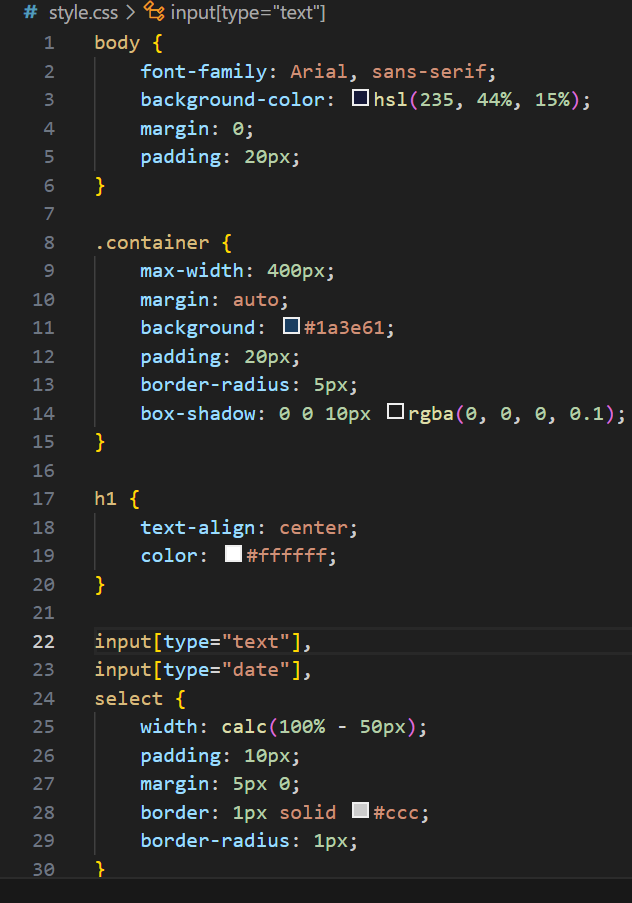
* First Result after running *HTML* code

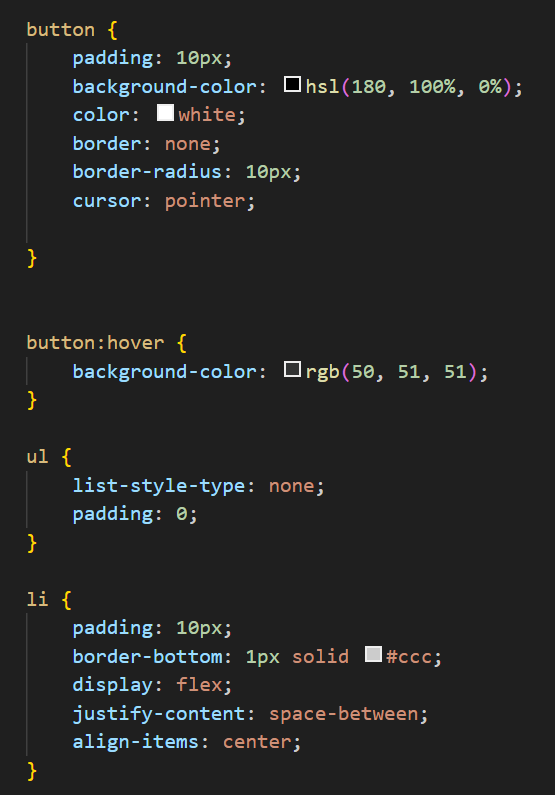


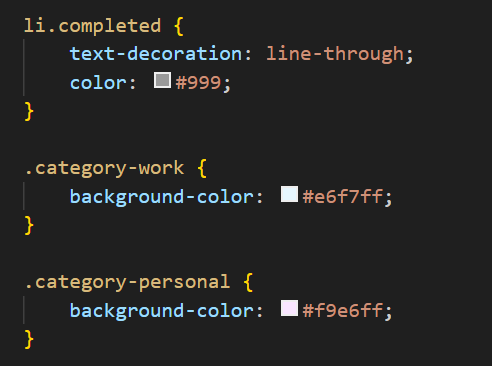
**2ND Started styling the *html* elements with *CSS*:**

*CSS* code:

1 2

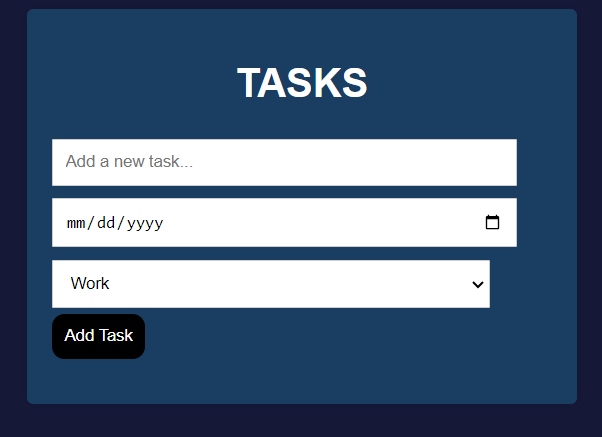






3

After finishing styling the html elements this was the result:

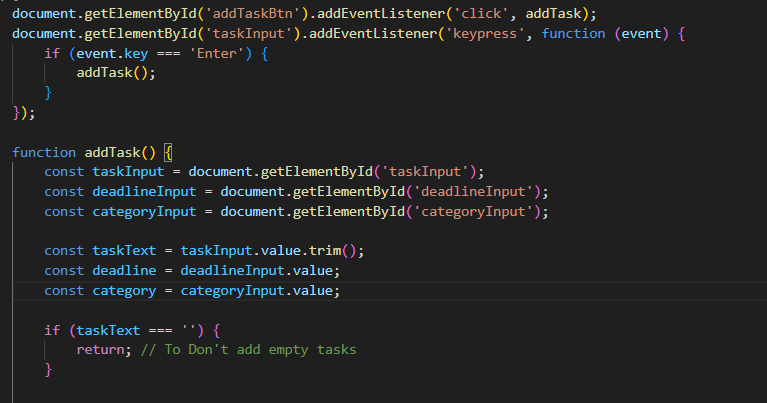


**CSS requirements (summery):**

1. I have centered the container and styled it with a background and added some shadows.
2. Concerning the inputs and buttons they are styled for good spacing using the margin elements and added to the button an interactive hover effect.
3. I have defined the maximum and minimum pixel sizes to ensure that no matter what screen size a user has, the app will be structurally well-designed. This prevents everything from simply shrinking without maintaining the proper size.

**Finally: *JS code & Functionalities:***

1. I have attached a click event listener to the *"Add Task"* button SO, When the user clicks the *"Add Task"* button, the add Task function is triggered, which will handle adding a new task to the list.
2. I have added a keypress event listener to the text input field. ITS functionality that When the user presses the "Enter" key (event. Key === 'Enter'), it calls the add Task function, allowing users to add a task by pressing "*Enter*" instead of clicking the button.
3. **1-** task Input: Gets the value entered in the text input field. **2-** deadline Input: Gets the selected deadline from the date input field. **3-** category Input: Gets the selected category from the dropdown.
4. *The.trim()* method removes any extra spaces from the start and end of the task text.
5. *The if condition* Prevents adding empty tasks. If the user tries to add a task with no text, the function stops (return;).



1. *li.className = category-${category}*: assigns a class name to the <li> element based on the selected category ("category-work" or "category-personal"), which allows the CSS to apply different background colors based on the task category.
2. Created a <*span>* element to display the task details.
3. Combined the task text and deadline into a single string, displaying them like: Task Name (Due: YYYY-MM-DD). By *(taskInfo.textContent)*
4. Added a *click event listener* to the *<li>* (task item).
5. When the task item is clicked, it ***toggles*** the "completed" class. This class is styled in CSS to strike through the task text and change its color, indicating that the task is complete.
6. The 2nd ***if condition*** purpose is if a deadline is set, the app will schedule a reminder to alert the user 1 day before the task is due.



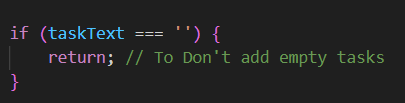
1. A notification will be adjusted 1 day before the deadline using:

notificationTime.setDate(notificationTime.getDate() - 1)

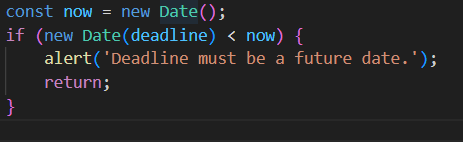
1. ***“setTimeout:”*** If the notification is in the future **“(timeUntilNotification > 0)”**, a reminder is scheduled using “***setTimeout***.” The reminder will alert the user with an alert saying that the task is due tomorrow.
2. **removeBtn = document.createElement('button')**: Creates a new button element labeled "Remove".
3. **removeBtn.addEventListener('click')**: When the remove button is clicked, the task is removed from the list.
4. **taskList.removeChild(li)**: Removes the corresponding <li> element (task) from the task list.
5. After a task is added, the input fields are cleared so the user can easily add a new task without manually clearing the text or date.
6. **taskInput.value = '';:** Clears the task text field.
7. **deadlineInput.value = '';:** Clears the deadline date field.

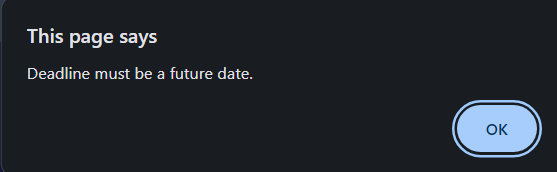


**Testing the app and fixing bugs:**

* When I have finished working on the *java script* functionalities, I went back revising the **HTML** and **CSS** codes to make and I have fixed some small bugs in the **CSS** (margins, paddings, btn hover and etc..) and the maximum and minimum pixel sizes to ensure that no matter what screen size a user has.
* ****Concerning ***JS*** I figured out while testing that an empty task can be added if the user didn’t write any thing in the input task it will be added, I fixed this bug by adding

* Also the app allows adding tasks with past deadlines without any warning or error, Cause: There's no check in place to verify whether the selected deadline is a future date , I fixed this bug by adding a validation to check if the selected deadline is in the past.



If the user added a task with past deadline a warning message will displayed.

As shown in the following images this is how the app looks like on different screen size.

