**To-do App  
  
Whats the scope of this project?**

The main scope of this project is making a To-do app

Main Functions:

1. Add a task
2. Remove the task
3. Edit item
4. Mark Complete
5. Sorting
6. Filtering
7. Search Functionality: Implement a search feature that allows users to quickly find specific tasks by entering keywords or phrases.

If this phase is successfully completed, I will move on to the next phase of this project where we will have the option to create other lists also

1. Groceries
2. Travelling
3. Eating
4. Shopping
5. Reading

I believe the first phase may be a little difficult but after the first things will get much easier. I will have to just clone the items and rename the list.

Once the App is created, I will add new features like dates and make it link to a word file so that once I add or delete the task it automatically updates the file too. It will be much easier to add and remove the task. I will also learn too much in this case.

**Rough Work/ Question/ planning**

**Font Awesome**

**I have connected a font awesome link successfully, whats the next step?**

1. My first task is to center and make the to-do list heading at the center and at the top
2. add icon along with heading to make it aesthetic.
3. By using the flex part, I have successfully done my heading part/ Font size/ position/ color/ img. All are just perfect.
4. The next step is to search for something from internet to create the input and button to add task in the list. (Research on UI/ UX).
5. The design that I have found so far is that I will add a Plus sign inside a circle that will (+) that when pressed will pull up a box that ask for the task to be added in side the list. When added those items successful will add them in my list.
6. Can I give the class and Id to the same button?

I have learned something new about how to hide the items in html using the style tag and also how to use getElementbyId () to apply JavaScript those Elements and make make the reappear. This is called an event listener that performs a certain action when an event occurs. For example, I click a button/ Pressed Enter key and certain action is performed, this is called Event Listener.

**26/3/2024**

I have understood the functionality to make it appear. Now I have a question about what if I want it appear on the to do app and not where it was already. Like everything goes in the background and input box and add task button appear. Making everything blur in the background.

It says that I have to bring some changes to my CSS only. Just add some properties that will change the input box and it will appear once th

I have come so far in my to do list that I have designed the header very properly including a logo, heading and the button. The button also plays the role of showing the button. So far, I completely understand the code. How it is working and whats the logic behind it. When I press the button the input box appear that takes the input and add it to the list.

**Next target:**

1. **My First target is to understand how the overlay work so that input will appear on the top of the existing content. (Done)**

I learned something very easy about how to create the input form and it’s very easy.

There is will be two divs. The first one will cover the entire page and relative to the parent we will position the child div and relative to the child div there will be there will be input form and the button add task.

I learned many things while creating the overlay so everything gets blurred once we create a input form. Learned about viewport also

Why we used two div🡪 one was covering the entire viewport and the second was the container for input form and button.

position: fixed; This property positions the overlay fixed relative to the viewport, meaning it stays in the same position even if the user scrolls the page. These

top: 0; left: 0; These properties set the distance from the top and left edges of the viewport to position the overlay at the top-left corner of the screen.

width: 100%; height: 100%; These properties set the width and height of the overlay to cover the entire viewport, ensuring it spans the full width and height of the screen.

background-color: rgba(0, 0, 0, 0.5);: This property sets the background color of the overlay to semi-transparent black (rgba(0, 0, 0, 0.5)). The rgba notation allows you to specify a color with red, green, and blue components along with an alpha (opacity) value. Here, 0.5 represents 50% opacity, making the overlay partially transparent.

backdrop-filter: blur(5px);: This property applies a blur effect to the background behind the overlay. In this case, a blur radius of 5px is applied, creating a blurred effect. This property is commonly used for creating frosted glass or blurred background effects.

display: none;: This property hides the overlay by default (none), ensuring it's not visible until triggered to appear.

1. **My next Target is changing the style of input form. (Done)**

Do some searching of the input tag and bring changes accordingly. Don’t settle for something similar but work on achieving the exact search box. My understanding says that it wont take too long, just need to change some CSS properties and everything will be good. I have create a visually attractive input form with a button.

1. **Next task is to clean the code once for understanding (HTML, CSS and Javascript)(Done)**

I will remove all those things from code. Previous html has been removed so the style should also be removed. The html part is already commented out. The point of this exercise is to understand the code and how to write a clear code and learn to keep things organize with commenting and reducing the code using efficient method.

1. **The next part is creating a list**

**Main functions:**

* Same font color (White background and black content)
* Oval Shape
* I will add a bullet so people can select the list item and take appropriate action
* When we press the button, the task gets added in the list

I used JavaScript to add functionality to code. I have successfully made a function that adds the data from the input bar to list. Code is working absolutely fine.

**Thinking Process:**

I think it has something to do with(“Add-task) button so when we create an event that onclick this button adds the content in the button in the form of list.

Another task is to add

**Step**

1. Create a list in HTML
2. Write CSS to edit to design the list
3. Write JavaScript code that will add the content to the list dynamically.
4. Also add a selector so we can select the item.

**Understanding the structure**

1. There is a body that covers the view port
2. Inside the body there is another container

* Logo
* Heading
* Button
* Overlay input

1. After the container there is the list item that we will create to store the list items

document.getElementByID(“addItemButton”).addEventListener(“click”, function() {

const inputField = document.getElementById('taskInput');

const taskText = inputField.value;

// Create a new list item element

const listItem = document.createElement('li');

// Set the text content of the list item to the value entered by the user

listItem.textContent = taskText;

**Today my main goal is to design the task list on my own just according to the Microsoft style**

1. Background color:White
2. Font-color: Black
3. Add circular Selector to
4. Create a separate rectangular box in the list item

**4/3/2024**

Today my task is to edit the list. First, I will have to create the spacing to distinguish the elements inside list. The current problem is that I am not able to do so. They all are inside the same box and I need to change that.

**I am successful in doing it.**

Another thing that I want to add in this project is to make a standard size of list items. It should be a long bar even if the task added is small in length.

**I am successful in this mission also. Just had to add the width and all the task items will be under the same size of bar.**

Next, I am trying to add a selector to select the item to perform the necessary action. I can use html and CSS but desired functionality won’t be achieved by this only. I need to understand and modify the JavaScript code to reach there.

**I am successful in adding the selector also. Its working absolutely fine. I am happy with my progress so far**

**Adding a Remove task (x) button: (Success)**

My next task is to create a button that will remove the task from the list. I will add this feature to bring changes to my list. My general understating is that I will have to create a button that will appear when I will select the item using the circular selector. It will automatically pop-up the cross button which will remove the data from the list. The button will appear once the item is added to the list.

I have successfully added the button that appears only when I select the selector. I set the default display property to “none’. It keeps the button hidden. The button appears when I select the selector and the button appears.

**Problem:**

* How and where to add a remove button?
* The problem I faced was the button appeared right next to the task in the list which was not looking.
* The selector doesn’t show any sign (mark/ color change) when we select the selector.

**Learning:**

I learned about the float (right, left) property and it keeps the item moving to either side and use the maximum available space.

The remove () function was not a very difficult function to understand. We created a button dynamically in JavaScript and kept the text content “X” symbolizing the delete property

I am trying to understand the

In my opinion it should be inside the list items already  
we will keep it hidden and once the selector is pressed, we will make it reappear.  
The appearance of the button depends on the query selector. When user will click on It. It will appear Once it’s appeared the user can press it to remove the task.

Where we add list item in the JavaScript code is where we will add list items.

**Q: Whats append child in the code?**

A: Append child is a method in Javascript that adds the new task item in the list. In this case we can use

Adding these features to make my list grow!

1. **Task Prioritization**: Assign tasks with high, medium, or low priority levels.
2. **Responsive design**
3. **Due Dates and Reminders**: Set due dates and receive reminders and notifications.
4. **Task Labels and Tags**: Organize tasks using labels, tags, or categories.
5. **Subtasks and Checklists**: Create subtasks and checklists for larger tasks.
6. **Task Dependencies**: Set dependencies between tasks, so one task can't be started until another is completed.
7. **Recurring Tasks**: Schedule tasks to repeat daily, weekly, monthly, or yearly.
8. **Collaboration**: Share tasks and projects with others, with real-time updates.
9. **Task Comments and @mentions**: Leave comments on tasks and mention team members.
10. **File Attachments**: Attach files, images, or documents to tasks.
11. **Customizable Dashboard**: Personalize the dashboard with widgets, charts, and graphs.
12. **Integrations**: Integrate with calendar, email, or other productivity apps.
13. **Task History and Archive**: Keep a record of completed tasks and archive them.
14. **Customizable Notifications**: Control what notifications you receive and how often.
15. **Task Delegation**: Assign tasks to others, with optional due dates and reminders.
16. **Progress Tracking**: Visualize progress using charts, graphs, or Kanban boards.
17. **Custom Fields and Forms**: Add custom fields and forms to tasks, such as text, date, or dropdowns.
18. **Automation and Workflows**: Set up automated workflows and task assignments based on conditions.
19. **Time Tracking**: Track time spent on tasks and projects.
20. **Focus Mode**: Temporarily hide non-essential tasks and distractions.
21. **Customizable Themes and Layouts**: Personalize the app's appearance and layout.

**Learning:**

1. How to connect font-awesome.
2. Learned about connecting a Script file too.
3. Applied JavaScript Event Listener Properties to add a task using click and also keyboard.
4. Learned about the flex property of
5. Display property: none. // using JavaScript to make the hidden part visible.
6. I have just learned about appendchild() that how this method can be used to add the data in the list. I also learned about .value= “ ”; to disappear the value from the code.