

Date: 04/09/2025

Project/Module: Midterm I

Goals for the Documented Session

- Learn specific details of CSS such as alignment, boxes, etc.
- Style the webpage using different CSS classes.
- Build a well-structured HTML document.

Task Completed

I worked on my Midterm I, which was to create a webpage based on a given image. The main objective was to strengthen my HTML and CSS skills and implement them correctly. In the end, I achieved a good result.

Use of Gen AI

I used GitHub Copilot for small details that I found difficult to implement, especially in CSS.

The first prompt I tried was asking if my already-written HTML code was well structured, making sure I didn't miss closing a `<div>` or something similar. The output confirmed that it was technically structured correctly, but not in the best way. Initially, I used only tables and `<div>` tags to organize everything. The AI suggested using semantic tags such as `<header>`, `<section>`, and `<footer>` for better HTML structure. I didn't just rely on the AI—I also checked the class presentations, which confirmed that tables are no longer efficient for structuring HTML. Following that advice, I reorganized my HTML using semantic tags.

```

<div>
  <table>
    <tr>
      <td>
        <h2>All your files in one secure location, accessible anywhere.</h2> <br>
        <p>
          File-it stores your most important files in one secure location.
          Access them wherever you need, share and collaborate with friends,
          family, and co-workers.
        </p> <br>
        <form>
          <input type="text" id="email" name="email" placeholder="Enter your email...">
          <button type="Get Started">Get Started</button>
        </form>
      </td>
      <td>
        
      </td>
    </tr>
  </table>
</div>

```

```

<div>
  <table>
    <tr>
      <td>
        <h2>Stay productive, wherever you are</h2><br>
        <p>
          Never let location be an issue when accessing your files. File-it has you
          covered for all of your file storage needs.
        </p> <br>
        <p>
          Securely share files and folders with friends, family and colleagues for
          live collaboration. No email attachments required!
        </p> <br>
        <p>
          <a href="https://www.anahuacalli.com.mx/" target="_blank">See how File-it works</a>
        </p> <br>
        <p>
          File-it has improved our team productivity by an order of magnitude. Since
          making the switch our team has become a well-oiled collaboration machine.
        </p> <br>
        <p>
          <table>
            <tr>
              <td></td>
              <td>
                <b>Kylo Ren</b> <br>
                <p>Founder & CEO, First Order</p>
              </td>
            </tr>
          </table>
        </p>
      </td>
      <td>
        
      </td>
    </tr>
  </table>
</div>

```

This is how I implemented with pure tables (This is wrong, now I learn the correct way)

Once the HTML structure was correct, I moved on to CSS. I started with what I already knew and used the class presentations as a guide. However, I struggled a lot with alignment issues, such as why an image was appearing below the text instead of beside it. For these details, I again used Copilot.

Example Prompt:

“Help to align this element to the left side of the page just beneath ‘see how File-it works.’”

Output:

- Remove the duplicate margin-top property.
- Ensure the parent container (likely .productive) uses flex layout.
- Set .ceo to align-self: flex-start; so it aligns to the top/left within the flex container.

Edits: I corrected the class, experimented with width and alignment through trial and error, and added classes to better visualize the sections.

Reflection: These small adjustments in alignment and width require trial and error. The key is to truly understand the concepts so you know what you are changing.

Another Prompt:

“Please help me explain why the ‘Enter your email’ box is not aligned with the ‘All your files in one secure location’ text.”

Output:

- Make sure both elements are in the same column (.location-text).
- Remove justify-content: center; from .location form so it aligns to the left.
- Optionally, set align-items: flex-start; for vertical alignment.

Edits: I added more pixels to the top margin, changed center to left, and adjusted the width values through trial and error until I liked the result.

Reflection: Small pixel changes can shift the entire layout. It's important to carefully check alignment so that elements are exactly where you want them.

What I Learned

I learned many things, but the most important was patience. Achieving a good presentation on a webpage takes time and requires making adjustments little by little. With practice, it becomes easier to estimate values, but at the beginning, it's challenging.

I also learned that using tables is not the best way to build the HTML base. It is better to use `<header>`, `<section>`, and `<footer>`, and within those sections, use `<div>` elements that are easier to style with CSS.

Challenges

The first challenge, as I mentioned, was changing my HTML structure. I deleted the tables and implemented semantic tags. It was not too hard, but it was confusing at the start.

Another challenge was adding Google Fonts to my code. I asked the AI for help, but the code it gave me looked strange and gave errors. Instead, I researched on my own and found a very helpful YouTube tutorial that solved the problem.

The biggest challenge, however, was aligning and arranging everything as I wanted, adjusting alignment, font size, colors, etc. I relied heavily on Stack Overflow, where I found explanations about alignment properties, CSS class attributes, and how to write them correctly. The explanations were very detailed, so I could understand exactly what each property did.

Resources used

- https://www.youtube.com/watch?v=zPj_bXPnPFc
- https://www.w3schools.com/html/html_css.asp
- <https://stackoverflow.com/questions/22400584/align-images-left-right-using-css>

(Note: I used more Stack Overflow and other website links, but I forgot to copy them before closing the tabs. The ones listed above are some of the key resources.)

Next Steps

- Organize my code sections better to avoid disorder in my CSS.
- Focus on writing cleaner and more reusable CSS classes instead of repeating code.
- Improve the use of semantic HTML to make the code more professional and accessible.
- Learn how to better debug CSS issues without relying too much on trial and error.

Personal Reflection

At the beginning, I felt very stressed because things weren't working as I wanted, and I lost a lot of time trying to figure out why. Honestly, I don't like relying too much on AI because it can sometimes confuse me even more. However, since I was wasting time, I decided to use it. Normally, I use ChatGPT, but my dad recommended Copilot since it works directly within your code. Using it helped me progress faster, but whenever I didn't understand something, I paused and researched online to fully grasp what the function was doing. That way, I not only fixed the issue but also gained knowledge to use in future projects.