Lappeenrannan teknillinen yliopisto

Industrial Engineering and Management

Software Development Skills Front-End, Online course

Anna Tiilikainen 001665431

LEARNING DIARY, FRONT-END

Date : 1.4.2025

Activity : Getting familiar with the course material and git

Learning outcome: I became familiar with the course content and how to complete it.

Date : 2.4.2025

Activity : Setting up git environment and choosing a code editor. Uploading images, preparing folders and VS Code environment.

Learning outcome: Learning how the vs code works

Date : 3.4.2025

Activity : Base HTML and Navbar HTML.

Learning outcome: Doing some basic HTML.

Date : 4.4.2025

Activity: Styling buttons with CSS and using utility classes. Creating testimonial section.  
Learning outcome: Learned how to build and style buttons with CSS. Got familiar with CSS Grid layout and how to align elements properly.

Date : 7.4.2025

Activity: Creating and styling a pricing card section.  
Learning outcome: Learned how important it is to write clean and correct HTML.

Date : 8.4.2025

Activity: Building a FAQ section and adding interactivity with JavaScript. Creating a footer section and styling an input form.  
Learning outcome: Learned how JavaScript can be used to show/hide content.

**Date : 9.4.2025  
Activity:** Implementing a responsive mobile menu with JavaScript and CSS media queries. Fixed CSS errors.  
**Learning outcome**: Understood the role of JavaScript in adding interactivity.

Date: 10.4.2025  
Activity: Customizing the footer and adding a personal logo. Adjusting layout using CSS.  
Learning outcome: Learned how image size and spacing affect layout. Practiced customizing layout for better balance and alignment.

Date: 16.4.2025  
Activity: Uploaded OBS studio. Made a video of my project and uploaded it to YouTube.  
Learning outcome: Learned how to use OBS studio.

**LEARNING DIARY**

1.4.2025

I familiarized myself with the course platform and the tasks required to complete the course. Since the topic area was quite unfamiliar to me, I carefully went through the course materials and installed Git on my computer. I watched the introduction video to understand the main principles and came to the conclusion that the course seems manageable with the help of the clear video instructions.

2.4.2025

I have chosen VS Code as my code editor for this course. Since it was also unfamiliar to me, I watched some YouTube videos to learn the basic use of the editor. I watched the first 10 minutes of the instruction video and downloaded the zip folder with images from GitHub as instructed. Then, I imported the images into VS Code and created the necessary folders for the project.

3.4.2025

I did the first part of the instruction video and based on it I I built a navigation bar for my page. I implemented the main menu selectors (Home, About us, Blog, Log In) according to the instructions, but I thought I might change them as the project progresses to better suit my own idea.

I chose a font for the page through the Google Fonts service. This was completely new to me, since I had never even heard of Google Fonts before. I learned that it is a free online service that allows you to easily choose and add different fonts to your website. After this step, I now understand better how the layout of a page is built and how different parts, such as fonts, icons and navigation elements, are linked together through HTML and CSS.

4.4.2025

I made the video section “Buttons & utility classes”. I made my own buttons for the page with CSS and learned how to use utility classes. I made different styles of buttons according to the instructions in the video, such as a basic button and buttons in different colors. I tried to make the buttons a slightly different color than in the instruction video, but the other colors didn’t really show up the same way as the blue used in the video, so I ended up making it the same.

I added the images to the page and arranged them the same way as in the video. For some reason I couldn’t get a dark background color like in the video, but it didn’t really matter because I thought a light background was better.

In the testimonials phase, I added customer feedback sections to the page. I used the CSS Grid feature to place the feedback cards next to each other in a row of three columns. I learned that the grid-template-columns structure does just that, and gap defines the space between the cards.

I also encountered a small error at this point. The feedback texts went over the images, and the layout didn’t look as it should. I quickly realized that the reason was that some of the feedbacks weren’t inside the same grid – they were placed outside the card element, so from the browser’s perspective the structure didn’t work as it should.

I also noticed that the .hero section wasn’t closed at all, because it was missing a </section> at the end. I only noticed this when I looked closely at the example and compared it to mine. At this point, I learned how important it is to keep the HTML structure clean and correctly closed, and how small mistakes, like a missing closing tag or a misplaced element, can affect the appearance of the entire page. I also learned more about using CSS grid effectively.

7.4.2025

I created a pricing card in the HTML section and styled it using CSS. At some point, I realized I had accidentally left out the second card. I also noticed that the text inside my card wasn’t showing up. After checking the code, I found a small typo in the HTML structure: the <ul> tag wasn’t properly closed, and some <li> elements were in the wrong place. After fixing this, the text appeared as expected. Once again, I learned how even a small mistake can have a big impact on how the page functions.

8.4.2025

At this stage, I added a FAQ section to my webpage. The structure was built with HTML and styled with CSS, and I used JavaScript to add interactivity—clicking a question reveals the answer while closing the others.

The HTML part was quite straightforward since the structure was logical: each question and answer were grouped into its own “faq-group,” with the question title and icon (+ or −) on top and the answer hidden underneath. In CSS, I learned how to hide the answers by default and show them when the correct class was added.

JavaScript was clearly the most challenging part so far. The instruction video moved fast, and I had to pause it several times to understand what each line of code did. Still, I managed to get the functionality work and learned some basic JavaScript commands.

I also built the footer section of the page using HTML and CSS, and divided it into four columns with the help of a grid layout. I created a newsletter subscription email field and styled it to look cleaner. I learned how form fields can be styled and why it’s important to define the input type correctly. It improves both usability and how the page functions.

9.4.2025

I added a mobile menu to my page that opens and closes when the hamburger icon is clicked. I built the structure in the HTML and CSS files, but the functionality was implemented with JavaScript.

I used the addEventListener command, which tells the browser to "listen" for a click on the hamburger button. When clicked, JavaScript tells the menu to either show or hide. I learned that while the structure and styling of a page can be done with HTML and CSS, adding interactivity requires JavaScript.

At one point, I realized that my hamburger menu wasn’t working as expected. I had been following the tutorial without checking my own progress, and noticed that the regular menu was showing twice. I found out that I was missing the .navbar .hamburger button section in the CSS @media query, which was included in the tutorial video. After adding that part, the mobile menu started working correctly.

At this stage, I also compared my CSS to the example version. I noticed that I had repeated some rules unnecessarily and that one selector had an error. I fixed these issues. To make the project more personal, I also customized the colors, fonts, and text content.

10.4.2025

I wanted to add something unique to my site, so I created my own Growth logo using Canva and added it to the footer section. However, I noticed a problem: the logo looked too big and took up space at the top, which pushed the other footer headings ("Company", "Resources" and "Contact") lower and made the structure look unbalanced.

I solved the issue by creating a custom class .footer-logo for the logo and setting its size to 100px through trial and error until it looked right. I also edited the image slightly to make it more suitable.

Additionally, I reduced the padding and margin values of the .footer .card element because it was taking up too much vertical space. After adjusting those values, the other footer content no longer shifted downward.

As a result, the footer now looks cleaner, the elements are aligned, and the overall layout matches the reference design more closely. I learned how small changes, like resizing an image and adjusting alignment, can significantly improve the visual balance of a webpage.