Lappeenrannan teknillinen yliopisto

School of Business and Management

Software Development Skills

Timo Hyvönen, 0600254

LEARNING DIARY, “CT70A9110- Front-End”- MODULE

**LEARNING DIARY**

I read through the instructions for the course and briefly checked out the content of the course. I started by watching the first video "Introduction to workflow and sass". I've used basic html and CSS before but Sass is completely new for me. It will be interesting to see how much code is required to create the CV-website in the tutorial video. I did not know that Sass has the feature like "mixin" which looked as a really useful feature. I already had the Visual Studio Code, Node.JS and Git installed on my computer. I created the first project folder and learned how to open the visual studio code to the current directory in the command prompt. I added the "Prettier", "Live Server" and "Bracket Pair Colorizer"-addons to my Visual Studio Code as suggested in the tutorial video. I learned how to create html tags quickly with visual studio code and created the project folder and files shown on the video. The live server-addon is really useful and using sass instead of CSS does not seem that difficult at this point. Seems to be like writing in Typescript which get compiled into JavaScript. I didn't use the git command line commands, instead I used the Git Gui-program to initialize a new repository, stage & commit and pushed the new repository to my github.

**Homepage and Core Sass/CSS [Part 2]:**

I learned a lot of shortcuts for coding in visual studio code. So instead of my old way of fighting against the random shortcuts, I used my time to see how to use them properly. Learning these really helped me to speed up creating the html-file in the tutorial video. There was a lot of new stuff about SCSS which included: Nesting in SCSS, rem units, vh, vw, mixin, transition, overlay, z-index and a lot more. I was able to follow along the tutorial video and copy the example project code. I missed few tags in the html part which resulted in some of the sass not working, but I paused the video few times and fixed the problems which I encountered. Commenting out parts of the SCSS really helped me to pinpoint the problem.

**Rotating Menu Button [Part 3]:**

I was surprised by the amount of control that SCSS can have on the DOM-elements. I would not have thought to use SCSS on animating simple elements to create powerful menu-animation. Even the 0.5s transitions between things add up to the feel of the html page. I learned how to add and remove class tags using JavaScript and how to use the SCSS transform to rotate and translate html-elements.

**Menu Overlay & Responsiveness [Part 4]:**

Previously I have done the responsiveness for websites using bootstrap. I have struggled with bootstrap changing things version to version and never actually reading the documentation deeply enough. It was nice to see that you can do the mobile optimization without any libraries, especially if you have a small and simple website. I do not fully understand in which order SCSS applies the different styles especially if you have multiple styles for the same elements.

**Page with CSS Grid [Part 5]:**

I learned the basics on how to use the grid system. Usually I have used the div-tags and bootstrap to create these kinds of grids, but I'll be using the css grids from now on. So much easier and clearer.

**Work and Contact Pages [Part 6]:**

Not that much new stuff in this part. Most of the things were a repetition of things learned before. I followed along the coding and made my final commits for the example project.

**Website Deployment [Part 7]:**

Publishing websites was already familiar for me. A new thing from the video was the use of gh-pages to publish websites to github-pages. It is a shame that the video does not go into detail on how it works.

**Final project:**

Designing the UI for a web page has never been my strong suit. I decided to ask a friend for a UI-design which I could transform from a picture to a HTML + CSS page. She is doing a UX design course on Adobe XD-program and she had a suitable web page design for me. She gave me the general layout picture of the web page and the source pictures used in the page design. I started by creating the header navigation panel with the navigation buttons and the company logo. I decided to split the page in to three different sections: Navigation header, content body and the bottom footer with additional links. The bottom footer gave me a slight headache because in the original design the text size was extremely small. I decided to take some artistic liberties and change the font-size a bit bigger. Next, I added the four blocks to the content section. All of the blocks were a bit different, but I tried to use the least amount of different classes for different blocks and sections in the blocks. My friend did not provide any reference photo for the responsive web page design, so I had to implement my own. I decided that the text portion of the blocks are the most important ones, so I scaled down the amount of shown picture for the text to be readable on smaller screens. I’m quite happy with the result, though I think that my friend’s idea of mobile optimized version of the page would be completely different.