**Lappeenrannan teknillinen yliopisto LUT**

**Software Development Skills (3cr)**

**Arttu Käyhkö**

**Learning diary**

**30.5.-30.7.2024**

**In your git repository there should be the following:**1. Your exercise materials  
2. Learning Diary  
3. Your Project  
4. ReadME how to run your project  
5. File including a link to a **video** of your project running (included into the README!)

To begin with I read through the Moodle info page before starting the course (something I finally learned to do) and was glad to see familiar items on the list such as GitHub and VS code. Easy to say those were my initial choices.

Next, I moved onward with the guidance of the setting up workspace page + first video lecture, and after some problems managed to get everything working as intended, great! After this there was nothing else left to do except learn about front end development and new code languages.

**[Part 1] - Introduction to workflow and sass (1. YouTube lecture video)**

1. You will learn to setup environment for coding

- The installation instructions, in addition to the courses Moodle page, were simple and easy to follow, but I stumbled upon some Windows related issues while doing this. Luckily the internet was very helpful on this and I managed to fix it easily by just installing other dependency.

2. What is sass and how to use it

- Sass is short for “Syntactically Awesome Style Sheets” which is an extension of CSS, which is used to make webpages. They control where and how objects appear on a webpage.

3. What responsive design looks like

- Responsive design means that the website elements adapt automatically to the current screen size. It looks “adaptive” for example when browser size, device, orientation, or screen changes.

**[Part 2] - Homepage and Core Sass/CSS (2. YouTube lecture video)**

1. In this part you will create the homepage HTML markup

- In this part I faced some unexpected and major problems related to me changing the name of my folders and files (silly on my side). After 1-2h of hot fixing luckily, I got it to work. Other than that, the video was great and informative.

2. You will create the core Sass/CSS for the project

- Sass and CSS functions as base code, which enhances the capabilities of original languages. This includes colors, icons, etc. often seen in today’s websites

3. You will learn about variables, nesting and more

- Variables are like memory sticks which you set to certain value and they keep it if not changed. Nesting means structure where the below information is inside certain item, and thus does not show elsewhere.

**[Part 3] - Rotating Menu Button (3. YouTube lecture video)**

1. In this part you will create pure CSS menu button

- This was something that seemed very complex at the beginning but was very clearly explained by Brad in the video. JavaScript seems to work very effortlessly with the website design, which of course is very good to know.

2. You will add the JavaScript-code to handle the rotating

- The code was professionally explained and worked as intended

3. You will learn to use CSS transitions to rotate the button into an X

- This functionality felt a bit niche for the starter video, but I guess it is a great showcase and many customers want to have it in their websites.

**[Part 4] - Menu Overlay & Responsiveness (4. YouTube lecture video)**

1. In this part you will create the menu overlay

- The overall overlay with the different backgrounds/see through effects in my opinion looks fantastic. Also, it was surprising how easy it was to implement in action.

2. You will also add media query Sass mixins

- Mixins function as kind of a “global” variables that can be accessed as functions/set values instead of the need to write the same code over and over again. In this case for the different screen size presets.

3. Learn to add branding and navigation to the site

- Branding has always fascinated me in the UI & UX design, and I think this video gave a great sneak peek into that and the basic possibilities in sass.

**[Part 5] - Page With CSS Grid (5. YouTube lecture video)**

1. In this part you will create about page using CSS Grid

- CSS grid, after looking at the great explanatory video about it, seems an excellent way to place objects into the website layout. Simple and effective, nice!

2. You will also add grid template areas to map out the layout.

- The template areas create a very freely flowing way of developing and designing websites, which is always what a developer wants to see.

3. You will also learn to add Sass styling and make the page responsive

- I was a bit surprised how easily the styling could be added to the pages, but hey not complaining.

**[Part 6] - Work and Contact Pages (6. YouTube lecture video)**

1. In this part you will learn more about CSS grid

- The CSS grid still seems very straightforward and easy to navigate

2. You will learn how to use Flexbox

- Flexbox seems like a very easy way to add and style a text box. I am sure you can add more into it if you want, but for now this will do

3. You will also learn the properties of both layouts

- The video explained this very well, and the overall layout functionality is easy to follow and understand. Kudos once again to Brad

**[Part 7] - Website Deployment (7. YouTube lecture video)**

1. In this part you will learn to deploy your website with GitHub Pages

- As earlier, the deploying process was explained very well, and I had no issues with the deploying process. Smooth as butter. One thought is that I would not like to have a website in the web that I don’t frequently update, so I guess that is something I have to do now. Glad the page looks cool.

Project – Summer cottage rental website

At the start of the project work I decided to make something that has been on my radar for way too long. My grandpa is renting three summer cottages, and I have discussed multiple times about making them their own website. Of course, the own website would need advertising which makes it less of an option when compared to big commercial websites like Nettimökki. But that being said, I wanted to see what I can come up with since now would be the perfect opportunity.

A screenshot of a web page

Description automatically generatedA screenshot of a computer screen

Description automatically generatedThe development process started from the drawing table with sketches about the “Home” page and other pages (5 pages/views in total):

Image 1. First draft of the home page

Image 2. First draft of the content pages

The base for the project is the “Modern portfolio” already developed through the course works and Brad Traversy’s videos. I knew many things needed to be changed, but then again what is the point of making an actual website if I would just make it bad. Also, it is worth mentioning that OpenAI’s ChatGPT was crucial tool in the development process. It is used throughout the development process, but it had some major limitations with grasping the big picture and my vision. Therefore, I would say that the development is done by me, and code structuring and first drafts developed using ChatGPT.

**Disclaimer!** I have permission to use actual pictures of the cottages and other relevant info texts from the cottage’s actual advertisement pages. The staff pictures are generated using OpenAI’s Dalle image generator and are not real images.

A screen shot of a computer

Description automatically generated

The development started with making the project folder and

structure which ended up being as presented in image 3.

From the main.scss everything is compiled into the

main.css file using the recursive logic in the package.json file.

Main.js contains the JavaScript functionalities.

Html files contain the page layouts.

Scss files make the main functioning logic for the html files

and contain the required classes used in the project layout.

Image 3. Folder/file structure

From various hours used in VScode, paint and ChatGPT came out something actually decent. the end results looks as follows:

1. Home page

A screenshot of a website

Description automatically generated

The “Home” page or landing page acts as a high-level view of the cottages available. The pictures/flexboxes all have on hover functionality and clicking moves user to the desired page.

2. Köllölä 1 page (oldest cottage)

A screenshot of a log cabin

Description automatically generated

This is the more thorough review page of the Köllölä 1 cottage. Images function as a photo-reel and contact form + map are interactive. Everything is responsive and uses CSS Grid. The contact form checks that every box is filled before letting the user send the form. Upon sending the user is informed of successful send via pop-up window.

3. Köllölä 2 page (middle cottage)

A log cabin on a river

Description automatically generated

This is the more thorough review page of the Köllölä 2 cottage. Images function as a photo-reel and contact form + map are interactive. Everything is responsive and uses CSS Grid. Upon sending the user is informed of successful send via pop-up window.

4. Köllölä 3 page (newest cottage)

A screenshot of a computer

Description automatically generated

This is the more thorough review page of the Köllölä 3 cottage. Images function as a photo-reel and contact form + map are interactive. Everything is responsive and uses CSS Grid. Upon sending the user is informed of successful send via pop-up window.

5. Extras and staff page

A screenshot of a website

Description automatically generated

A screenshot of a contact details

Description automatically generated

The Extras and staff page presents the available extra services and display further details on a pop-up window when clicked. Below is a short staff introduction and contact details as flexboxes.

**The project objectives were:**

1. Page with navigation

- This page is the “Home” page, which has links to Köllölä 1-3 pages and navigation menu to move to the desired page. The menu is also displayed on the other pages.

2. Responsive styles with CSS

- All of the pictures, texts and other objects are made responsive using CSS.

3. Flexbox and CSS Grid

- Multiple flexboxes are used to give reader information. CSS Grid is used to maintain the layout on different screen sizes.

4. Deployed Website

- Website is deployed using Github’s deployment functionalities. See Github repository readme.

**Reflection on the course:**

Overall, I found the course to be well structured and the time-to-study points relation to be well justified. Great course to give a quick sneak peek on the front side development related to websites!