Lappeenrannan teknillinen yliopisto

LUT UNIVERSITY (School of Technology)

Software Development Skills Front-End, Online course

Ekong Rejoice, 001494262

LEARNING DIARY, FRONT-END MODULE

Date : (8.5.2024)
Activity : Course Overview
Learning outcome: I Learnt about git repository, git commit, git push, vs code addons.

Date : (9.5.2024)
Activity : Introduction To Workflow And SaaS
Learning outcome: I Learnt Node.js and npm installation, introduction to SaaS

Date : (9.5.2024)
Activity : Homepage And Core SaaS/CSS
Learning outcome: I Learnt about HTML markup, simulations of SaaS and CSS mixin, @include, viewport height (vh), rem, variables in SaaS, nesting

Date : (9.5.2024)
Activity : Rotating Menu Button
Learning outcome: I Learnt on for-each method, transform and translate in SaaS, Font Awesome Icons, partials, @import

Date : (9.5.2024)
Activity : Menu Overlay And Responsiveness
Learning outcome: I Learnt translate 3d, for loop in SaaS, media query

Date: (10.5.2024)
Activity: Page With CSS Grid
Learning outcome: I Learnt grid layout, SaaS calc function, grid gap.

Date: (10.5.2024)
Activity: Work And Contact Pages
Learning outcome: I Learnt about linking several html pages, CSS grid.

Date: (10.5.2024)
Activity: website deployment
Learning outcome: I Learnt on git push again, deploying the website, getting custom URL.

Date: (10.5.2024 – 12.5.2020)
Activity: Project development and deployment
Learning outcome: I developed my project, faced some issues, solved all issues and deployed the project website.

LEARNING DIARY (8.5.2024 - 10.5.2024)

COURSE OVERVIEW

8.5.2024

I reviewed the general material and grasped the course's main goal, which is to discover my passion as a software developer and create a unique project to showcase my abilities. I chose the front-end module since it was the most intriguing project available. I've also tried to configure my environment. I created a git repository and make my first commit; everything went well when I clicked the banner to view an introduction to GIT.
I choose VS Code as my code editor for this course because that is what I use, and I used to it. I refreshed my memory on addons and put the ones that were provided in the youtube video. I started to watch the first part of the example project to understand the technologies better. I did my second commit, and it went very well. I decided to learn more and how to push my files to my github repository and it went well the first time. Hours later, I forgot the command and I had to go back and read it again.

INTRODUCTION TO WORKFLOW AND SAAS

9.5.2024

I started watching the introduction to workflow and SaaS video and I followed smoothly until the part where I had to download node.js and npm also. My coding environment was set and ready by just watching about 5 minutes of the video. I refreshed my brain on what sass was because this was the first time in a long time that I used it, so I had to remember what it was again. I admired the youtubers' responsiveness. They were so clean, but I have worked a lot with responsiveness in other programming language, so it seemed easier to learn but interesting as well. I already had node.js and just installing npm was the problem I had. I tried installing npm and getting it to work but unfortunately it didn't work, and I did this for several hours until I found out that the problem was that I was using PowerShell in my vs code terminal and instead of the command prompt terminal which worked immediately I tried using it. This took a lot of time out of me, and I would have had time to go for another topic, but this drained me just to find the solution to this problem. After that, I continued to follow to the end and did the Git commit and the Git add for all the files that I created for this course material. Every other thing went well as we did not push the files to the Git repository yet.

HOMEPAGE AND CORE SAAS/CSS

9.5.2024

This section was very beginner friendly and very easy to follow and maybe I will say it is because I have already had little experience with all of this, so it was okay for me to follow along. I refurbished my brain on how to create a HTML markup and found out that it was quite easy to do. I learnt more about the SaaS, and it is almost like CSS and very good to follow, I found out that everything that was happening in the main SaaS file was being converted and sent to the CSS file and this was what was making the website update from the live server extension that we downloaded in the first lesson. The video explained mixin properly for me to understand and it was explained as a block of code that lets us group CSS declarations we may want to reuse throughout our site. I can pass in values to make your mixin more flexible. The @include is used together with the mixin as it lets it know that its body will be used by the whatever is calling, it is like calling a function. I finally understood what vh which was a CSS unit representing "viewport height". It is based on 1% of the viewport's height. For example, if the viewport height is 900px, 1vh is equal to 9 pixels. It is useful for building responsive designs that scale with the viewport height. Then the other unit for measurement that I finally understood was rem and he explained in a very detailed manner, but from what I understood it was that rem only applies to the root (or html) font size. If the base font size is 16px, then 1rem equals 16px. It is useful for ensuring proportional spacing and sizing in your designs. I had an idea of how variables and nesting, so it was smooth to understand how they worked.

ROTATING MENU BUTTON

9.5.2024

This section was one of the most difficult that I have watched but this was mostly because of the JavaScript, and I struggled to understand very few things and how it works but by the end of the video I understood quite well. The area in which I was struggling was the For-each method which I later I understood through stack-overflow as is a built-in array method in JavaScript that executes a provided function once for each array element. It's a way to iterate over the elements of an array. I don't usually use the syntax, but I found out it is quite easier than using the for loop or the while loop or the silly if…else statement. Speaking of the if…else statements were also used in the tutorial, and I knew a lot about. The transform and translate syntax were ones that I got to play around with because, I did not think of using that to change an icon and I always just change the icon instantly. But it was fun to see the transition from a menu button to a cancel button and it is very understandable. We also got to import Font Awesome icons, and this was where we used the transform and translate for them. I am not really a big fan of JavaScript, but I am still knowledgeable quite a bit in the subject matter and I got to use the document object module to alter the behavior of various HTML and CSS objects as well. In the JavaScript file that we made used of in this lesson we mainly made a function for the menu button to alter its behavior. However, in the previous lesson and in this one as well we made special types of SaaS, and they have to start with an underscore (_) and they are called partials. The underscore lets SASS know that the file is only a partial file and that it should not be generated into a CSS file. Partials are used with the @import directive to include the content in other SASS files, which is what we used in the main.scss file.

MENU OVERLAY AND RESPONSIVENESS

9.5.2024

As the heading suggests, in this section we covered almost everything about making a menu and the menu overlay with the addition of making it responsive. We used design styles like translate 3d, made the website responsive for laptops, medium phones, big screens, and small phones and after checking the responsiveness for almost every phone in the chrome developer tools, I would say it is quite good and very responsive even when I put the phone to have a very small width and a very small height as well. I found out that it is possible to do for loop in a SaaS file which is still intriguing, and we used it for the transition delay of the menu options, so that they can show 0.1 seconds after its predecessor has been displayed and which gives a very smooth transition. I tweaked the settings for the for-loop and added some more features I found on other YouTube channels. The branding of the menu was very fun, and it was not head scratching, this made the navigation of the site very fun to work with. Overall, I created the menu overlay according to the tutorial and did some mixin for media query which was for the phone sizes.

PAGE WITH CSS GRID

10.5.2024

In this section we started by adjusting the text color based on the background color's lightness. This was done with the set text to color function and this function takes an initial color variable and returns either black or white depending on the lightness of the passed color. The function is implemented in the main.scss file which replaces static color values with calls that are made to the function where any color can be passed as an argument. We also delved into creating the about page which we began with setting up the HTML structure which took a lot of time because we had to set the grid layout. I considered the grid and the grid-gap syntax which I played around with before I could understand its behavior. The content structure for the About page is outlined, including sections for bio, jobs, and a footer. I thought of other ways that the footer could have been sticky to the bottom of the page, but I think the mathematics that was applied in the CSS was also a fun way of doing it. The sticky footer was implemented using the calc function in sass to ensure it stays at the bottom of the page regardless of content length. Also, Specific grid template areas were defined for different sections of the About page, including the bio image, bio text, and job listings. We put the imported mobile at the end of the main sass file size because we wanted it to run last as that is also how CSS takes in information and process it which is from top to bottom. Overall, the page was very responsive.

WORK AND CONTACT PAGES

10.5.2024

This section was quite easy to follow along as I learnt the flexbox and how to manage and use it with the responsiveness of the page. I forgot to commit once I was done with the about page and went on and started the HTML file of the contact page but once I remembered I committed it with the contact page that I already designed when the video committed. For the Font Awesome icons with the Fab and Fas, I forgot that the social media icon will need Fab, so I was so confused until the tutorial found the problem and fixed it easily. I would say I learnt from experience and will try to avoid that mistake in the future. I altered the grid template columns by making a situation where we had more than 5 projects and how it will look and be responsive and I must say I was quite surprised of the power of sass when it comes to responsiveness because it was quite very good. The grid template columns reduced by 1 as the screen became smaller and smaller and I expected this to happen because that is how I understand it. The projects section was styled using CSS Grid, with adjustments made for responsiveness. Images are set to 100% width of their container, and a grid gap was introduced for spacing. I went on to research different kinds of hover that I could use for my personal projects and I have some in my itinerary. Navigation links were updated to correctly link to the Work and Contact pages, ensuring users can easily navigate the site. Initially they were with slashes which just brought them out of the webpage and into a JavaScript section that the HTML has. I committed again at the end of the tutorial, and I am getting used to doing that as I go along.

WEBSITE DEPLOYMENT

10.5.2024

All that was done in this last section was the deployment of the website as the title says. I created my git repository, which is not my first time, so I quite knew and understood my way around it. For the deployment of the website this was where my problem was. I installed the gh pages and this went well. After I put my homepage URL and the deploy script in the package.json I tried pushing it to my main and I kept receiving errors but I did not understand what this errors meant until I searched the internet and then I understood that I was pushing my project to a main which did not exist and I was suppose to push it to my master and when I did, everything came together and began to work. Then I used the npm run deploy and initially, when I put my URL in chrome, I was receiving 404 errors and once I tried again with the run deploy it worked. Then I had noticed that my sass file was not updating my CSS file because I forgot to run sass at the start of the day when I continued the project. I had to go back and add and commit all of it again after I ran the sass file for the CSS to be updated. I was thinking of getting a custom domain and I realized that it would cost money, but I plan to buy some in the future. Finally, everything went as planned and went well and I think it was a very good and wonderful series.

PROJECT DEVELOPMENT TO DEPLOYMENT

10.5.2024 - 12.5.2024

Starting this project, I did not know what I was going to develop and I felt like I should do the portfolio page like the one that was done in the course material video, change somethings and make additional features to the website. After a long while of thinking and contemplating, I decided to make a website that sells items. This was because I knew it was going to enable me to practice more on responsive pages using Sass. I started with the design of how I want the website to look as this is the first step I must go through before the development of any software, and it is the planning and design phase. So, after about an hour of designing and planning, I decided to start the coding of the software. I started with git init and created the necessary html files for my project and I also created the corresponding scss folder with the main.scss file for the project. As I went along with the developing the index file which is going to serve as where the homepage is going to appear, I decided not to do a side navigation bar and I decided to stick with the traditional top down navigation bar even and no navigation with bigger screens like laptops and so on. I designed the navigation bar or in this case the header that was going to be a navigation bar on smaller screens as I envisioned on my notes. Furthermore, I decided to keep the use of the navigation button that was used in the course material. I wanted to practice how to make it by myself and after multiple attempts I went to look at the javascript of it and I found out that I made one or two mistakes and it was case sensitive, which I forgot it was but later fixed.

I went to read the course project requirements again and got confused in where it said ‘Responsive styles with CSS’. I thought this meant that you can also make the project without the Scss, then I decided to email the lecturer and from what I understood from the email that I got back was that I could not just use plain CSS but I could use Bootstrap as well. This became very clear to me and I decided to create a new project (this was not really a new project, it was the same project but with another file and styles). I made the project and installed bootstrap and all its necessities for it to work. I have worked with bootstrap and I think it was one of the easiest to use when developing responsive software, But with this project I did not create a local repository and git add all the files and git commit as well, because I thought it was only going to slow me down and I decided to write all my milestones in a paper because this was my preferred way to work. Moved straight to developing the home content which just shows a sale going on with a button that will take the user to the product page and the user can buy anything on sale from that page. I decided to write the preview of the product page in this homepage because after getting inspirations from websites like amazon and e-bay, I noticed that a preview of the products page can be shown in the home page. I also realized that it was going to be a more efficient way of working because making it also determines how fast that the product page is going to be developed alongside. So, I decided to start the product cards. I had not made any predetermined sizes for the cards and so I went on to play around with different styles and I finally agreed on one and this was the best that worked for me. I then wanted to start making the responsiveness of the little page that I had because it was also a factor that was going to determine how the product cards are going to be structured. I played around with my navigation button and then I chose that it should be shown on smaller screens and medium screens, then the navigation bar should kind of be a dropdown that will show all the navigation pages and it was still going to use the navigation and cancel button that was used in the course material video. After working with the responsiveness for the navigation bar I decided to move to the home content once again and fixed it responsiveness, because the image I put there was not responding properly on smaller and medium screens. I worked and played around with different things until I finally got one that I really liked with the bootstrap code and the CSS media screen which is also an important factor when building responsive pages, I changed the font sizes and height in various styles that I thought was going to work with me while programming. After developing the product cards and its responsiveness, I went on to writing codes for the offers which shows all the customer services that this company provides like free delivery on products with a certain amount of money and also free returns. This was where I decided to install icons and Font Awesome icons was one of them, I also went on to use google icons and I thought they were easier to use because their naming system is very simple and less hard to make a mistake unlike Font Awesome's icons. However, I still used Font Awesome's icons. I also decided to add icons to the product cards because this was going to make things even more realistic and I am glad that I did this because it filled up some spaces that I was willing to fill up with other complex things and this saved me a lot of time, I then added the responsiveness to the offer area with the column and row grid system in bootstrap which is what I mostly used with other responsiveness in the website.

Began to write the code for the newsletter and which was the section in which users will subscribe to the newsletter that the company provides 'weekly'. This was another way of working with my responsiveness skill and I think this was kind of easy, but my brain was still getting used to it. I moved on to the footer which I thought was going to be a nice touch to the end of every page, this means that I wanted every page to have the nav bar and the footer with their content in between both of these features. I added the content of the page which was initially going to be just the copyright and the year of development; however, I got even more inspiration from the Amazon website, and I decided that they should be a top part of the footer and they should be a bottom part where the copyright and the year of development will be. I added things like terms and services and the companies and customer privacy policies. This was a good time to add more of the contact ways to reach out to us the company and then I thought this was only going to clog up the whole footer and I decided to try and keep it simple. Then I got the idea to add the social networks in which how they user can reach out to us with icons of the social network, and I thought this was very decent and did not choke up the footer area. Below the footer code is where I added the script codes to the script file, this contained the script for the navigation button. Then I moved onto the product page where I mostly duplicated the header, the footer and the product card that was already in the homepage. Straight into the contact page, I made the flexbox for the contact for the contacts, and I did it with bootstrap and CSS a little. I added the header, footer, new later input area where users can send messages with their own name and contacts information and alongside a message. The about page was no different from the others with the header, newsletter, footer, and the offer. Although, I decided to add a little than just that to the about page. I wanted to test my skills in responsiveness with images and text that are side by side together and then I added some lorem ipsum text, and this text basically shows the about of the company and what the website is all about. Then I decided to add an image and this image has its description which is also a lorem ipsum text but basically that is all about it. I went to make the nav bar sticky so that it sticks to its position when scrolling and I must say this took some time and research on YouTube, but I finally got the answer to my question. I also decided to add another JavaScript code in the product page where half of the products will not be displayed initially but there will be a button that says ‘Show More’ and when that button is clicked the rest of the button will show and the button will say ‘Show Less’ and after a long period of time I was done with implementing this feature to the website and everything turned out great.

It was time to deploy my website project to GitHub and initially everything was going very well. I created a new repository on GitHub and installed the necessary and changed the branch from master to main which was what GitHub suggested. Then I created a local git repository with git init and then I had to install gh-pages which came in with folders like node-modules and also the third JavaScript apart from the one that carries git installation package. I added the homepage which holds the URL to the project and I then added the deploy to the script tag of the package.json and then I of course had to install gh-pages for every other thing to come in. Then I added gitignore to hold for the push to ignore the node modules file. I then decided to add all files to the git repository and then do my first git commit and all of this went well and then I did my first npm deploy and seconds later my deploy was ready. When I ran it I found out that it did not run with the CSS and java script, I realized that I did not put all of this in the dist folder which is what holds all the HTML files and then I decided to add all of this into the dist folder, did the second git add and git commit for this change to apply. I pushed this to the GitHub repository and then deployed it. Everything was working properly, I later found out that one camera image was not showing and the Show More was not working properly. I took one problem at a time, and I decided to check if it was working properly locally, and it was. After several git adds and git commits, I found out that the name of the file did not match the image that I brought in, well it did but it was case sensitive. After noticing this, I altered the change and then did another deploy and the image I was missing began to show. Feeling more motivated, I went on to attack the other problem which was why the Show More button was not working and after multiple inspects, I found out that the same java file which was working for the navigation button was returning a 404 error for this other button. I did several gits add and git commits and npm deploy after making several changes and nothing seemed to work even though it was working accordingly locally. I decided to make a new JavaScript file containing the code for the Show More button and then suddenly, I was surprised that this time it worked. So, I stuck to that and as the saying goes “if it is not broken, don’t fix it”. In conclusion, I have a working website that works how it is supposed to and this is all thanks to the course video and instructions.