Lappeenrannan teknillinen yliopisto

School of Business and Management

Sofware Development Skills

André Filipe Ferreira Miranda, <Insert student number here>

LEARNING DIARY

31-03-2022

Introduction to workflow and sass Module

Today it was my first day at the course.

Through the lesson it was possible to learn and understand how a simple project can be started.

For writing the code, we need a code editor. It is recommended the use of VS Code. On this code editor, there are a lot of great and useful extentions that will help to code faster and better,. In this lesson we learn about Live server, which provides an immediate update of the browser after saving changes in VS Code. We learn also about Bracket Pair Colorizer which give colors to different parts of the code, helping us to don’t get lost about what it’s what, and also Prettier Formater, that will automatically format our code after we saved it.

We learned as well about the importance of Node.js. This will be present in all our projects since through it we can install many packages that we will use in our projects, through the npm command.

Also git is mentioned. Used to version control.

In this lesson is showed how to start using the sass. First we need to install it with the npm install node-sass. Then we learned how to organize the folders and starts writing some sass lines and how it affects the appearance of the project.

05-04-2022

**Homepage and Core Sass/CSS**

During this course I learn about how to build the homepage using html and sass, making use of classes and id’s.

I learn also how to link the JS file to the html through the use of a script tag in the html file.

I learn how to use icons, by first import the script from the source used to the html file, then, giving it a i tag with some specific classes.

Regarding to sass I learn how can I use nested classes or id so that I can avoid over repeating of same terms.

Also the creation of variables and mixings for transitions was learned during the course.

An interesting and important part was also how to create a simple overlay for a background image.

06-04-2022

**Rotating Menu Button**

During this lesson we learn how to make the menu button rotate. For that to happen it was needed to work together with javascript and css.

While in css (or scss) we could edit our classes and apply the nice transitions, it was needed to use javascript to handle events and allow the classes to be added or removed according the clicks from the user.

During this process I got some difficulties with the transform property on css but I got to understand what I was doing wrong. A missing of a colon was not allowing the property to take effect.

Also I got to understand that repeating properties was overriding the property.

**Menu Overlay & Responsiveness**

I learned here how to do the overlays and animate them. Important tasks here was to define the display as flex so that both divs would come side by side, and also we defined that each div should be 50%, so they would occupy both same with available.

Then the use of translate3d property was useful to make the divs slide from bottom or top and display. Also needed the use of the transition for this.

We learned also in this lesson how to use loops in sass. In this case we made a loop that could be compared to a for loop on javascript to animate all the items of the navigator.

In this lesson I learned as well how to make the homepage responsive. For that it was used media queries, for four different device sizes.

In this media queries we used the min or max width to define what device queries are used in some device.

We created a new scss file to edit all the media queries, what make our code much more organized and structured.

**Page With CSS Grid** 

We start learning how to create functions in scss and how to use it to change dynamically the color of text in the webpage, according what kind f color is our primary color.

In this lesson we started building the about page where we use the grid display to mage the position and display of our content.

For that we have to set the display of the div we want to “grid”. After that we set the area and column as we want the page to be ‘divided’.

The elements will use as much space a«possible has our ‘grid preset’ allows it.

I learned has well to make a sticky footer, using the calc() property to set the minimum height of the div to be at least the size of the screen except the height of the footer.

For media query we need then to change the grid-template-area, and grid-template-columns to fit the screen as we want.

07-04-2022

**Work and Contact Pages**

On this lesson we continue the learning of using the grid to have control over the projects.

We learn as well how to extend classes in sass. This is pretty useful we pretend to attribute many same properties to different classes;

We used as well the flex-box system instead of grid. We tested some of the flex properties to control the display and responsiveness of the contact boxes.

**Website Deployment**

In this lesson we learn how to create a repository and commit and push the project to it in github. Then we learned how to publish the project in github pages making use of the cli.

Important do not forget to add the homepage key-value in the package.json file.

We learn as well about the possibility of using a custom domain and also some sources we can use to buy the domain.

09-04-2022

**Project - Homepage**

This day I started the project. First, I had to choose what would my project be about. I decided to take a simple approach and make a simple and small website about the season of the year.

This way I divided my project in two parts. The homepage and the seasons page.

Today I did the homepage. To this homepage I split it into two phases. The layout page and the menu.

The layout was quite simple to do, and is taking a similar style to the exercise homepage layout.

To the button, I took a bit of same approach, with the three div’s creating the button and on click making the rotation to an X.

Although, changed the behavior of the menu. I didn’t want it to occupy all screen and coming from two direction. I wanted it coming from the right and occupying only part of the screen.

Also, since my portrait was already in the homepage, I didn’t want it in the menu, but instead, a logo, on the top.

So, for that I had to reduce the width of the menu for 30vw, and also change the translate3d according what I needed.

After the menu has been finished, I just made it responsive to the different scree sizes. So for small and medium screens, the menu will occupy all screen and also I made it so, that there’s no opacity. To large screens it will occupy 50% of screen and extra-large, the base 30% of the screen.

10-04-2022

**Project – Season pages**

For the season pages (one page for Summer, Autumn, Winter and Spring), I wanted It divided in four parts.

The top part should be just a title with the name of the season and a small describing sentence.

The second part would contain a picture and also a description of the season.

For that I used the grid to give 1/3 of space to the photo and 2/3 to the description.

The third part would be a facts section and a slideshow with images.

For that, I used as well the flexbox to give 50/50 space to both. The facts div is a simple unordered list. On the other side, the slideshow required a bit more work and the use of javascript.

To implement this slideshow I had to create some variables:

1. index: would start with value of 0, and is used to control what picture to display.
2. pictures: this would be an Array containing all the pictures with the pictures with the class ‘seasonPic’. This array would contain all the picture that would be available on the slideshow.
3. imageToShow: this would define what picture would be the one to show. So basically this is ‘pictures[index].
4. slidePic: this would be referring to the div where the picture would be shown in the html file.
5. arrowBack and arrowFront: This would represent the arrows in the slideshow where the user can click to change the picture.

After the variables have been defined, I created the Event Listner on the arrows.

So on click, the function created would first check if the index would be higher than the length of the pictures array. If that would be the case, then it would make the index to be -1. After that it would increase the index in one value, so to 0, showing then the first picture of the array.

If index was not higher then the array length, then it simply increased the index in one value.

Then, slidePic.src would be settled to be imageToShow.currentSrc. This currentSrc is the key attribute in where is stored the value containing the link to the image on the web.

The function to control the change the pictures backwards has the exact same process in a reversed way. We decrease the Index in one value, and also we check if the index is lower than 0. If yes, then we increase it to be array.length-1.

The last part of the page would contain a gallery with all the pictures.

This part was done using flexbox.

It was given a display of flex to the container. And each item would occupy 30% of the space available.

This would allow that three pictures are shown per line.

11-04-2022

**Project – Season pages - responsiveness**

Now the remaining thing to do was make this pages responsive. Both flex and grid are a good help for it. For the second part of the page I changed the grid-template-area and grid-template-columns so that there would be only one column per line occupying all 100% width.

This would make the picture would appear above the description.

For the third section I turned the flexbox direction into column, so I would achieve the same result as I did in the grid.

Then, for the Gallery section, I wanted that for large and medium screen it would be 2 pictures per row, so for that I screen sizes I set the width of the pictures to around 50%.

For small size screen, only one picture will show for row, so 100% of width used in the pictures.

I also made some changed in the text size and some margins so that text would display better according the size of the screen.