**TASK 1**

**introduction to workflow and Sass (Part 1)**

**7.5.2022**

The course accomplishment started by checking out the descriptions presented on the General Course Information page; according to this page, it was tried to be generally familiarized with the other course assignments as a preliminary overall course review. On the Environment Setup page, instructions regarding the installation of Version Control (Git) and code editors were skipped entirely. Since Git commands are sometimes challenging to use, especially whenever there is a conflict between commits or pushed tasks, another Git course from Tampere University was added to the course tasks as an external reference. The main programming languages used during the course were well-known *HTML*, *CSS*, and *Javascript* languages, but the Sass package and other code editor extensions were used for the first time. Therefore, related packages were added to the VS code, preparing to begin coding tasks.

**8.5.2022**

The project's local repository was made via the command *mkdir* , and VS was called to add two folders by the names of *dist* and *scss;* then, an *HTML* file named *index.html* was added to the *dist* folder and a *CSS* file named *main. scss* to the *scss* folder as well. During the opening of the VS, an error message related to *node.js* and *Sass* was encountered, and the problem was solved thorugh reinstalling the *node.js*accordingly. *Node-sass* was appropriately acting to compile *Scss* files to*CSS***.** Once all essential tools were prepared, the *node-sass* package was used to convert the *main.scss* file to the *main.css* file containing styles to be added to the *index.html.*

**9.5.2022**

Version control course (Tampere University) contains four modules as the introduction, basics, intermediate, advanced, and GitLab, respectively. Tha primary purpose of this Git course accomplishment is to review related contexts and practice more. The introduction of the Git course has been studied, and according to this module, the Git structure could be depicted in the following figure 1, showing the main commands and data flow paths between repositories.

Diagram

Description automatically generated

Figure 1:Git version control and main commands

**11.5.2022**

During this day, GitHub and Gitlab remote repositories were being checked out to be ready to save the course example project and other tasks such as the final project. Git basic commands used to move course files between local and remote repositories were reviewed, the example project was initialized, and some files and changes were added and commited to push the first version of the example project to the remote repository.

**TASK 2**

**Homepage and Core Sass/CSS (Part 2)**

**16.5.2022**

After primary project configurations, the main coding task was started by developing the <**body**> tag of the index.html file, the first page of the project, by adding a <**header**> tag containing <**nav**> and <**div**> tags. The new project version, including the last modifications, was committed and pushed to the remote repository.

**17.5.2022**

In the **<body>** tag we added a **<main** id=” home”**>** tag which contains **<h1** class=” lag-heading”**>** and**<h2** class=”sm-heading”**>** tags and a **<div** class=” icons”**>** containing <**a**>tags and related **<i**> tags as well.In this project we used the Awesome icons which would be include into the Html files via a <**script**> tag written into the end of **<body>**tags containing the awesome website’s link as the following Html code(see Figure 2):

<script src="https://kit.fontawesome.com/a9f8805286.js" crossorigin="anonymous"></script>

**Diagram

Description automatically generated Figure 2: DOM tree for the Html structure of the example project**

Another <**script**> tag containing the directory of the project JavaScript file (main.js) has been added to the end of the **<body**> tag as well.

**17.5.2022**

After completing the Html structure of the home page, it was time to change the style of the tags by adding the styles to the main.scss file. Briefly, the main adjusted styles can be listed in the following cases :

1. text font size and font-weight, and other related styles.
2. The <body> background picture was changed, and an overlay was added to increase the readability of other objects.
3. <header> position was fixed at the top of the page.
4. A hover style was added to icons, and a transition function was defined for
5. Overflow for the <main id= ”home” > was adjusted as hidden.
6. The sass variable and functions were moved to a new module of \_config.scss

All new project versions developed during the coding tasks were commited and pushed into the remote repository.

**Rotating Menu Button (**[**Part 3**](https://www.youtube.com/watch?v=7WaohfclZRs)**)**

**18.5.2022**

According to the example project, in this task, we used javascript to work on Html tags and included better styles for the menu button**.** First of all, we locate the button in the top right corner of the header. The javascript syntaxes needed to learn during this task since the javascript cheat sheet would be the best reference for the overall review of the main.js file. Through the task, we can better understand how Html, CSS, and javascript can be combined to handle a web programming project. This task shows how an event such as clicking on the page objects (here menu button) can be related to tags and styles. Another highlighted aspect of the task is rotation and translation to add motion to the project.

**Git Basic & Intermediate Level**

**19.5 - 24.5.2022**

during the course basic level (Tampere university course), we reviewed the following discussions, and tasks were accomplished and submitted to be evaluated:

* [Creating a local repository](https://plus.tuni.fi/tie-git/v4/02_basics/03_repository/)
* [Staging & creating commits](https://plus.tuni.fi/tie-git/v4/02_basics/04_staging/)
* [Basic usage of version history](https://plus.tuni.fi/tie-git/v4/02_basics/05_using_commits/)
* [Exercise Log\_&\_checkout](https://plus.tuni.fi/tie-git/v4/02_basics/05_using_commits/#chapter-exercise-1)
* [Synchronizing repositories](https://plus.tuni.fi/tie-git/v4/02_basics/06_push_pull/)
* [Exercise for push &\_pull](https://plus.tuni.fi/tie-git/v4/02_basics/06_push_pull/#chapter-exercise-1)
* [GitHub](https://plus.tuni.fi/tie-git/v4/02_basics/07_github/)

One of the most highlighted and crucial functions of the version control systems in the programming project is taking part in a team of colleagues or joining project branches as well as merging the branch history with another one (mostly with the main branch). It should also be considered that merging the branch histories could cause conflicting situations; hence, these circumstances were discussed in more detail during the intermediate level of the Git course. The main subjects of this level could be summarized as the following ones.

* [Working with Others](https://plus.tuni.fi/tie-git/v4/03_intermediate/01_intro/)
* [Teamworking in Git](https://plus.tuni.fi/tie-git/v4/03_intermediate/02_working_in_group/)
* [Branches and remotes](https://plus.tuni.fi/tie-git/v4/03_intermediate/04_branch_remotes/)
* [Moving and merging branches](https://plus.tuni.fi/tie-git/v4/03_intermediate/05_merging/)
* [Submodules](https://plus.tuni.fi/tie-git/v4/03_intermediate/07_submodules/)
* [Tagging (or naming) versions](https://plus.tuni.fi/tie-git/v4/03_intermediate/08_git_tag/)

**Better CSS With Sass**

**25.5 - 29.5.2022**

Text, letter

Description automatically generatedSince, in this course, we have to write out a lot of Scss (Sass) rules, an overall review covering the core literature regarding the Sass (Scss) would be the best choice in order to proceed better through the course context. One of the accessible sources would be **Better CSS with Sass,** written by Cole Henley (see Figure 3).

The book covers all essential syntaxes through many examples sorted into the following content.

Chapter 1: Getting started with Sass

Chapter 2: The building blocks of Sass (Variables, Nesting, Extends, Selectors )

Chapter 3: Logical Sass (Mathes,Mixins,Functions ,Conditionals)

Chapter 4: Organising Sass(import and partials, File organization, Controlling output, Source maps)

Chapter 5:Level up your Sass (Dealing with errors, media query bubbling,built-functions, Sass maps)

**Figure 3: book used to review Scss syntaxes**

**Menu Overlay & Responsiveness (**[**Part 4**](https://www.youtube.com/watch?v=92J-3ajM0dI)**).**

**30.5 - 31.5.2022**

Video related to this part was reviewed a few times to perceive all details. During the task, we would get familiarized with media queries as well as responsive web design. Simultaneously, new syntaxes such as translatet3d and transition-delay were applied to add more new advanced styles to the menu overlay. On the other hand, to implement more complicated version control (Git) practices, a new branch has been created for each file added to the project. All changes and modifications related to a specific file would be committed and pushed into the local and remote repositories through their own branches throughout the example project.

**Page With CSS Grid (**[**Part 5**](https://www.youtube.com/watch?v=QOdnrMC9O7A)**).**

**1.6.2022**

Text

Description automatically generatedThis part of the example project would be helpful for iterating previous practices; in other words, a new about.html file to be added to the project directory for which we have to define styles. Except for the main.js file, There seem not to be any more javascript tasks to be handled throughout the example project. Similar to previous practices, During this part of the project, all implemented or modified styles and Html tags were simultaneously commited and then pushed into local and remote repositories. In order to exercise more sophisticated Git tasks, more branches were extracted from the main branch of the project and merged back into the main branch. (see figure 4)

**Figure 4: a typical slice of the Gitlab repository of the project**

**Work and Contact Pages (**[**Part 6**](https://www.youtube.com/watch?v=oxTG7TVr7PQ)**)**

**2.6.2022**