Lappeenrannan teknillinen yliopisto
Software Engineering (School of Business and Management)

Software Development Skills Front-End, Online course

<Zhenghan Li >, <001755419>

1 LEARNING DIARY, < Software Development Skills: Front-End 2025-26> MODULE

LEARNING DIARY

3.10.2025

Activity:

I read the general instructions of the Software Development Skills Front-End online course on Moodle. I made sure I understood the main requirements: I need to provide practice materials, a learning diary, my own front-end project, a README file explaining how to run the project, and a file containing a link to a demo video. I also checked the example project and compared it with my own TechFlow project idea.

Learning outcome:

I understood that the real goal of the course is not only to follow tutorials but to create my own project that demonstrates my skills. This helped me to think more critically about what parts of my work are original and how I can use the course materials as a starting point instead of copying them. I also became more confident that my TechFlow project can be accepted as a unique project that builds on the example.

5.10.2024

Activity:

I set up my development environment for this course. I chose VS Code as my code editor and installed useful extensions like Live Server and Prettier. I created a Git repository for the course work and pushed an initial version to GitHub. I also organized the folders into project/ for the final project, exercises/ for tutorial practice, learning-diary/ for this diary, and docs/ for additional documentation.

Learning outcome:

I learned how to prepare a clean and practical environment for front-end development. Using VS Code with the right extensions and Git version control already made my workflow smoother. I also realized how important it is to structure the repository clearly from the beginning, so that others (and my future self) can easily understand what is practice work and what is the actual project.

7.10.2024

Activity:

I followed a front-end tutorial series (for example, a project-based tutorial by Brad Traversy) to review HTML, CSS and basic JavaScript. In the exercises/ folder I reproduced some small demo pages, focusing on layout, navigation bars and simple interactive elements. While doing this, I wrote down notes about tags, CSS selectors, Flexbox and DOM manipulation.

Learning outcome:

By rebuilding the tutorial examples, I refreshed my memory about the essentials of front-end development. I learned again how semantic HTML improves structure, how Flexbox can be used to build responsive layouts quickly, and how JavaScript can be used to react to user actions. More importantly, I started to see these examples as a toolbox that I can reuse and adapt later in my own project.

8.10.2024

Activity:

I started building the structure of my own project, TechFlow, which is a responsive SaaS landing page. I created the basic sections: hero, features, pricing, testimonials and contact. I focused on writing clean HTML with meaningful class names and keeping the structure similar to what a real-world landing page would need.

Learning outcome:

I learned how to plan a front-end project beyond just copying code. Designing the sections for TechFlow made me think about user flow: what visitors should see first, what information they need, and how to guide them towards a call-to-action. I also saw how a good HTML structure makes later styling and JavaScript easier.

10.10.2024

Activity:

I worked on the main layout and responsive design for TechFlow using CSS Flexbox and Grid. I defined different breakpoints for mobile, tablet and desktop widths, and tested the layout by resizing the browser window. I adjusted font sizes, spacing and alignment to keep the design readable and consistent on different devices.

Learning outcome:

I deepened my understanding of responsive design. I learned how to use media queries to change layouts at certain screen widths and how to combine Flexbox and Grid depending on the section. I also understood how small details like padding, margins and font scaling can make a big difference in the visual quality of a page.

12.10.2024

Activity:

I implemented interactive elements in the TechFlow project. I added a simple image or testimonial carousel, a smooth scroll effect for navigation links and a parallax effect for some background sections. I wrote JavaScript code to handle user events, update the active slide in the carousel and add or remove classes from elements based on scroll position.

Learning outcome:

I learned how to connect JavaScript with my HTML and CSS to create a more dynamic user experience. Handling events and manipulating classes in the DOM helped me understand how front-end interactivity works in practice. I also learned to keep the code organized so that it remains understandable instead of putting everything into one big script.

14.10.2024

Activity:

I added a contact form to the project and implemented client-side form validation using JavaScript. The validation checks for required fields, email format and message length, and shows friendly error messages when something is wrong. I also styled the form and error states with CSS to fit the rest of the TechFlow design.

Learning outcome:

I learned how front-end validation improves user experience and data quality before any server-side processing. Writing the validation logic helped me think carefully about edge cases and how to communicate clearly with users when input is invalid. I also noticed how easy it is to introduce bugs if the DOM selectors or state variables are not named clearly.

17.10.2024

Activity:

I did a round of refactoring and cleanup on the HTML, CSS and JavaScript code. I removed duplicate styles, grouped related CSS rules together and renamed some confusing classes. In the JavaScript files, I split larger functions into smaller ones, added comments where needed and tried to follow a consistent style.

Learning outcome:

I experienced how important refactoring is for front-end projects. Even if everything seems to work, messy code makes it hard to maintain and extend the project. After refactoring, I felt more confident that I could add new features or fix bugs later. I learned that clean code is also part of professional software development skills, not just getting the correct output.

18.10.2024

Activity:

I tested the TechFlow landing page in different browsers and on different screen sizes. I opened the site in Chrome, Firefox and Edge, and also used the developer tools to simulate mobile devices. I fixed several visual issues, such as misaligned buttons and overlapping text at certain breakpoints. I also slightly optimized images and removed unused code.

Learning outcome:

I learned that cross-browser and cross-device testing is necessary even for a relatively small project. Some layout problems only appeared at specific screen widths or in certain browsers. This taught me to be more careful with assumptions and to always verify the design in multiple environments. I also understood the value of small performance improvements like optimizing assets and cleaning up unused code.

20.10.2024

Activity:

I wrote documentation for the project: I improved the main README.md file with a clear project description, instructions for running the project and an explanation of the repository structure. I also created a project summary document to describe the main features, technology choices and my learning experience during the course.

Learning outcome:

I learned how to communicate my work through documentation. Writing the README and project summary forced me to think from the perspective of another developer or

teacher who sees the project for the first time. I realized that good documentation is essential for collaboration and for showing my skills in a professional way, for example when applying for jobs.

23.10.2024

Activity:

I planned and recorded a demo video of the TechFlow project. In the video I showed the main sections of the landing page, the responsive behavior on different screen sizes, the carousel, the parallax effect and the contact form validation. I uploaded the video to an online platform and created a separate file in the repository containing the link to this demo video, as required by the course.

Learning outcome:

I learned how to present a software project in a clear and structured way. Preparing the demo script helped me to highlight the most important technical and design decisions instead of showing everything randomly. Recording the video also made me more aware of the overall user experience, because I had to see the project as a complete flow rather than isolated features.

26.10.2024

Activity:

I did a final review of all course materials: practice exercises, learning diary, project code, documentation and the demo video link file. I also checked the Moodle instructions one more time to make sure that my Git repository is public and contains everything that is required. After that, I prepared a short submission text with the repository link to upload to Moodle and marked the course as ready for grading in the “Course completion” section.

Learning outcome:

I learned how to finalize and package a self-study course project in a professional way. This included not only coding, but also organizing files, documenting my work, reflecting on my learning and following the submission instructions carefully. Overall, the process helped me to see the bigger picture of a software development project, from first setup to final delivery and reflection.