

Lappeenrannan teknillinen yliopisto
Tietotekniikka

Software Development Skills

Daniil Feoktistov, 002601245

LEARNING DIARY, Full-Stack 2025-26 MODULE

LEARNING DIARY

Introduction

06.02.2026

I reviewed the background information and understood the main goal of the course: to find my calling as a software developer and create a unique project that showcases my skills. I chose Module Full-Stack because I considered it one of the most important of the proposed projects. Moreover, I believe Module Full-Stack will be very useful for developing my skills. Since I already worked with VS Code and am familiar with it, I chose it for this module without hesitation. I am also already familiar with Git, so I immediately created a public repository.

NodeJS

06.02.2026

I'm already familiar with NodeJS from the Advanced Web Development course, and I'm familiar with most of what was covered in the video. I know that NodeJS is necessary for running JavaScript for the server and application logic. The only thing that was new to me was creating an HTTP server from scratch without Express. I hadn't encountered this before, or I'd simply forgotten. It's never too late to refresh your memory.

MongoDB

06.02.2026

I'm also familiar with MongoDB from the Advanced Web Development course, and I know how and why to use it. MongoDB is extremely useful for web applications and can handle huge amounts of data. It can add, update, delete, and create documents and dates. Again, I didn't learn anything new.

ExpressJS

07.02.2026

Since I already knew how NodeJS worked, I understood why Express was useful. NodeJS can handle server-side logic, but Express speeds up development by providing tools like routing and middleware. While some of this was already familiar, it was a good reminder of why frameworks like Express exist.

React

07.02.2026

Again, I'm already familiar with JavaScript and front-end development, so most of the information about React seemed clear to me. I know that React is a JavaScript library used to build user interfaces. Because of this, developers often combine React with other tools for things like routing or state management. While much of it was familiar, it was still useful to review.

MERN-Stack

08.02.2026

After watching four videos on how to use the MERN stack to build an app from scratch, I'm even more knowledgeable and ready to build my own project. The only thing I didn't know is that this stack is called MERN, which stands for MERN = MongoDB + Express + React + NodeJS. I know MERN is primarily used for building web applications using only JavaScript, but I'm going to try using TypeScript to develop my skills in that area. Is that allowed?

PROJECT

PART 1

08.02.2026

Today I started building the backend for my project. It took a bit of work since I haven't worked with the backend for a long time. At this point, I've implemented all the basic backend requirements: Node.js + Express backend, MongoDB, user registration and authorization (JWT), and authentication-protected routes. Additionally, users can now create, read, update, and delete documents. Each user has their own disk, and their data is stored in a database. The backend has a clear structure, so checking files step-by-step should not cause any problems.

PART 2

09.02.2026

Today I started working on the frontend of my project. After an exhausting amount of work, I finally implemented the core features of both the backend and frontend. AuthContext for login/register, Drive page, Editor page with locking, and PublicView page for access without login.

MongoDB → mongodb://127.0.0.1:27017/testdb

Backend → localhost:5000

React → port 3000

Express + MongoDB + React used. Moreover, API tested properly and CORS handled correctly and perfectly.

At this stage of the project, there are no critical bugs or errors, everything works perfectly.

PART 3

10.02.2026

Today, I believe I'm finished with my project. I made the final adjustments and changes to the core backend and frontend files, plus added some new ones.

The end result is this: User is able to sort the documents in drive based on the name, creation/last edited timestamp; Documents are moved first into recycle bin and user can then clean the trashes when he/she sees fit; User can upload documents (it is enough to allow only images) to his/her drive; Translation of the whole UI in two or more languages; Document can be download as PDF, and some more.

I had to put in a lot of effort for this project, but the best part was that I enjoyed the process and learned and developed my skills.