

Lappeenrannan teknillinen yliopisto  
Tietotekniikka – Software Engineering

Software Development Skills

**Henri Hujanen**  
**613717**

**LEARNING DIARY, Software Development Skills: Full-Stack 2022-23**

14.6.2023

I started the course. I began watching the Node.js crash course and verified that I had Node and npm already installed. I learned some basic information about Node and how it works and started coding along with the examples in the video, learning about some useful modules related to path and file management.

5.7.2023

I continued the Node.js video. I got some understanding of how different modules like url, events, html and os work. Following the guide, I also created an http server and a simple example website, that was to be deployed to Heroku. I finished doing that except the few last steps which would have required verifying my Heroku account with payment information, but I understood the process regardless.

14.7.2023

I started the MongoDB tutorial. I was not familiar with it, and as I have only used relational databases before, there was much to learn. There were some problems with getting the mongo shell to work, as the version used in the tutorial had been deprecated. I managed with the built-in shell provided with the MongoDB Compass GUI, but later had to just download the shell on its own when connecting to the cloud.

After getting everything to work smoothly, I started following the tutorial. I learned how to add and delete new databases and navigating them, how to add new collections, and how to search the databases using different criteria. The examples were quite straightforward, but I learned the basics, and I learned some differences between relational and document-oriented databases. Finally, I connected the database to the cloud using MongoDB Atlas, and learned how to navigate and edit databases there.

18.7.2023

I started the ExpressJS crash course. I was familiar with Node and npm from the earlier tutorial, and the course seemed to build on the earlier NodeJS tutorial. I had to install Postman as I had not used it before, but it seemed useful for testing APIs quickly.

I started by creating a simple server and website and downloaded nodemon to speed up creating changes. Following the tutorial, I created a simple API with names and emails in JSON format. For testing it, Postman proved useful as I could save GET requests and see changes quickly. I also learned about third-party Express middleware and how to write my own in the form of a logger. Finally, I learned how to create views for the application using handlebars by using their templates.

20.7.2023

I started the Angular tutorial. As I had worked with Node and npm in the previous modules, I thought I could hop right into the course, but some problems appeared. My node.js version was out of date for Angular, and after updating it to the current version, npm stopped working at all. After uninstalling and reinstalling node and deleting the old folder manually I got it to function. Finally, I got to work on the actual course, and I downloaded the Angular CLI and the example app and started with the simple hello world portion.

In the next parts I learned how to create components, and how to use Angular services to be used by multiple components. After adding multiple components, I enabled routing to support navigation between views, and added the details view for inspecting different houses. To allow for customers to reserve houses, I integrated Angular forms for sending applications, and a filtering function. Finally, I added a mock JSON server so that the housing information doesn't have to be hard coded into the components and can use data via HTTP requests. Overall, I feel like I learned the basics of Angular well and am ready to begin the MEAN-stack project.

26.7.2023

I started working on the MEAN Stack project. First, I worked on the backend adding routes and creating a user model. Some problems arose when using mongoose. In the tutorial, callbacks were

used for registering, authenticating, and accessing user profiles. However, mongoose threw an error stating that it no longer supported callbacks, so I had to refactor the code to use promises with the `async/await` commands. I verified the new code using Postman and got it to work accordingly.

27.7.2023

Really struggled to get the project Angular frontend to work. As I had used Node version 16 for the beginning of the project, downloading the old version of angular-cli proved hard. I changed my Node to version 12, but doing so made the backend stop working. I refactored the code again to remove my previous updates for Node 16 usage to match the video tutorial. I also had installed a new version of MongoDB for the earlier parts of the course, and I had to downgrade that as well. I changed the rest of the dependencies in the `package.js` file to use old versions that I estimated the tutorial used in 2017.

The final obstacle was a strange file in the angular-src files, that had something to do with the version of TypeScript being used. I found a solution from Stack Overflow to comment out a line of code in this file. Had I found this solution earlier, perhaps I wouldn't have had to downgrade absolutely everything that I used earlier on the course, but I hope this proves worth it for the rest of the project, and at least the frontend seems to be working for now.

28.7.2023

I continued building the project. I added the angular components and routes, and some simple example html for each component. There was some confusion when working with the components, as my code editor kept warning me about problems with the code. I tried my best to remove the problems, but turned out that the application compiled correctly and ran without issues. I don't know what causes the issue, but for now it's just cosmetic, as files and lines of code keep flashing red. I also imported a bootstrap style for the frontend.

Next, I added the register functionality. I improved the html, and imported and added some flash messages to warn about problems during registering. Then I added an authentication service and made registering a new account work together with the previously built backend `addUser` function. After this, I worked on the login component. When login was successful, I needed to hide the register

and login functions. I added a logged in condition to the navigation bar html, which could also be used to hide the dashboard and log out buttons. However, a logged-out user could still access the dashboard by manually typing the address. For this, I added the auth guard, and made the dashboard and profile pages able to activate them.

After this, the project was ready for deployment. The tutorial wanted to make use of Heroku, but as it requires credit card information for even the basic functions these days, I decided to not deploy it to the internet. I used the 'ng build' command to build the frontend, and the project can now be run by simply using the 'npm start' command and going to <http://localhost:4200/>. I also did not add a remote MongoDB database, and the app uses a local database instead.

Now, all that's left is to publish the coursework on Github. Overall, I felt that the course was successful. I learned a lot about Node, MongoDB, Express and Angular, as I hadn't used them much before. As a critique, I would have liked for the course to use newer versions of the technologies, as many things wouldn't work from one module to another, and during the final project having to downgrade almost everything caused frustration.