Lappeenrannan teknillinen yliopisto

School of Business and Management

Software Development Skills

**Ilona Engblom, 000255952**

LEARNING DIARY, FULL-STACK 2021 MODULE

**LEARNING DIARY**

**Full-stack 2020-2021**

15.1.2021  
I gained access to the course and went through the hassle of once again activating new user ID for another university. Read through the overall information of the course and checked my setup, which already included Git as well as VS Code, so there was no need to install anything. I also superficially checked the course materials, which included external videos on all the topics.

I chose this course because I want to focus on full-stack development, not being able to decide between backend and frontend but also because it is an asset on the job market to know both. I have some previous experience via courses on both frontend and backend development, but those have not really been enough to form a whole picture and deeper understanding on how things work.

16.1.2021  
Started with the Node.js crash course video by following the examples and writing the code along with the instructor. The Node.js course I’d taken before focused on using Express and did not explain anything on why things were done as they were, so this opened my eyes quite a bit. I am also starting to get a grip of the deeper essence of Javascript, coming from a Pythonic background.

Spent quite a while (selecting a new colour theme for the VS Code editor and) trying to make the .gitignore file such that it would actually ignore node\_modules. Not much success even though I made the file before making the folder a git repository. Getting quite familiar with *git rm -r --cached node\_modules* by now.

Quit halfway through the video, eagerly waiting to see what we’ll actually be able to do besides the demo code snippets, which though have been useful so far.

18.1.2021

Went through the rest of the Node.js video (and changed the VS Code colour theme back to Pink Cat Boo). The small ‘server’ done in the end of it was impressive in terms of how quickly the instructor did it and how one could do it ‘from scratch’, i.e. not using Express. What this Node.js crash course taught me was a lot on how I could have done things better in my previous projects, based on the learnings from another Node.js course, for example routing the paths and handling the content types. I also realized I hadn’t really known what the *(req, res)* stuff even meant, instead I’d just typed them in the code without thinking.

This course is already making me hate the way the previous courses have been based on using ready-made libraries and modules for backend and frontend even for things that would have been simple to make yourself.

I also watched the MongoDB video and typed along following the instructor (not being able to keep up with his speed even though you’ll have it hard to find a typist as quick as I). I had never used MongoDB before, so I needed to install it, and being on a Windows machine against my will, had some trouble with the Windows Powershell to get the Mongo shell to operate. Let’s just say I wouldn’t use this shell for building a proper database… MongoDB itself feels nice and simple compared to SQL databases, of which I’ve mostly used PostgreSQL, and the cloud solution Atlas that was presented in the end of the video is something I plan to deploy in my own projects in the future.

22.1.2021

Took the Express.JS part, which again made me realize how good it was to finally get someone to actually explain these things to me instead of just following ready-made code and copying and altering the parts I need, without really knowing why and what they do – even if it’s just some (probably not-so) random American dude on Youtube.

I wish the videos would soon start making use of real databases for the examples instead of hard-coded arrows, because the handling of data is so different. What I personally also hope to gain, especially for own projects, is authentication. I’ve tried Auth0, but didn’t really get much grasp of it. Now this Traversy Inc. instructor talked about Passport and actually Youtube is recommending me their video on just using Passport in a Node.js project, so I think I’ll watch that after this course is completed, if it doesn’t come up in the following parts.

I liked that the video took up template engines as well, besides ‘only’ building an API, because the previous course only mentioned them in a sidenote and used Pug, the oversimplified syntax of which just felt utterly stupid to me. Glad the instructor here agreed and introduced us to Handlebars. I also got to use Bootstrap for the first time in my life – this is how you learn.

Fought with Git again to get my part 1 (Node.js) files actually visible from their submodule. And .gitignore still doesn’t work with node\_modules, so have to manually remove them from every commit before committing. I’m actually enrolled on an study-at-your-own-pace Git and version control online course, but I’m beginning to be less and less convinced on whether it can actually offer me something new (that I haven’t had to Google and read about in Stack Overflow already).

23.1.2021

Went through the Angular tutorial for building the Tour of Heroes in the 4th part of the course, starting yesterday after updating the learning diary. Having used React in my previous front-end applications, the Angular framework was completely new to me and still, after the tutorial, feels strange and the syntax extremely complicated. This may, again, be because the tutorial didn’t explain everything that thoroughly – or I myself didn’t read through the explanations so that I would have actually comprehended everything. Will have to get back to them once more to hopefully get some more insights as well as read the rest of the guide.

Creating components from the command line and automatically getting all the classes, templates and .css files for each and every one of them also feels a little unnecessary and I fear it makes the app heavier than needed. Not that excessive use of different React modules / libraries is a more efficient solution though.

What I didn’t like about the tutorial was the use of a simulated data server but not explaining to the detail how and in which parts of the code it differs from using a real one, if at all.

I managed to solve my .gitignore problem by adding one containing *node\_modules* in every subfolder that has that folder.

23.-25.2.2021

Struggled my way through the MEAN stack Youtube course, which introduced us to authentication and clarified the whole mess in my head called the Angular framework a lot.

While I still think the different Angular components with all their separate files are a bit confusingly too much, now I start to understand how they work, what the servers are for and maybe, just maybe am able to build an app of my own with Angular.

Even more positive was to finally get a grasp on the whole authentication process from both server and client side as well as how the tokens can be handled (and hashed). The threshold of implementing authentication/authorization in my own projects has been rather high thus far, because all the material I’ve found has looked complicated enough to require a lot of studying – the time to which I’m a bit short of. This tutorial helped with it a lot.

Too bad the videos were outdated in crucial parts regarding the Angular framework and modules, and trying to get everything to work in their older versions really asked for some nerves as well as quite a few rounds of deleting *node\_modules* and *package-lock.json*, fiddling with the dependencies in *package.json* and rerunning *npm install*. Have to admit it also taught me more about how the dependencies etc work. Some oddities got answers in the videos’ comments, but some problems caused by more newer versions of different modules required quite some Googling and trying.

I didn’t deploy the app in Heroku yet and I am unsure whether I want to do it at all, since my Heroku is filling up with these nonsense apps and I already know the deployment process (even with React apps that don’t build without external buildpacks…). I may try some other server though, if the instructor in the final video I’ve yet to watch uses something that could be of use for myself in the future as well.

For the course project, I think I’m going to go for the Observation diary app + API + database I’ve earlier made a REST API and a PostgreSQL database for but got stuck at the authentication phase, but instead of the old API and database make a new one and use MongoDB Atlas. And after all I think I **am** going to use Angular (which, as far as I understood, isn’t required in this course) but stick to the newest updates and versions of everything I use to minimize the fighting with dependencies.