Lappeenrannan teknillinen yliopisto
School of Business and Management

Sofware Development Skills

Eric Liski, 000452179

LEARNING DIARY, FULL-STACK MODULE

LEARNING DIARY

1.12.2023

Today I'm starting this full-stack development course. I chose the full-stack development module because I have heard a lot about full-stack development online and became interested in it and wanted to learn more about it. I think this course will be perfect for accomplishing this goal. I also like self-study courses a lot because I feel like they support my way of learning since I like to go at my own pace often. Now to get started with my code editor I will be using VS Code because that's what I have always been using and I have no complaints about it. I also have already set up a lot of extensions on it so I don't have to worry about that too much either though if there are introduced new extensions that could be useful, I will be sure to add them to the collection. Starting with the first module/task "NodeJS". I'm already a little familiar with node.js but am not an expert by any means. I already have node.js installed so I don't have to worry about that. I also am familiar with git, so it was easy for me to set up my git repository for this course as well as make my first commit. I'm currently watching the first tutorial/video I chose the updated one because I thought it would make more sense about node.js and already have learned a few new things and node.js built-in modules for example about the path and fs and how they can be used to for example create directories and files as well write and read them which I can see being useful in many cases like creating a web page as shown in the tutorial with the help of HTTP module or for example for creating crash reports for when application crashes could also be useful in some cases. I also got a chance to refresh my memory on the things I already knew about node.js and JavaScript beforehand. The first tutorial/video was fairly straightforward I had some problems where it would not let the CSS for the pages load but after some digging, I realized that I didn't change the 'Content-Type' to 'contentType' variable and it was instead text/html which leads to it not loading the style.css as CSS thus not styling the pages as expected at first. I also unexpectedly learned something new about Google dev tools like network monitoring which allows me to check response Headers and other stuff which definitely will come in useful later on. Another cool new thing I learned is Heroku which allows me to easily upload my website to the internet as well as use the terminal to use git since I have used the VS Code interface for git commits and setting up repositories before.

3.12.2023

Today I will be continuing with the next task/part about Mongodb. Thankfully I already have some experience with MongoDB as well, so I already have done the installation which was fairly straightforward for Windows. After finishing the tutorial/video about MongoDB I can safely say I have learned something new. The tutorial was mostly about the use of MongoDB shell which I in prior have not used all that much. All of my experience with MongoDB was through node.js/express.js as well as MongoDB compass instead of using the shell. Now I have learned how to use the shell to some reasonable extent which will come in useful later on. I also learned about MongoDB Atlas which allows you to upgrade your database from a local database to a cloud database which is useful in some scenarios.

4.12.2023

Today I will be continuing the full-stack course module with the next task/part express JS. I have used express JS a little bit beforehand when creating a back end for a website in another course and I know that it's a framework for node.js. Hopefully, after this lesson, I will learn more about express JS that I might have missed before. Also, after trying to create a web application in the first lesson using only node.js, I can without a doubt say that express JS makes it much easier comparatively since you don't have to worry too much about the content types and such which results in a much smaller chunk of code and thus takes less time and stress. I will begin watching the video/tutorial now and note anything new I learn in the diary. A few things I have learned so far are how to properly use 'module.export' to make your index file less cluttered which is something I have had problems with in the past I would just put all the GET, POST, PUT, and DELETE requests all in index.js and it would be hard to navigate. The second new thing I'm currently learning is 'express-handlebars' which is the first time I have heard about this view engine. I have always used the Pug view engine and I know about the Ejs view engine but express handlebars is a completely new view engine to me so this will be interesting to learn a new view engine. I'm almost done with the lesson but there is a problem I'm currently having with 'express-handlebars'. For some reason doesn't allow me to set 'app.engine' to 'handlebars' and keeps giving me an error (TypeError: exphbs is not a function) which I'm trying to figure out as of right now hopefully I will come back with good news soon. After a little bit of trial and error and researching on Google I found a Stack Overflow thread (javascript - TypeError: exphbs is

not a function - Stack Overflow) with the same problem and apparently, the solution was to use “exphbs.engine({ defaultLayout: “main” }) instead of exphbs({ defaultLayout: “main” }) like shown in the Traversy Media video. Not sure if I made a mistake somewhere or if the video is a bit outdated since it’s 4 years old already but I believe it works now after that small change so I will continue following the video further. The fix I found seems to be working just fine thankfully. Another thing introduced in this lesson is Bootstrap which I know what it is but haven’t got much experience in using it. I had a small problem where when I tried to make the web application display **Members** it displayed it like this instead <h1>Members</h1> after rewatching a bit back I noticed that I used double curly brackets {{body}} instead of triple curly brackets like this {{{body}}} so it didn’t work as expected at first, but I got that fixed now as well. I’m finally done with the video which was quite long I must say almost as long as the first video. But I definitely learned something new which is quite nice since that’s why I'm studying. Overall, the key things I learned were a bit of the usage of Bootstrap with an express handlebars view engine to create a nice and clean web application. Express handlebars was quite confusing at first but I’m starting to slowly get the hang of it.

5.12.2023

Today I will be continuing the full-stack module and next up on the task list is the Angular task. Unlike the previous tasks, I haven’t heard or worked with Angular before so this will be completely new to me. Meaning I will be learning a lot of new stuff today. I think I will be now starting the task. After clicking the “Angular TOS” hyperlink, it looks like this time around there is no video tutorial instead there seems to be a tutorial in the form of documentation. I clicked on the “First App Tutorial – Angular Homes” hyperlink on the page to continue. After setting up for Angular i.e., making sure that I have the correct node version installed (node --version) and installing the Angular language service extension (Angular Language Service - Visual Studio Marketplace) for Visual Studio Code I think I’m ready to continue. After some googling, I found a Traversy Media video on Angular which is about 2 years old, and I feel like I would learn better from a video so I’m going to send an email to the teacher for now and continue this Angular task later after I get an answer.

11.12.2023

Today I decided I will be starting to learn about MEAN-Stack since I haven't heard back from the email yet, I will do the Angular task after I receive some clarification on the Angular task from the email I sent. Not quite familiar with what that is yet, but I will be learning that today hopefully. I will be starting with the first part of the MEAN-stack series from Traversy Media and come back to the learning diary after that. After watching the first part of the MEAN-stack series I learned that MEAN-Stack is a web development stack that consists mostly of previous modules that we have learned throughout this course i.e., Node.js, MongoDB, Express.js, and Angular. It seems to enable developers to build end-to-end web applications that use JavaScript for both the server and the client side. I also got a look at how a quite basic MEAN-Stack authentication app for example works and how it is constructed. I will be now moving to the second part of the MEAN-Stack series which seems to go through how to set up express.js in MEAN-Stack, sounds exciting. I'm just getting started with the second part, but something interesting I observed already is that you can write down the dependencies in the packcage.json file like "name": "*" and after "npm install" it will install all the wanted newest dependencies which can be quite useful and a time saver instead of writing "npm install (name)" for each dependency you want to install like I'm used to doing. Anyway, moving on now. After finishing the second part I have set a good start for a web application project with Express.js and MongoDB. A new module was introduced named "cors", which I had not previously seen nor used. We enabled CORS for all routes, which seems to make web applications more secure by controlling cross-origin requests to prevent common security vulnerabilities like Cross-Site Request Forgery (CSRF) and Cross-Site Scripting (XSS). We also used a separate config folder with a database file to store the database path, which I usually haven't done thus far.

12.12.2023

I will be now going to the third part of the MEAN-Stack series. Ran into a small problem at the end of the third part, which seems to be that Mongoose functions save() and findOne() no longer accept callbacks ("error: throw new MongooseError('Model.prototype.save() no longer accepts a callback');"). After some research, I found a Stack Overflow thread (https://stackoverflow.com/questions/75586474/mongoose-stopped-accepting-callbacks-for-some-of-its-functions) talking about this same error and issue with Mongoose

With this information, I changed the users.js file "/register" POST request function so that it is asynchronous and awaits the addUser() function. I also changed the user.js file so that it now handles promises. After a little bit of tweaking around and googling I got it to work like intended in the series. Now I can continue to the fourth part of the series. I have encountered my first problem in this part. The problem was that I needed to initialize and install express-session dependencies for login sessions to work in express which I don't remember Traversy Media doing in his video. I have encountered yet another issue in this part where like in the previous one about the save() function the same problem remains with the findOne() function that it doesn't accept callbacks in the newer version of Mongoose. Since I already know what the issue is I can go straight to fixing it. After some tweaking and changing I got it to work like intended. I changed the getUserByUsername function in user.js a bit so now it is asynchronous and uses promises instead of callbacks. I also changed the "/authenticate" POST request function to asynchronous so that it awaits the getUserByUsername function and stores the user in a variable. I also had to change the comparePassword function a bit and make it, so we await the response as well when calling it. I'm now at the end of part four of the series and had a little problem with the passport.js authorization strategy file not authorizing correctly but after a little bit of experimentation I got it to work like all others by making it asynchronous and using promises. Onto part five now about what it looks like the Angular module. After starting the video, I instantly encountered a problem that I have now been stuck on for quite a long time, and I finally got it to work after very much of googling and reading Stack Overflow threads. The problem was that it wouldn't let me "npm install -g angular-cli" and it kept giving me tons of errors. I read on the Moodle page that you need to use node version 12, so I was trying to change the node version using nvm (nvm use 12) and it seemed to change by printing out "now using node v12" but for some reason, it didn't change the version from node 20.0.0. After I realized that it didn't change the version I found one Stack Overflow thread (nvm use does not switch node versions - Stack Overflow) with the same issue and I realized that the problem was the NodeJS folder in Program Files and I needed to delete it, which it wouldn't let me at first. Eventually, I downloaded the node.js installer and uninstalled it through that. Now it let me change the node version to which I ever wanted. Then I also had a problem where it kept saying that 'ng' already exists, but it would give me an error when I tried to use it to do anything, so I deleted/uninstalled all angular-cli and @angular/cli folders/files I found from

“node_modules” and tried again and it finally worked. It’s getting late so I will be continuing the fifth part tomorrow.

13.12.2023

I will be now continuing from the fifth part where I left off. There is another huge issue I just encountered. When I try to run “ng serve” it gives me error (Invalid ‘reference’ directive syntax Webpack: Failed to compile) and I’m currently trying to solve this issue. After a lot of testing and searching I found a helpful YouTube comment under the fifth part of the MEAN-Stack series which said to go to the angular package.json file and change the typescript version from ~2.0.3 to 3.5.1 and then “npm install” and “npm audit fix” which worked for me and now I managed to finally run “ng serve” successfully and everything compiled at least for now. Next up I installed the navbar to the component folder using “ng g component navbar” but I had a small issue where it said, “No app module found.” and wouldn’t add it as a dependency and import it to the app.module.ts file. I managed to easily fix that by manually just adding the import statement and dependency to the file and it worked. I have the same problem with installing the login and register components so I will probably have to manually just add the dependencies and import them to the app.module.ts file until I find a fix. I had an issue with the CSS of the navbar not working but I realized after reading comments on the video that it was because I was trying to use the new bootstrap (https://bootswatch.com/5/sandstone/bootstrap.min.css) instead of the older version of the sandstone bootstrap CSS (https://bootswatch.com/3/sandstone/bootstrap.min.css). I’m now done with the fifth part and will be moving on to the sixth part of the series. I just finished the sixth part, and it was fairly straightforward, and I didn’t thankfully encounter any big issues for once. I learned more about Angular and how to use it to create different pages of the web application. I also learned how to validate the email and check that input fields aren’t empty as well as new and visually pleasing ways to display error messages on the page using flash messages. Overall, this was an enjoyable part, now I will be going on to the seventh part. Another short and rather simple part with no issues personally, which taught me how to create user registration functions from the front end so that they communicate with the back-end APIs previously created and can register users to the Mongoose database. Everything is really coming together now which is quite satisfying to see. I will be continuing to the eighth part tomorrow.

14.12.2023

Today I will be watching parts eight, nine, and ten and hopefully finish this MEAN-Stack task. After finishing part eight I learned how to setup login so that it uses the POST API in the express back end which sends back the authentication token which will then be set in local storage in the browser so that we can authenticate user based on the token and they don't have to constantly login again to access different pages. I also learned how to clear the browsers local storage so that we can essentially log out. I will be now moving on to part nine. In part nine I learned about how we can set up our web application in a way that only logged-in users can see and access certain routes such as profile and dashboard. Overall, everything was clear and straightforward though I had to use the suggested fix in Moodle (tokenNotExpired("id_token")) since I couldn't get it to work without it. I also learned about how to use 'ng build' to compile our Angular application and package it into a more distributable format that can be deployed into a web server. I will now be watching and doing the last part. Part ten seems to be about web application deployment to Heroku which is one of many cloud platforms for deploying applications. After finishing the video, I learned how to deploy web applications to the internet in general but in this case, using the Heroku cloud platform and the mLab MongoDB which is a cloud-based database service that provides MongoDB database hosting services that we use to store registered users for the web application. After finishing the whole MEAN-Stack series I've gained some valuable insights into web application development, particularly with the MEAN-Stack (MongoDB, Express.js, Angular, Node.js). Exploring the ins and outs of each component and understanding their collaborative work in building dynamic and scalable web applications has been very interesting and enlightening. Additionally, deploying these applications to the internet using cloud platforms like Heroku or services like mLab MongoDB added practicality to the learning experience. This MEAN-Stack task has broadened my understanding of end-to-end web development and has been a great learning experience.

23.12.2023

Got an answer for the email I sent about Angular task Traversy Media video, and it seemed to work just fine so I will be watching that and coding alongside it since I actually quite enjoyed watching them on the previous parts and thought they were well made and helpful. Since I already completed the MEAN-Stack module I actually got bit familiar with the

Angular so I don't think I will have too much trouble following this module. When I was doing the MEAN-Stack module since the crash course for that is already 6 years old it used quite an old version of Angular so now, I have to install the newer version of Angular again since I had to delete it in order to make the old version work properly. I had to install and use node version 20.10.0 since that's the current latest LTS version, which also supports the newest Angular version. I also had to uninstall the old "angular-cli" and ng since they were interfering with the download. After using "ng new angular-crash" I realized that it for some reason doesn't create the "app.module.ts" file, so I did a little bit of research and found an official Angular GitHub issue thread (https://github.com/angular/angular/issues/52751) where the issue is discussed and found that "ng new angular-crash –no-standalone" fixes this issue since they have changed it so it by default is now standalone application meaning it doesn't generate the "app.module.ts" file. At 25:40 in the video, I had a problem with the variables; text and color not having an "initializer", but it was easy to fix by adding them to the constructor. I have learned so far how to better utilize and generate the components of the Angular framework ('ng generate component') and how to combine them to make a clean and nice-looking interface. I also learned about the Angular Font Awesome component which allows you to use free icons of many kinds which are useful to make a clean-looking interface. Font Awesome was supported for the newest version of Angular (17. x) and I had a choice between Font Awesome versions 5 and 6, I went with version 5 since that's what is used in the Traversy Media video. I also learned about observables and how they work better. At first, I didn't understand them too well but it's starting to make more sense. They handle asynchronous operations kind of like promises in JS ".subscribe()" is works kind of like ".then()". Something new I learned also is "Json-server", which is a NodeJS-based tool that allows creating a simple API for testing purposes during development. At 1:05:12 the "onDelete(task)" wasn't working for me so I had to specify the type of task "task: Task". Something I also just realized is that you can initialize a variable (set default value) without using the constructor function. I also learned more about two-way binding and how useful it can be. I finally finished the long 2-hour video on the Angular crash course. Another thing right at the end I also learned is how to use RouterModule and Routes to set routes in your web application. Definitely learned a lot of new stuff and reinforced old knowledge with this crash course and can say without a doubt it was helpful in my learning journey. Now I will

be officially done with the task modules and will be moving on to my project next for this course.