Lappeenrannan teknillinen yliopisto
School of Engineering Science

Sofware Development Skills

Joonatan Kallio, 0607202

LEARNING DIARY, FULLSTACK MODULE

LEARNING DIARY

28.3.2024

I enrolled to this course yesterday and started today by going through all the general course information on Moodle. After that I watched the first lecture video about Node.js from YouTube.

I knew quite a lot about Node.js before enrolling to this course from other courses at LUT University and also from my own projects. I still learned quite a bit new information about Node from the video. For example, I knew about the syntax and how to setup Node.js server and also I know how to use node package manager and have used it a lot before. Something new I learned is how to use Node.js without express since I have always used the two together.

10.4.2024

I watched the first nodejs video to the end and learned about http, os, event and url packages. I created a logger that used event package to log the events and after that I learned how to create simple http server without express. I learned also how to create more sophisticated server with only http, path and fs packages and then I tried to deploy that to Heroku, but it did not have any free tiers I could use for free so I did not deploy it to Heroku but did run it locally.

24.4.2024

I started to watch the MongoDB crash course. I already knew a lot about MongoDB from projects I've done before. I knew how to insert data to the database, how to update the data and how to remove data. I did still learn how to use the $set syntax to only update one variable from the post. The $inc and $rename operators were also something I learned to use. After that I deployed the test cluster to MongoDB Atlas which I had also done before in other LUT courses.

24.4.2024

I started to watch the Express JS crash course. Before watching the course video, I knew the basics and some more advanced features of Express because I have used it in multiple fullstack projects. Most notably in the Advanced Web course from LUT where I learned to use it and the basics. From the start of the video, I also knew HTTP status codes, JSON, different methods and I knew something about JS arrow functions.

25.4.2024

I finished watching the video and did the same application as in the video. I knew all the different ways to handle requests in Express such as get and post. However, I learned how to use the view engine called Handlebars with Express. I had used Pug before, so the concept of view engine wasn't new, but I had not heard from Handlebars before but now I know the basics of it.

26.4.2024

Today I started watching the Angular crash course. Before watching the video, I knew that it is a frontend framework which is similar to React in a way that it uses components and Angular is also very popular. Angular was the reason that I wanted to go through with this course, since I had not used it before but I wanted to learn it.

4.5.2024

I continued watching the video and started doing the programming parts along the video. I had some problems with getting the versions working because all of the package versions used in the video were deprecated and I had to change Node version to get the website working with all the packages. First thing I noticed was that the file system differs a bit from React. For example generating the components is simpler with the cli command. Also I learned how to use the event emitters to create a custom button component that can be reused within the app, because the function isn't defined in the component itself but it is defined when it is used in other areas of the app. I also created a way to delete the tasks and update the reminder state with the emitters. I found it kind of hard to learn the emitter concept because I have only used React and it does things very differently. Finally I got the website working and now I started watching the project videos.

The setup was pretty simple for the Node server with Express and also Mongoose since I already know how to use these packages. I had no problems in this phase and got the backend "skeleton" working.

I did the authentication with jwt and used bcrypt to hash the password. I had a problem with the brypt.compare() function since I couldn't get it to match even if the password was true but after a long time I found out that I had put the username in to the password variable. After that I had no issues and everything worked.

5.5.2024

I watched the API authentication video. I knew a bit about authentication before watching the video for example how to do it with jwt and how it works. I still had some issues with the bearer token since in the video it was sent to the browser with "JWT" + token, but I got it working only with "Bearer " + token, in the sent json. I also had to put the jwt payload to this variable for it to work: jwt_payload.user._id. After that I got the authentication working as it should.

I watched the Angular component and route part of the video, since I had just yesterday watched the Angular crash course it was pretty easy to follow this tutorial and I got it working without problems. From this video I learned more about bootstrap since I had forgotten a lot of it. I had problems with getting the layout right but after checking the bootstrap documentation I got it working. I had to apply a padding-top to the body in styles.css to prevent any content from hiding behind the navbar and I learned how to use the navbar with bootstrap.

I followed along the video where I created a register functionality to the page. I learned more about the services and how to use them in an Angular application to help with the different login and register functionality. I did not have any major issues with this part of the tutorials. I also learned about the flash messages package and how to provide information to the user with these messages.

From the Login and logout video I refreshed my memory about storing the tokens in the browsers localstorage. The login was similar to the register functions so I could use almost the same flash messages in the login functionality. I had no major problems with this part of the videos.

I finished the project videos and had many problems with the jwt angular package, since it doesn't support my angular version from the earlier tutorial. I had to not use the package and instead made a check that checks if the token is null or not to return to the navbar.html file. I think this is acceptable way to go around this problem since the authentication doesn't depend on this functionality only how the page looks like to the user if he is logged in/not logged in. Otherwise I got the authentication working.

I learned a lot from Angular from this project video tutorial. For example how to create routes with AuthGuard and also how to conditionally render the different components if needed. Now I will start my own project by first copying the project that I made from the videos and expanding it to create my own application with the login/register functionality.

Since Heroku is not free anymore, I didn't deploy the project there but ran it locally using the static public folder.

5.7.2024

I decided to make the project on this course to be a todo application where the user can create todos and this way manage their time better.

I was very busy during the two last months, so I got the time to continue this course just now. Today I implemented a way to add todos for the project, remove them and change the todo status. I also added support for viewing the amount of todos in profile. The profile todos are shown in two different categories, one which shows the amount of done todos and another one which shows the amount of not done todos.

In the project there are also multiple backend api endpoints which the frontend can access. The endpoints that modify the data in any way are authenticated endpoints meaning that the user must be logged in to modify the data. The todos must also be owned by the user to be able to modify them, this is to prevent any kind of malicious api calls.

I also installed SASS for the project because I think it made the styling a bit easier. For the styling I used bootstrap like in the tutorial videos together with SASS. This way I could use the styled buttons and such but for example style divs with flexbox easier since I find it much easier to use normal css styling compared to the bootstrap classes.

6.7.2024

Today I finished the project, I added a new service for my TODO API calls which handles now all of the todo related calls. I also recorded the project video from which the application can be seen. I also added ReadMe to the project files. Overall, the course gave a good insight into angular 2 which was the framework I had not used before. I think that the tutorial videos were good, but there were problems because they are so old and Angular 2 is also already deprecated and not used in new projects anymore. Still it is good to also learn how to manage legacy code.