Lappeenrannan teknillinen yliopisto

School of Business and Management

Sofware Development Skills

Tuomas Kiviranta, 000244125

LEARNING DIARY, Back-End

**LEARNING DIARY**

**Beginning**

9.2.2021  
  
I introduced myself to REST API, what it generally is and how it functions. After I knew the concept of it, I moved on to NodeJS. I got to know about node modules, initializing a Node project, and running a local http server.

11.2.2021

Next chapter was MongoDB. I installed it along with Compass to make it simpler to use. I created a new database for my project. Then, I added few collections that would be needed. I wrote some json files to import data to the collections.

I installed more modules which I thought were going to be used. These included Express, http, mongoose, and few others.

12.2.2021

For mongoose, I made a model for each mongodb collection. I’m planning to use database aggregation with node, rather than making the queries myself.

**Advancing**

17.2.2021

I added database query files that pages would use to return information. Authentication would be used for users to create reviews for books. Users with special rights should be the ones to delete or modify database information, others could only post and get.

18.2.2021

I made database aggregation queries easier to access. A user system was finished, which I will use to deny certain to actions. For sessions, I used a token made by using a bcryptjs module. The tokens consist of id and name of a user.

One problematic situation occurred when acquiring information from the database. Using async and await in the function, fixed the problem.

26.2.2021

I added a review posting capability. A new query file was added for reviews, which could be used to view all reviews from a specific user.

**Finishing**

26.2.2021

For the ending, I did a logout part which would wipe user’s header token. The token is created after a user has signed in after being registered. Only signed users can post reviews and delete them, admins can control all the books and comments.

There is authenticate.js file that exports functions used for verifying, signing, and returning users. It limits the users performing certain actions. The routing files use authenticate.js in midst of get/post/delete operations.

Configurations for mongo and user token come from config file.