Lappeenrannan teknillinen yliopisto

School of Business and Management

Anytime-course: Software Development Skills: Full-Stack 2021-22

Bipika Amatya, 000850551

LEARNING DIARY, “Full-Stack 2021-22” MODULE

CT70A9140 Software Development Skills: Full-Stack

**LEARNING DIARY**

**NodeJS**29.03.2022  
I checked the NodeJS video. In that video I learn that the NodeJS is JavaScript runtime. It is not a framework or a language. I downloaded node from nodejs.org and installed in my machine. Then started by creating package.json file which was done by typing `npm init` in the terminal and default setup was chosen. Then a new js file index.js was created to tryout js codes. I printed a simple string using `console.log()` command which allows us to print anything to the console. After that again a new js file person.js was created to demonstrate the js module exports for reusability. Then logger was also tried out extending EventEmitter class which emitted the message to console when called. Then event, file system, http, os, path and url examples were tried. Server was also created that show different html page for different url patterns. The specific content type was use for specific file type for example, for file with js extension ‘text/javascript’ was used and so on. 404.html file was created in case user makes a request that is not recognized by the server.

**MongoDB**

07.04.2022

In this video about mongodb, I learned to install the mongo in local machine and connect with the cloud mongo server. I referred the latest mongodb video version. I first signed up to mongodb atlas cloud and the connection string from this cloud is used to connect from the local mongodb compass, mongodb shell and visual studio code using mongodb extension. Like RDBMS, mongodb provide all the function to queries the database easily using javascript friendly codes. Nodejs code can be used to connect with the mongodb atlas and visualizing the data from database.

**Express JS**

07.04.2022

Express is web framework for Nodejs. It is a server-side/backend framework. It makes building NodeJS web applications easier. It is light, fast and free. In this video, I create a simple server webpage running at port 5000. Then an array of member was listed at endpoint ‘/api/members’ to list the json data of member. This endpoint was queried using postman desktop application to show the list of example members. Single member detail was also obtains using the id value to the endpoint. For endpoint requesting non-existing ids, 400 status and message was returned by the backend.

Later express router was used for routing to list members, create a member, update a member, delete a member with the unique id. First the endpoint was tested in postman and the server returned json files. Then express-handlebars templating was used to create the view layout. Bootstrap was used for styling.

**MEAN-Stack**

07.05.2022

The project introduction videos illustrate the use of technology for building the project. We will be building rest api using node.js/express to create the endpoint, token generation and authentication for authorization, Cors enabling, Mongoose ODM for using user models, for frontend use angular 2 and angular-cli, angular router, http module, angular2-jwt to handle the tokens, Auth Guard for authenticating routes, angular flash message module and then at the end we will compile and deploy the created application in the cloud.

In second video, authentication application using node.js and mongoose. I started by installing mongodb and node.js in my local machine. Then I created simple server application using the express js to run at localhost port 3000. The mongoose was used to connect to the mongodb database. Express routers were used to create routes in order to get data from the server to display to client or web browser. A simple response string was sent from the server for the endpoints to test that it was working.

08.05.2022

In the third video, I learned to create the models for the users.js that was previously created. Here I created user schema with name, email, username and password. The function to get the user by id and username was created then addUser was created to add new user. I use bcryptjs to hash to user password. The user registering was tested using the postman. After adding the user, mongodb database meanauth’s users collection was query to check the hashed password that was saves in the database.

Then I used the passport js to authenticate if the password requested matched with that in the mongodb. I first find the user by the given username and then the given password is compared with this username password details from the database. If these both password is a match, then I return the token and user details as json response format as result in the postman and if there was no match among the passwords then I return message “Wrong Password”. Then at the users/profile end point, I use the jwt token as authorization to get that user details as result in the postman application.

In fifth video, I installed angular cli in my local machine. After that I created a folder “angular-src” inside the previous backend project I’ve been working with. Inside of which I create new angular project. Here the index.html inside src folder is the main file for the frontend. And we have app folder which contains components and main angular modules. The file app.module.ts is the main file which registers all the working components in it. Then I created “components” folder inside of which I created 6 new components using the angular command “ng g component <component name>” which comes with inbuilt css, html, typescript and spec files inside that component folder. Then I included those newly created components to the app.module.ts which is the main app module. I created a navbar using the bootstrap and added the routerLink values to the navigation bar menu such that on nav menu click it takes us to that route defined in that menu button.

Next video continues with the frontend development with angular 2 creating a form to register and input data validation and information messaging using flash messages. I connected each input value to the models and created a function that check for input validation.