Lappeenrannan teknillinen yliopisto

School of Business and Management

Sofware Development Skills

Jhuma Kabir Mim, 000327482

**LEARNING DIARY**

**Learning diary 1**

10. April.2022

I checked the general information and understood the main focus of the course, which is This lecture introduction about NodeJs and its application, which is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine, developed in c++, and executes JavaScript code outside a web browser. NodeJs lets developers use JavaScript to write command-line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Also, this lecture talks about the core fundamentals of JavaScript. I learned that javascript is an asynchronous language, how nodeJs can be set up, and how we can install different packages and frameworks in node js with the 'npm install' command. Also, I learned how to create a node server, a small micro API service, and get output in a JASON format.

To practice package installation I have test npm install command to install vue.js. To do that I have used npm install -g @vue/cli which has globally install vueJs framework in my pc.

**Learning diary 2 - MongoDB**

05. May.2022

This course is about MongoDB, which is a database, but it's a NoSQL database. It is scalable and very popular with the JavaScript community since it has similar JavaScript syntax to work with. Working with Mongo DB is simple; we just need to create a database name and collection and connect to the server. For practice, I'm using the localhost default port. Creating a collection is like creating a Table similar to an SQL database. MongoDB can be used as a cloud service. For my final project, I will use the MongoDB cloud service provided by AWS. Both cloud versions and the local version of MongoDB support the same command for creating, updating, or deleting databases. It's better to know some basic syntax for database management using the command line rather than visual mode. I'm trying to learn command line syntaxes, and I found it interesting since it's similar to JavaScript syntax.

**Learning diary 3- Express Js**

12. May.2022

This lecture is about express js, which is a back-end web application framework for Node.js, released as free and open-source software under the MIT License. Before working with any JavaScript framework, it's better to know, for example, JavaScript function, array, object, data type, and different kinds of operation, statements, and conditions. I learned basic JavaScript before learning ExpressJs, which helped me further learn about advanced arrow functions, map, JSON format, etc. First, I have learned basic route handling, HTTP request handling, query string, and URL parts. So to use it first, I need a node server that I have already installed, then I have to use POSTman for API testing. In my project, I will use express as routing the different pages, redirecting (for example, redirecting a successful user to new pages), and showing HTTP responses (for example, 404 page not found, 500 server error). Though I have learned a little advanced, for example, API creation with express js where I have used four fundamental commands GET, UPDATE, POST, and DELETE. He gets a request to get information that can show output from a file ( i.e., Html file), folder, or JSON format data.

**Learning diary 4- Angular js**

12. May.2022

**Project- MEVN**

12. May.2022

For project I have create small blog page where I can create a post and delete it. I have used Mongo DB[[1]](#footnote-1) for database, Expressjs for API creation , Vuejs for frontend management and Nodejs for server handling.

Frist I create an API that can support GET, POST and Delete request.

// Get Posts

router.get('/', async (req, res) => {

const posts = await loadPostsCollection();

res.send(await posts.find({}).toArray());

});

// Add Post

router.post('/', async (req, res) => {

const posts = await loadPostsCollection();

await posts.insertOne({

text: req.body.text,

createdAt: new Date()

});

res.status(201).send();

});

// Delete Post

router.delete('/:id', async (req, res) => {

const posts = await loadPostsCollection();

await posts.deleteOne({ \_id: new mongodb.ObjectID(req.params.id) });

res.status(200).send({});

});

async function loadPostsCollection() {

const client = await mongodb.MongoClient.connect(

'mongodb+srv://jhuma:Common@12!@cluster0.wsaic.mongodb.net/?retryWrites=true&w=majority',

{

useNewUrlParser: true

}

);

return client.db('Cluster0').collection('posts');

}

After that I have create a client that can show the content and perform delete operations of post. I have also used a third party weather API to show weather data. To test the project we have to run both server and client.

1. https://mlab.com/ [↑](#footnote-ref-1)