

Lappeenrannan teknillinen yliopisto

Software Development Skills Full-Stack

Jasper Tarvainen, 002575575

LEARNING DIARY

NodeJS part 1

26.10.2025

I have chosen VS code as my code editor for this course. Today I followed along the NodeJs video (updated) for 60 minutes and did the tasks in the video.

First I learned what node.js is and what is the difference between front-end and back-end. Then I installed the node.js and learned to use node commands and run how to run .js files on terminal with “node “filename”” command.

Then I also learned how to use node’s http module to build and run my own server that responses on the browser. In this part I also understood how req & res worked.

Then it was time to learn how to install packages with npm and how to use nodemon. And I learned that nodemon is very good for this kind of tasks, since it restarts the server automatically when you make changes to it, helping the developer to save a bit of time.

Also learned that not all files need to be in git repository, and how to hide files from there with .gitignore.

NodeJS part 2

27.10.2025

Today I followed along the rest of the nodejs video (updated).

I started from creating a simple API, that return user data in JSON form. This made me understand the fundamentals of backend and how frontend could collect data from the server.

Then we moved to middleware and handlers. I learned that they are functions which are run between requests and responses, and they help making the code clear, reusable, and easy to expand.

Then I learned how to write, read and append files, using fs module. I understood that fs.readFileSync is for smaller files and fs.readFile for bigger files, since readFile doesn't block other requests.

Then we learned to use path module, to handle directory paths and to handle folders. It automatically makes the path right, whether you are on windows/linux, so it makes the paths right changing from / to \ depending on your operating system.

Then OS module, to seek information about your systems cpu, memory, user etc.

Also learned about url modules. Learned that it can be used to analyze url's path and its parameters.

Then learned about crypto module, I understood that it can be used for security. Used for encrypting, hash and generating tokens. I learned how to make encrypt and decrypt data with AES algorithm.

Then I learned about emitting events with Node.js EventEmitter, and with it I was able to create events and listeners, to make the code reactive.

And lastly there was process object. With it you are able to seek information on Node process's environment variables and arguments.

MongoDB

27.10.2025

Today I watched the video about MongoDB, and created my own cluster. I learned that MongoDB is a cloud database, where I can manage and store data. I understood how to set up database on Atlas and how to connect to it using the connection string.

I learned about CRUD, to create the command is `insertOne()` / `insertMany()`, to read: `find()` and it can also be used with filters to find specific types of documents / data. To update: `updateOne()` / `updateMany()` and to delete: `deleteOne()` / `deleteMany()`

I watched the whole video, but didn't perform all the operations performed on the video, but I did some of the as I mentioned earlier, I did the set up on database, and connected to it, and also went through the installation processes.

ExpressJS part 1

28.10.2025

Today I watched 40 minutes of the course video about ExpressJS and did tasks provided on the video. I learned that Express is a web application framework for Node.js, and it makes it easy to build HTTP servers.

I installed the expressJS, and did a setup on a basic server. I learned how to send files and static content with `res.sendFile()` -command. With the static webserver I was able to for example serve a whole folder easily and I understood that express handles JSON data (`.json()`).

I also installed a helpful tool for web development, postman. It's a simple tool I installed as a vscode extension. With it I can test API routes without a browser.

I also did learn about params and query -commands, and setting up custom status responses. Lastly I learned about routing, I understood that it is possible to send only one response / request. Also understood that it is recommended to separate routes into different files, to make the code clean.

ExpressJS part 2

28.10.2025

Decided to follow along the rest of the video today. I continued from the POST request section. I learned how to create new resources in the backend using POST. Then I learned how to update old resources with PUT method, and how to find particular resource with id and to edit its values. Also learned about how to delete resources with DELETE method.

Then I learned to make custom error handler, and used middleware to check the status. Overall I learned new things about error handling.

Then we moved logic from routes to controller functions, to make the code easier to read and routes clearer, and this is also important to make the code scalable.

Then I learned how to make requests from frontend with `fetch()` API, to make requests to backend. I also used `fetch()` to send form data to API.

In the end I installed EJS engine, and I understood that it allows you to insert JS directly to HTML with its own tags. It allows us to create pages that change based on data from backend, for example if server sends data about current user's name (Mikko) then the EJS could response with Hello Mikko!

ReactJs

29.10.2025

Today I went through the video about ReactJs crash. Although I followed the video closely, I didn't complete all the tasks, since the tutorial was somewhat outdated and using the same style of code caused errors with the newer version. However, I still watched the whole video and understood the main idea of ReactJs.

I learned how to setup create React app with "npx create-react-app". I learned about JSX, I noticed it looks like HTML and describes UI elements, but it allows using JavaScript inside curly braces. I also understood that React apps are built using reusable components.

Then I learned about Props, and that they are used to pass data between components. For example, a button can receive its colors as a prop. And PropTypes are a way to check that props have the correct type, and they are not mandatory, but helpful to prevent bugs.

One important thing I learned about was state. With state, we can make the UI respond to user's actions. For example, if the user clicks "add" button, the thing the user added will be added to the shown list without having to refresh the page. This is also used on forms, when on HTML user sends something, the browser will send the message to server and then the page reloads, but on React the app will update the state and state changes and then it will run into the same situation without reloading the page.

Lastly I learned about routing with react-router-dom library. With this, it was possible to navigate between different pages on the site, without reloading the browser. It was also shown how layouts can be reused with routes, in this situation it was to show footer on every page.

Mern Stack Video 1

1.11.2025

Today I watched the first video of example project about MERN stack. I learned how to setup MongoDB database and how to connect it to Compass and VS Code using Mongoose. I also learned how to use the Get, Create, Update and Delete, and how to

use them as designed. I found that `.remove()` is not anymore working command. Then I asked AI if there is some other command nowadays and found out that its now `.delete()` to delete all or `.deleteOne()` to delete a specific one, so I used `goal.deleteOne()` to delete specific goal found by id.

Mern Stack Video 2

2.11.2025

I continued the series by watching the second video. I learned about JWT, an url-safe token used for secure transport of information between client and server. It was used for authentication and authorization. I learned how to define user schema (name, email, password) and how to securely store password with hash.

I understood how to create routes for registration, login, getting information and deleting information.

Also learned about protecting routes, to make some routes require authentication. The protected routes use the middleware we created in the tutorial, and it verifies the JWT token before allowing CRUD (Create, read, update, delete) operations. Get only Users Goals worked the same way, it showed currently logged-in user's goals, not anyone else's, preventing users accesing other's data. Also did same kind of thing for the all the CRUD elements.

Summary:

JWT provides token for secure authentication

Middleware makes it possible to have an authentication logic for multiple routes

Hash passwords before saving, to make more secure

Now that we have created 2 models, I understand how to make multiple models work together, to create more complex dependencies.

Mern Stack Video 3

2.11.2025

Today I moved on to the 3rd part of the tutorial series. I created the frontend, page components, header & navigation, pages, and did a set up on react routes. Also did a setup on Redux toolkit. Learned about updating state after API calls and setting up concurrently to run frontend and backend simultaneously. Also learned about authSlice and authService, which are part of redux toolkit. AuthSlice handles users authentication state, like is user logged in and authService communicates to backend, like register, login, logout.

Also first time in this course, I really faced challenges. I initially tried to use proxy for API calls in package.json, but it didn't work, I always ended up in error 404 when trying to register. I tried a solution found from StackOverflow link ->

(<https://stackoverflow.com/questions/70374005/invalid-options-object-dev-server-has-been-initialized-using-an-options-object>). But it didn't work either. Then tried to solve the problem with AI, it also gave the same solution as Stackoverflow, and also another solution, which I found working. I installed CORS package to handle the backend and now it worked smoothly.

Mern Stack Video 4

3.11.2025

Today I went through the last episode of the Mern stack series. Basically, we did the same thing we did in last episode, but now we just did a set up on goals (slice and service). Also did more frontend building, made the dashboard display goals, and goalForm to add new ones.

Then we added getGoals thunk to fetch all goals belonging to logged in user, to make it work, we need to send Auth header with the token. Also did the same, but to delete goals, for this we also needed to add button "X" to make the delete.

Then the deployment. In the tutorial, they used Heroku as a cloudbase to deploy the site to internet. I didn't use it since it wasn't free to use anymore. So I asked AI a list of services like Herkou but free, and ended up using Render. Then I just had to give render the git repository of this project and inside of the .env file, and also did a Postbuild script. Then I pushed it to render and tested it and it worked.

