Lappeenrannan teknillinen yliopisto

School of Business and Management

Sofware Development Skills

Pablo Oca Galeano, 002532699

LEARNING DIARY, UPOFITNESS MODULE

**LEARNING DIARY**

24.01.2025

I checked the general information and understood the main focus of the course and also, I have understood what the main focus of the course is. I chose UPOFITNESS module because I love to do sports and also the management of sports facilities and their users. I already had the environment set up so I had no problems with that. I have chosen Visual Studio Code.

25.01.2025

I have seen the Node.js Course, Express Course and started the React course.

30.01.2025

I started programming the front-end using React. I have made the Header and the Footer of the web page as well as the CSS related to both. Also I have done the first commit of the GitHub repository.

01.02.2025

I have made the log-in form and also the CSS related to the form.

04.02.2025

Today I have started programming the backend of the web. I have had some problems related to the connection with the MySQL database but after doing research on the internet, I have settled the problem.

Other problems have arisen as I have been programming, such as not being able to use a method in another file. I have investigated but have not been able to solve it, so I will continue investigating.

With all the progress I have made so far, I have made the second commit on GitHub.

24.02.2025

After watching the videos of the Mern-Stack example, all the necessary methods for both the relational database (MySQL) and the non-relational database (MongoDB) have been laid out on paper, so it has been decided to start the entire backend from scratch in order to follow the same directories as in this example.

25.02.2025

The remaining methods for the rest of the SQL database entities have been implemented, they just need to be tested through Postman. In addition, the encryption of the password and the iban for both users and workers are still missing. The token for the login and logout function is also missing.

26.02.2025

The methods used yesterday have been tested and two of them do not work, so I need to investigate why they do not work and how to fix it. In addition, the functions for the non-relational database have been created, although this one does not work either, so I need to investigate the same as with the other functions.

14.03.2025

Creation of the diet pages to display the client's diets, the About Me page to display user information, and modification of the necessary parameters on the login page.

15.03.2025

All backend routes are being checked. All are working, except those using the MongoDB database.

Two functions have also been implemented to encrypt user and employee passwords and IBANs.

The final step today has been the implementation of tokens for both user and employee registration and login.

06.05.2025

I'd been trying to fix a bug for several days, but I finally managed to do it. The problem was due to a host issue when connecting to the proxy on the frontend. Although it hasn't been uploaded to GitHub yet, the bug has been fixed.

10.05.2025

The backend methods were tested to introduce protection methods. All methods work except for listing by appointment worker. Methods that are not yet implemented, such as the logout and update methods for both users and workers, also do not work. The appointment update method has not yet been implemented.

12.05.2025

The methods have been protected and verified to work.

When testing the method for adding a new gym (only workers can do this), an error occurred because the function for retrieving a worker by ID didn't return anything because it was returned via an HTTP request. This has been modified to work.

The methods tested were those for adding, deleting, and updating a gym, as well as listing all users. These methods can only be run by gym workers.

The function for retrieving users by ID has also been changed to avoid the same problem as before.

15.05.2025

All appointment functions have been checked, and they all work, both those requiring worker authentication and those requiring user authentication.

The only odd thing is that in the methods for listing by date, worker, and user in Postman, you have to enter the date, worker ID, or user ID in the URL instead of in the Body section - x-www-form-urlencoded in Postman. I need to investigate why this is happening.