

# ASSIGNMENT FOR MEDEI

For any questions please contact me at [cos\\_raul@yahoo.com](mailto:cos_raul@yahoo.com). I would be more than happy to get in contact with you. Thank you for your time reading.

## Task

For this assignment I was asked to implement a small client-server web application with an AngularJS frontend communicating with a RESTful API written in NodeJS and ExpressJS that uses MongoDB as database calling Mongoose operations.

## Requirements

The system has the following requirements:

- Create a user (not restricted)
- Authenticate a user (not restricted)
- View information about the user that is logged in (requires authentication)
- Edit information about the user (requires authentication)
- An administrator user shall be able to view all users (requires authentication, admin rights)

## Result

The outcome system is rather convenient, as it met all the requirements. A user is able to register a new account, log in, edit user details, log out and delete his account. A user that is logged in with an account that has administration rights is also able to view a list of all registered users. I also added the possibility for the admin to be able to give admin permissions to any other use account from the list, as otherwise the creation of a new account with administrator rights is impossible through GUI. Instructions for how to run the project can be found in the last article of this report.

## Coding standard

Considering the layout, I usually write only one statement or declaration per line, I indent with one tab stop and I leave one blank line between method definitions and properties. I comment my code both on separate lines and at the end of lines of code. When not in a hurry, I would begin commented text with uppercase letter and end with a period. For naming vars I would always use a longer word with meaning rather than short names. I like the architecture of a software to be divided into 'backend' and 'frontend', don't judge me.

## My choice

As IDE I used JetBrains WebStorm for all the coding and as for browsers I used Google Chrome, Firefox Nightly and Microsoft Edge.

Regarding MongoDB, I chose to use MongoLab<sup>1</sup> (now calling themselves mlab) because they use MongoDB as a service and it's a lot easier to test and handle it, not having to work with multiple command prompts and 'cd-ing' all the time for the right path. It basically hosts your database on cloud, just as you would link the dbpath to them, and the nice part is that it's always up and running.

## Testing

Having a rather small full stack MEAN application had me thinking if Postman is enough or I should go for Karma framework. Reading a lot about how everything works and how to setup, I realized that the project is too small and that it's not worth implementing Mocha for server testing as the files for the api are not that complex and there aren't so many routes. Also having Jasmine for client side would require more effort and time to set everything up than to actually test it with the Postman, Chrome's plugin. Of course Postman won't ensure good quality of the code and won't cover everything, but for such a small project, some functionality tests are enough. Therefore, after each route declared in the server api, I tested it with Postman and made sure it works as it should, and after all backend was at its place, made the controllers and services connect with the server. I know that more testing would have gave the code more quality, and probably if I were to do it again I would use Karma.

## Conclusion

In conclusion, I think the final version of the application meets the expectations, with some small exceptions here and there. The logic behind user Schema is not the best, having a Boolean property 'isAdmin' to identify if a specific user is administrator or not, but I worked around with this and it seems to be working just fine. It's the first time I am using tokens for logging in, as I decided I should leave cookies behind, so there might be some problems with the tokens, but I couldn't find any problems as far as I searched for them.

## Instructions

You need to have NodeJS installed on your computer, then open cmd or power shell and navigate to the project root folder and start the server by typing `>npm start` or `>node server.js`. You will notice it will probably log that it successfully connected to runner's port and to database. Further, open a web browser and navigate open <http://localhost:3000/>. There should already be the login page. To log in as an administrator you can use email: "admin@admin.admin" and the password, obviously: "admin"; or you can use "[cos.raul@yahoo.com](mailto:cos.raul@yahoo.com)" with the password: "mypassword".

---

<sup>1</sup> <https://mlab.com/> - Their website