

Coursework II

Petru-Marian Burlacu

40285539@live.napier.ac.uk

Edinburgh Napier University - Web Technologies (set008101)

1 Introduction

The Web Technologies second coursework focuses on designing a blog web application platform that contains a server and a client side. This coursework is a great opportunity to learn the process of creating a web application using MEAN stack development. The application provides an option for user and blog posts creation. The data will be stored in a database and will be editable. A blog post can be edited or deleted and retrieved in a user interface. User authentication is provided using the database connection and both server and client sides. The approach used in this project are mainly from documentations of MEAN stack elements thought by David Acosta in his YouTube Tutorial Series and Envato Web Development learning series. Bootstrap was used for a visual overlay.

2 Software Design

For this coursework, my main plan was to learn, understand and use AngularJS JavaScript framework as I will be working with it starting this summer at my student placement at BlackRock. After researching my possibilities and remembering the lectures I decided to develop my web application with MEAN.Js Stack (MongoDB, Express, Angular, Node.js). I followed a YouTube tutorial series that offers a very detailed explanation of everything one needs to know about MEAN Stack development using the default techniques found in documentations of all MEAN components. I truly learned to understand the development process behind a web application development and used what I considered useful for my blog. Using full stack JavaScript seemed like a good approach to reach the requirements for this coursework and to learn more about this environment for my future. The blog's server platform would use Node.js with Express web server framework. Angular is used on the client side of the application. To store blog posts and users the application uses MongoDB database. The connection between the database and server side is made by Mongoose package.

This Blog MEAN Web Application requires all four components of MEAN stack: MongoDB, Express, Angular, and Node.js. Additionally, it uses packages to interact or enhance the process. To interact with MongoDB database, I used Mongoose NPM. To parse request parameters, I installed Body-Parser Middleware. To pass the data from the Development server to back end, I used Cors NPM. Json WebToken NPM was installed for web token to recognize user sessions. I used Nodemon to update my server after saving any changes without having to do it manually.

Robomongo application is an interface that helped me interact with my database.

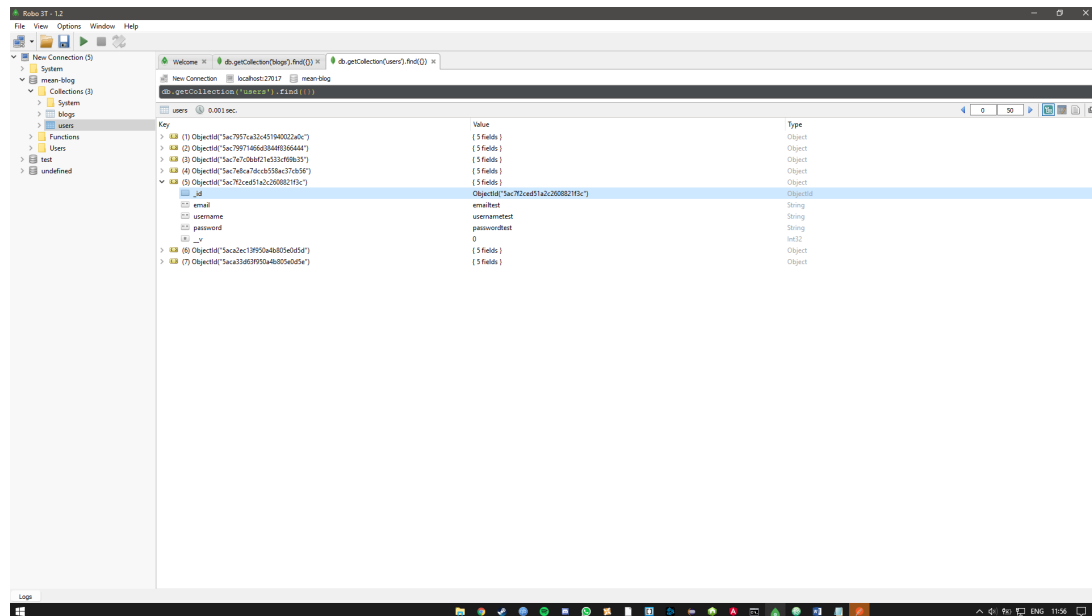


Figure 1: Robomongo User Interface for - MongoDB

I used Postman application to interact with the requests on the server side between requests and database and to check my connection and look for errors.

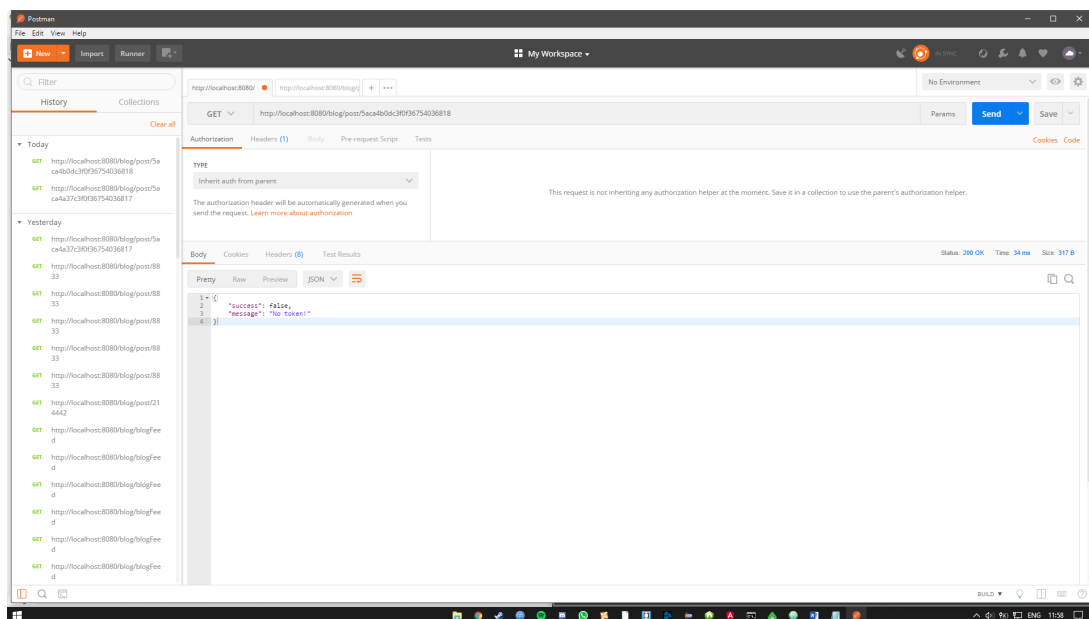


Figure 2: Postman Application- Requests Check

To connect to Node Server from Angular application side I had to set a proxy (client/src/proxy.json).

To interact with MongoDB using Mongoose, I had to define a Schema and create a model on the server side.

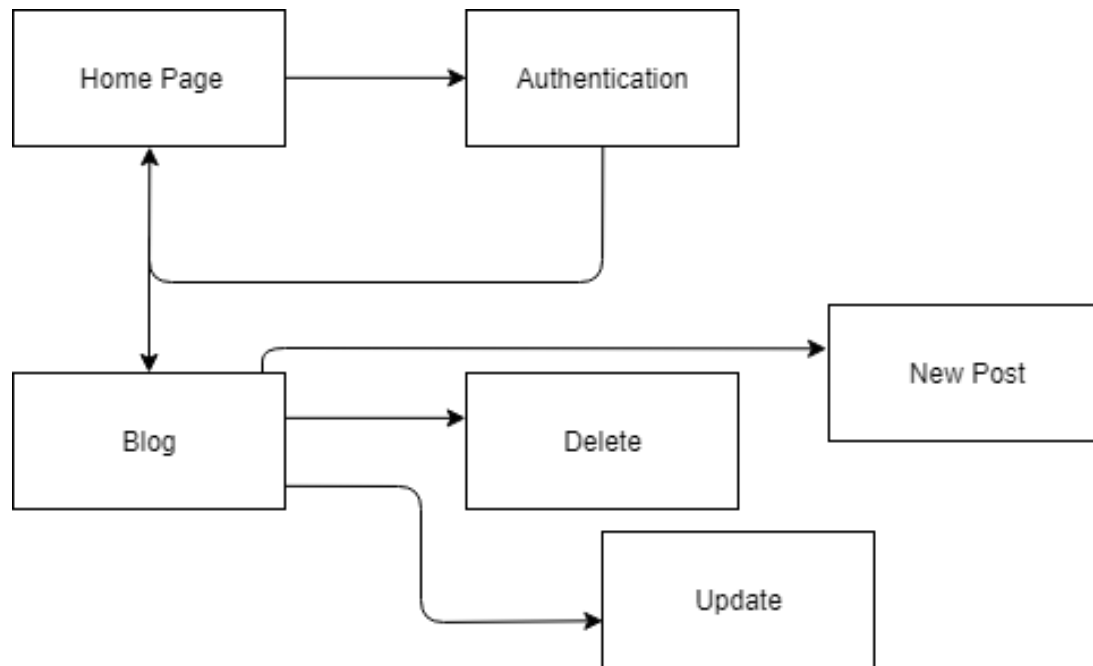


Figure 3: Main Routes

3 Implementation

On the Angular client side, the main app component calls dynamically other components using the routes. The navigation is always in the app component.

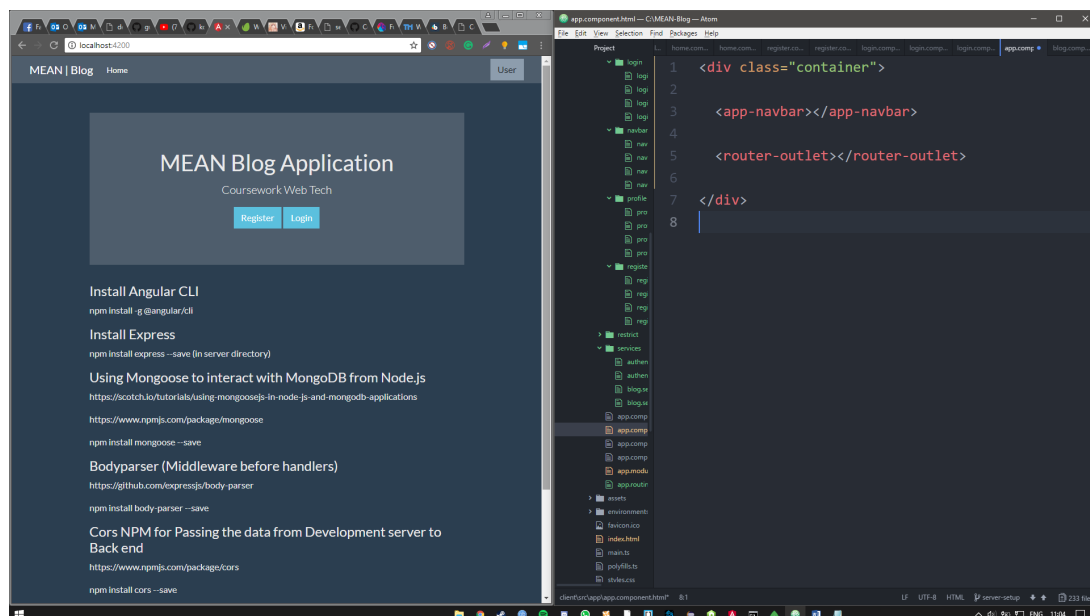


Figure 4: **router-outlet** Calling components

Depending on the user navigation, the HTML router-outlet element in the app component will call different elements.

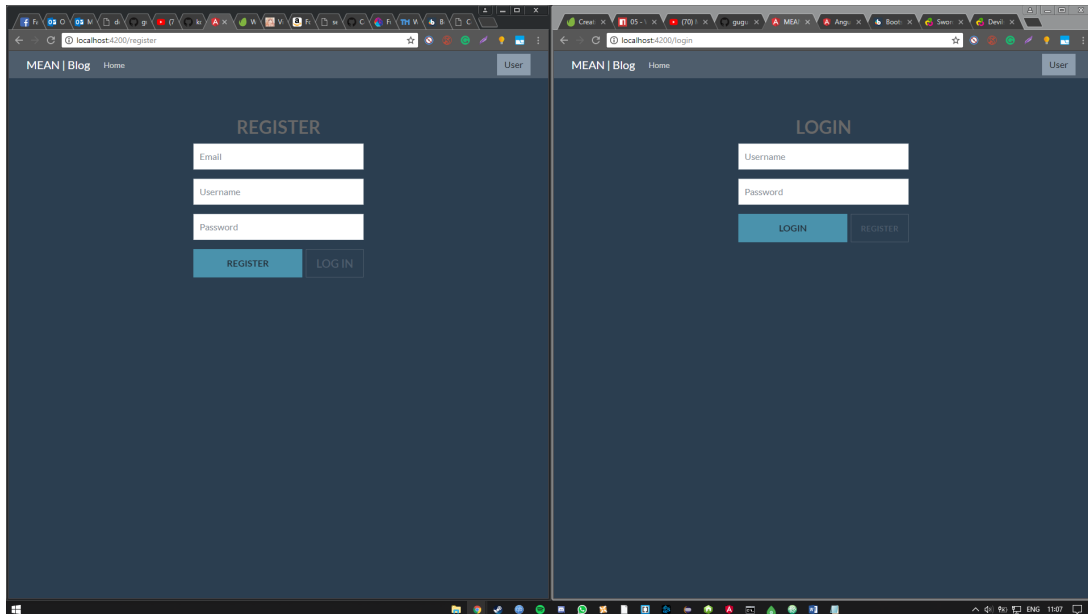


Figure 5:

After creating an account and successfully logging in using the database, the user is redirected to the home page with the option to access the blog component or access user options (log out / user profile).

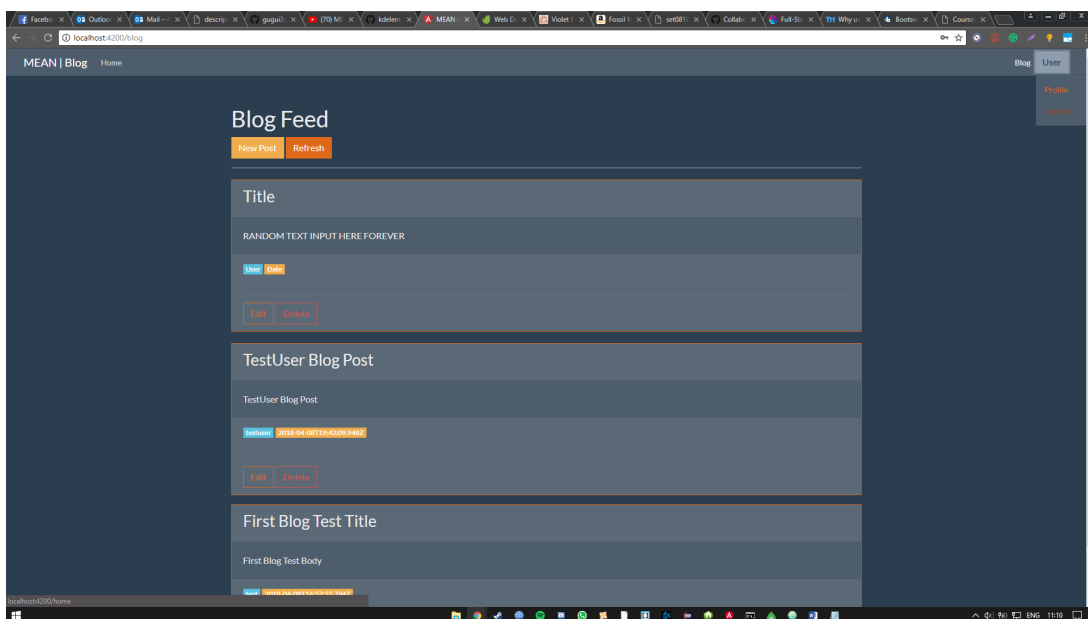


Figure 6:

The blog page allows users to view the blog feed, create a new post and to edit or delete a post.

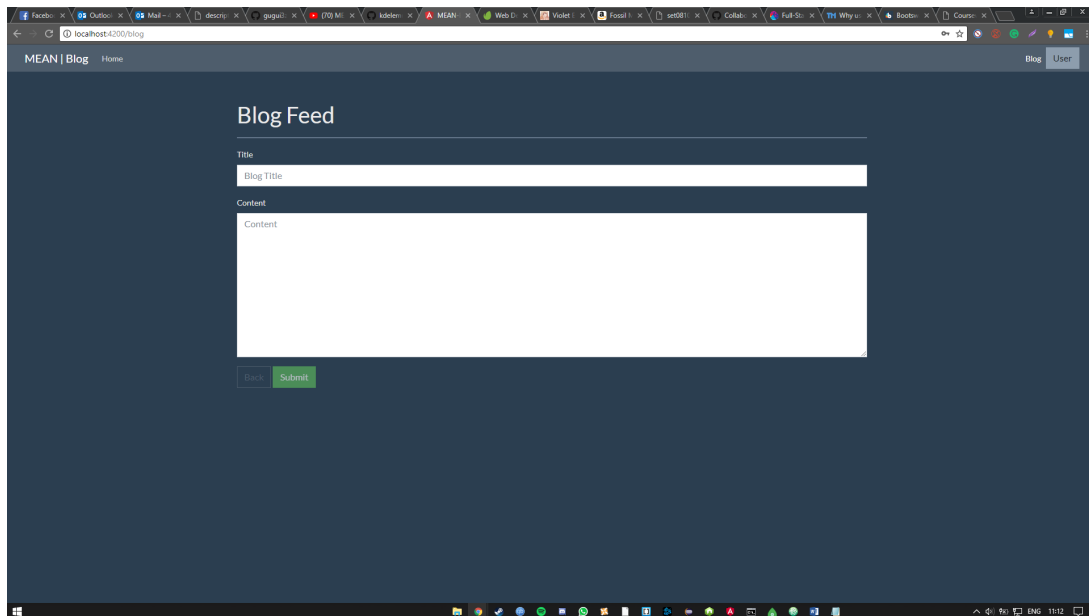


Figure 7:

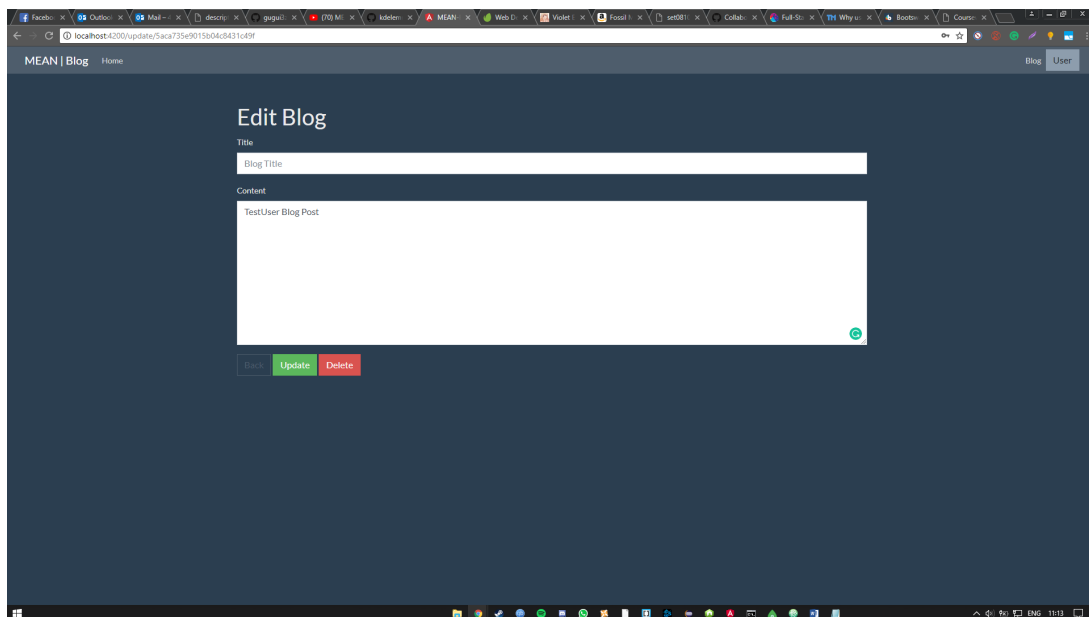


Figure 8:

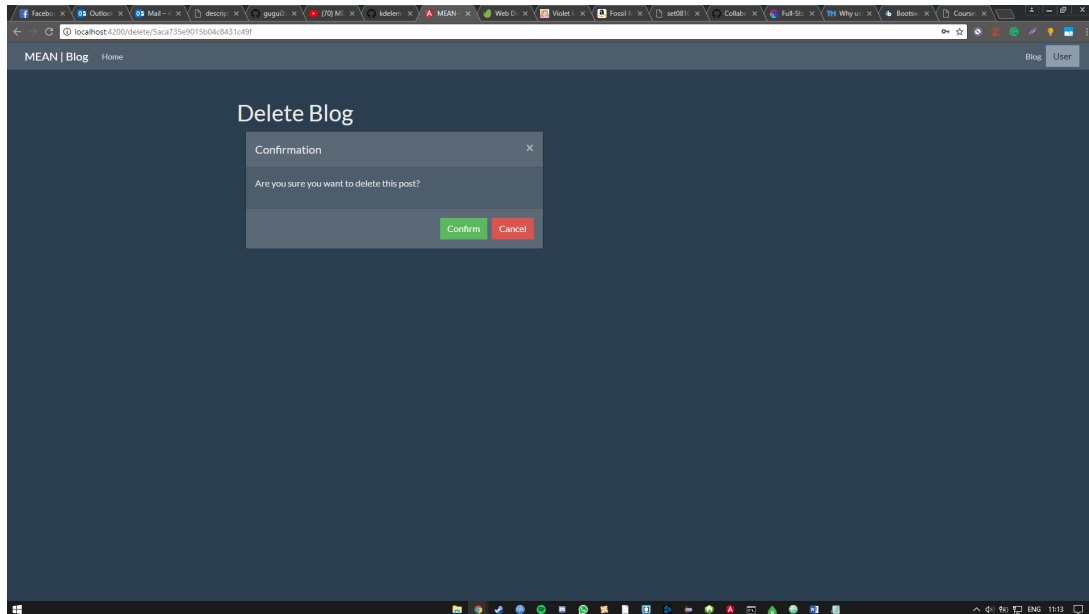


Figure 9:

4 Critical Evaluation

As any blog, the application can display a blog feed. This blog application is basic and allows users to only post, delete or update a blog post. Additionally, there is a level of simple security added to the application, using authentication methods the user can register and login with an account to access the blog main features. For the CSS and customization, I used Bootstrap framework to make a simple and fast application with predefined stylesheet templates.

I would consider in the future adding more features that are common in blogs. For example, a feature to post a comment and like blog components. To implement a search for posts option and more consideration for the user. A dashboard to manage user posted posts and manage profile. Add more security to the application and allow users to filter content by categories and customize their homepage.

For this coursework, I could have made a more serious validation for user registration/login using conditions and regex. I would make sure that every user that registers in the database is formatted exactly how we need in order to avoid future bugs or complications. Maybe I could have implemented an encryption system for passwords to make it more secure. Learn more about web tokens and make user sessions fluid. Unfortunately, I did not have enough time to work on minor features and make the application run smoothly without bugs. A common situation that the user can encounter is being randomly logged out of the system. From my research and attempt to fix, this is caused by web tokens not being properly stored and read in local storage. A solution to this problem would be implementing a method that allows us to read the local storage the right way and take consideration of the user's session duration. Another thing I would love to implement is form validation for words and restrictions and add interactive visual responses for user's commands, such as pop-ups or hidden divisions. I would consider social media integration, adding the option to share the posts or implement social media features to a post.

5 Personal Evaluation

Taking this module made me understand that there is so much more that I don't know about the web technologies. I am truly appreciative of this opportunity to learn about web applications development. My main focus was to get some experience in using Angular and the whole MEAN stack development. It was a new and challenging experience and I learned to understand the environment and what I need to focus on to improve. For the first time, I had the opportunity to work on a whole project with both client side and server side parts, understand the interdependencies between elements and how JavaScript is in the middle of it all.

When I say the development process, I refer to it all. From the appropriate file naming and organization to the actual web application deployment. I learned to properly use the documentation provided by MEAN stack elements and where to look for when I need to find something. The documentation, to my surprise and satisfaction, that it is provided with each element is very well written and organized, making it easy to search and understand how to implement a feature. I learned how to set up a MongoDB database and connect it to my application, what tools to use, such as Mongorobot, to interact with it, get feedback and finally found out where do I need to use the techniques learned in classes for it.

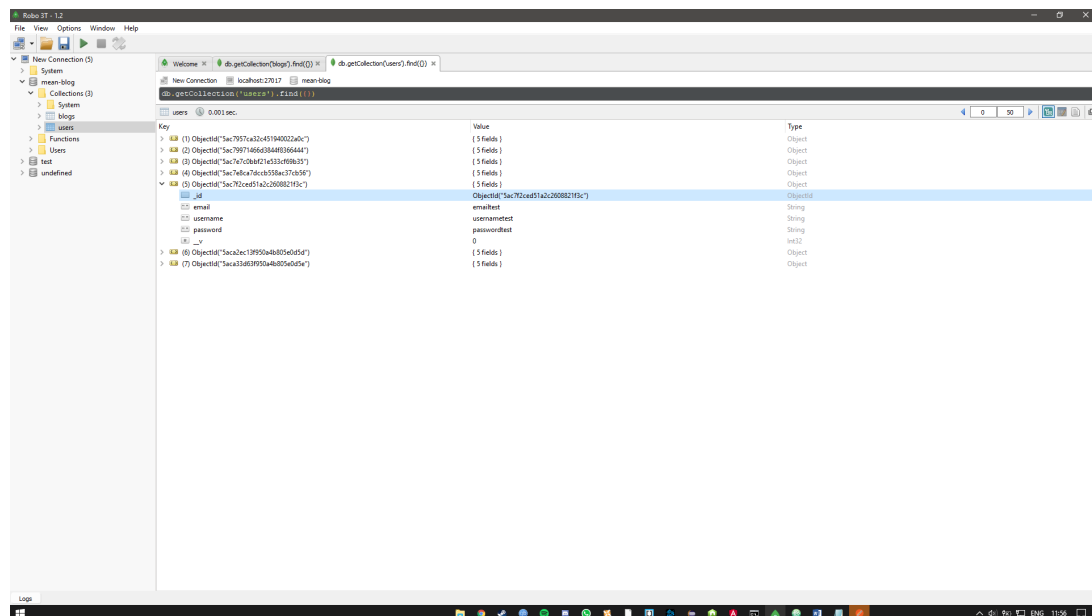


Figure 10:

I learned the importance of routes and requests. To test my request and connection with the database on the server side I used Postman application. This helped me understand what error can appear in different situations and how to handle them. It provided a feedback on my requests to see if it actually works in order to continue with my development process.

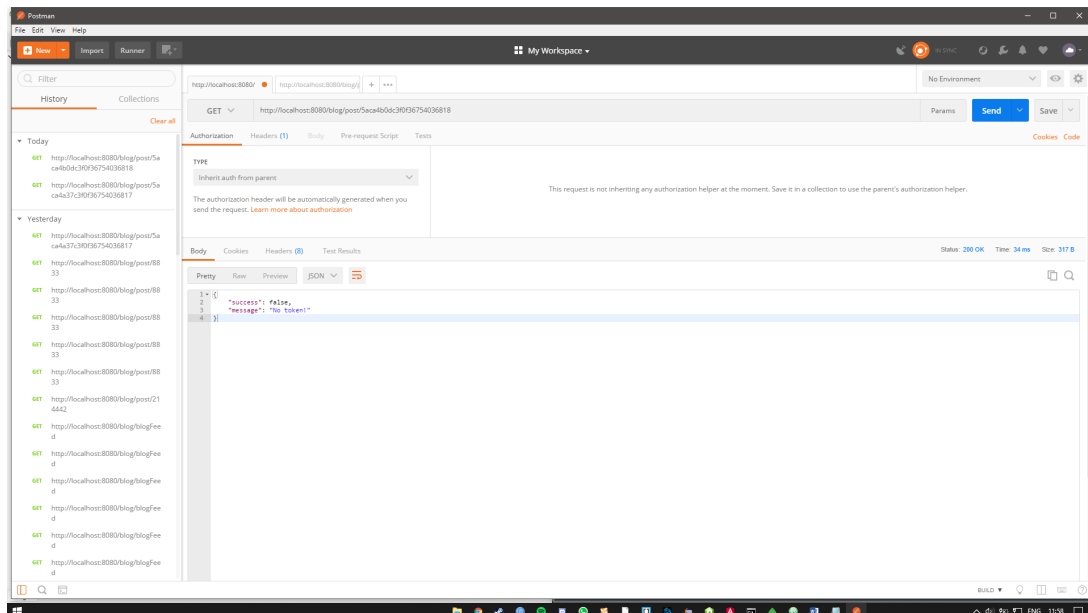


Figure 11:

Another great tool that I learned to use was the browser inspect element. It provided a great support in solving countless errors, issues and helped me understand how it works. Accessing the browser's application section and I understood where the local storage is and how it works.

Even if I followed a tutorial, it is based on documentation materials of MEAN stack elements and to better understand and develop this blog, I encountered an unprecedented amount of errors and issues. From misrouting to simply not handling properly request errors or database validation. The most important thing was that I learned how to approach these problems. I would always start by testing my request, sending JSON messages every step of the development. I used console log to state every conditional step and print/ understand errors. Another important step for me is to learn how to use the command control and how helpful it is once you log every step of the actions. It provided great feedback in case of an error and provides a good direction of where I have to look at. I believe that I have spent more time fixing issues and reading the documentation in my application development process. I had to research in order to understand how something works so I could fix it in many tutorial steps.

I believe it was the right thing for me to approach. It opened a door for me exactly for what I need to learn in order for me to give my best in my placement and continue with my goals. I genuinely felt it was a challenging experience and made me realize the potential of web applications development. I now understand how many websites work and I had the possibility to connect the teachings from this module with a practical experience to understand it better. I am planning to continue working on this project after the coursework to implement some of the stated above features and learn more about it. For the first time, I had a glimpse of a direction for my honors project and what I need to focus on in the future.

References

YouTube Tutorial Series by David Acosta

https://www.youtube.com/watch?v=G_xHi0jywmclist=PL3vQyqzqjZ62zMxa0veah_CHMgwOM4JWDI

<https://code.tutsplus.com/series/creating-a-blogging-app-using-angular-mongodb--cms-1249>

Bootstrap

<https://bootswatch.com/superhero/>