Report by Scott Murphy

Introduction

This assignment in which the task was to create a website containing several ciphers with encrypt and decrypt functionality was expanded in part two where we had to run the site on a server with login and messaging functions. This site has been updated to have a login page and also a sign up page to allow the user to create an account. These will connect to a database using Mongodb which allows one to push data to a database which in the case of our site will store the user's login details. For the view engine, Pug was used which is one of several that can be used which meant translating all html files into the respective view engine.
Reading up on these I used the lab tutorials which provided a good amount of context when writing the site into node.js and also helps in understanding how it should work. However I struggled to implement many of the features

Software design

Once deciding on which extensions to use and what view engine to use, my next step was to create a location for the application and reinstall node.js in an easy to locate area. Subsequently I created a Mongodb account and created a cluster for the data.

From there, planning an order in which to create the functions where the most appropriate seemed to be register, login and then the message function. With specifications laid out, it was time to commence the development as such things as a requirement list would have been ineffective as during the development stage I changed many of the items I used such as Ejs to Pug.

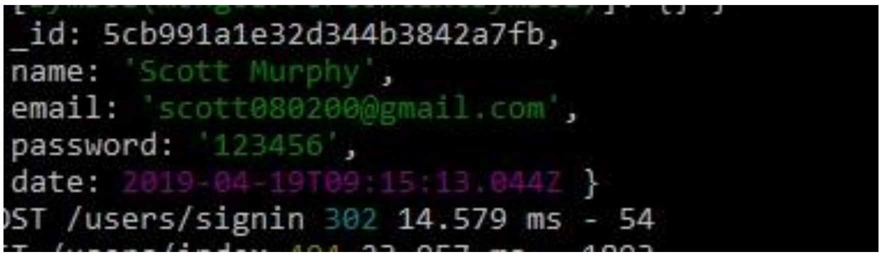
Implementation

Upon implementing the website into node.js the process was relatively effective however upon translating the site over a few parts of the design seemed to change as the sidebar is now a very similar colour to the page contents as well as it being not as wide. The text editor brackets was very helpful in translating into pug due to its error detection feature where it made it possible to find these problems which could then subsequently be searched for online. For the input areas, the best method would be to use forms as a way to then retrieve your data as you could then save the data on mongo db.

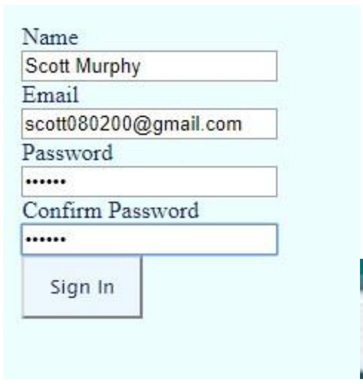
Using express, mongoose and pug as the main packages helped create the basic functionality of the site such as the register and login function and to authenticate, passport and passport-local would have effectively done this.

After this, it was time to create the pages which there were several that were dealing with data retrieval and uploading to mongo which was all written in JavaScript. Within the file "User" we would create an object ready for transfer over to the database using mongoose.

This would then go into the “users” file and when the sign in button is pressed, it subsequently saves the user to the database.



```
_id: 5cb991a1e32d344b3842a7fb,
name: 'Scott Murphy',
email: 'scott080200@gmail.com',
password: '123456',
date: 2019-04-19T09:15:13.044Z }
ST /users/signin 302 14.579 ms - 54
```



Name
Scott Murphy
Email
scott080200@gmail.com
Password
••••••
Confirm Password
••••••

localhost:3000/signin

Critical evaluation of your implementation

Upon implementation I would change many of the features I used such as mongo db Atlas as when testing the application it numerous times would stop functioning and this caused testing to take a long time. Pug has both advantages and disadvantages however in the end I decided to use this as it was now more familiar than the other languages. Overall I believe it would have been better to increase the total functions that this application uses and potentially implement with Ejs instead as once I changed over it seemed more complicated.

Personal evaluation

Through this process I have learned a lot about how “node.js”, allowing for server side programming and “mongodb”. Overall most problems that arose came down to external factors which consumed a lot of the time. If not for this there would be several more features such as the messaging system that would have completed the program.

References

Labs 7 to 10

https://stackoverflow.com/questions/30924859/unable-to-connect-to-mongolab-gettingmongoerror-auth-failed Saving Data to MongoDB
https://www.youtube.com/watch?v=2oYtk83FZCA