

# Web Technologies Coursework 2 Report

Fraser Yuill - 40318856

[40318856@live.napier.ac.uk](mailto:40318856@live.napier.ac.uk)

Edinburgh Napier University - Web Technologies (SET08101)

## **Introduction**

For this project, I have been tasked with updating my previous webpage by allowing server functionality which includes being able to persist data about messages and users for future use of the website. There also must be client-side updates, which are supported by the server-side elements, that include; a sign-up system, a login system, the ability to send messages to other users and the ability to read messages sent by other users. Before starting to plan out how I would approach this task I read multiple sources for a more advanced look on how this whole project could be done efficiently and to a high standard[ CITATION Bas14 \l 2057 ], [ CITATION Aza14 \l 2057 ]. These sources allowed me to gain more of an insight into how 'node js' and 'express js' are used and how I could implement the strategies discussed into my own project with my own takes on them. This was all done using the 'node js' and 'express js' server which allowed me to put my site on a localhost server that hosts the site and allows for both client and server side elements for each user. On my site, I have implemented all the features that I was tasked with doing. Firstly, when entering the site, you are brought to the login page which allows you to either login using your entered email and password or signup for the webpage. The signup page allows the user to enter their first and last names, their email address and a password, which, if no errors are found, is then stored. When a user logs in, they are then taken to the home page which is the page from the last coursework but updated with 'send message' and 'inbox' buttons. These buttons take you to the message and inbox pages respectfully, message is where a message can be sent to another user and inbox is where a user can look at the messages sent to them.

## **Software Design**

Before planning how I will approach this task I believe I will need to investigate what the requirements are for this task and how I will be tackling them. The main objectives are to create a site with client and server-side functionality, and with that have a login and signup system and a messaging system. This whole task could be done mainly using the node js setup as it allows you to run your website on a server and because

it allows you to use node modules that can enhance your capabilities. Now that I have planned how I will approach this task; I believe I should sketch the pages I plan on creating. The login, signup, messaging and inbox pages all need a sketched design that can be used as a basis for the website. Firstly, the login page (see appendix 1) is simple and easy to read as it is just a login page and so isn't the core part of the site which allows it to be easily used by any user. Next, we have the signup page (see appendix 2), this is clearly quite similar to the login page and is meant to be as they are similar kinds of pages and so I believe that keeping a consistent model keeps the user aware of where they are on the site. Now we have the messaging page (see appendix 3), I'm not exactly sure if this is how the page will be, but I plan on having this layout with all the elements needed to send a message readily available to the user on this page. Finally, there is the inbox page (see appendix 4), this is designed to be as easy to read as possible so that the user does not get distracted by anything other than the messages that were sent to them and to keep personal for each user.

## **Implementation**

The site's implementation was actually more difficult than I thought it would be. Firstly, I created the login system/page (see appendix 5) and signup system/page (see appendix 6), this was done by using a database which stored all the user's data within which happened on the signup page. When the user first goes on the site, they are taken to the login page and have to go to the signup page from there, in the signup page are all the necessary fields required to create an account on this system and once the account is created the user is taken back to the login page where they are prompted to enter their login info. Attempting to signup or login both trigger a post event which the system handles by going to the index.js file that runs the page and its elements. For the messaging and inbox pages (see appendixes 7 and 8), I used another database which has a primary use of storing messages, their decoded forms, who they're from etc. Within the messaging page the user is told to enter a recipient and a subject then when they submit this the message entered on the cipher page is taken as the message and stored in the database. On the inbox page, the user's email is taken and searched in the database to find any messages that are associated with that address and if so then the details of that message are displayed for the user.

## **Critical Evaluation**

To compare my project here to the list of requirements that were set out before me, I believe that I have achieved everything that was asked of me but could've still improved some elements of my website. Firstly, the task

was to create a website with client and server-side features, this task was completed as all the features were implemented. The login system, signup system, messaging, and inbox were all created and added to my site for use by any user who wishes to do so. The features are all, in my interpretation, laid out exactly as they were intended but they still could have been improved with regards to design and security throughout the site. If I had more time, I would've attempted to implement much more reliable security features such as the use of cookies and/or the use of the bcrypt module. Either would allow any user a more safe and enjoyable experience on my platform and would reduce any risk of a breach in the security or a hacker attempted to steal people's information from the databases. I do believe however that I did slightly improve the aesthetics of my site as I was told from the first project that an issue with my site was that the colour scheme was quite odd and that it would be quite unappealing to any users, I changed this by keeping the whole site one colour (which is a light pink) and by keeping the design of each page consistent. If I had more time still then I would've also attempted to have admin capabilities which would allow a certain user(s) to do things such as delete accounts, messages etc. A great example of this is shown on the bootsnipp website [ CITATION boo19 \l 2057 ], which has many examples of admin pages, their layouts, what features they should have etc.

## **Personal Evaluation**

Overall, I feel I have performed okay but I have room for improvement when it comes to web design and implementation. Throughout this project I have learned many valuable things in relation to web site creation, mainly to do with the server-side of things and how they all work. I learned all about using a server hosting platform such as node js, which allowed me to understand how a server could be implemented in the real world and the difficulties that come with keeping it running and maintaining it. The server elements that I learned about also brought about many of the challenges I faced throughout this project. I struggled quite a bit with understanding how node js works and how it should be used in hosting sites, I studied up on this quite a bit and eventually overcame this challenge by looking up what the best techniques were to use node js properly [ CITATION nod19 \l 2057 ]. I also struggled with using express js which can be useful with this kind of task as it can allow you to run a site more efficiently and quickly, I also overcame this using information available to me on the internet [ CITATION exp19 \l 2057 ]. Once again, my skills were tested on this project but I believe I have shown I am capable of doing such a task, but I still have some things to learn.

## **References**

(2019, April 2). Retrieved from bootsnipp.com: <https://bootsnipp.com/tags/admin>

(2019, April 4). Retrieved from expressjs.com: <https://expressjs.com/>

Mardan, A. (2014). *Starting with Express.js*. Apress, Berkeley, CA.

nodejs. (2019, April 5). Retrieved from w3schools.com: <https://www.w3schools.com/nodejs/>

Syed, B. A. (2014). *Beginning Node.js*. Apress, Berkeley, CA.

## Appendix

### Appendix 1:

Login

email enter box

password enter box

Signup

login

### Appendix 2:

# Signup

Enter first name box

Enter last name box

Enter email box

Enter password box

Signup button

## Appendix 3:

# Message

Enter to email box

Enter subject box

choose cypher button

Enter message box

send message button

**Appendix 4:**

# Inbox

Area for messages to appear

**Appendix 5:**

Login

Email:

Password:

signup

Submit

**Appendix 6:**

## Signup

First name:

Last Name:

Email:

Password:

Submit

## Appendix 7:

## Message

To (email):

Subject:

Submit

## Appendix 8:

### Mail

- To: fraser@1990
- From: fraseryuill99@gmail.com
- Subject: qwer
- Message: DODO
- Decoded Message: fwfw
- To: fraser@1990
- From: fraseryuill99@gmail.com
- Subject: Chris
- Message: NFE
- Decoded Message: vhk
- To: fraser@1990
- From: fraseryuill99@gmail.com
- Subject: hiya
- Message: NRENS ONWWTON...
- Decoded Message: Enter Message...