

# Coursework Report

Ross Muego  
40280659@live.napier.ac.uk  
Edinburgh Napier University - Web Technologies (SET08101)

## 1 Introduction

My blog is a generic blog with no specific category or genre of posts in mind. This will allow for a wide range of users to use the site and not limit it to a specific user group. The main function of the blog will of course be to make blog posts. Posts will contain a body of text being the content of the post, along with a title and an accompanying cover photo. Users will be able to use the site if they are not logged in or registered, however in order to access all the features of the site, they will need to be signed in with an account. Users are able to make accounts and choose a username, their location and a profile picture. They will be able to update some of these details once their account has been created.

A users profile will display all of these details in a central area. On the users profile also you will be able to view all of the posts made by that user on their account. From a users profile also, other users will be able to follow and unfollow the user. If the user is on their own profile, the follow/unfollow button will be an edit profile button.

There will be a main homepage of the site where all posts made by all users are displayed. There will also be a section of the site dedicated to just showing posts by users that the logged in user follows. Users that are not logged in or registered will not have access to this feature.

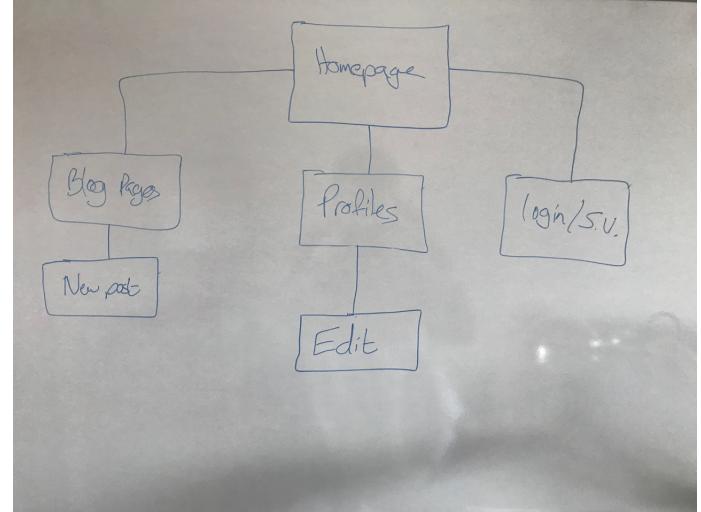
Posts will be able to be commented on by logged in users. Comments will display under the post and will show who it was that has posted the comment. Comments are displayed with the newest at the top, meaning you will see the newest comment first as you naturally scroll down the page.

My background reading for this project has mainly consisted of w3 schools in order to understand how node and mongodb worked, as well as the tutorials provided for this module to understand the basics of express. I have previously created a mock social media site in which I seen some similarities in this project so I also looked back over that to give me an idea of how I could implement some of the features.

## 2 Software Design

### 2.1 Site Structure

The first thing I done was make note on my whiteboard the basic structure for the site. This was just to get an idea of what pages I would need in order to provide the desired functionality.

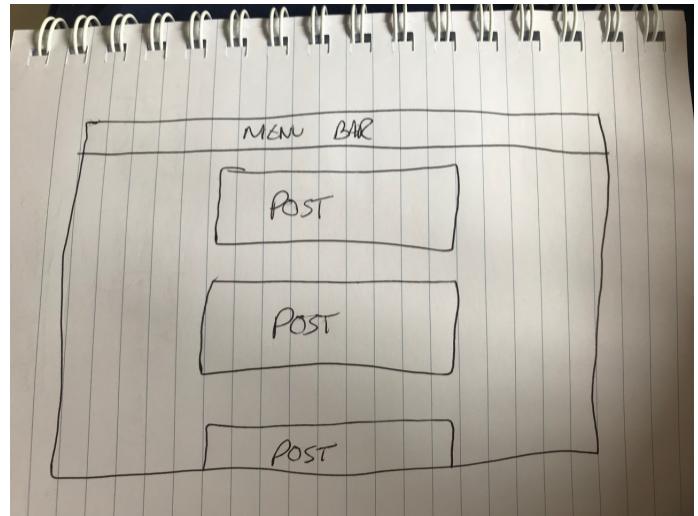
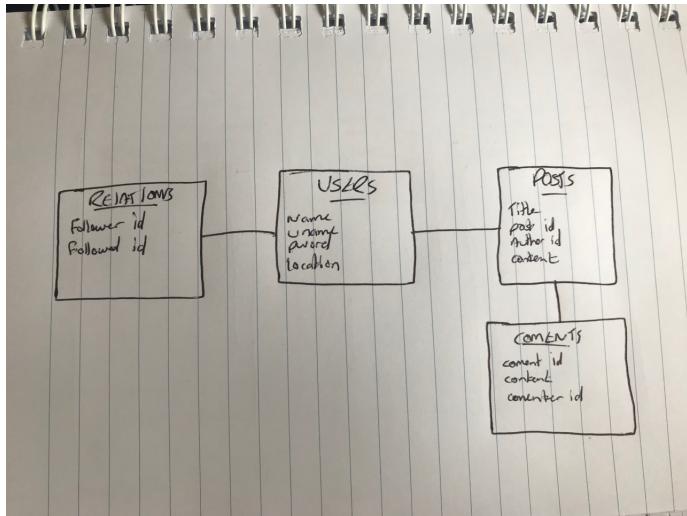


From this I established I would need to create a homepage, a profile page, a page for an existing post, a new post page and a signup/registration page.

### 2.2 Database Structure

Next, I moved on to how I would retain the date from the site and make it so people could access posts and user data etc. To do this I decided to research and implement MongoDB. I am familiar with SQL, however MongoDB's object notation appeared to be easier to use in a JavaScript environment. This paired with the opportunity to learn about a new technology was ultimately what made me decide to implement a MongoDB database to store my data.

I knew from previous experience roughly what MongoDB collections I would need to execute a project like this. I decided upon a 'users' collection, responsible for storing user data such as usernames, passwords, names etc. All the users information would be stored in this collection. Next, I was going to need a collection for the posts made on the site. This would contain the title of the post, the content, a unique ID generated for that post and the user ID of the author of the post. A new collection would be created in order to store the comments made on posts, and this would be constructed of the ID of the post the comment has been left on and the content of the comment, along with the user ID of the person who posted the comment. The final collection required would be for relationships between users. This would keep track of which users follow one another. This only requires the ID of the user being followed, and the ID of the person who is following that user.

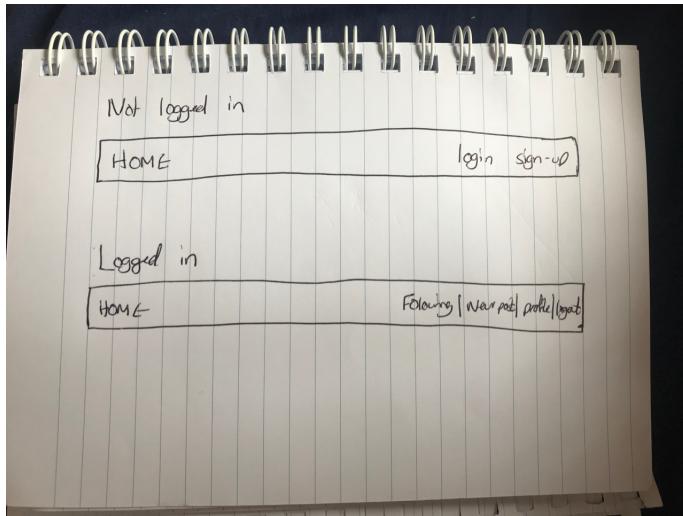


## 2.3 Page Design

Once I had came up with a basic site structure and database design, I turned to designing the individual pages of the site and what I envisioned those to look like.

### 2.3.1 Menu Bar

There would be a menu bar across the top providing access to the homepage and giving the users the option to login/register if they hadn't already. This would be consistent across the whole site, and would change depending on the users login status.

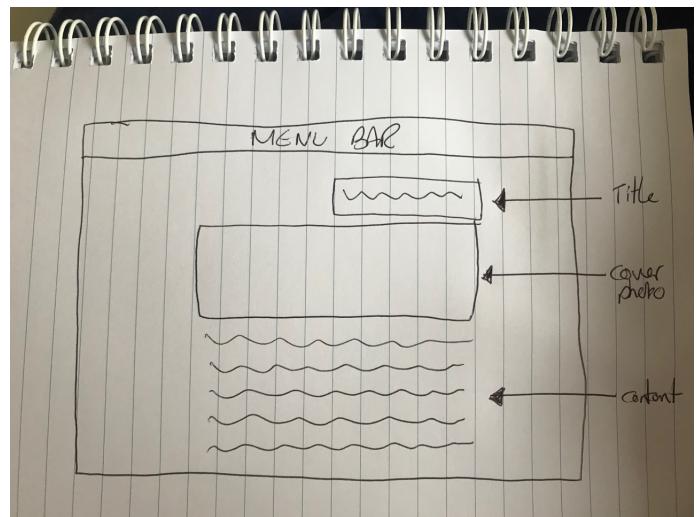


### 2.3.2 Index/Homepage

For the homepage I envisioned it to be very simple and basic, with just a list of blog posts from all users. The post elements will contain a small snippet of the content of the post, along with the title and a smaller version of the cover photo that will feature on the posts actual page.

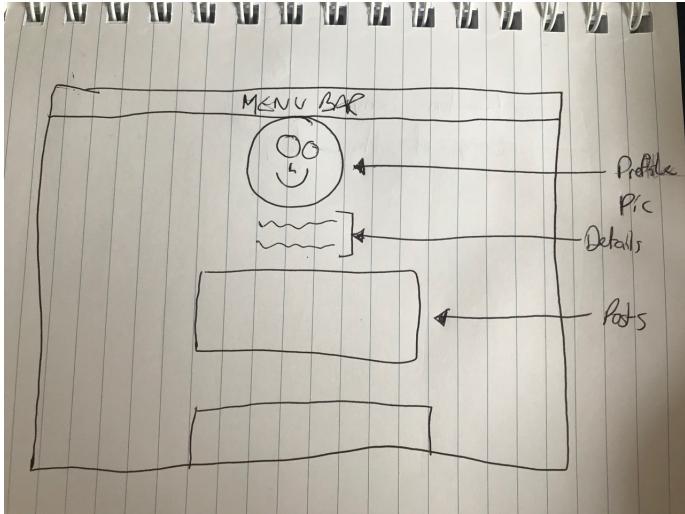
### 2.3.3 Posts

The post pages are relatively simple in the sense of there are not a lot of elements to it. At the top there will be the title of the blog post, followed by the cover photo and lastly the content body of the post. Under this there will be the comments section where logged in users can post comments.



### 2.3.4 Profiles

A users profile must display all of the information they have decided to share with the site, along with all of their posts and the opportunity to follow/unfollow or edit the logged in users profile. If you try follow a user and you are not logged in, you will be redirected to the login/registration pages.



### 2.3.5 Login/Registration

This will be split up into 2 pages. One for logging in for existing users and one for registering new users. Both will have a similar layout with the form being in the center of the page allowing users to enter the information they provide.

## 2.4 Tech Stack

For this project to be a success I will need to use the following technologies;

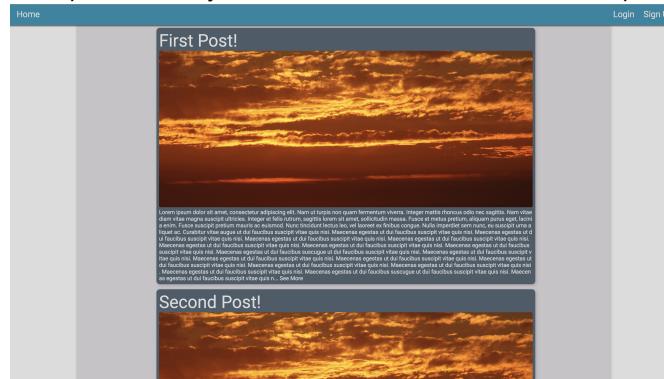
- HTML
- CSS
- JavaScript
- Node.js

## 3 Implementation

The implementation took me approximately 2 weeks to complete.

### 3.1 Homepage

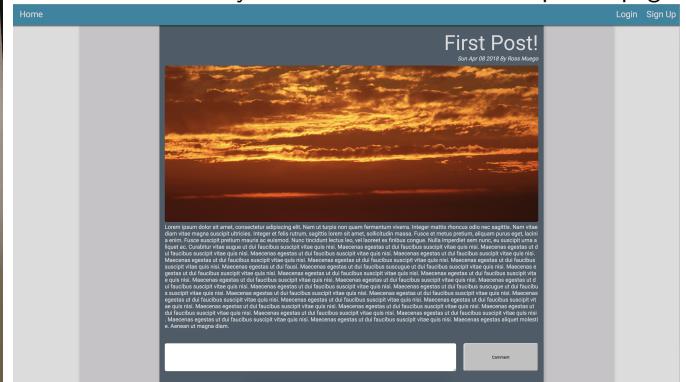
The Homepage of the site is exactly how I imagined it in the planning stages. By clicking on the 'see more' at the bottom of the post you are taken to the page for that post where you can comment and see the full post.



### 3.2 Posts

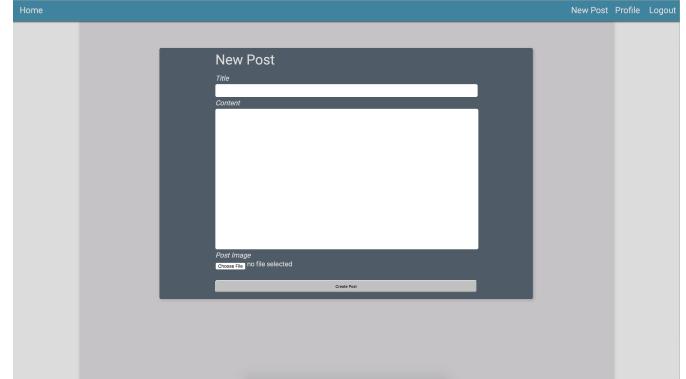
The Post page is also extremley similar to how I envisioned it during planning. At the bottom of the page

there is spaces for comments. I also added under the cover image for the post some details such as who made the post and when they made it. By clicking on the users name you will be taken to their profile page.



### 3.2.1 New Posts

Below can also be seen the form to create a new post. This form can only be accessed if the user is logged in. It requires you add a title to the post, a body of text and an image before you can submit the form.



To generate a random post ID for each post, I execute the following code;

---

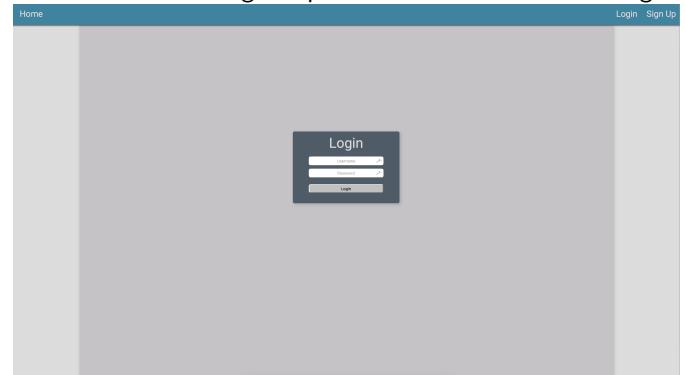
```
1 Date.now() + Math.floor(Math.random() * 100)
```

---

This snippet of code effectively gets the current time in milliseconds and adds a random number between 1 and 100, creating a truly random number. The same process is used to generate unique user IDs

### 3.3 Login/Registration

The login and registration pages of the site are exactly how I had envisioned them looking. They are simply yet get all the information required for a user to sign up for the site or to login.



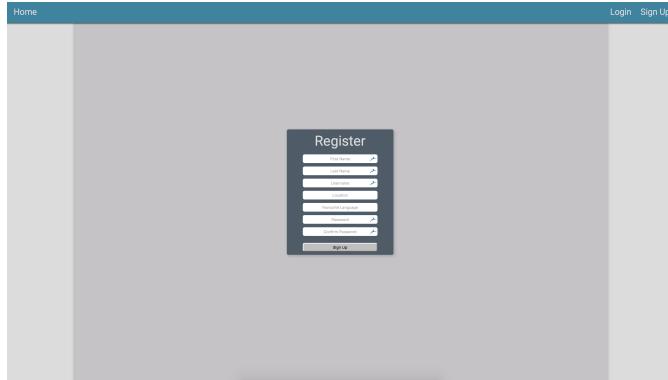
All the usernames and passwords are md5 encrypted so that they cannot be seen in the database in plain text. The server takes the input from the user and changes the password to md5, then queries the database for that username and password combination. If there is a match then the user is redirected to their profile page, otherwise they are directed to the registration page where they can signup for an account.

If a user is successfully logged in, a new session on the server is started with that users Id and their username. The profile pages user usernames as all usernames are unique and the database uses user IDs to store posts and relationships etc.

```

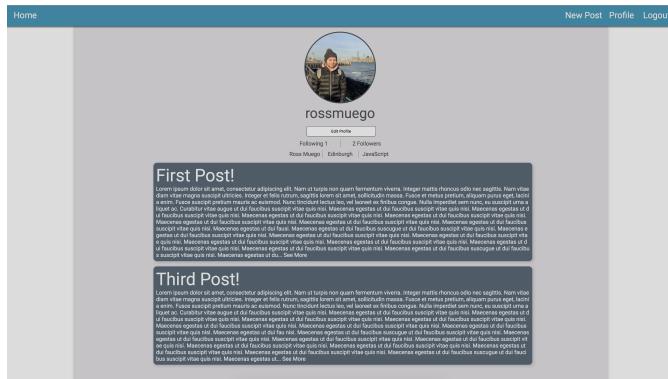
1  var query = {
2    username: req.body.username,
3    password: md5(req.body.password)
4  };
5  dbo.collection("users").find(query).toArray(function(err, ←
6    result)

```



For the registration of users, all fields are also required. The user also has to confirm their password and if these are not correct they are prompted to reenter them. When a user registers, the username they entered is queried in the database of existing users to make sure that is is unique. If it is unique, then the users details are added to the database and the account created. Following that the user is redirected to their profile.

### 3.4 Profile Pages

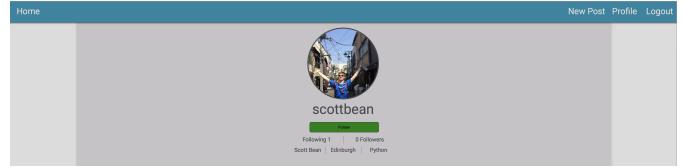


As seen above, I followed the plan for the profile pages exactly and I am happy with the result. You can clearly see the users profile picture, as well as their username and other details displayed underneath the profile picture. There is a button for users to edit their own

profile, and the edit profile page can be seen below.



If the user is on a profile that is not their own, they will be given the option to follow that user. When the follow button is clicked, it will turn into an unfollow button and vice versa. (See Below).



## 4 Critical Evaluation

In regard to the specification set out for this project, I feel I have met the requirements well and delivered upon what was required. I have used node.js to run a server, as well as using express and a node package 'handlebars' to deliver dynamic content rendered on the server. I have used HTML and CSS well in the implementation with all elements working correctly as they should be. I have also included features such as comments and the ability to follow users that have allowed for my solution to be slightly more than the specification has asked for.

In terms of what I could have done differently or implemented, there a few points I could make. I would have liked to have added an edit/delete function for posts, as well as for comments on posts. Another would be a redesign of the sites homepage to maybe provide recommended posts or hot posts on the site, rather than just having a long list of all the posts made. A notification system would be something else I would liked to have introduced, to alert users when somebody they follow makes a post or when one of their posts receives a new comment, along with being notified when they receive a new follower.

## 5 Personal Evaluation

In terms of my own personal evaluation, I feel I have performed well with this task. I have allocated my time well, and have enjoyed researching the new technologies asked of us. I had some knowledge of node.js before but developing this blog has helped me implement what I already knew as well as having to research new ways to do things. I feel my blog is a good standard with a few missing features that could be relatively easily implemented in the future if so desired.