Coursework Report

Darwon Rashid
40280334@napier.ac.uk
Edinburgh Napier University - Web Technologies (SET08101)

1 Introduction

1.1 Brief History

Since the creation of the internet back in the 90s, people have been trying to figure out ways to share information with another across the globe. Initially blogs were considered just personal web pages, but then the term blog came about in 1999, and it stuck ever since. In 1999, there were around 23 blogs, but by 2003 there were around 50 million different types of blogs worldwide. Blogs suddenly became very popular for people to use to sh are their stories and/or other types of information, before Facebook came around, blogs were the closest thing to an idea of a "social-media". They are still popular these days and are used across a very diverse demographic of people.

1.2 Aim

This assignment involves the creation of a dynamic blog website, that allows for adding posts, removing posts, and editing posts. On top of that, it allows for multiple user functionality; users can register an profile, write their own posts, view other user's posts, and even comment on them. Posts are also categorized, so users can view posts that are category specific. Users can also edit their profile or if they wish to, can delete their profile.

1.3 Blog Features

The features chosen for the website are as followed:

1.3.1 Posts

Once a user has logged in, if the user wanted to, they can add a post, edit a post, and delete a post. A post includes a title, category, body, and an optional image. Once a post has been added, it will appear in the home page, and the my posts page. It is in the my posts page, where the user can edit or delete each post they have added. Posts displayed on both the home and my posts pages do not show all of the body, it limits the body text up to five hundred characters. This is so you can view numerous posts at a time. If the user wants to view the entire post, they can click on read more and it will go on the show page and display the post in its entirety. Users can also add comments to each post if they wanted to.

1.3.2 Users

The first step the user has to do to access the website is to log-in, if the user has no profile, they can register and then log-in. A user profile consists of a name, email, user name, password, and optional profile image. Every user name must be unique so no two users can have the same user name. In addition, every password is hashed, so their passwords are secure and safe. Once a user has created a profile, they can edit or delete it, and if they do choose to delete their profile, all their posts will also be deleted.

1.3.3 Categories

A small feature, but a feature nonetheless, users can create categories, and once a user has created one, that category becomes available to be picked by all users when they are adding a post. Categories also allow for users to view posts specific to that category. If a user wanted to, they can view posts that are category specific. This lets users narrow down posts to ones that they are more interested in.

1.4 Background Reading

Background reading for this assignment involved websites such as Stackskills, Express Node, Stackoverflow, and Bootstrap.

2 Software Design

2.1 Initial Plan

Initial design of the website first started on paper to figure out before any code how it should look and function to the user. Custom CSS was applied at first as a placeholder to create basic functionality to see how the website should flow. For navigation, the idea was to have a single navigation bar that had the most important and relevant pages for users to navigate across.

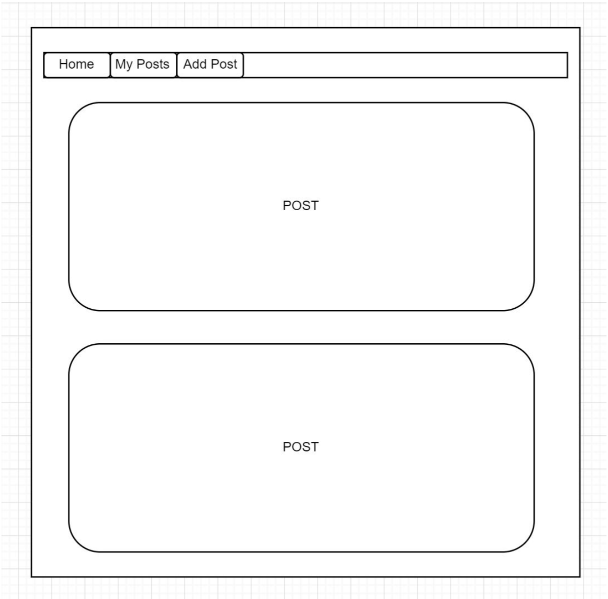


Figure 1: Initial Navigation bar

The idea of having users create profiles came in later after figuring out how to display posts in a way that made it easy for any users to view and navigate. Initially there was no log-in page, but after establishing how posts will be laid out, it was time to add in user-profile design into the website. Initial design of log-in page is as follows.

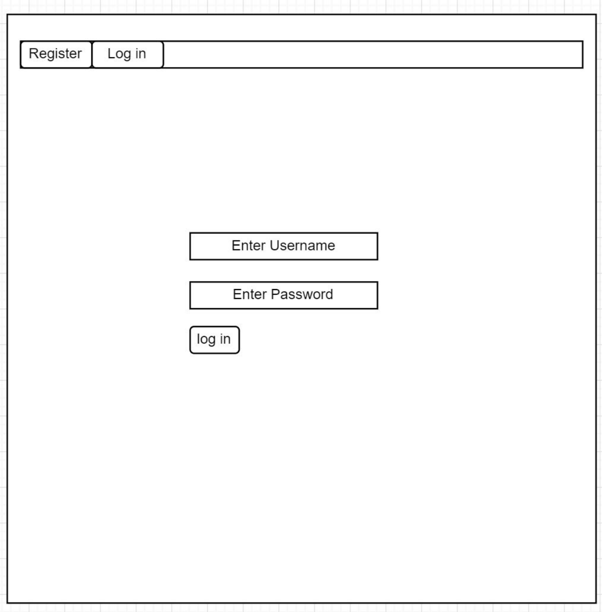


Figure 2: Initial Log-in Page

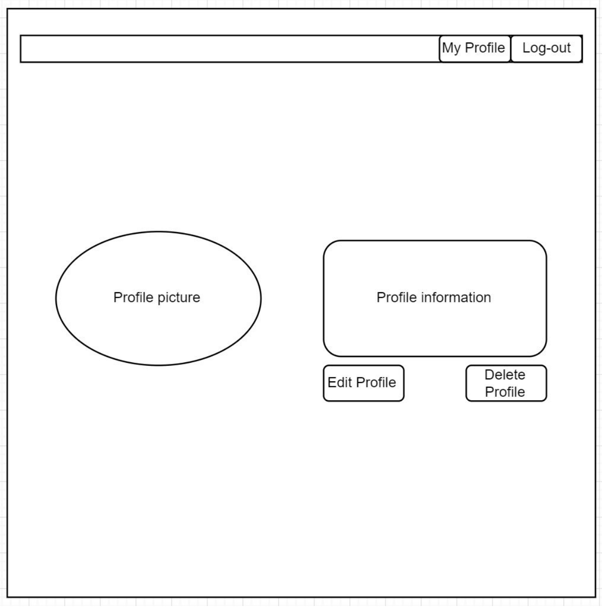


Figure 3: Initial Profile Page

2.2 Design

Once the design of the website was complete, it was time to decide on whether to do all the CSS in-house or go for a CSS framework that is already available for use. The final decision was to go with the popular Bootstrap framework for the style of the website.

2.2.1 Bootstrap

The reasons to go with Bootstrap rather than doing all the CSS in-house are due to the fact that using Bootstrap speeds up the development process. Developers do not have to spend so much time on stylizing but instead focus on adding more features and etc. Bootstrap is reliable, mobile friendly, and is compatible for all browsers. If the decision was to do all the CSS in-house, there would have been no time to develop multiple users functionality.

Going with Bootstrap allowed for no time to think about types of font, colors of buttons, and etc. Bootstrap's approach to design is also very flexible, it's look and feel is compatible with almost anything any developer wants to create. Its design is one most users recognize and know how to use.

2.2.2 Final Design

Later on, navigational bar was redesigned to have links to the home page, my posts page, add post page, and add category page that are positioned on the left side while the right side has the my profile page and log-out link.

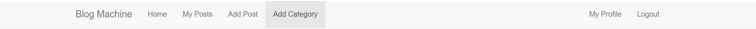
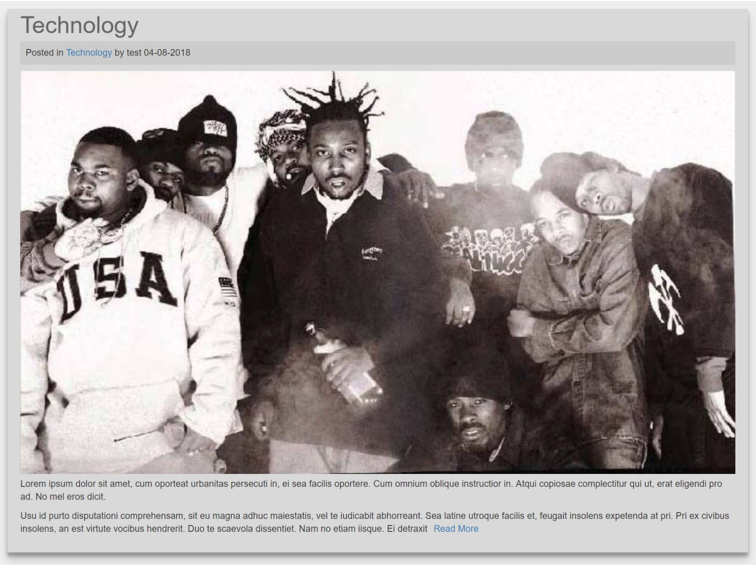


Figure 4: Final navigational bar design

As for posts, each post is contained in a card, so when users are browsing through the home or my posts pages, they can easily tell when a post starts and ends, and once they click on it to view the entire post, the card disappears and only that post is shown, a design queue to signal to the user that they have left the previous page. The colors of the website are simple and are chosen not to distract the user from what's important. The background color and the color of the current active link on the navigational bar are the same for it is a design queue to let the user know what page they are at without thinking about it.



Technology

Posted in Technology by test 04-08-2018

Lorem ipsum dolor sit amet, cum oporteat urbanitas persecuti in, ei sea facilis oportere. Cum omnium oblique instructior in. Atqui copiosae complectitur qui ut, erat eligendi pro ad. No mel eros dicit.

Usu id purto disputationi comprehensam, sit eu magna adhuc maiestatis, vel te iudicabit abhorreant. Sea latine utroque facilis et, feugait insolens expetenda at pri. Pri ex civibus insolens, an est virtute vocibus hendrerit. Duo te scaevola dissentiet. Nam no etiam iisque. Ei detraxit Read More

Figure 5: Post Example

3 Implementation

3.1 Node.JS

Node JS is a popular framework that is used to make web applications, and this website was made using this framework. Node JS supports a lot of different modules that do a lot of different things, because there is a lot of modules, this means developers spend less time building websites. It is also very fast for it is built off Google Chrome's V8 JavaScript Engine. Certain functionality for the website completely depended on certain modules that support Node JS. The most important module that the website completely depends upon is the Express framework. It sets up the entire project and is responsible for a lot of the functionality of the website. Other modules also depend on it. It offers all the HTTP utilities that are used for this website and allows for middleware. Another powerful feature of the Express framework is that it allows for alternative ways of writing HTML for the developer. The template engine used for this website was one called Jade, and it is superior to HTML in the case that it allows for many dynamic functionality. It can use conditional statements, and display JSON objects in all sorts of way, both of which are used heavily in this website.

3.2 Organization

Once the design of the website was planned out, the implementation was next. Since this involved the usage of many different technologies the first step was to organize the files in a manner that is easy for anyone to understand when viewing the source code. Luckily Node JS allows for a good project structure so anyone who views the project can easily see where everything is in a clear manner.

3.3 Bcrypt and Passport

When a user makes a profile, they require an user name and a password for them to be able to log-in to the website. When a user creates a profile, the password of their choice must be encrypted so if there was a breach to the database, their profiles would still be secure. Bcrypt is a node module that hashes the users password so that it gets encrypted. When the user eventually tries to log-in the website needs to check the user's input to see if it is valid or not. To achieve this, the module Passport allows for this functionality. Passport allows you to create a strategy that handles authorization.

Listing 1: Using Bcrypyt to hash password

```

        module.exports.createUser = function(newUser, callback)
        {
            bcrypt.genSalt(10, function(err, salt)
            {
                bcrypt.hash(newUser.password, salt, function(err,hash)
                {
                    newUser.password = hash;
                    newUser.save(callback);
                });
            });
        }

```

3.4 MongoDB

A huge part of the website is its ability to store data once the website has been closed. This was achieved by using a NoSQL Database called MongoDB. Anytime the website grabs posts or any sort of data, it pulls it from the database set up in MongoDB and sends the data in packaged objects called JSON. The modules Mongoose and Monk take care of all communications between the website and database. Mongoose allows for the creation of a Schema, and Monk handles all queries to the database.

Listing 2: Using Monk to Query a user's posts

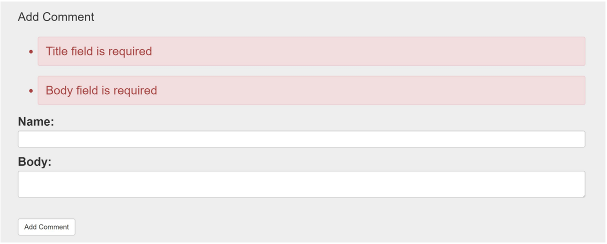
```

        var posts = db.get('posts');

        posts.find({username: req.app.locals.username}, {}, function(err, posts)
        {
            if(err)
            {
                throw err;
            }

            res.render('myposts', { posts: posts, title: "My Posts" });

        });


```

3.5 Express Validator

Another important module that is used in the website is the Express validator. It checks for validation for all the textareas that the user types input to. From validating emails to passwords, the express validator has a wide variety of check methods that can be used to validate user input. It also allows for custom validation that the developer can write to validate for more specific fields. Once it checks for validation, if there are errors, it compiles a list and then displays it to the page by parsing the errors into the jade file for that specific page.

Listing 3: Using Express Validator to validate adding a comment

```

        // Form Validation
        req.checkBody('title','Title field is required').notEmpty();
        req.checkBody('body', 'Body field is required').notEmpty();

        // Check Errors
        var errors = req.validationErrors();

        if(errors)
        {

            var categories = db.get('categories');

            categories.find({},{}, function(err, categories)
            {

                res.render('addpost',
                {
                    errors: errors,
                    categories: categories,
                    user: req.app.locals.loggedInUser
                });

            });

        }

```



Add Comment

- Title field is required
- Body field is required

Name:

Body:

Figure 6: Comment Validation Error

4 Critical Evaluation

4.1 Comparison Against The Requirements

The minimum requirements of this assignment involved making a simple blog website that could add, edit, and remove blog posts.

The website has met the minimum requirements and has a bit more functionality added to it. Instead of having a blog dedicated to one person, multiple users can create profiles and then can add, edit, and remove blog posts. They can view other user's blog posts and even comment on them if they wanted to. On top of this, users can also edit and delete their profiles. The website is also responsive in design, so it can be viewed on mobile devices.

As for evaluation of the website, after allowing some real-life users test the website, they have concluded that it is alright and gets the job done. Other blog websites have more features and look better, but not all allow for multiple user functionality. As for design, it follows Bootstrap's design style, and none of the real-life users said anything negative or positive about it. And that is the point of Bootstrap, no distractions and blends in with the interface.

4.2 Possible Improvements

At the beginning I was really ambitious and thought I can add a lot of features, but then I realized I have no time for adding everything in. There are many improvements that can be made to the website.

For beginners, make the website look better, I do not like how the my profile page looks at all, but I had no time to make it look nice. I wanted to also add a search functionality where you can search for users and/or posts. It would have been cool to have the ability to view other user's profiles too. I thought I had time to add a forgot password feature but nope, so that would have been cool to have as well. Make how posts view also look better, for they look a bit weird.

5 Personal Evaluation

I enjoyed this coursework way more than the previous one for it was developing a dynamic website. I still do not like JavaScript, and I think I never will, but that's fine as long as I can bear with it. This assignment has taught me how to make a web app from scratch, and also how to be dependent on my own for background reading (I barely came to uni). I have learned how other professional websites that are built by Node work, and it is cool. The design aspect I still did not enjoy much, but I must say, working with Bootstrap was a good experience.

Most of the challenges and frustrations I faced through this course work had to do with mostly with JavaScript and its asynchronous nature. Since I have not worked this much with an asynchronous environment, it took a while to get used to the nature of callbacks and returning promises. I am still not fully used to it, but I will defiantly come back to this so I can learn. I needed to get used to all the HTTP methods and how they are called and all the different types and what each does was all new stuff to me, but I think it was worth it.

I am still not head over heels on web, but this is much better than the last coursework, and Jade is a blessing. I will look more into Node JS in the summer hopefully.

6 References

[1] https://stackskills.com/ - stackskills

[3] https://www.w3schools.com/ - W3Schools

[4] https://expressjs.com/ - Express JS

[5] http://jade-lang.com/ - Jade

[6] https://stackoverflow.com/ - Stackoverflow

[7] https://getbootstrap.com/ - Bootstrap

[8] https://cdn.shopify.com/s/files/1/2262/1373/files/wu-tang-logo_7ca3cc3b-9135-435d-9597-e3899ac82f53.png?8958264219539425520 - Wu-tang symbol

[9] https://www.webdesignerdepot.com/2011/03/a-brief-history-of-blogging/ - Information on blogs

[10] https://github.com/ctavan/express-validator - Info on Express Validator