Web Tech SET08101

viana3030

April 2018

1 Introduction

The purpose of this coursework was to create simple blog engine using the following some of the technology languages practised in labs and at home. Within this report details of how the application was design and implemented with be explored and explained.

1.1 Scope of the coursework

The scope of the coursework was detailed as creating a simple blog engine that allowed a user to create, read, update and delete profile or content posted. Within the scope other areas such as usability, functionality and learnablity of the application were to be considered.

2 Research

To better understand how to create and implement a blog, I decided to conduct a mini research on how top blog engines such as wordpress and blogger are created. I started my own blog experience on blogger, to better understand how a person would use a blog engine. I wanted to understand what worked, what didn't and how I could implement one similar with less functionality. log.PNG

`blogger log.PNG`

log.PNG log.PNG This is an image of login page for blogger engine by Google. woodpress.PNG This is the word press landing page which is interactive, suing a combination of languages such as PHP, Javascript, HTML and CSS.

3 Requirements

Understanding my own basic or low level ability in programming I decided that was best to focus on finding online resources that allowed me to practise creating the blog engine as I learnt.

- the ability for the user to create and log-in to the engine
- to edit post created and shared
- to delete post and to keep unpublished post private
- build the blog using Html, css and node.js

4 Software Design

The planning process for the blog involved looking at how servers are built in order to data to be stored and retrieved from it. Looking at the technologies required for blog engines I moved quickly to learn Node.js as it will be the main language to support the blog. although I wanted to avoid building a database, I needed to focus on planning what I wanted to collect from the blog engine. The important data I needed to collect:

- basic user login (first and last name)
- post detail such as title, body, comments and date of post
- image upload
- deleted files and edited posts

4.1 Page flow

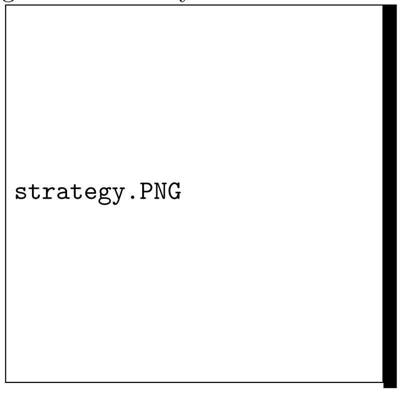
- Log-in page needed to allow the user to Log-in or register was the blog engine
- Posting section to allow the user to add new posts
- Post editing section to all the user to edit, delete or save content for later posts.

5 Blog Engine implementation

The ambition to create a blog engine similar to word press evaporated quickly once I started to learn full indepth course on node.js with mongoose and mongodb. The verity of course available made the process daunting as it fragmented the plan of starting with a database.

5.1 NPM

The creation of the Engine required a mixture of NPM installs, that had to be connected node using bash and window command prompt to load and connect the files within the online server and local host. The tutorial I was coding was was two years old and there were sections that had changed dramatically. For example using the google oauth20 cause multiple crashes.



A couple of stake overflow later and it transpired that I have put the comma in the wrong place.

5.2 Mongoose and MongoDB

Connecting the server to the application was the worst and most difficult experience. This application was deleted, terminated and crashed many times. However the tutorial I later used introduced me to Mlab, was fantastic. The only installation I needed was to add the MongoURI to the keys.js file. Later I was able to store dated I have specified to be collected. This supported to with ensuring that other security elements such as post privacy were honoured.

`mlab.PNG`

5.3 Screen shot of Application

`welcomeName.PNG` `menu.PNG` stories.PNG

stories.PNG stories.PNG `public stories.PNG`

6 Critical Evaluation

6.1 Comparison of requirement

The implementation of the application changed a lot during the implementation stage. This was due to me own lack of knowledge in node. js, mongoose, mongodb and javascript. To avoid disappointment I decided to follow the top tutorials online as I found books to be outdated due to the rapid change of technologies.

6.2 Comparison of other platforms

The blog created is very different to main blog engine available as Its not as colourful and interactive as I would have liked. However the file engine provides the user with the CRUD requirement list in the coursework scope.

6.3 Improvement

The blog engine can be improved by adding functionality such as image upload, font and style selection to allow the user more customisation options. Future improvement will also allow for the user to login without Google. The passports.org offered a large amount of authentication options with over 500 strategies. The final part of the improvement is to deploy the application on mlad server and allow for other users to access the website away for the localhost:5000.

7 Personal Evaluation

The biggest challenge of this coursework has been learning how to link HTML, CSS, JavaScript and nodejs. I found myself feeling frustrated with the lack of clear information on how to implement the blog engine. I usually rely on books to understand a subject, however on this occasion I choose to only use online tutorials as I was pressed for time between work and childcare. I used my little note book to write down all the passwords for the various application support such as mlab, Heroku, google and github. I then proceeded to note down all the keywords that I didn't understand or where my application crashed. Slowly understood how stakeoverflow works to solve issues.

Where I improved, was in using bash within the text editor to activate nodemon to run the application on the local host, to search for files within git or heroku using the command line and I was able to use tools such as materialised css, font awesome, JQuery and google to make the application look nice and achieve the required scope.

8 Reference