A simple Blog Platform

Alba Ghigi Pino
40284061@live.napier.ac.uk
Edinburgh Napier University - Web Technologies (SET08101)

Abstract

Web Technologies is a term referring to the many *programming languages and multimedia packages* that are used in conjunction with one another. Through these packages we can create components and process data to produce the many dynamic web pages that we now see and use everyday.

Keywords – *Web Technologies, API, Git Bash, mongodb, Node.js, js, html, css, bootstrap, Blog platform,Edinburgh Napier University*

1 Introduction.

The aim of this report is to produce an overview on how I created and implemented a simple blog platform with both server and client elements.
This means I had to create a user interface to allow the user to create and upload a new blog post and to delete or remove an existing one (CRUD functionality).

To build this project I implemented HTML, CSS and Javascript for the client interface and Node.JS because it's an open source cross-platform for executing code on the server-side. Node processes incoming requests in a constant event stack and sends small consecutive requests without having to wait for the responses.

2 Implementation

The first approach I took to this project was searching for several videos on YouTube so I could at least have an idea on where to start. After watching a couple videos, I understood that there was more than one way on how to do it, but all similar, so I stuck to a tutorial that to my knowledge had all the basics I needed; the use of the command prompt to connect to the server, how to install and use MongoDB.

First I will install Gitbash instead of using the command prompt, no particular reason why, just preference. Then I'll have to download Node.JS, because as I mentioned before, I'll use it as the server on which my blog will run.

Then I will install express on Gitbash (npm install express). Express is a framework for node that will make writing my code for the server a lot easier.

Angular is going to be implemented in the front-end framework. It's going to be the interface between the HTML and the JavaScript. Angular will be set up in the index.HTML file same as Bootstrap. Bootstrap will give our page a responsive layout, as well as making it look more aesthetically pleasing.

For the database, to store all the data, I'll be using MongoDB which is a NoSQL database, which means it stores it's data in JSON format.

I'll also install mongojs through Gitbash (npm install mongojs) to be able to connect our server to our MongoDB database and to also be able to interact with it, which is the restful API part of the project. Restful API basically refers to how we are going to interact with the data in the database.

Last but not least, we must install another module called Body-Parser so that we are able to send the input data to the server and to also be able to make the console log print the data it receives to the console.

And of course I will be using JavaScript for both the back end and front end frameworks along with HTML, and CSS to stylise the page.

```
1  Download Gitbash.
2  Download Node.js.
3  Download express (from Git).
4  Download Mongodb.
5  Download mongojs (from Git).
6  Download Body-parser (from Git).
```

3 Website Design and Development

3.1 Design and Implementation of features

As I mentioned before, I used a tutorial to guide me through my project, but this tutorial was about creating a "Contact List App". It contained three different fields; the name, email and number of the contact. Because of this reason, In the beginning my blog looked like a contact app stylized by Bootstrap. I now know I shouldn't have, but to change the aesthetics a bit I combined it with CSS.

The blog is very simple, it has the three basic commands which are; add, delete and edit. For the design, I chose to implement three input boxes; one for the Author's name, the second one for the title of the post/publication, and a third one containing the actual post/publication.

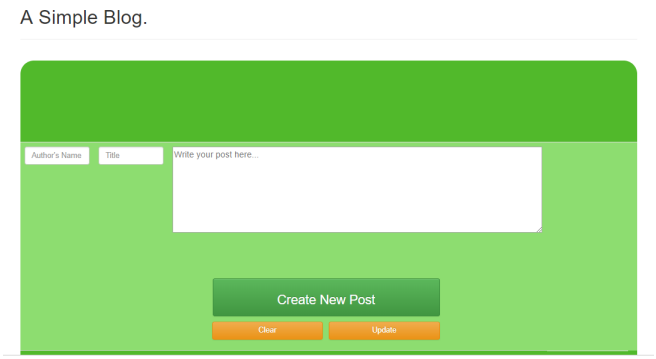

Figure 1: Blog

Figure 2: Blog

Once you have entered these details, you press "Create a new post" and it will save it to the database and appear on the screen visible to the user. Each time you add a new post it will add itself to the publication list below the input area.

I also added a "delete" button to be able to remove the post and an "edit" button. If you want to change your post you press edit and the whole publication will appear in the input boxes ready to be changed. If you want to clear the whole publication without having to create a new one press "Clear", write what you want instead, and then press the "Update" button, it will automatically save as the same entry you created initially but with the new data the user just entered.

4 Evaluation

4.1 Critical Evaluation

The requirements needed for this Coursework were; for a user to be able to navigate between pages and to be able to read the latest blog post and create new ones. My blog not meets the requirements, but it also includes a "delete", "edit", "clear", and "update" button. Unfortunately it only contains one page (index.html), I wanted to create a Log in page but I ran out of time.

Even though it meets all the requirements, if I compare it to other blog platforms, the functions/features a very limited and aesthetically it could be much better.

If I were to do it again, learning from my errors, I wouldn't have combined Bootstrap and CSS together. Because of this, I wasn't able to stylize the blog how I wanted. I would have also liked to increase the number of pages to be able to navigate through the blog platform. One of these pages I would have liked to have is a login page to also increase the security of the blog, and maybe for future projects be able to login with a user and not having to enter the "Authors name" it already recognizes you as a user, so the "Authors name" field would be populated accordingly.

I would have changed the aesthetics of the page. Function wise I wouldn't change it.

4.2 Personal Evaluation

For me the whole Coursework was challenging. However, the most challenging part was where I had to connect to the database and ensure it assigned each entry with a unique ID, which would allow me to find that entry and later carry out the corresponding task.

I also struggled a lot using bootstrap and CSS together. Every time I would change the size of the window everything would move from its position or even disappear from the screen. The problem was that I didn't fully understand what Bootstrap was until I already was far along with the CSS, and when I started changing it to correct it, the blog started failing so I stopped playing with it, but I am very disappointed with the aesthetics of my blog, in that respect.

I've learned more resitting it now in summer than when I started it during the course. Now I know how to create a local server and how to connect to it and its corresponding database to store the data I want to collect. I've learned the basic controls of gitbash and about bootstrap that it is different from css... Being in Spain forced me to do this coursework all on my own and I'm very proud I achieved it, but I know now, with the knowledge I've obtained, I could do it much better.

5 References

Video, 15 Feb 2015.
https://www.youtube.com/watch?v=kHV7gOHvNdkhttps://www.youtube.com/watch?v=kHV7gOHvNdk.