Edinburgh Napier UNIVERSITY

Coursework Report

Cool Student
4008000@napier.ac.uk
Edinburgh Napier University -

Abstract

For Coursework 2 of the web technologies module we have been asked to create an online blog platform, consisting of a server and client, with CRUD capability using HTML, JavaScript and NodeJS. The platform must also be able to persist the data stored within the server using a database system. I decided to have a relatively simple platform with the basic Create, Remove, Update and Display elements using a single html page

1 Design

Features As possible inspiration for the features I wanted to include in my blog platform, I researched and observed other blog platforms available. The main platform, as it is one I am familiar with, was the online platform Tumblr.

Tumblr has many advanced features, such as having secure user accounts, personalisation of blog pages, the ability to upload blog posts in multiple formats such as audio, video etc.(Figure 1)

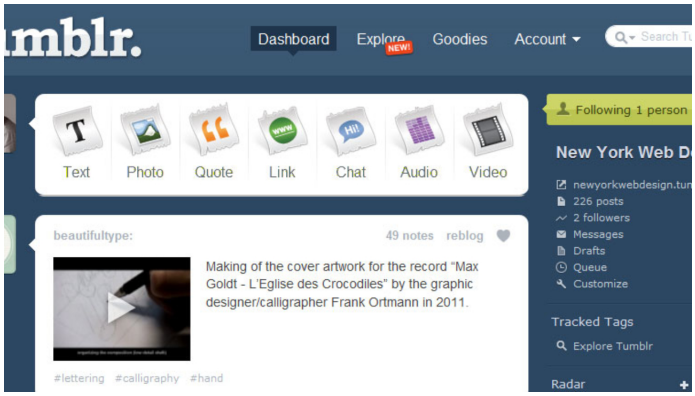

Figure 1: Tumblr Dashboard

However, the extent of Tumblr's features is not practical to apply in the coursework due to the sheer volume of time, and code it would take to make a replica of such a large application

Appearance As a result, I decided to take more inspiration from the aesthetic side of Tumblr. Tumblr employs a simple, graphics based interface to ensure it is easy to manipulate by the user as well as it being visually appealing as it avoids unnecessary clutter which would disorientate the user.

2 Approach and Planning

After some research I found most simple applications that consisted of a blog platform were created using the MEAN stack. The MEAN stack consists of MongoDB, Express, Angular and NodeJS. These four modules work together allowing a cohesive program to be created with more extensive coding features than simply HTML and JavaScript.

2.1 NodeJS

NodeJS is an open-source platform which allows developers to create JavaScript which can interact with a server and serve up HTML element. It also allows the developer to install "modules" through the command line to add extra functionality to their applications e.g the use of MongoDB.

2.2 Express and Angular

Express is the main web application framework used with NodeJS to create web applications. It adds extra functionality to the application along with Angular.
Angular is a framework, mainly developed by Google to enrich HTML applications. It does so by allow two-way data binding which permits dynamic content within the HTML page thus allowing real time synchronisation between the business and presentation layers.

2.3 MongoDB

MongoDB is a NoSQL based data management system intended for use in CRUD applications which wish to persistently store data. Some of its features include but are not limited to: File storage, Indexing and Query operations, all which can be utilised through JavaScript operations.

3 Implementation Self Analysis

3.1 Walkthrough

When implementing the app, I decided for the interface to have a simple single page application which would contain the form through which the user would interact (Figure 2). The form consists of a title containing the name of the platform - Bloggr - and two text areas, one for the title of the blog and one for the blog content. Below the form is the two buttons (submit update) as well as the display area for submitted posts. Through the use of the Angular library I programmed the buttons, using "ng-controller" to allow the HTML to communicate to a separate JS file called "app.js". It is in here that I had the middle man of communication between

Figure 2: Main page of Bloggr

the client and server side elements. The "ng-click" commands would contain a function which when invoked would carry out the task associated with each individual button i.e. Update or Submit. These functions would then pass over to the server side which would then implement the required interaction would MongoDB to store the data persistently, or recall the data from the database. When recalling the data it is then displayed below the form in date order from oldest to newest. From here the user has the capability to edit and delete posts. The interface for this part of the program was allowed through the use of glyphicons, which is included in the bootstrap cdn, from which I gained the CSS sheet for the platform.

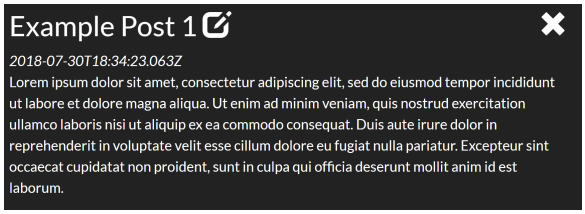

Figure 3: Blog Display Area

4 Analysis Conclusion

While simple in design and functionality, I am very pleased with the end-result from this coursework as, given my limited experience with coding in JavaScript, I have still managed to create a functional application which satisfies the CRUD criteria. I also found that by using other node libraries such as angular I have seen a more in-depth presentation of the capabilities of coding in JavaScript and the poly-functionality the language possesses.
If I were to do it again (which I hope I do as I've discovered a fondness for web applications) I would definitely add more features, such as having security on post adding and deleting as well as having user accounts along with that security. I would also have liked to have added the ability to add more than text to the posts such as image/video/audio files or permitting a rich-text format when creating the posts.