|  |
| --- |
| Photo displaying partial image of two pie charts on a canvas-textured page |
| WEB-TECHNOLOGIES COURSEWORK 1REPORT  Ruaraidh Curran ‘40209895’ |
| |  |  |  | | --- | --- | --- | | ruaraidh curran | 4/4/18 | Web-Technologies | |

Contents

[Node.js Blog Report 2](#_Toc510725768)

[Introduction 2](#_Toc510725769)

[Software Design 2](#_Toc510725770)

[MongoDB database 2](#_Toc510725771)

[Node.js Modules 3](#_Toc510725772)

[Node.js and JavaScript 3](#_Toc510725773)

[Navigation-bar 3](#_Toc510725774)

[Website visual design 4](#_Toc510725775)

[Implementation 4](#_Toc510725776)

[Critical Evaluation 5](#_Toc510725777)

[Personal Evaluation 5](#_Toc510725778)

[References 5](#_Toc510725779)

[Appendices 5](#_Toc510725780)

# Node.js Blog Report

## Introduction

For this coursework I have been tasked with creating a simple code based responsive blog using node.js. This simple blog needed to have the key features of posting new blogs, editing previously posted blogs and deleting blog posts. Using the previous knowledge from the last coursework in JavaScript, css and html as well as using a couple new languages and applications such as express, node.js and mongodb I was able to achieve this. This report will be alongside my code files as well as a README.txt file that will explain how to successfully launch my webpage on other machines. I believe have been able to create a Semi-Professional looking blog site that has met all the required functionality for the coursework hand in. This report I will cover any external learning sources that I used to come up with my web-site (which will be held in References), my software design, Implementation, critical evaluation followed by my personal thoughts on my final creation.

## Software Design

In the early stages of my design I realised that this blog was going to require at least 3 different pages with maybe a couple dynamic pages as well (for editing and viewing individual posts). I found my last website I made for coursework one quite appealing so I used the design concept as that. Although implementing this design through node.js and express to create a responsive website caused some difficulties that I had to overcome.

### MongoDB database

From the beginning of creating the website I knew that I would need a database for the Blogs to be stored on. I knew that these blogs needed to include multiple variables such as Title, Auther, Post and a timestamp. I decided that mongodb was the best way to create and implement a database for my website. To start off I created the database through the command line namesd “blogdb” and then continued to add an entry for testing in the websites early stages of development.

When telling my website to access the mongodb database I simply added the mongoose module [[1]](#footnote-1) and using the function mongoose.connect and then directing it to the database and storing that in a global variable simply called db.

### Node.js Modules

During the whole creation of this website I needed to use a number of extension modules to allow me to carry out a number of tasks ranging from saving and fetching from the database to creating adaptive pages with different html links (for individual posts). For example the body-parser and json[[2]](#footnote-2) which both allow the proper use of the ajax function which I used to gather the ID# for each post and match it to the corresponding entry in the database to create individual pages for each post as well as delete each post. In the later stages of creating this app I tried to integrate responsive error messages for the user using express messages, connect flash and express-sessions, but unfortunately I wasn’t able to get these functions working properly. I researched finding that these modules are easiest used when using a view engine alongside for example moustache or handlebars but I personally enjoy creating my own view on a website using a .css stylesheet. I didn’t want to compromise individuality for the stereotypical layouts of well used view engines.

### Node.js and JavaScript

To create a web-based responsive website we were shown during the practical’s the proper use of a java engine called node.js. through-out the creation of the website I regularly needed to go out from lab workbooks and go research from external sources on how to carry out some of the more advanced tasks [[3]](#footnote-3). Some of the more simple actions like the app.use, app.get and app.post was used during the creation of my website allowing the user to request and get responses and to create/ edit posts from the server side to be posted onto the users side html page. Other than the node.js there wasn’t much generic JavaScript needed to create my final web-site. The only JavaScript I needed to hardcode was the Ajex request that I needed to create for the delete function that I created for each individual post.

### Navigation-bar

I found that the most popular and easiest way to navigate through majority of websites is by using a navigation bar located at the top of my website. Luckily I had already designed a functional navigation bar that I adopted slightly for the use in this website. It only needed a couple of links between the views. Using href’s within an <a> tag to link the buttons to the appropriate pages. I tried to integrate a seamless feel to the navigation bar by using a slight fade effect when the user hovers over each button, which gives feedback to the user letting them know that they are in fact hovering over a usable button.

### Website visual design

When going over the design of my website in my head during the early stages of development cycle. I always imagined it being a bright website with a lot of whites, blues and yellows. I believe that this is probably because I was taking inspiration from other more popular blog sites. But as I started to actually code and got into the later aspects of css I decided against the bright backgrounds and stereotypical blog page colours. I found that having a dark theme to a website gives it a much more professional look, with minimal distraction away from the main features of the blog.

I enjoyed playing around with the alpha in some parts of the website. I wanted it to feel like it wasn’t split up into different sections and that everything flows seamlessly with each other. I believe that I was able to create this by adding a slight transparency into each part of the webpages. Allowing the users focus to still be where it needs to be but without it being an eyesore. I complimented this alpha with a simple white color text with the font family of “Lato” sans serif. I believe that the combination of these have created a quite appealing looking site.

## Implementation

During the implementation of this program the first thing that I had to get up and running was the mongodb server. I first started by downloading mongodb[[4]](#footnote-4) and creating a database file in my C drive. I then continued to create a database folder and a log folder. I added a simple database to the “db” (database) for which was going to house by blog’s database. Which was simply called “blogdb” I used the command line tool running as admin to create this database. Once directed into the mongodb folder I placed on my c drive I navigated to the bin folder and created a directory path using the following command line script “—directorydb –dbpath C:/mongodb/data/db –logpath C:/mongodb/log/mongo.log –logappend –install” which created the directory’s I needed for my database. I then simply added in the command line “net start mongodb” to start the mongodb service to run in the background. Following that I actually created my database by using the line “use blogdb”. I was able to add a couple test entries into this database through the command line to allow for testing during the creation of the website.

Before I started coding my website I needed to create a folder to house everything. After creating the folder I got the command line tool and directed it to the project folder to allow for easy module installation and server updates.

One thing I found quite irritating when coding and testing my page was the fact I constantly needed to stop and reload the server to allow me to see any visual changes I made to the web-site via the javaScript code. I was able to find a way around this though by using the “Nodemon” which was easlily installed through the command line by typing “npm install nodemon” which allowed the server to automatically restart each time after I updated my code.

## Critical Evaluation

When comparing my website against the requirements set out my coursework descriptor I know that I have created a website with all the functionality asked for. And I believe that I have also done this in a very visually appealing way. With the use of multiple pages for displaying posts and editing posts. I believe the integration of dynamic pages has shown my ability and skill with javascript, express, node.js and html. I found myself constantly learning throughout the process of creating this website. and I believe that my creation shows off quite well my abilitys. Although with some more time for research I would have liked to create a messaging system for the users for when they create/edit/delete a post, where an information box will display on the relevant page telling the user what they have just done. Alongside that I would have liked to include a User register/login system that recorded users names and passwords to allow for log-ins. With this I would have created the website to only alloy registerd users to edit and delete their own posts. I believe a module I would need to use for this is express-validation, express-passport and express-session. But unfortunately I was not able to get these to work properly before the hand in date.

## Personal Evaluation

From Start to finish of this module I have been constantly learning. I have found that I actually quite enjoy working with web-pages. Depending on how my marks go in this module and with me taking the advanced web-technologies module next year it could quite possibly be something I would like to go into as a career obviously with a lot more studying to go underway before-hand.

## References

## Mongoose module: Internet Accessed. 01/04/2018 <http://mongoosejs.com/>

* Body-Parser Module: Internet Accessed. 01/04/2018 <https://github.com/expressjs/body-parser>
* Mongodb : Internet Accessed. 01/04/2018 <https://www.mongodb.com/download-center?jmp=nav#atlas>

1. Link for mongoose module git page added in References [↑](#footnote-ref-1)
2. Source link given in references. [↑](#footnote-ref-2)
3. All external learning material will be included in the references [↑](#footnote-ref-3)
4. Download link will be help in References [↑](#footnote-ref-4)