Coursework Report

Cristian Anastasiu
40280345@napier.ac.uk
Edinburgh Napier University - Web Technologies (SET08101)

Abstract

Simple, one page, blog that allows users to post and delete blog posts, storing them in a local database.

Keywords – blog, website, nodejs, mongoose, mongoDB, express, post, bootstrap, angular

1 Introduction

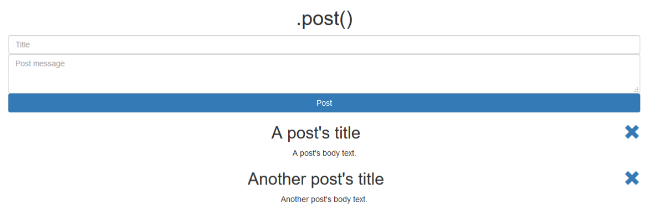
For this coursework we were asked to create a blog website that allows users to post. The approach was a rather simple one, constraining everything to one HTML page. The design was supplied by Bootstrap and the basic commands by Angular.

2 Software Design

The approach was easy to understand, there are just two .js files, server.js and app.js, and one index.html page. The CSS has been supplied by Bootstrap as it is free to use and gives the web page an impeccable design.

3 Implementation

The user interface was set to be a minimalist one. Each post that is added is appended to the list(at the bottom) of posts. For usability, not only when the page is loaded the list refreshes, but also every time a new post is added or removed.



.post()

A post's title

A post's body text

Another post's title

Another post's body text

Figure 1: **The index page user interface**

3.1 Header Section

The header section, as mentioned before, is referenced here. Both Angular and Bootstrap are linked here, as well as the App.js file which handles deleting and creating/storing posts.

Listing 1: Head referencing

```
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.5.7/angular.min.js"></script>
    <script src="app.js"></script>

```

3.2 Server.js

The server uses it's own API, simple as it's use is limited, though the notations are descriptive and should be easy to understand.

Listing 2: Server.js API

```
app.post("/api/blogpost", createPost);
app.get("/api/blogpost", getPosts);
app.delete("/api/blogpost/:id", deletePost);

```

4 Critical Evaluation

The developed website does include the base features requested. In order to post it does not require an account, though anyone could delete the posts. A server.js file was developed, and the HTML is easy to understand and navigate, as it is not packed with features.

4.1 Possible Improvements

Because the approach was so simple, there are loads of possible improvements that can be added. One of the most basic feature has to be adding the date and time each post was added, simply because it's already stored in the blog post database. Another signature one could probably be adding an author to each post, just like most blogs do. Though there could be loads more, such as creating threads, having an index page that allows users to register, adding comments to posts or up-voting/down-voting posts.

5 Personal Evaluation

While finishing this coursework I have gathered information regarding Angular and Bootstrap, services I knew about but never actually knew how to use. Creating the database was quite easy to understand, as I have worked with JSON before, though using NodeJS was a whole new experience. It required external reading, most of the information being supplied by Google Sheets. Finding information was not a challenge as there are plenty resources online, though understanding NodeJS, especially the Mongoose and Express packages, was not an easy task.