Edinburgh Napier UNIVERSITY

Coursework Assessment 2
CRUD Blog

Oliver Thornewill von Essen
40210534@napier.ac.uk
Edinburgh Napier University
Web Technologies (SET08101)

9 April 2018

Abstract

The goal of this project is to design and implement a simple blog platform. It will compromise server and client elements.

The client element must present a user interface, enabling at least one user to add a new blog post, to edit or view an existing blog post, and to delete an existing blog post. The server element will persist data related to the blog, will serve up the user interface, and will also provide a create, read, update, delete (CRUD) API that the client element will utilize in providing the blog's features.

This project must select and apply appropriate technologies aiding it to reach the goal. Specifically HTML, CSS & JavaScript must be used for the client interface, while Node.JS must be used on the server. Additional libraries including Angular.JS and Express have been used.

Contents

1 Introduction

Introduction: The scope of this project is to build a RESTful blog platform hosted through Node.JS that will allow at least one user to Create, Read, Update, Delete(CRUD) blog posts.

2 Software Design

This application is going to be produced through Express meaning that will be dynamic pages. The advantage of a dynamic page is that it uses templates allowing the pages to look like they are in unison. Angular.JS allows one to populate the various pages with different content

2.1 The approach to creating this website

This website is going to be created using Angular.JS, and Express which are both modules of Node.JS

2.2 Node Libraries Used

2.2.1 Angular.JS - https://docs.angularjs.org/guide/introduction

AngularJS is a "structural framework for dynamic web apps". It enables templates to be created using HTML and further extents HTML's syntax allowing one to express the blog's components in a clear way. Not every app is suitable to be using Angular.JS as flexibility gets diminished through a higher level of abstraction (breaking down of components) to the developer. However, as AngularJS was produced for CRUD applications, it is suitable for this blog platform.

2.2.2 Express - https://www.tutorialspoint.com/nodejs/nodejs_express_framework.htm - https://expressjs.com/

"Express is a light framework for Node.JS providing a robust set of features for web and mobile applications"(express home page). Features include middlewares when responding to HTTP requests, routing tables based on the URL, dynamically render HTML pages (tutorialspoint).

2.3 Sketches of design

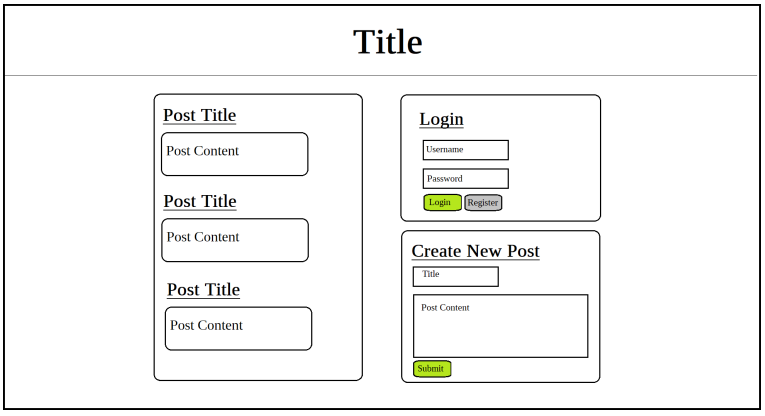

Figure 1: Initial sketch for home page

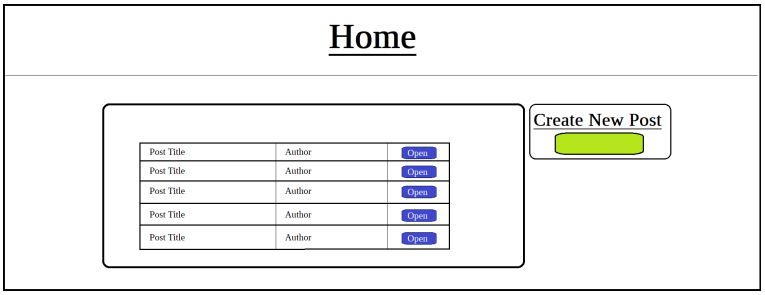

Figure 2: Redesign of the home page

2.4 The requirements to be fulfilled

- At least one user
  - Administrator User
    - Create posts
    - Update posts
    - Delete posts
  - Other Users
- Consistent visual design such that the pages look similar varying in content. Among others, examples include the header.

3 Implementation

Some text on implementation

4 Critical Evaluation

4.1 Checklist against plan:

☑ Item

☑ Sub Item

4.2 Possible improvements:

The improvements to be taken into consideration are:

4.2.1 Login System

The current implementation only allows for one user. This restricts the usability as it prevents multiple users from submitting their own content. A very critical issue with the current log in system is that the password **is not hashed**. The administrator user that exists is hard coded into the login page on the database side. When the user clicks "Login" the server simply checks the user input string against the one stored and if it is the same then permission is granted to the dashboard page. This is why there is no "Register User" page and also means that the administrator cannot change their password unless they have access to the server source code.

4.2.2 Search bar

In a situation where there are many blog posts on the platform, there is currently no way to search or sort the posts. At a lower scale implementation of the blog platform this is not so much of an issue to scroll through 10 posts to find a blog post. If the platform were expanded to allow multiple users to log in then many more posts would be created and such a sorting / searching mechanism would be required.

4.2.3 Commenting

Expanding on the search bar possible improvement, a system that would allow users to comment on the separate blog posts would be nice. In order to achieve this, the login system must be further developed in order to enable users to register an account in order to comment.

4.2.4 Others

Placeholder in case I think of anything else

5 Personal Evaluation

This section will take the opportunity to not only reflect on what I learned but also the challenges I faced, the methods I used to overcome these challenges, and how I feel that I performed.

5.1 Lessons Learned

Through the use of HTML, CSS, and Javascript I was able to make an interactive website from scratch. Examples of lessons learned include section off items inside the browser window, allow buttons/cursor to change their appearance when hovered on, create classes in CSS, or use JavaScript to build ciphers.

5.2 The Challenges that I Faced

Things about Angular, Node.JS, Express, Handlebars.

5.3 The Methods that I used to Overcome the Challenges

Pretty much YouTube

5.4 Evaluation of my Performance

VV Good