CS312-001

Aug. 8th, 2023

Ian Dennis

Jacob Olson

Gavin Russell

Phase II Report

Deliverables:

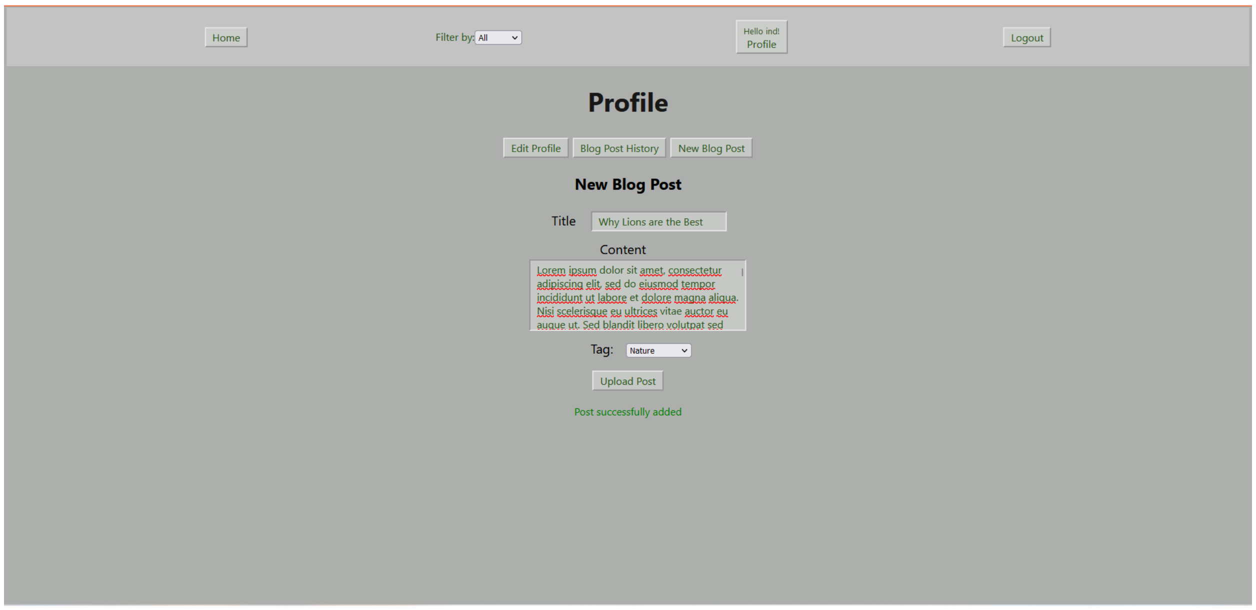
Below are the tasks accomplished during the Phase II development cycle of this project.

Backend Implementation Work

In Phase 1, the backend work included login, signup and cookie functionality. In Phase 2, we implemented all post functionality, such as adding and deleting posts, displaying the posts on the main page with different filletring options such as by tag or showing all the posts, allowing the user to see their profile page and edit their info, displaying the users post history in their profile, and adding comment functionality such as adding and deleting comments. Adding and deleting the posts were simple, we created a new database for posts and when the user filled out the create post form in their profile it was added to the database. When a user deletes a post, the ID of the post is sent to the database for it to be deleted. Both of these actions were done by a http request. Displaying and filtering were just as simple. When a user loads on the main page, a request is sent to the server which gets all posts in the database and sends them back as a list to the webpage. Filtering is the same process, but except for finding all posts it finds all posts with a specific filter. The same method was again done for displaying the users post history; filtering by username. The profile page was implemented which simply takes the cookie token name which is their username and fetches their info in the database. The user can then edit their profile info on the profile page. Comment functionality was implemented on the back end along with the update functionality for blog posts. This is because the database stores the comments as a field of the post to maintain association between post and comment. When a new comment is created, the server will receive a request to update the post that holds the comment. The request holds all the data for the updated post, meaning that the server can update the post document in the mongo database with the new data from the request. After processing the request, the database will have the updated post with an additional comment, and a success message and status code is sent back to the frontend client.

Frontend Implementation Work

While Phase I of the project focused on setting up the website's navigation and skeletal structure, Phase II focused on implementing the features laid out by this structure. After Phase II user's are able to edit their personal profile information and update it in the server. They can also create new blog posts and assign them a tag of their choosing (shown below). All created posts are rendered on the home page by default. Alternatively, the posts can be filtered so that only posts relating to a tag are rendered.



New Blog Post

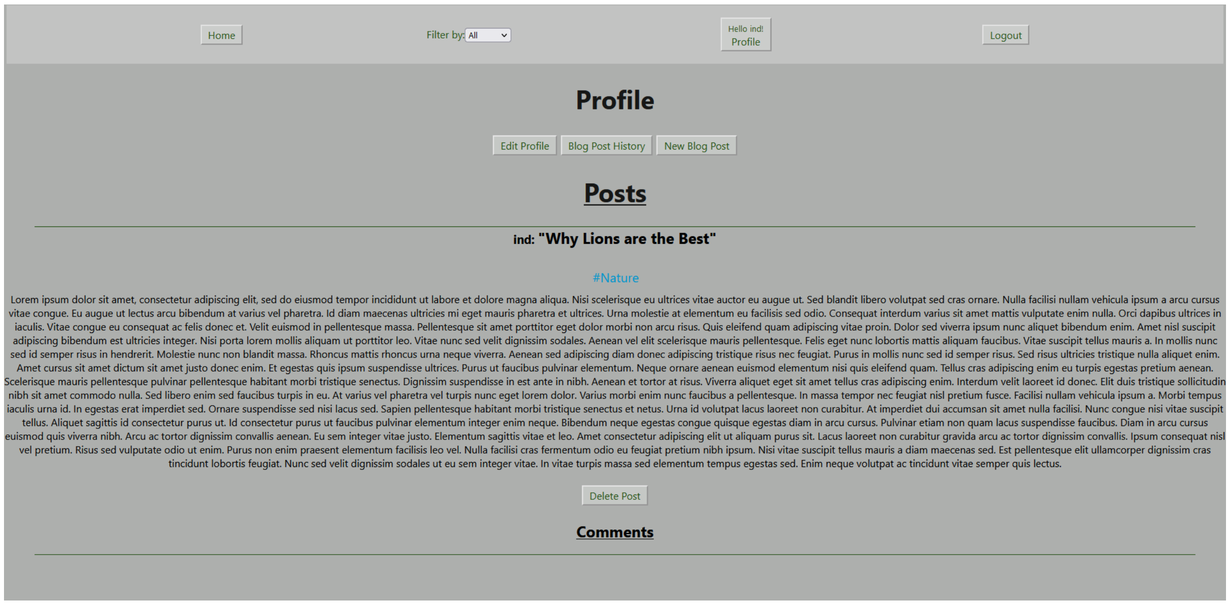
Title: Why Lions are the Best

Content: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Nisi scelerisque eu ultrices vitae auctor eu augue ut. Sed blandit libero volutpat sed

Tag: Nature

Post successfully added

One feature present in Phase I that was dropped during Phase II development was filtering posts by username on the home page. It was decided that this didn't make much sense in the spirit of the website. However, the same filtering technique was used to display user's post history in their profile page. This page was where users could read exclusively their own posts, along with any comments attached to the post. This is also where they could choose to delete their posts as desired.



Profile

Posts

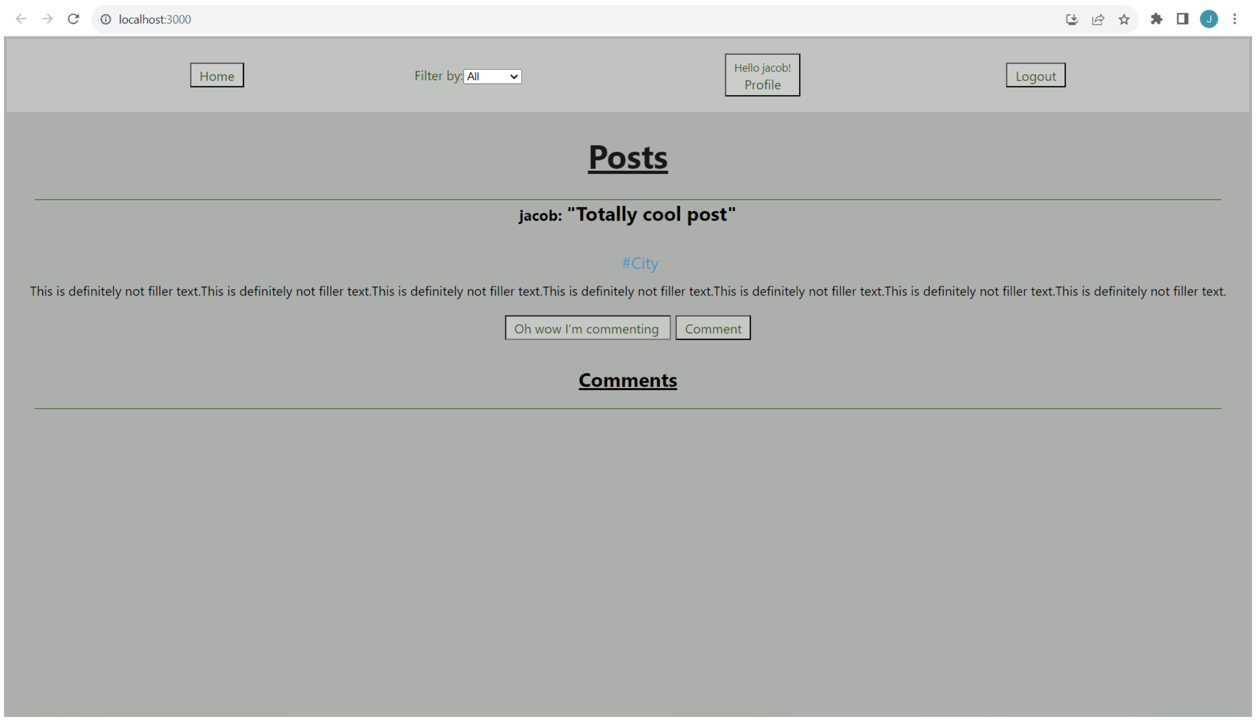
ind: "Why Lions are the Best"

#Nature

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Nisi scelerisque eu ultrices vitae auctor eu augue ut. Sed blandit libero volutpat sed cras ornare. Nulla facilisi nullam vehicula ipsum a arcu cursus vitae congue. Eu augue ut lectus arcu bibendum at varius vel pharetra. Id diam maecenas ultricies mi eget mauris pharetra et ultrices. Urna molestie at elementum eu facilisis sed odio. Consequat interdum varius sit amet mattis vulputate enim nulla. Orci dapibus ultrices in iaculis. Vitae congue eu consequat ac felis donec et. Velit euismod in pellentesque massa. Pellentesque sit amet porttitor eget dolor morbi non arcu risus. Quis eleifend quam adipiscing vitae proin. Dolor sed viverra ipsum nunc aliquet bibendum enim. Amet nisl suscipit adipiscing bibendum est ultricies integer. Nisi porta lorem mollis aliquam ut porttitor leo. Vitae nunc sed velit dignissim sodales. Aenean vel elit scelerisque mauris pellentesque. Felis eget nunc lobortis mattis aliquam faucibus. Vitae suscipit tellus mauris a. In mollis nunc sed id semper risus in hendrerit. Molestie nunc non blandit massa. Rhoncus mattis rhoncus urna neque viverra. Aenean sed adipiscing diam donec adipiscing tristique risus nec feugiat. Purus in mollis nunc sed id semper risus. Sed risus ultricies tristique nulla aliquet enim. Amet cursus sit amet dictum sit amet justo donec enim. Et egestas quis ipsum suspendisse ultrices. Purus ut faucibus pulvinar elementum. Neque ornare aenean euismod elementum nisi quis eleifend quam. Tellus cras adipiscing enim eu turpis egestas pretium aenean. Scelerisque mauris pellentesque pulvinar pellentesque habitant morbi tristique senectus. Dignissim suspendisse in est ante in nibh. Aenean et tortor at risus. Viverra aliquet eget sit amet tellus cras adipiscing enim. Interdum velit laoreet id donec. Elit duis tristique sollicitudin nibh sit amet commodo nulla. Sed libero enim sed faucibus turpis in eu. At varius vel pharetra vel turpis nunc eget lorem dolor. Varius morbi enim nunc faucibus a pellentesque. In massa tempor nec feugiat nisl pretium fusce. Facilisi nullam vehicula ipsum a. Morbi tempus iaculis urna id. In egestas erat imperdiet sed. Ornare suspendisse sed nisi lacus sed. Sapien pellentesque habitant morbi tristique senectus et netus. Urna id volutpat lacus laoreet non curabitur. At imperdiet dui accumsan sit amet nulla facilisi. Nunc congue nisi vitae suscipit tellus. Aliquet sagittis id consectetur purus ut. Id consectetur purus ut faucibus pulvinar elementum integer enim neque. Bibendum neque egestas congue quisque egestas diam in arcu cursus. Pulvinar etiam non quam lacus suspendisse faucibus. Diam in arcu cursus euismod quis viverra nibh. Arcu ac tortor dignissim convallis aenean. Eu sem integer vitae justo. Elementum sagittis vitae et leo. Amet consectetur adipiscing elit ut aliquam purus sit. Lacus laoreet non curabitur gravida arcu ac tortor dignissim convallis. Ipsum consequat nisl vel pretium. Risus sed vulputate odio ut enim. Purus non enim praesent elementum facilisis leo vel. Nulla facilisi cras fermentum odio eu feugiat pretium nibh ipsum. Nisi vitae suscipit tellus mauris a diam maecenas sed. Est pellentesque elit ullamcorper dignissim cras tincidunt lobortis feugiat. Nunc sed velit dignissim sodales ut eu sem integer vitae. In vitae turpis massa sed elementum tempus egestas sed. Enim neque volutpat ac tincidunt vitae semper quis lectus.

Comments

This phase also included creating capabilities for comments, giving the users the ability to add on comments beneath any post (shown below).



Posts

jacob: "Totally cool post"

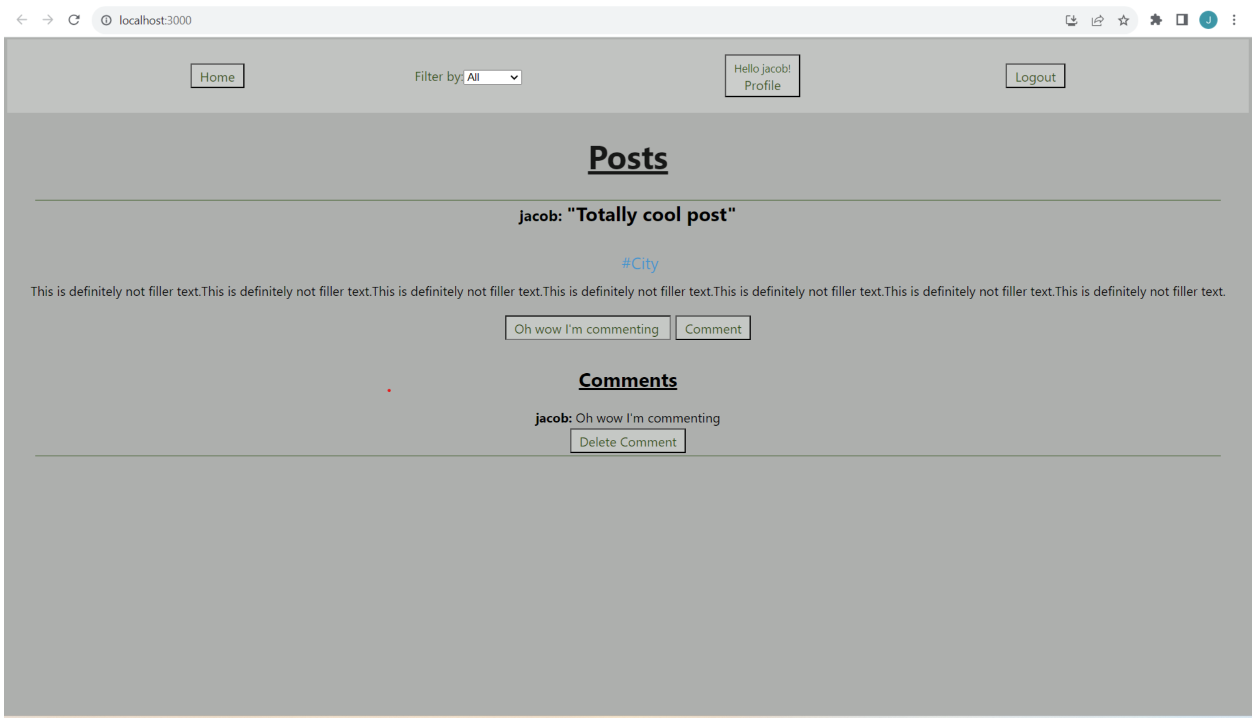
#City

This is definitely not filler text.This is definitely not filler text.This is definitely not filler text.This is definitely not filler text.This is definitely not filler text.This is definitely not filler text.This is definitely not filler text.

Oh wow I'm commenting

Comments

The user can click the comment button once they are happy with the text, and the comment will be added as part of the post data, and rendered with it onto the page.



Posts

jacob: "Totally cool post"

#City

This is definitely not filler text.This is definitely not filler text.This is definitely not filler text.This is definitely not filler text.This is definitely not filler text.This is definitely not filler text.This is definitely not filler text.

Comments

jacob: Oh wow I'm commenting

Now in the comments section the user can see their comment displayed below the post. The comment functionality is achieved by mapping through a list of comment data objects that are in the post's state data. In the map we render a comment component

Challenges and Moving Forward

Phase I of this project primarily consisted of technical challenges for the team while they learned how to integrate a complete MERN system properly. Phase II, on the other hand, taught the team the necessity of proper planning and standards prior to diving into implementation using such a system. The team found that they all went about implementing the same features on the site in wildly different ways. While each methodology worked well enough, they also led to a great deal of time and energy being wasted trying to make three separate visions for the React structure compatible. In some cases the team was able to overcome this challenge by sitting down and talking through the problems/bugs until a working solution emerged. In other cases however, the team found they're implementations to be too drastically different and they weren't able to fully implement some features on the given timeline. The biggest of these features being the ability for users to edit comments and blog posts.

If this project were to continue, including user editing abilities would be the next step in the process. First however, the team would sit down together and create much more strict and comprehensive guidelines for how they would approach building the site. These guidelines would include a more thoroughly drawn out system architecture than was originally created as well as more specific details for coding conventions including variable names and commenting.