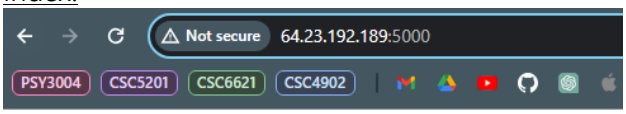
Xavier Robbins
CSC 5201
Final Project
5.6.24

Github:
https://github.com/robbinsx/csc5201_final_project/tree/master

Application Description :
The application built for this project is a Twitter copy-cat. The application provides functionality to post, look at your feed, explore all posts created by all users on the platform, and follow or unfollow users. The application is built in flask with a MySQL database run remotely in a separate docker container. The overall application and associated containers are composed with docker compose and then hosted through digital ocean on a droplet server. All endpoints, except index, login and register, require a user to be logged into an existing account.

Application Demo :
Index:

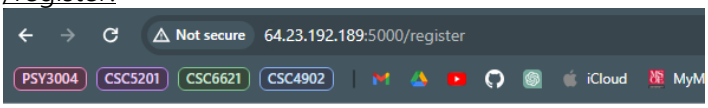


Welcome to Social Media

Login Register

Basic index landing page for the application, allowing a user to log in or register for an account.

/register:



User Registration

Email:

Username:

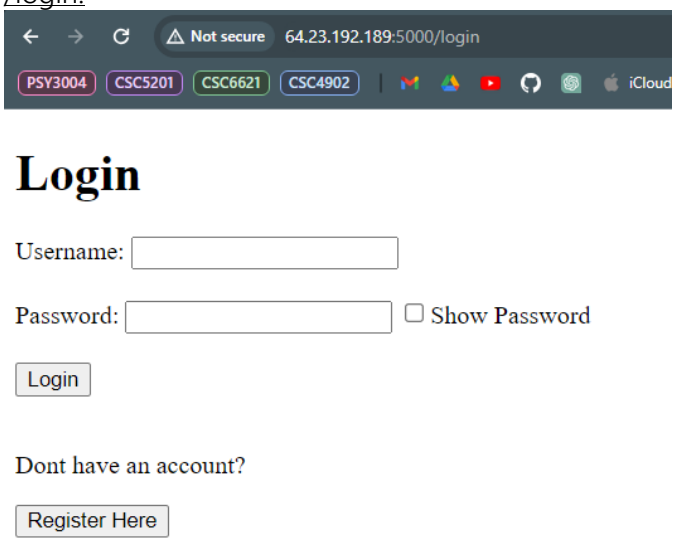
Password:

Confirm Password: Show Password

Register

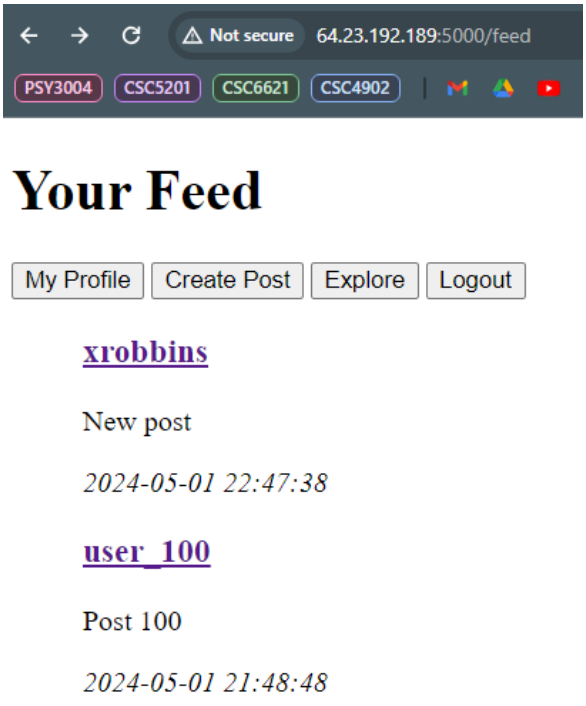
Registration page for the application, requires a valid email address, unique username and a password, which must be confirmed twice.

/login:



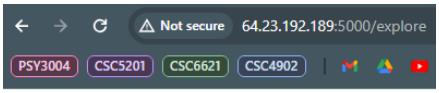
Login page for the application, allowing users with an existing account to login to app. Users who successfully register for an account are redirected to this page to login with their new credentials.

/feed:



Page for users feed, showing posts from the current user (in this demo xrobbins) and other users that the current user follows. This is the landing page for a user after they login. From here the user can go to their profile, create a post, see the explore page, or logout. The user can also select a user from the feed to see their account.

/explore:



Explore

My Profile | My Feed | Logout

xrobbins

New post

2024-05-01 22:47:38

user_100

Post 100

2024-05-01 21:48:48

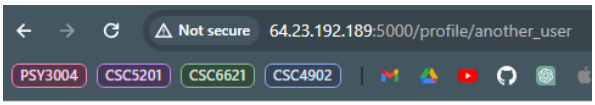
another_user

Hello everyone!

2024-05-01 21:45:52

The explore page is shown above, which shows the current user all posts made on the platform. From here the user can see their account, their feed, logout, or click on a user's username to see their account (clicking on the users own username will take them to their profile).

/profile/<username>:



another_user's Profile

Back to My Feed

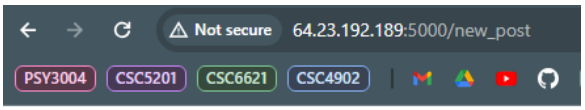
Unfollow

Hello everyone!

2024-05-01 21:45:52

This shows an example of what is seen when clicking on a users profile from the explore page (or the current users feed). If the user is following the user they select, an unfollow button is shown – which will unfollow the user and remove their posts from the current users feed (using the /follow endpoint, though this endpoint does not render an actual web page). If the user is not following the user the select, a follow button will be shown. This will add the user to the list of accounts the current user is following and add their posts to the current users feed.

/new_post:



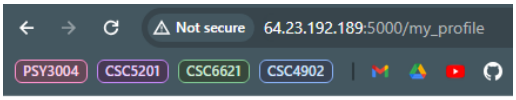
New Post

Content:

Post

This endpoint allows the current user to make a new post, the user can type their post into the content box and post to their account using the post button.

/profile:



Your Profile

My Feed
Create Post Logout Analytics

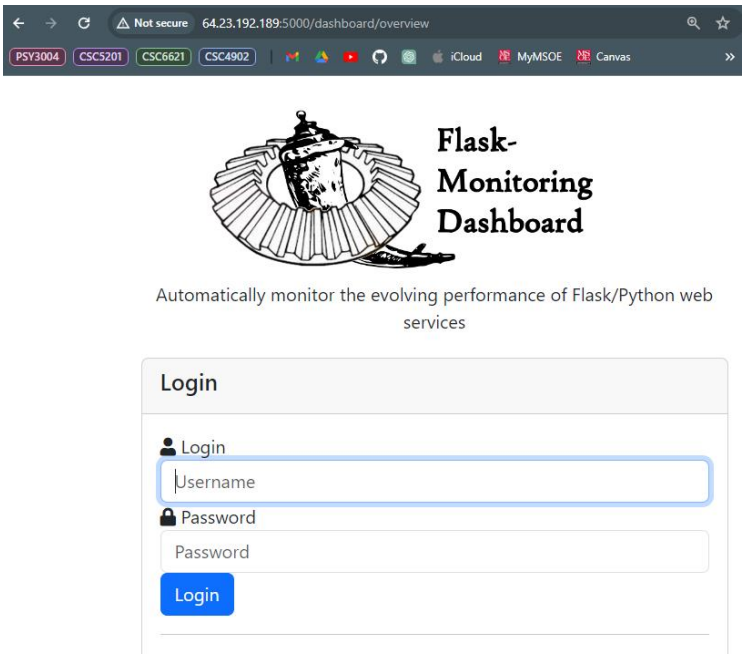
xrobbins

hello

2024-05-01 21:42:26

The profile endpoint allows the current user to see their profile, which shows all posts they have created. From here the user can see their feed, create a new post, or logout. Admin users of the application will also see an analytics button, which will direct them to the analytics dashboard login page.

/dashboard:



The dashboard endpoint shows the login page for the analytics dashboard which can only be accessed by admin users of the application through the analytics button on their account.

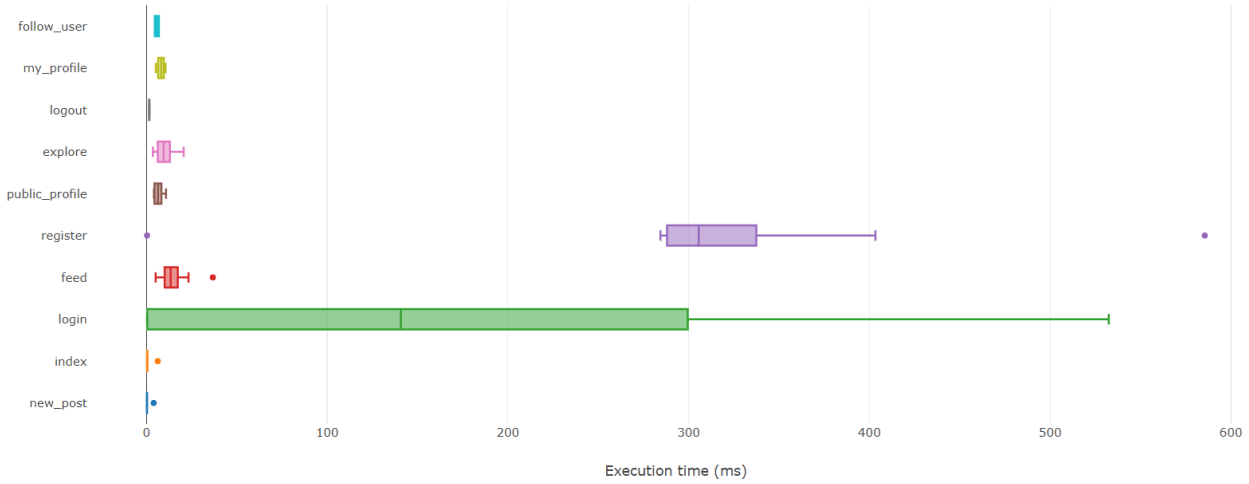
Performance Metrics:
The application was tested using a hand made testing suite to make requests to endpoints of the website. Response times and request numbers made are shown to admin users logged into the applications monitoring dashboard.

Median Request Time (ms)

Number of hits | Median request duration (ms)

Show 10 entries Blueprint:

Endpoint	Today	Last 7 days	Overall	Last requested	M
index	0.4	0.4	0.4	4 hours ago	
new_post	0.3	0.3	0.3	7 minutes ago	
login	2.5	2.4	2.4	17 minutes ago	
feed	14.9	13.6	13.6	7 minutes ago	
register	310.8	307.4	307.4	20 minutes ago	
public_profile	8.3	6.4	6.4	9 minutes ago	
explore	14.3	9.5	9.5	13 minutes ago	
logout	0	1.4	1.4	1 day ago	
my_profile	8.2	8.2	8.2	5 minutes ago	
follow_user	0	5.8	5.8	1 day ago	



Number of Requests Made

Show 10 entries Blueprint:

Endpoint	Today	Last 7 days	Overall	Last requested
index	306	312	312	4 hours ago
new_post	302	312	312	12 minutes ago
login	128	151	151	22 minutes ago
feed	64	81	81	12 minutes ago
register	66	73	73	25 minutes ago
public_profile	2	8	8	14 minutes ago
explore	3	8	8	18 minutes ago
logout	0	7	7	1 day ago
my_profile	2	4	4	10 minutes ago
follow_user	0	2	2	1 day ago

Project Post Mortem:

Throughout this project I learned a lot about building web applications in python as this was easily the most complicated flask project I have ever completed. One thing that was very new to me was implementing navigation throughout the site using buttons on the page. This was handled using code written into the HTML templates used to render the pages for each endpoint, which would redirect users to the proper endpoint when the button was activated. This functionality was also used for things like following/unfollowing users and logging out, which performed functionality for the application but did not actually render an endpoint of its own. I also gained useful insights into using docker to containerize custom applications as well as compose containers together and get them to communicate properly over the network that docker runs the containers on. One thing I actually found fairly easy with this project was deploying my application through Digital Ocean. I was very easily able to set up a droplet application in Digital Ocean and get my app running on it quickly. I had initially attempted to deploy the application on AWS using an elastic beanstalk,

however I very quickly ran into issues. Ultimately, I felt over my head in trying to determine what I needed to do to get deployment in AWS working and switched to Digital Ocean where I found the interface easier to understand and follow along with. I also spent considerable time trying to get a Prometheus database to work with my application to collect metrics for monitoring. I tried getting metrics to push to a docker container running the Prometheus database. However, I had multiple different issues at different points in the process of trying to get it to work and used multiple python libraries to try and implement my metrics and was unable to get anything working. This led to my use of the flask monitoring dashboard for my metrics.

Feedback:
Overall, I enjoyed this project. I learned a lot through this project, and I came out with a lot more understanding of several tools, especially docker and SQL. I also used peewee, a python based ORM, to interact with my database, and I found it made writing queries and fetching data much easier. I also liked using this library as it made my application able to easily switch between databases, which I used when initially building my application. I initially built my flask application with a local SQLite database to ensure all my functionality worked properly and that my database queries were returning the correct data. Once that was finalized, I incorporated the MySQL docker container and to switch my app over I only needed to change the line of code initializing the database connection. With this in mind, I liked the flexibility of this project a lot as it allowed me to explore new tools like the peewee library. This also allowed me to change and alter the tools that I used to achieve certain requirements when I encountered problems with the ones I initially planned to use. I also feel that I enjoyed this project more because I was able to create something that I was interested in rather than having to build something to follow a specific set of instructions.