David Britton
CPSC:307
Final Project Report
12/4/2023

Intro:

PastExchange is a social media platform for history and anthropology questions, blog posts, or link sharing. Essentially, it operates as a mini-Reddit or StackExchange. I got this idea because most history blogs online are on individual websites, and this makes it difficult to find new history content that a user might enjoy. This is a solo project. I did 100% of the work.

The project's stack is a React Frontend, a Node.js application server, and a locally hosted MySQL database.

There are a broad range of functionalities that I've included, such as thread creation; post creation, editing, and deletion; a likes system; a profile dropdown for users to see their threads; pagination of results; as well as a search bar. The search bar allows for searching by author, post content, and post title simultaneously.

Database:

The associated MySQL database has the following schema:

users (username, password, first_name, last_name, email),

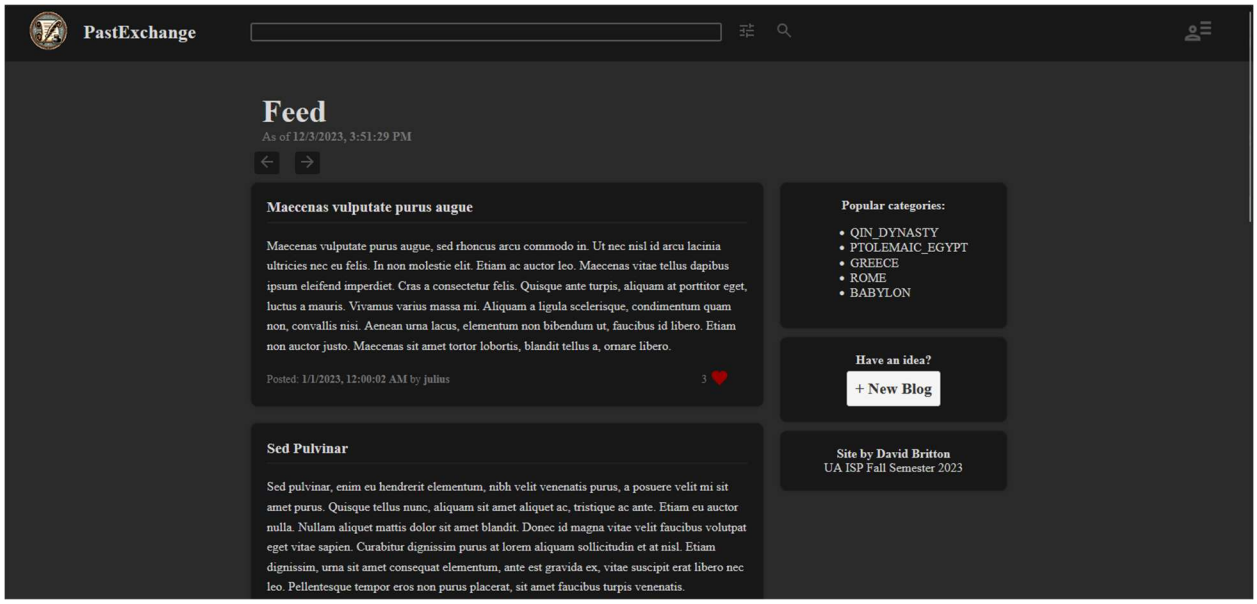
blogs (blog_id, title, owner_name, category, created_at, updated_at)

posts (post_id, blog_id, post_title, owner_name, content, created_at, updated_at)

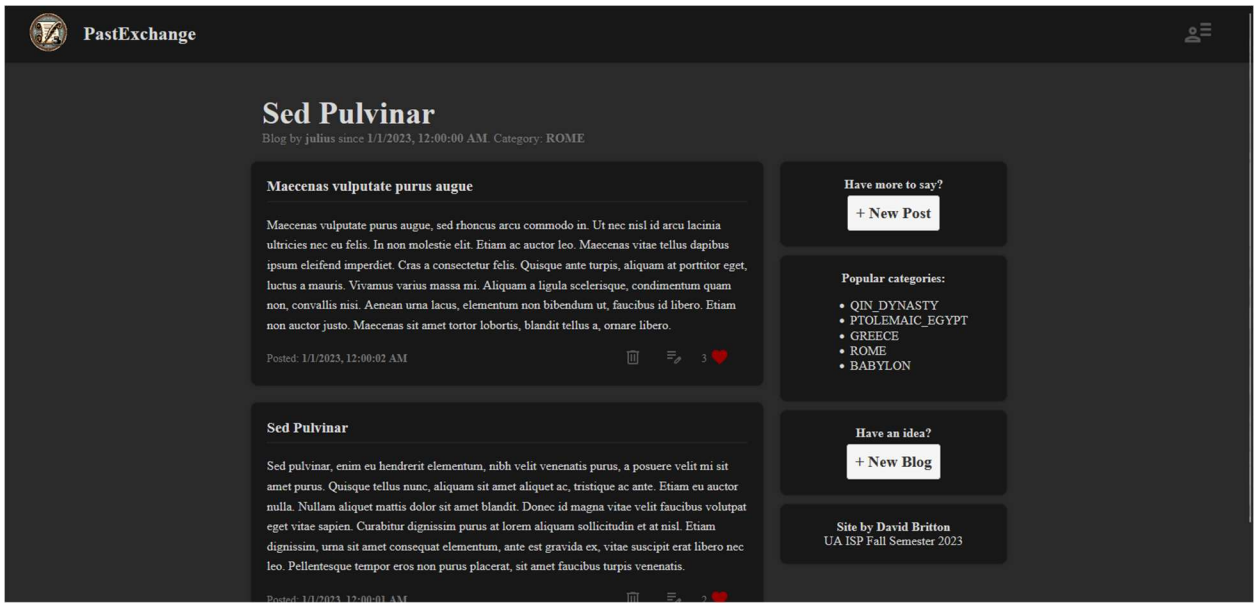
post_likes (post_id, owner_name, created_at)

Screenshots:

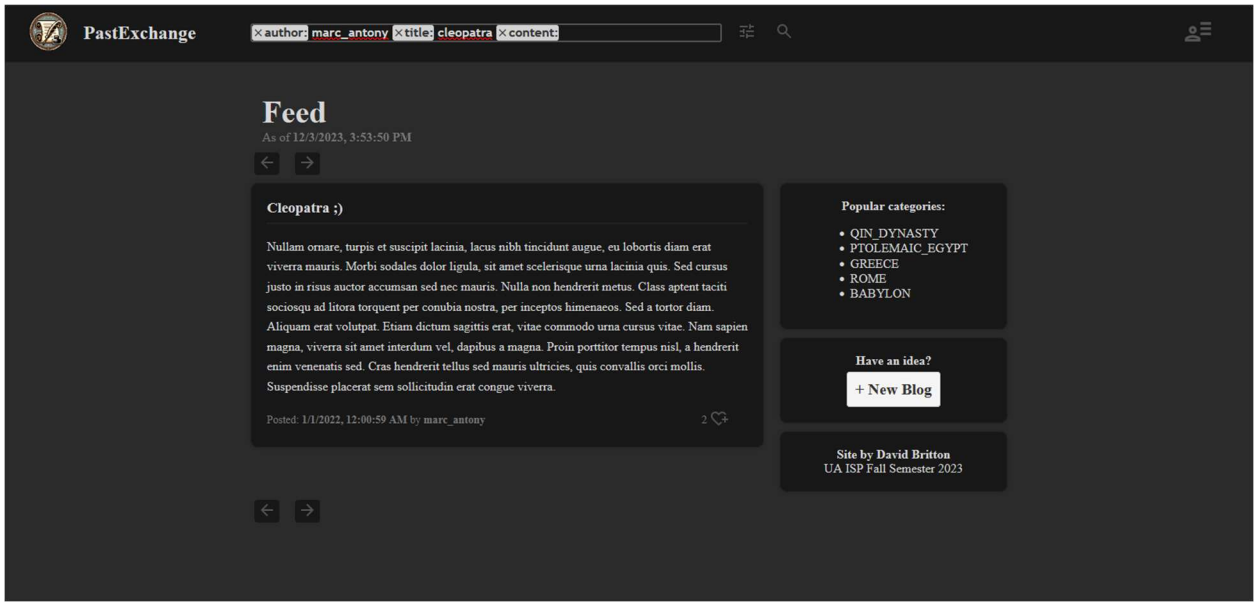
On the main feed page:



On a blog you own:



An example of the multi-parameter search bar in action:



What I learned and improvements to be made:

With both this project and my Database Management project made considerable progress with JavaScript. This is mostly thanks to the fact that the entire front-end is made with React. SQL is another thing this project helped me make progress with, especially the dynamic queries in the search bar and pagination.

The project, however, is far from perfect. Currently, my project lacks string sanitization, which unfortunately leaves it vulnerable to SQL injection attacks. Other methods for further safety include constraints other than foreign keys in the database design and even parametrized statements. User experience is also lacking, specifically when it comes to the posts themselves. Currently, you cannot embed images, and you cannot format text. Introducing Markdown interpretation of user input could solve this problem, and I know that React libraries exist that do exactly that.