NETS2120 Pets.com Project Report

Overview

Pennstagram is a social media app with the ability to sign up/ log in, set/change your profile, link with your actor-look alike, add/remove friends, receive a custom feed with smartSearch capabilities, and chat live with others & your AI assistant!

Technical Description

Pennstagram is built on node.js and React with external endpoints to:

- AWS RDS Database for persistent data
- AWS S3 for images and other large files
- Spark for cloud & distributed computing of algorithms (Post adsorption)
- Kafka for connections to other social media apps & Twitter Feed
- ChromaDB for actor-image linking, natural language search & AI chatbot.

Frontend-Backend are connected by REST API, as well as Websockets for the Chats page.

Design Decisions

BACKEND

REST routes

- Includes routes to all external endpoints (RDS, S3, Livy, Kafka, Websockets, ChromaDB)
- Tracking who is online uses a middleware function that runs before user calls to the REST API.

Websockets

- Holds sockets of active users & associated chat rooms in a Map
- Fires on events from users like sendMessage, receiving invite, accepting invite, rejecting invite, leaving chat

RDS Database

- Key'd on id where possible (user_id, chat_id, post_id) that are primary keys, not null, and auto-incrementing for consistency

S3

- Users can create posts with optional captions and images. If an image is uploaded, we create a unique 32 byte image id which we save in the RDS database and use as a key to upload to the photo's S3 bucket.

Kafka

- Have a consumer that subscribes to FederatedPosts and Twitter-Kafka topics. Each time a post is received and Kafka is turned on, we insert the post into our local RDS database under the Federated Posts and Twitter Kafka users respectively
- We also have a producer that sends a message to the FederatedPosts topic each time a user posts.

ChromaDB

- Have created a posts-2 database that stores post and user data. We have done this by turning some elements of the mysql databases users and posts into a Chroma db embedding. It uses vector similarity to compare query, and return the most similar embedding. The database is scalable

because it embeds a post when it is created, never in bulk. AI does not have access to sensitive information for security reasons (no passwords, etc.)
    - Run Chroma with chroma run --host 0.0.0.0 --port 8000 --path ./data/chromadb
- Have created the face-api-database that uses k-similarity to return 5 most similar actors.
    - Run Chroma with chroma run --host 0.0.0.0

Post Rank (Spark)

- Created adsorption algorithm for post ranking based on user interactions. Utilizing Apache Spark for efficient processing, it analyzes user activities and prioritizes content relevancy.
    - Data from users, posts, likes, hashtags, and friends tables are fetched from an AWS RDS database and loaded into Spark DataFrames.
    - A graph represents users, posts, and hashtags as vertices, with edges indicating interactions. Weights are assigned to these edges based on interaction metrics using user-defined functions.
    - Adsorption algorithm is run, initialized with a state that starts at the user nodes, value of 1, and the label for each value equal to the user node. On completion of algorithm, we look at the post nodes and add the post node, user label, and value ("weight") to the feed database

FRONTEND

Signup/Login

- Users can sign-up. Their provided info will be compared against existing records, and added to users database if no conflicts / sql attacks are found. After, users can simply log in granted they have provided the right information.

Profile

- The user is able to upload a profile photo and get back 5 most similar actors in terms of appearance. If the user clicks on the presented actors, they are added as a link and can be viewed on the Profile page across sessions.
- The user is allowed to add Hashtags they're interested in.

Friends

- Unset online times default to Jan 1st 2024
- Online times of recommendations also show up

Home

- Create Post:
    - Ability to create posts with optional captions and images. We create an image_id which we store in the database along with the caption, while the image is stored in S3 with the image_id as the key.
- Likes, comments:
    - Each post has the ability for users to like and comment and displays like counters and previous comments from other users. Each time a user likes or dislikes a post, we send a post request to add (or remove) the like from the likes database (similarly with adding comments).
    - Used to get likes and comments by sending a get request for each individual post. However after Twitter-Kafka populated the feed with 1000+ posts, we redid the feed query to join likes, comments, and post content together into just 1 query.

- Hashtags:
    - Users are able to pick hashtags they want to subscribe to at the beginning when they register for an account– which is inserted into the database. Each post posted (or received by Kafka) and comment is parsed for hashtags and each match is added to the hashtag table.
- Feed:
    - Get an SQL query that combines the PostRank table (result from adsorption algorithm) with the general posts table to help with infinite scroll. By using LIMIT, we are able to load in a few of the posts at a time, which is increased when users scroll 95% of the way down the page (infinite scroll). The posts are joined with tables for like count, whether a user has liked the post, and comments. Therefore, we can get the post, comment, and likes in one query.

Chats

- Invites persist in RDS database (they are cleared when handled (accept/reject))

Changes & Lessons & Design Decisions

We learnt a lot about teamwork & modular software development in the team. It was gratifying that everyone was able to take classroom knowledge into practice and hone in on a particular microservice framework. Github was at the core of the technical teamwork– working through merges and conflicts from different group member's local repositories to integrate each microservice into a full-fledged application was rewarding. Additionally, coordinating group member's responsibilities, helping each other work through bugs, and staying up past 2AM in Huntsman GSRs bonded us as a team.

Implementing Langchain with RAG was another challenge. The first design idea was to query the whole posts database in each call to chat, and then use an sql agent. This idea was put aside due to concerns around scalability. Instead, the posts database was turned into a Chromadb embedding, one at bulk at the beginning and then each time a new post was created. This relates to the challenges of thinking about scalable, 'bigger' projects.

Additionally, we had to overcome significant challenges in understanding the adsorption algorithm. The breakthrough was learning that the "labels" travel with the values. We illustrated all of the spark computation before programming the algorithm.

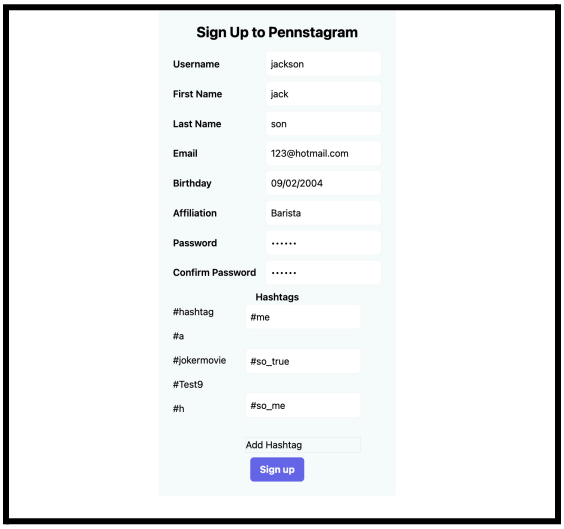
Extra Credit Features

- Websocket for chats

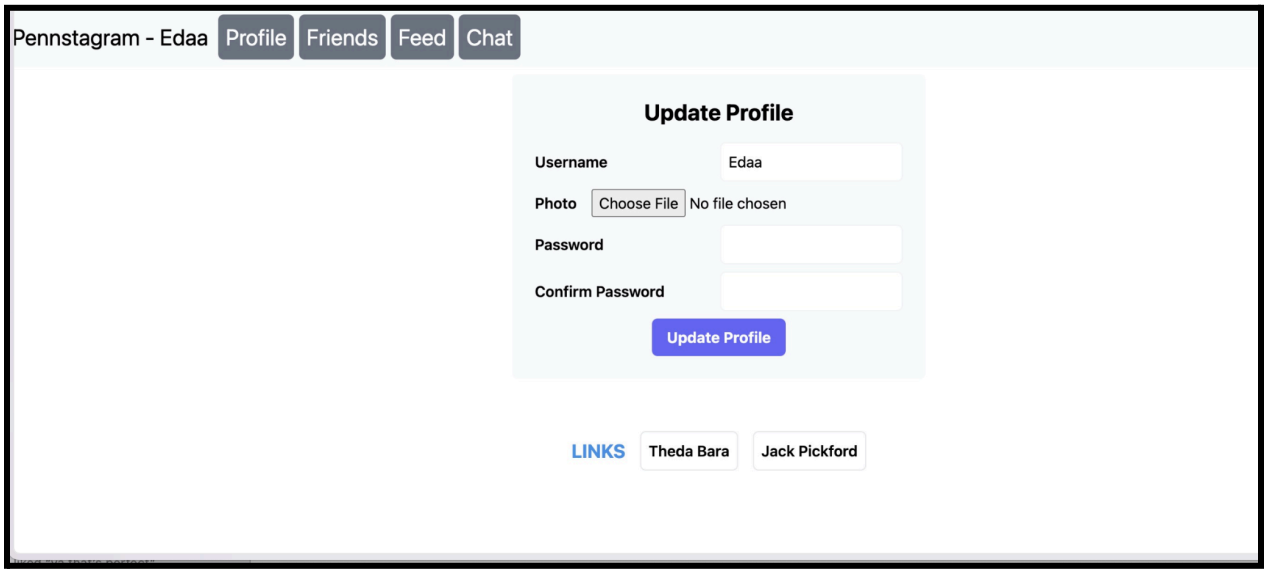
    - Mirrored server state of active users in the chat to handle sending invites, messages
    - Two-way communication between server and chatters: (when a chat is created via accepted invite both the inviter and invitee are automatically added to a chat room)
- Light/Dark Mode
    - Controlled by the dark state in App.js, which we pass as a parameter to Home. We define the colors in light, dark2, and text arrays, and the dark boolean determines which of these colors are displayed.
- Infinite Scroll
    - We load in 30 posts to start out with. When users scroll 95% of the way down the page, we load in 30 more posts. This happens infinitely (or until we run out of posts– which is like 20,000 due to Kafka).
- User search with AI chat (RAG)
    - Users mysql db is turned into an embedding and stored alongside posts in the same collection posts-actors. AI both uses vector retrieval and similarity search.

Screenshots

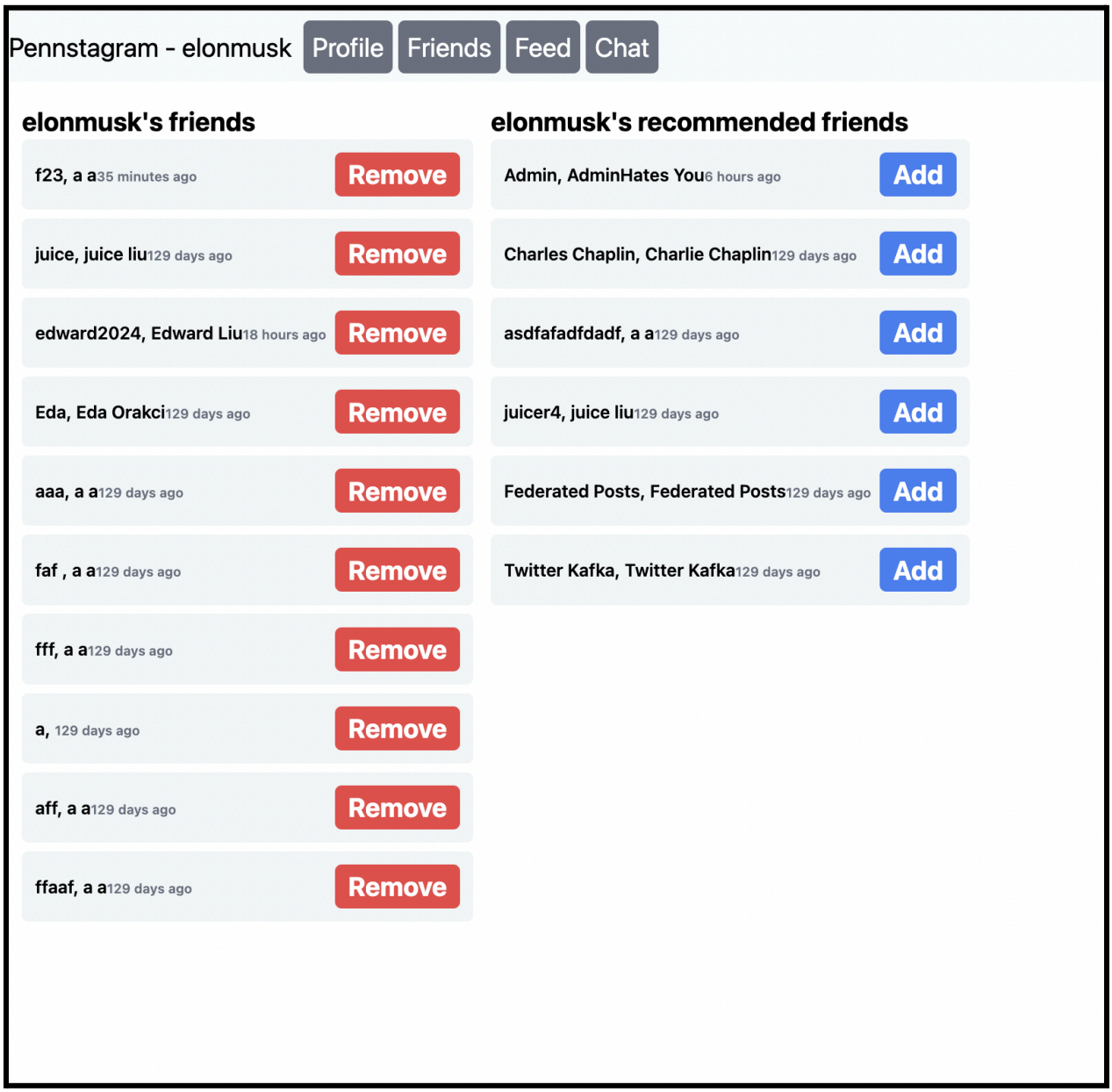
Signup



Profile


Pennstagram - Edaa
Profile
Friends
Feed
Chat
Update Profile
Username
Edaa
Photo
Choose File
No file chosen
Password
Confirm Password
Update Profile
LINKS
Theda Bara
Jack Pickford

Friends


Pennstagram - elonmusk
Profile
Friends
Feed
Chat
elonmusk's friends
f23, a a35 minutes ago
Remove
juice, juice liu129 days ago
Remove
edward2024, Edward Liu18 hours ago
Remove
Eda, Eda Orakci129 days ago
Remove
aaa, a a129 days ago
Remove
faf , a a129 days ago
Remove
fff, a a129 days ago
Remove
a, 129 days ago
Remove
aff, a a129 days ago
Remove
ffaaf, a a129 days ago
Remove
elonmusk's recommended friends
Admin, AdminHates You6 hours ago
Add
Charles Chaplin, Charlie Chaplin129 days ago
Add
asdfafadfdadf, a a129 days ago
Add
juicer4, juice liu129 days ago
Add
Federated Posts, Federated Posts129 days ago
Add
Twitter Kafka, Twitter Kafka129 days ago
Add

Home (Feed)

@Admin posted

First Post

Gee, I should think of something to say quick.

1

Comments:

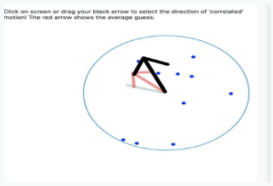
889: Great photo!

Content

Create Comment

@Eda posted

does this work?



hi

1

Comments:

889: So helpful!

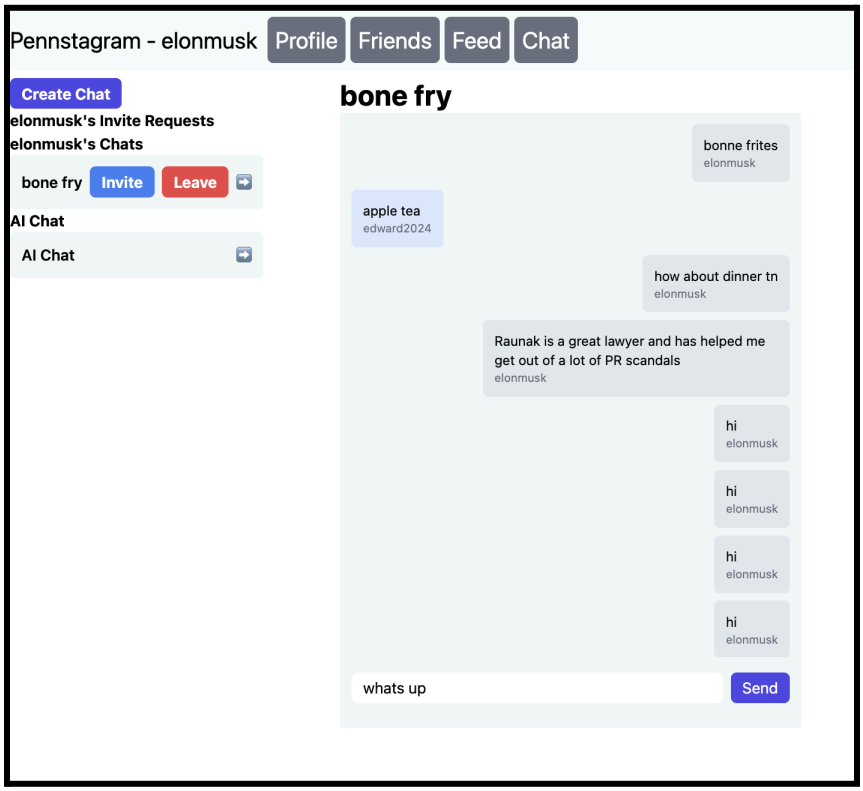
Content

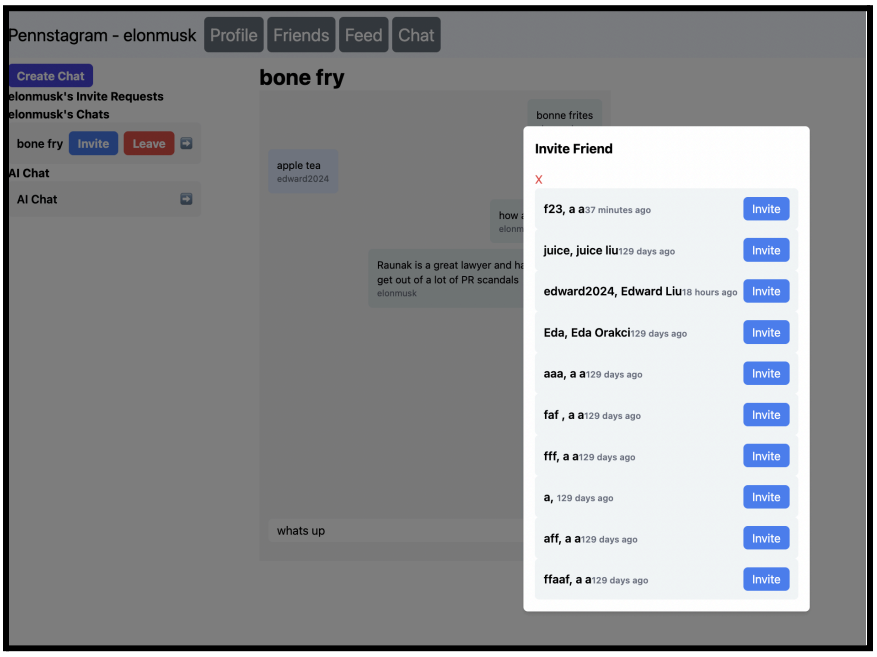
Create Comment

@Eda posted

d

Chats


Pennstagram - elonmusk
Profile
Friends
Feed
Chat
Create Chat
elonmusk's Invite Requests
elonmusk's Chats
bone fry
Invite
Leave
AI Chat
AI Chat
bone fry
bonne frites
elonmusk
apple tea
edward2024
how about dinner tn
elonmusk
Raunak is a great lawyer and has helped me get out of a lot of PR scandals
elonmusk
hi
elonmusk
hi
elonmusk
hi
elonmusk
hi
elonmusk
whats up
Send


Pennstagram - elonmusk
Profile
Friends
Feed
Chat
Create Chat
elonmusk's Invite Requests
elonmusk's Chats
bone fry
Invite
Leave
AI Chat
AI Chat
bone fry
bonne frites
apple tea
edward2024
Raunak is a great lawyer and ha
get out of a lot of PR scandals
elonmusk
whats up
Invite Friend
X
f23, a a37 minutes ago
Invite
juice, juice liu129 days ago
Invite
edward2024, Edward Liu18 hours ago
Invite
Eda, Eda Orakci129 days ago
Invite
aaa, a a129 days ago
Invite
faf , a a129 days ago
Invite
fff, a a129 days ago
Invite
a, 129 days ago
Invite
aff, a a129 days ago
Invite
ffaaf, a a129 days ago
Invite

I recently got into Penn and I want to connect with users who are affiliated with Penn. Can you give me such users?
eda

Hello! I can connect you with users who are affiliated with Penn. One user you might be interested in is Joe D, who is affiliated with Penn. Would you like me to provide you with more information about Joe D or help you connect with them on Pennstagram?
chatbot

Give me one post made by user with username edward2024?
eda

I found a post made by user edward2024 with the title "My first post" and the caption "me in nets." It seems like they are sharing a photo of themselves in a sports setting. Is there anything else you would like to know about this post or any other posts on Pennstagram?
chatbot

I've been looking for a post. It goes like: It's a beautiful day to fly
eda

Title: It's a beautiful day to fly Caption: @Xfinity takes a reunion of Navy veterans on a journey they weren't expecting ✈ Author: Twitter Kafka
chatbot

@edward2024 posted

My first post

me in nets

0

Give me information about user Admin. What is their first name?
eda

The first name of user Admin is AdminHates. How can I assist you further?
chatbot

Can you give me a post that includes #MetGala?
eda

Here is a post that includes #MetGala: Title: net.esinx.netstagram-c74cee59-093a-4665-a091-4e7d1f9c7871 Caption: The 2024 #MetGala was full of jaw-dropping fashion moments! By: Federated Posts
chatbot