Overview

This report details the development of a comprehensive Instagram like social media web application designed to support user registration, social interactions, and content sharing. Key features include user authentication, profile management, friend recommendations, real-time chat, and a dynamic feed for posts and comments. The system integrates several advanced functionalities, including face matching and social rank computation.

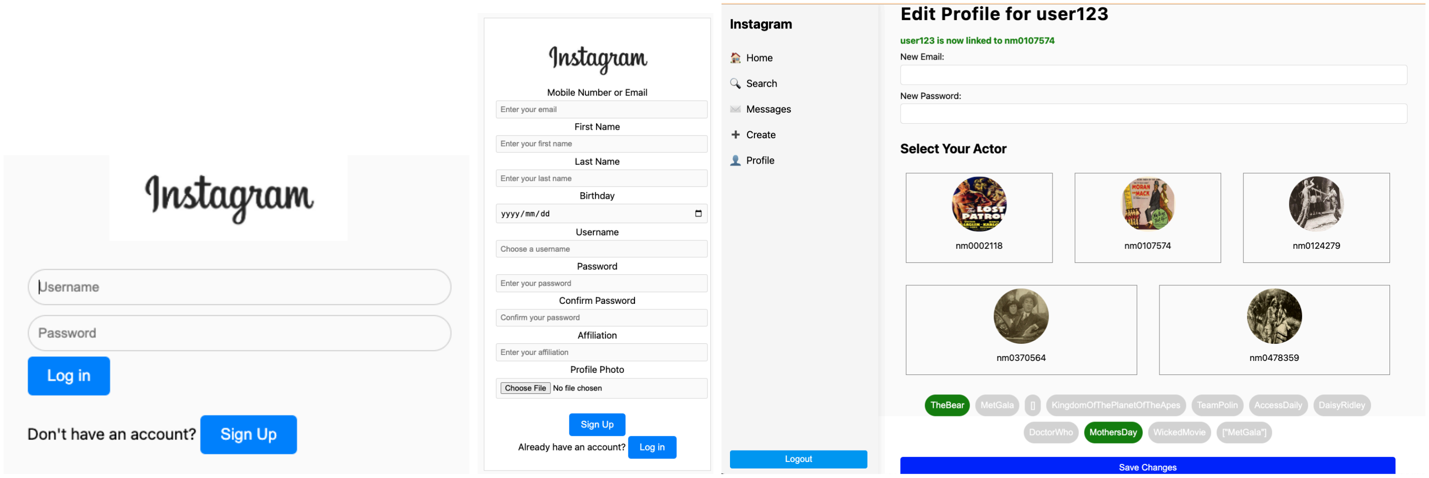
Components and Technical Details

1. Registration & Face Matching

a. User Guide: The registration process is divided into two distinct parts. The first part requires users to input personal information such as email, full name, birthday, username, password, affiliation, and profile photo. In the second part, users select hashtags of interest and choose an actor from among the five 1920s-era Hollywood actors with the highest embedding similarity for a linked account.

b. Technical Details: User info data is stored in a **MySQL** database hosted on **Amazon RDS**, utilizing a **users** table for all user information and a **user_hashtags** table for selected hashtags. For image uploads, we use **Multer**, which stores the uploaded profile picture as file metadata locally under ‘uploads/’. During the second registration step, the Face API indexes images of 5,000 IMDB actors, comparing these with the profile picture embedding to identify the top 5 similar actors. All images are stored on **AWS S3**, with the resulting URLs saved in our MySQL tables for efficient access.

c. Design Decisions and Changes: To avoid overwhelming users with too many input fields, the signup process was split into two parts. This approach lets users complete registration quickly and provide additional information at their own pace. The backend manages multiple API calls to process profile pictures, ensuring clear instructions and a seamless experience, even with potential latency.



user123 is now linked to nm0107574

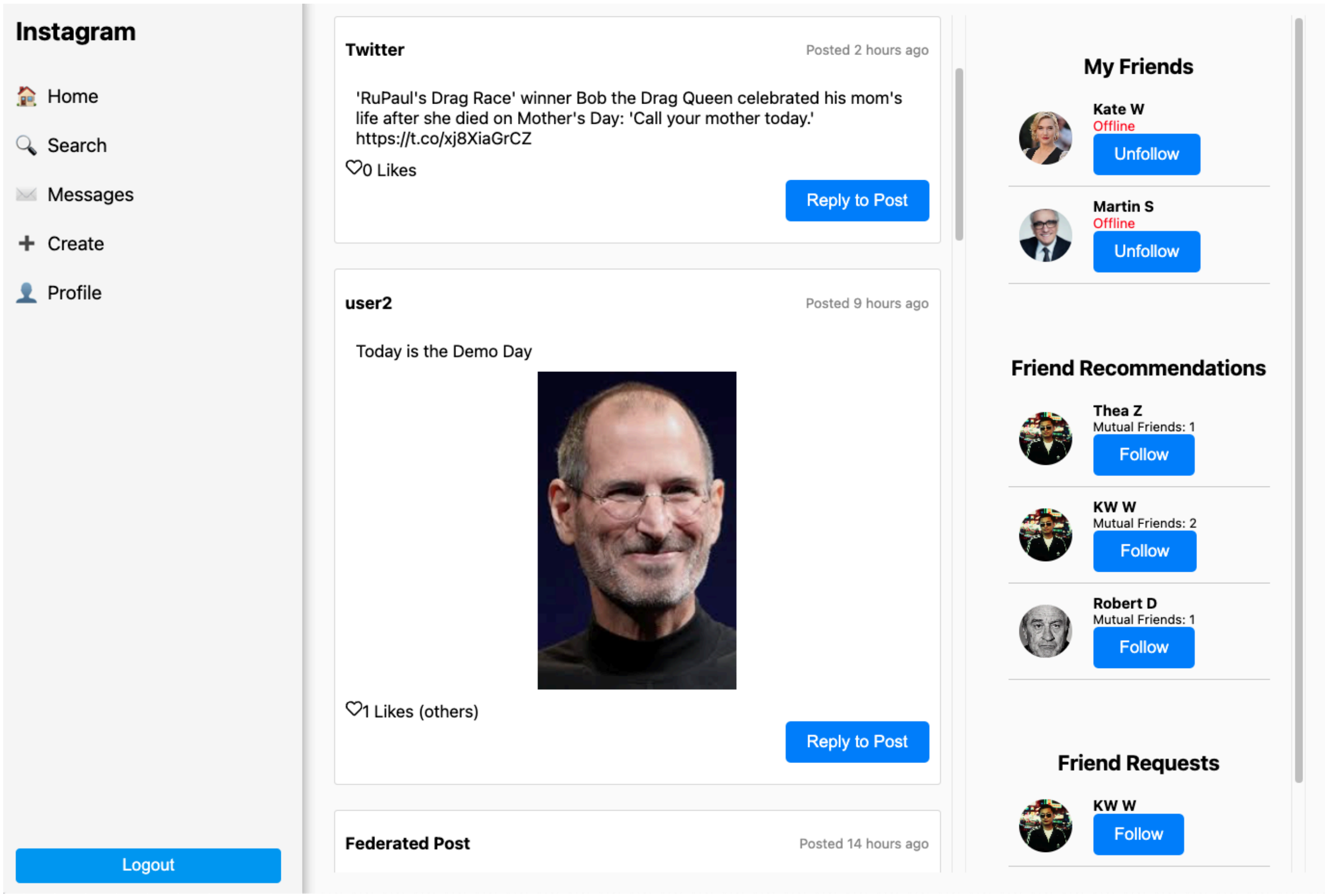
2. Login

a. User Guide: The login component serves as the entry point for user authentication. Users enter their username and password to access the system, which ensures a secure gateway to the main features.

   b. Technical Details: This component validates credentials against the **MySQL** database, ensuring user identity and secure access to the application's main features.

3. MainPage

   a. The MainPage acts as the main container for the user's homepage, combining the Sidebar, Feed, and three Friend components to create a comprehensive user interface.



Twitter — Posted 2 hours ago

'RuPaul's Drag Race' winner Bob the Drag Queen celebrated his mom's life after she died on Mother's Day: 'Call your mother today.' https://t.co/xj8XiaGrCZ

♡0 Likes

user2 — Posted 9 hours ago

Today is the Demo Day

♡1 Likes (others)

Federated Post — Posted 14 hours ago

4. Sidebar

   a. User Guide: The Sidebar is a fixed navigation bar that provides links to various sections of the application. It remains present and fixed to the left of the screen across most pages, offering a consistent navigation experience.

   b. Design Decisions and Changes: Including the Sidebar on all pages ensures easy and accessible navigation. We included essential features while keeping the navigation concise and clear.

5. Feed & CreatePost

   a. User Guide: The Feed & CreatePost components provide an interactive interface for users to view, like, and comment on posts. Users can also create new posts by entering text content, uploading images, and adding hashtags.

   b. Technical Details

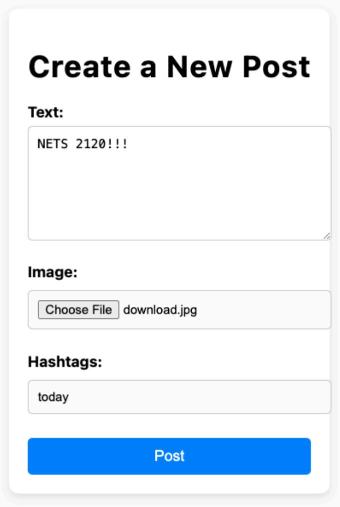
      i. Posts: This component displays post details, including the creator, images, and content.

      ii. Likes: Handles updating like counts and user lists dynamically.

      iii. Comments: Manages nested comments and replies, ensuring clear interaction.

      iv. CreatePost: Implemented as a separate component for easier backend integration.

c. Design Decisions and Changes: Initially, the feed was a single component handling all functionalities. We decided to divide it into Posts, Likes, and Comments to improve maintainability and integration with backend endpoints.



Create a New Post

Text:

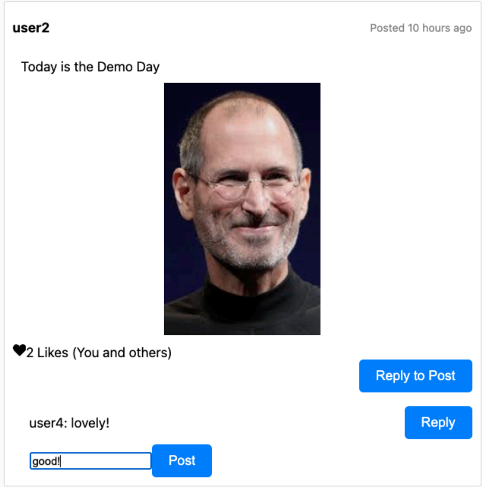
NETS 2120!!!

Image:

download.jpg

Hashtags:

today



user2

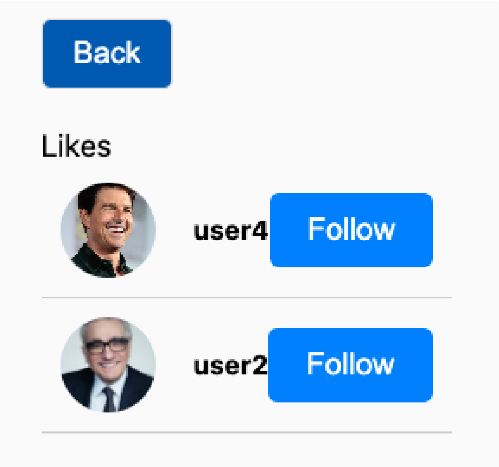
Posted 10 hours ago

Today is the Demo Day

♥2 Likes (You and others)

user4: lovely!

good!



Likes

user4

user2

6. Friends

a. My Friends:

i. User Guide: FriendsList allows users to view their mutual friends, see their online status, and follow or unfollow friends

ii. Technical Details: The component displays only the current user's friends and uses **WebSocket (EC)** for real-time online status updates.

iii. Design Decisions and Changes: Switching to a WebSocket approach for handling online status minimizes database load and ensures real-time updates, enhancing performance and user experience.

b. Friend Recommendation:

i. User Guide: The FriendRecommendations component displays suggestions for new friends, ranking them based on a social rank algorithm.

ii. Technical Details: Friend suggestions are determined by the social rank score, ensuring relevant and meaningful recommendations.

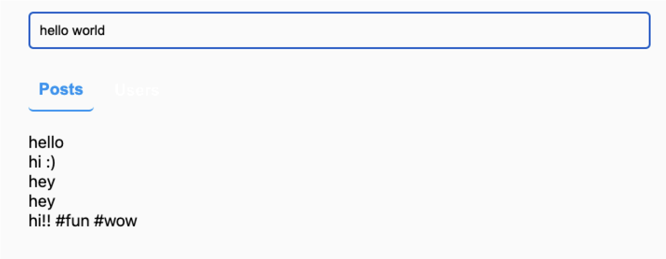
c. Friend Request (EC):

i. Technical Details: The FriendList component is reusable across scenarios, like displaying followers and following on the profile page. **WebSocket** is used to handle online status, avoiding database overload and ensuring real-time updates.

ii. Design Decisions and Changes: To avoid cluttering the main page, a separate column was created for the friend lists, allowing users to explore suggestions in a dedicated space.

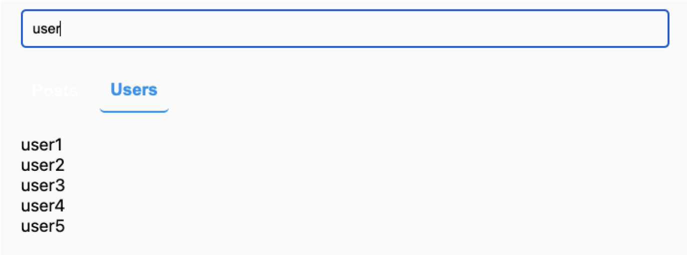
7. Search

a. User Guide: The Search component allows users to search for posts or users, providing a tab-based interface for easy navigation. The search uses an **LLM**, paired with embeddings from **chromadb**, to interpret the query of the user and provide similar results.



hello world

hello
hi :)
hey
hey
hi!! #fun #wow



user

user1
user2
user3
user4
user5

8. Chat

a. User Guide: The Chat component enables users to invite friends to chat sessions. Friends receive notifications of the chat, and a session is then created. Group chats work for both one-on-one chats and larger multi-person group chats.

b. Technical Details:

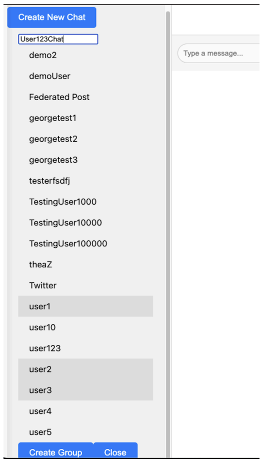
i. Friend Notifications: Ensures friends are promptly notified of chat invites.

ii. Chat Icon: Users are able to upload chat icons which are uploaded to **Amazon S3** and displayed to the other people in the chat.

iii. Messages: Users in the chat can send messages that will promptly update for recipients using **Websockets (EC)**.

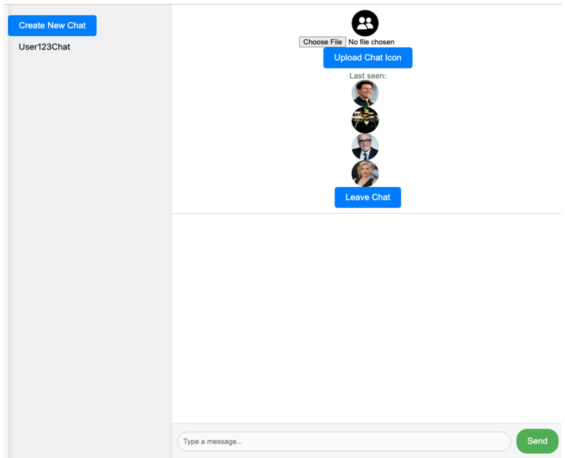
iv. Chat group (EC)

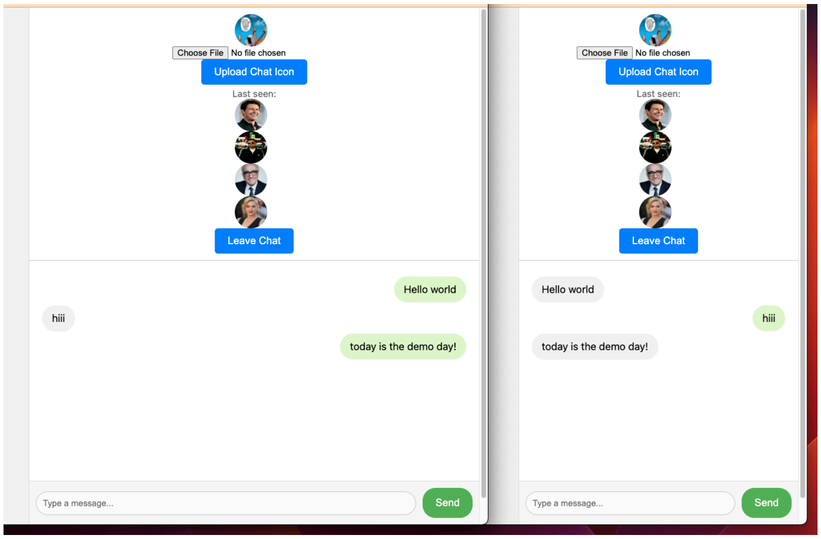
v. Leave Chat: Users can opt to leave a chat, which will delete their access to it and remove it from their view. When a user leaves, other users are promptly notified using sockets by their profile icon disappearing. When a user leaves a chat that they're the last member of, it deletes the chat from the backend.



User123Chat

demo2
demoUser
Federated Post
georgetest1
georgetest2
georgetest3
testerfsdfj
TestingUser1000
TestingUser10000
TestingUser100000
theaZ
Twitter
user1
user10
user123
user2
user3
user4
user5





Hello world
hiii
today is the demo day!

Hello world
hiii
today is the demo day!

9. ProfilePage

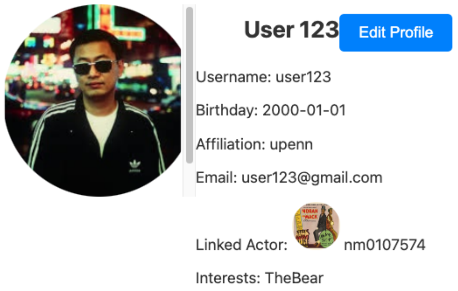
a. User Guide: The ProfilePage displays user details and allows for profile editing. Users can update their email, password, actor link, and selected hashtags.

b. Technical Details:

i. ProfilePage: Shows user details and photo gallery.

ii. EditProfile: Manages profile updates, ensuring changes are saved and reflected correctly.

c. Design Decisions and Changes: A separate edit profile page was created to simplify profile updates and provide a focused experience. This approach ensures clarity and user-friendly interactions.



Username: user123

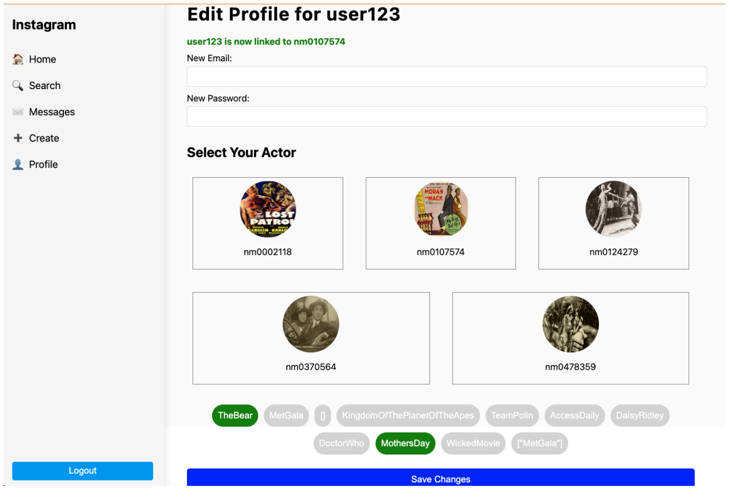
Birthday: 2000-01-01

Affiliation: upenn

Email: user123@gmail.com

Linked Actor: nm0107574

Interests: TheBear



Edit Profile for user123

user123 is now linked to nm0107574

New Email:

New Password:

Select Your Actor

nm0002118 | nm0107574 | nm0124279 | nm0370564 | nm0478359

TheBear ✓ | MetGala | [] | KingdomOfThePlanetOfTheApes | TeamPolin | AccessDaily | DaisyRidley | DoctorWho | MothersDay ✓ | WickedMovie | ["MetGala"]

10. Twitter / X Feed (Kafka)

a. Technical Details: The **Kafka** topic serves as a hub for consuming real-time Twitter posts. Tweets are continuously published to the Kafka topic by a process integrating with the Twitter API. The system checks if the tweet author is registered; if not, a new user is created. Tweets are added to the post table, ensuring data integrity and consistency.

11. Federated Post (Kafka)

a. Technical Details: The Kafka consumer in the Federated Post system listens to messages containing post data on a specific topic, processes and validates the content, and constructs a multipart/form-data payload to integrate the post into local databases. The Kafka producer publishes messages to a dedicated topic, encapsulating post data in JSON format, and ensures message delivery by confirming successful transmission and managing connections.

12. Absorption Ranking Computation

a. Technical Details: Using **Apache Spark** and **Livy**, the absorption ranking computation updates hourly, clearing the socialRank table before each run. The algorithm considers follower-followed relationships and computes user ranks iteratively, handling sinks and backlinks effectively.

13. Security and Scalability

a. Technical Details: We utilize **cookie** settings to enhance website security and manage authentication processes. By implementing these settings, we ensure that user sessions are securely maintained and that authentication credentials are handled safely throughout the user's interaction with our site. This approach helps in safeguarding sensitive user information and maintaining the integrity of our web services.