MODERN APPLICATION DEVELOPMENT II

SPONSOR COORDINATION PLATFORM

BY RISHABH INDORIA

Student Details

Name: Rishabh Indoria

Roll No: 21F3001823

Email: 21f3001823@ds.study.iitm.ac.in

About Me: My passion lies in Data Science. I'm captivated by the world of data and its endless possibilities, striving to unlock insights that drive meaningful decisions. Additionally, my curiosity extends to the realm of Artificial Intelligence, where I'm dedicated to pushing the boundaries of what's possible with intelligent systems.

Introduction

The project aims to provide a platform where sponsors and influencers can coordinate with each other. Sponsors can find influencers to promote or advertise their product, and Influencers can choose who to collaborate with and receive monetary compensation for the same.

Approach

For the project, I started by building the backend. Due to experience gained in making the project for MAD I, I was able to quickly set up the backend. Then I learned Flask-Security, how it works and how the models should be set up. Flask-Security, bcrypt and HTML5 validation were used for ensuring safe and secure login functionality. Next, I needed to learn flask_restful to build an API, I segregated functionality of each model into different files to make it more structured and easy to access. I also explored pagination from

Flask-sqlalchemy which allows limiting the number of results shown in the page, this made sure that the user is not overwhelmed with a huge number of campaigns or ad requests.

For the front end, Vue CLI was used. Since, I had never used Vue before, the majority of my time was spent building this. Even though I had HTML templates from my MAD 1 project, I had to add and change a lot of things to incorporate the flask API. For styling and aesthetics, I ended up choosing cards for displaying all the details and modals to display all the forms, I also used various alerts and buttons to give the application a more lively feel.

For Backend jobs, I learned celery for scheduling and workers, as well as Redis for caching. I made a daily reminder template for influencers where if they have any pending ad requests, they would be sent an email. I also made a monthly report template where every month, all the sponsors would be sent how many campaigns and ad requests they created in this month and how many of those were accepted or rejected in the form of graphs. Finally, I also made CSV generating function, where the CSV contains details about all the campaigns for a particular sponsor. The sponsor can request the CSV generation and CSV would be sent to their email.

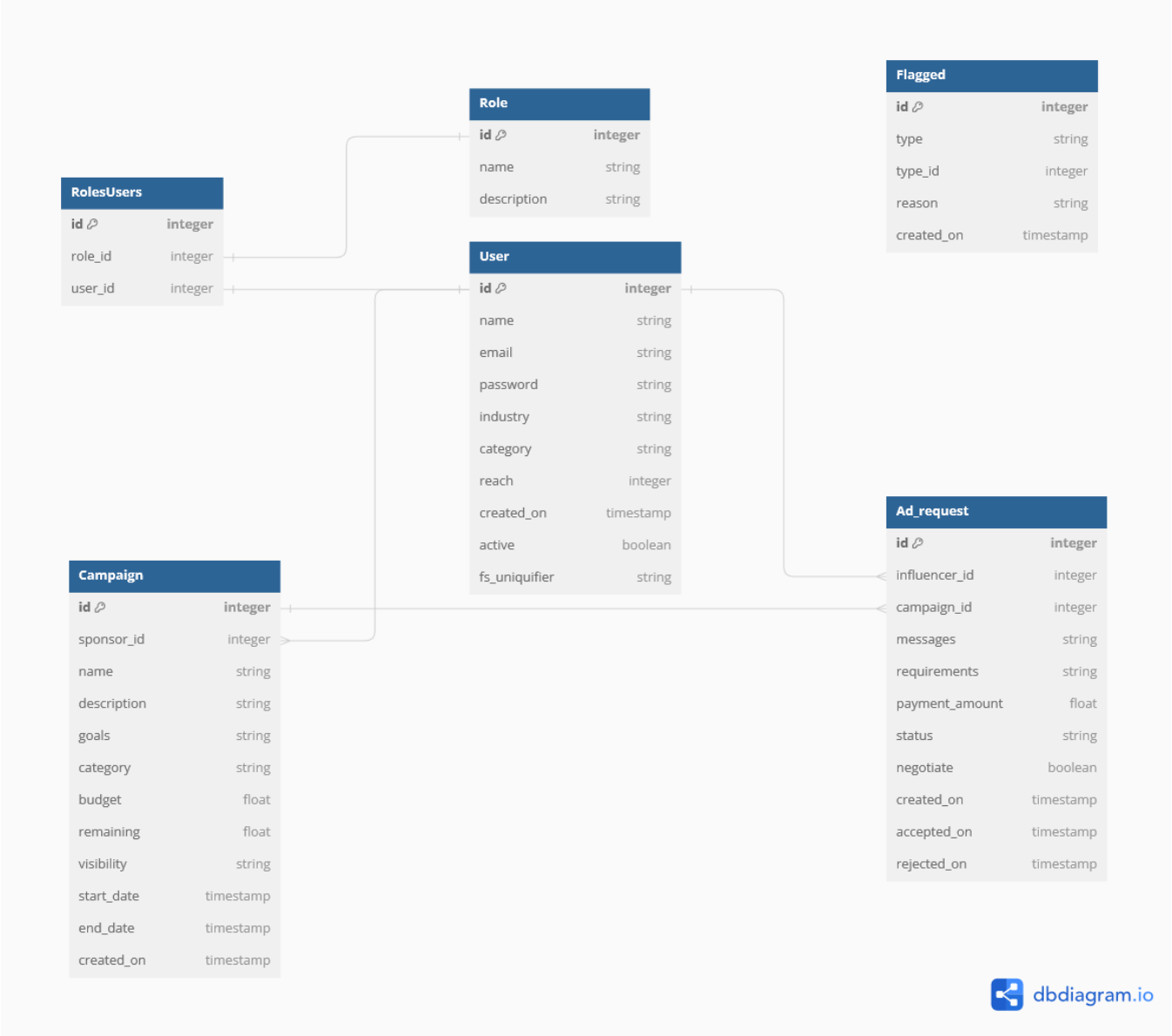
All the different graphs were built by using matplotlib which are then converted to a base64 encoded string which can be displayed by HTML. I also used caching for the statistics page, so if any of the users go to their statistics page the same page would be displayed for 6 hours, after which it would be refreshed.

Frameworks and Libraries used

- Flask: Backend framework used for building the backend
- Flask-sqlalchemy: Part of Flask framework used to easily build and manage SQL tables
- Flask-Security: Part of Flask framework used to authenticate and validate users
- Flask-Bcrypt: Part of Flask framework used for password hashing
- Flask-Cache: Used for caching
- Flask-Restful: Used for building API
- Flask-CORS: Used for handling Cross Origin Resource Sharing
- Redis: Used for caching
- Celery: Used for backend jobs and scheduling

- datetime: Used for handling date and time operations
- re: Regular Expression library used for pattern matching in form validation.
- HTML: Frontend technology for displaying user interface
- Vue: JavaScript framework for building user interfaces
- Bootstrap: Frontend framework used for styling
- Matplotlib: Used for making graphs
- faker: Python library that generates fake data

ER Diagram



API Endpoints

GET, PUT, DELETE /api/campaign/<campaign_id>

- Functionality: Get Campaign Details, or Update details or Delete Campaign.

POST /api/campaign

- Functionality: Create new Campaign

POST /api/campaign/sponsor

- Functionality: Get Campaigns associated with specific Sponsor

POST /api/campaigns/search

- Functionality: Search Page for Campaigns

POST /api/campaigns

- Functionality: Get All Campaigns

GET, PUT, DELETE /api/ad_request/<ad_request_id>

- Functionality: Get Ad Request Details, or Update details or Delete Ad Request

POST /api/ad_request

- Functionality: Create New Ad Request

PUT /api/ad_request/revert

- Functionality: Respond to an Ad Request(Accept or Reject)

PUT/api/ad_request/negotiate

- Functionality: Negotiate an Ad Request(For Influencers only)

POST /api/ad_requests

- Functionality: Get All Ad Requests

GET, PUT, DELETE /api/user/<email>

- Functionality: Get User Details, or Update details or Delete USer

POST /api/user/search

- Functionality: Search Page for Influencers

POST /api/users

- Functionality: Get All Users

GET/api/user/stats/<email>

- Functionality: Get Statistics for a particular User

POST, GET, DELETE /api/user/flag/<email>

- Functionality: Get Flag, Delete Flag, or Create Flag

POST /api/users/flagged

- Functionality: Get All Flagged Users

GET /api/auth/test

- Functionality: Test for blueprint functionality

POST /api/auth/login

- Functionality: Flask Security based login

POST /api/auth/register

- Functionality: Register new User

GET /trigger_daily_reminder

- Functionality: Trigger celery worker to generate and send daily reminder emails

GET /trigger_monthly_report

- Functionality: Trigger celery worker to generate and send monthly report emails

GET /trigger_create_resource_csv/<email>

- Functionality: Trigger celery worker to generate and send email containing a CSV attachment with campaign details to a specific sponsor.

Presentation video

https://drive.google.com/file/d/1Cxu3QUi4NYI53LdLSvj4uJhqQNRZxePF/view?usp=sharing