Project Modern Application Development - 1

Name: Challa Rupak Vardhan
Email: 210040038@iitb.ac.in

Problem Statement: Influencer Engagement and Sponsorship Coordination Platform - Basically, It's a platform to connect Sponsors and Influencers so that sponsors can get their product/service advertised through campaigns and ad-requests for each campaign and the influencers can accept, negotiate or reject the requests.

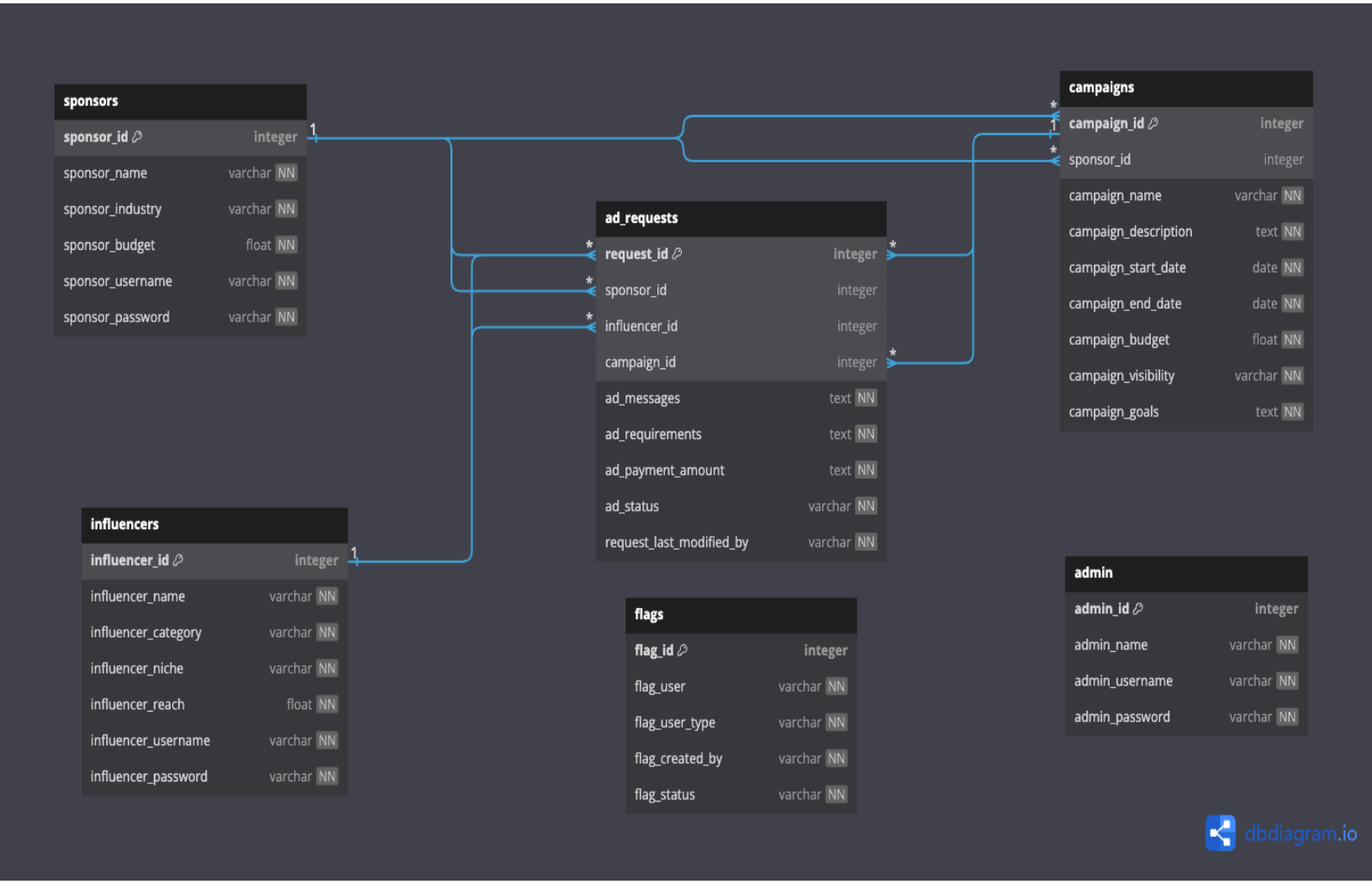
Approach:

1. First, I created a sponsor coordination portal. In that I created login or signup forms and then I have created the database tables for sponsors and then followed the same procedure for both influencer and admin.
2. I have created the database tables for influencer, campaigns and ad requests.
3. I have also created the table to monitor the flags. And there is one table for admins, just in case we need multiple admins for the website.
4. Then I created the core logic of managing the input form data for different purposes like creating ad requests, creating campaigns, editing them, login forms etc… to add, delete or modify them in the tables.
5. I have also added conditional checks on status of ad request and display appropriate actions respectively for the user.
6. The same is done for the sponsor and is given access based on the status of ad-request.
7. I have added a feature to flag a user (sponsor/influencer) and I made sure that only a user can be flagged by another user for 1 time. They can't spam flag them.
8. I have also created a session based Login system, such that users cannot enter into the site without logging in. He can't access the website solely based on URL/paths. He needs to be logged in to visit them.
9. I also made sure that admin flagged users can't login anymore.
10. I also made sure some part of backend validation in code for login systems and to do particular operations within the website.

Frameworks and Libraries Used:

1. Flask for application code
2. Jinja2 templates + Bootstrap, HTML and styling
3. Flask-SQLAlchemy for Database management
4. Flask-Session for sessions
5. Flask-Migrate for migrations
6. datetime for converting the dates into appropriate types

ER Diagram of the Database:



API Endpoints: In my website code, I haven't used any API service. All my routes are web routes. They are directly used by clients to interact. None of the view functions returns json data or any format that can be used by others.

Drive Link of the video:
https://drive.google.com/file/d/1KjAq8beF7fyCEvOv-sBVQCNDZdOrU3IO/view?usp=sharing