Influencer Engagement and Sponsorship Coordination Platform

Final Project Report

BY VIKAS RATHORE

22f1001805 | Modern Application Development-1 | May-2024, Term

AUTHOR

Name – Vikas Rathore

Roll No – 22f1001805

Email – 22f1001805@ds.study.iitm.ac.in

About me - I'm a data science student with a strong foundation in web frameworks, UX design, ML and Python. I have experience in frontend development and Flask, complemented by a solid background in mathematics.

DESCRIPTION OF PROJECT

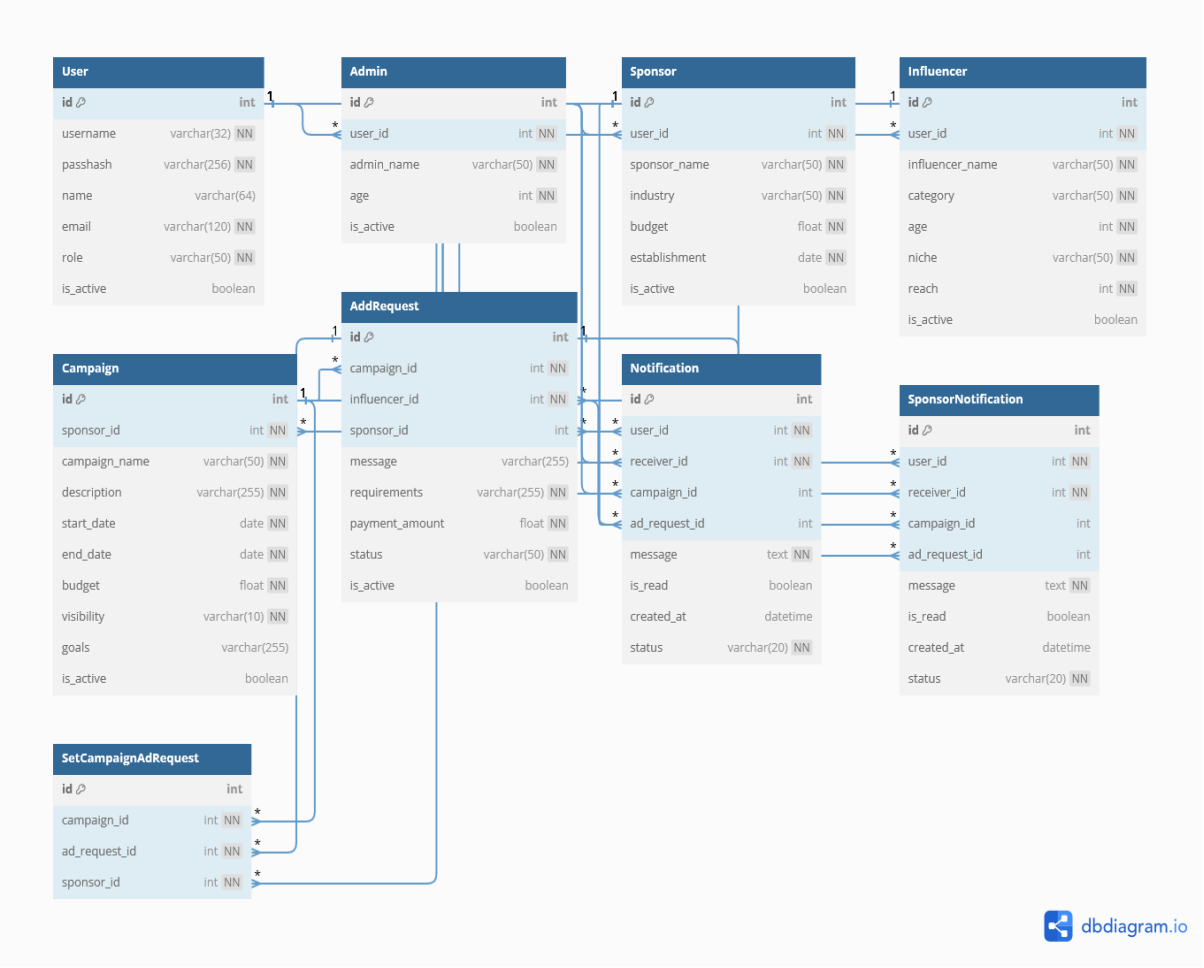
This platform connects companies and influencers, facilitating collaboration between them. Companies can register, create campaigns, and either assign influencers or send campaign offers for product promotion. Influencers, such as YouTubers or actors, can register to search for campaigns, engage with companies, and earn money by advertising products. Essentially, this software serves as a bridge to streamline and enhance partnerships between sponsors and influencers.

TECHNOLOGIES USED

- **Flask**: Python web framework for building and managing the web application.
- **SQLAlchemy**: ORM for database operations and schema management.
- **HTML**: Markup language for structuring web pages.
- **CSS**: Stylesheets for designing the web interface.
- **JavaScript**: Adds interactivity and dynamic features.
- **datetime**: Handles date and time functions.

DB SCHEMA DESIGN

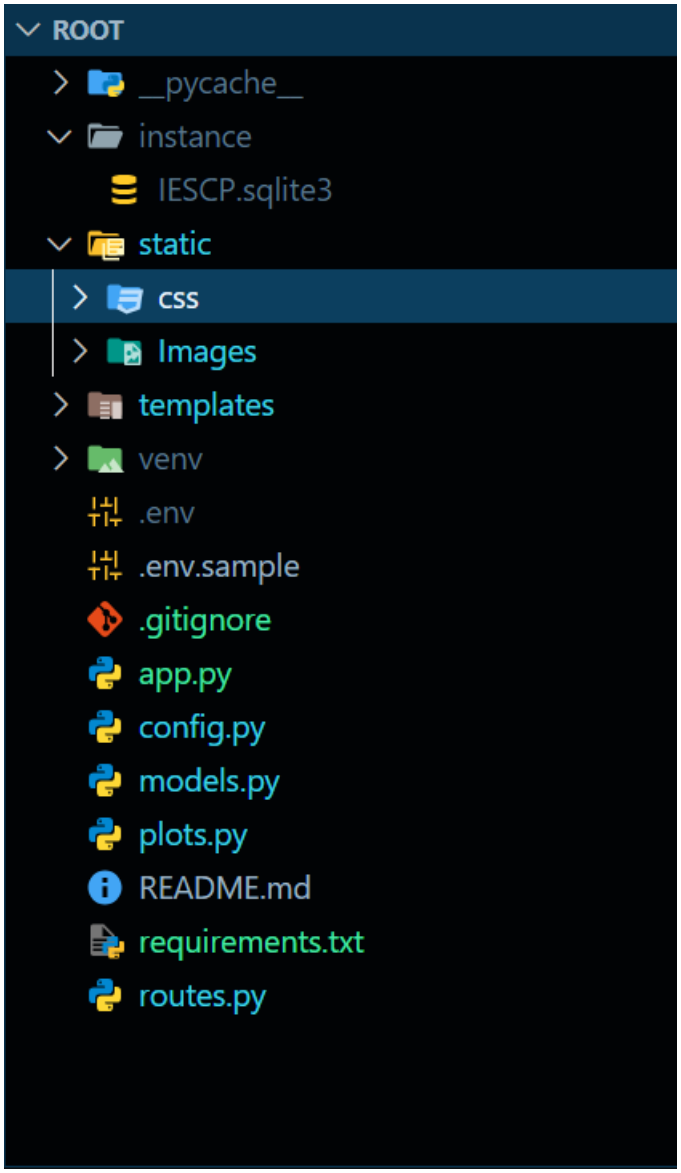
The database schema includes tables for User, Admin, Sponsor, Influencer, Campaign, AddRequest, Notification, SponsorNotification and SetCampaignAdRequest. These tables are interconnected to manage various aspects of the platform, such as campaign creation, ad request handling. Key fields track user profiles, campaign details, ad request statuses, and notifications. This design ensures efficient management and interaction within the sponsor-influencer ecosystem.



ARCHITECTURE

The app.py file is the main entry point for running the web application. The project is structured with the following components:

- **app.py**: Contains the main code to initialize and run the Flask web application.
- **models.py**: Defines the database models and their relationships.
- **routes.py**: Manages the application routes and view functions.
- **config.py**: Configures application settings using environment variables.
- **.env**: Stores environment-specific configurations such as database URI and secret key.
- **static**: Contains CSS and image files for styling and static content.
- **templates**: Includes HTML templates for rendering views.



FEATURES

User Management:

- **Registration and Authentication:** Allows users (sponsors and influencers) to register and authenticate.
- **User Roles:** Differentiates between sponsors and influencers to control access and permissions.
- **Profile Management:** Users can manage their profiles, including details like name, email, and contact information.

Campaign Management:

- **Campaign Creation:** Sponsors can create and manage campaigns, detailing objectives, budget, and duration.
- **Campaign Assignment:** Sponsors can assign influencers to campaigns for promotion.
- **Public and Private Campaigns:** Options to set campaigns as public or private based on visibility and access.

Ad Request Management:

- **Request Submission:** Influencers can submit ad requests for specific campaigns.
- **Request Review and Approval:** Sponsors review, approve, or reject ad requests from influencers.
- **Status Tracking:** Track the status of ad requests, including approval or rejection.

Notification System:

- **Request Notifications:** Send notifications to sponsors when influencers submit ad requests.
- **Campaign Updates:** Notify influencers about updates or changes in campaign details.

VIDEO LINK

Video demonstration of my project is available here

https://drive.google.com/file/d/1b748XeAothtJr2YgpPbqtuar16ycuXgy/view?usp=sharing

Thank You