Project for Modern Application Development II Course

TITLE: IESCP App

AUTHOR:
Name: Tejas Patare
Roll No: 22F1001500
Email: 22f1001500@ds.study.iitm.ac.in
CS Enthusiast | Currently a college student

1.Description:

The project is designed to create a platform that connects sponsors and influencers to facilitate advertising campaigns. Sponsors can create campaigns and request influencers to engage in them, while influencers can accept or negotiate terms for these campaigns. This platform includes role-based access control for different users: Admin, Sponsor, and Influencer.

- **Admin**: Monitors users, campaigns, and statistics, with the ability to flag inappropriate content.
- **Sponsor**: Manages campaigns, requests influencers, and tracks ad requests.
- **Influencer**: Manages their profile, searches for campaigns, and accepts/rejects ad requests.

Approach:

To solve the problem, I have developed a web platform using the following technologies:

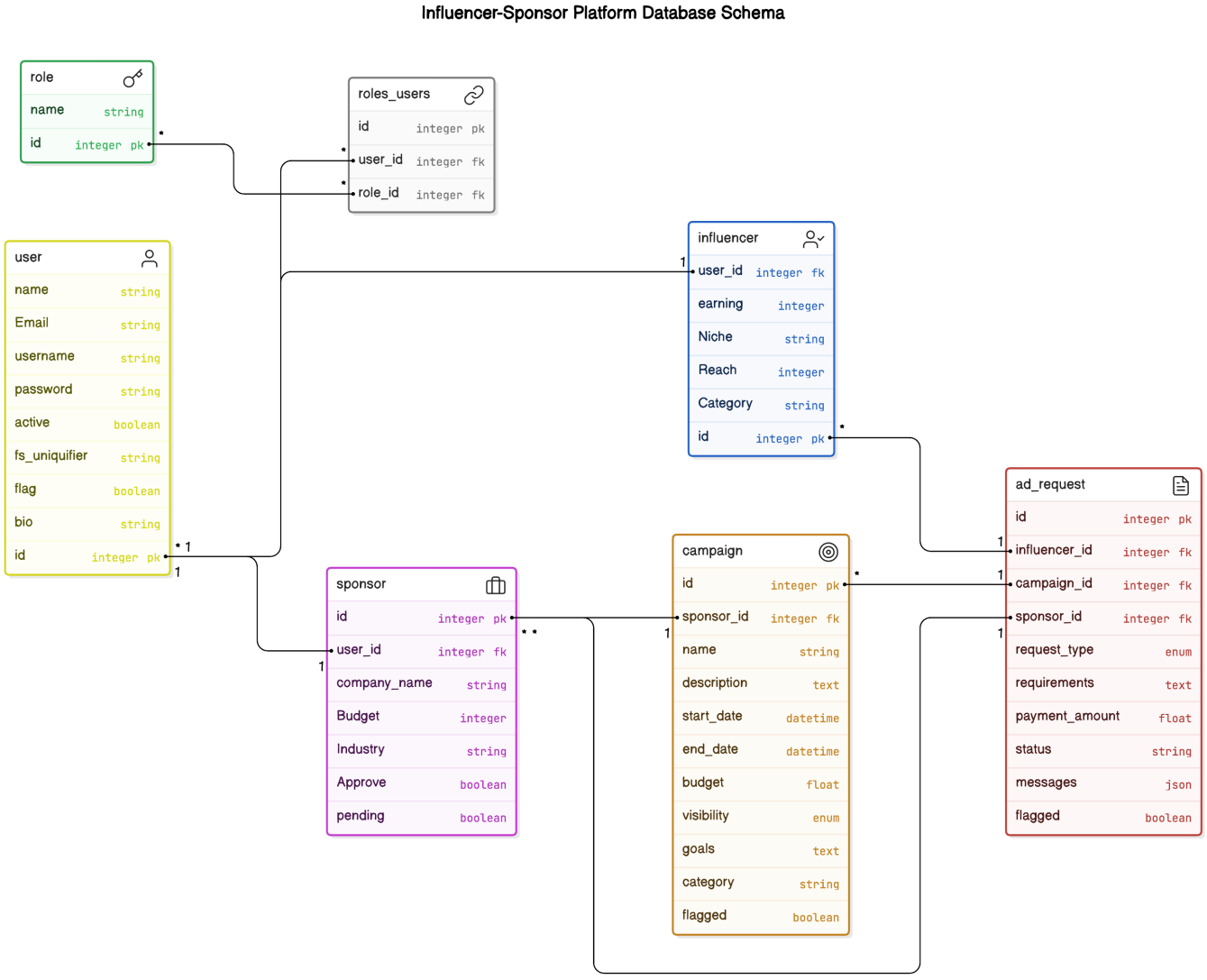
- **Backend**: Flask for API, SQLite for data storage, Redis and Celery for backend jobs, and caching.
- **Frontend**: Vue.js for dynamic UI with Bootstrap for styling.
- **User Roles**: Role-based access control for Admin, Sponsor, and Influencer.
- **Database**: SQLite to store user, campaign, ad request, and other related data.

2. Technologies used:

i. **Flask:** used for building the web application.
ii. **Flask-SQLAlchemy:** extension of Flask, used to handle database connections across the app.
iii. **Flask-JWT-Extended:** is an extension for Flask that provides JSON Web Token (JWT) authentication support. JWTs are a secure and standardised method for transmitting claims between parties, often used to implement stateless authentication in web applications..
iv. **CSS:** CSS is employed for styling the web pages and enhancing the user experience.

v. **HTML**: HTML is used for structuring the web pages and creating user interfaces.
vi. **ChartJs**: Uses for creating different types of charts on admin dashboard
vii. **SQL lite**: Database management system for storing application data.
viii. **Jinja2**: Template engine for rendering dynamic HTML content.

3.DB Schema Design: (ER Diagram)



5. Features:

1. The **flask app** opens at the main page. Where person can

i. Login: username and password are required fields in form. User should exist in the database or he/she can register himself/herself.

ii. Register user: first name, username and password are required in form.

Username should not exist in the database.

If incorrect password or username is given a prompt is displayed invalid credentials.

2. Users

a. Sponsor:

- Can login and create campaigns, update it, delete campaigns.
- Search for influencers and send Ad-request influencers to participate in their respective campaigns.
- They can accept of reject request sent by influencers to participate in campaigns.

b. Influencer:

- Can login, update their profile.
- An influencer will handle ad requests by accepting or rejecting them, and they can send revised ad requests back to sponsors.
- They can also search for active campaigns based on criteria like name and category and choose to participate. Additionally, they can delete their requests.

c. Admin**:**

- Can login, and see all the statistics in dashboard.
- An admin can monitor all the users/campaigns,
- Admin have ability to flag inappropriate users, and also view and delete it.

6. Video:

https://drive.google.com/file/d/1gQfPEY2i6C9FoRAmi__2TcKlRvySIt0B/view?usp=sharing