MAD 1 Influencer and Sponsorship Coordination Platform

Name – Sahana Srinivasan

Roll – 21f3002750

Email – 21f3002750@ds.study.iitm.ac.in

I am Sahana Srinivasan. I am a student pursuing BS in Data Science and Applications from IIT Madras, currently in diploma level. I have keen interest in coding, and Machine Learning.

Project Description

This project is based of our MAD 1 theory course. The project is about building a platform that connects Influencers and Sponsors.

AdVeri is a platform that is intended to connect influencers with sponsors. Sponsors being the one creating campaigns to market their products, and influencers joining those campaigns to advertise the products. The sponsors can keep track of their campaigns, and request influencers to join their campaigns. In a similar fashion influencer and request a sponsor to join their publicly made campaigns. Admin has the power to manage all the users, and campaigns.

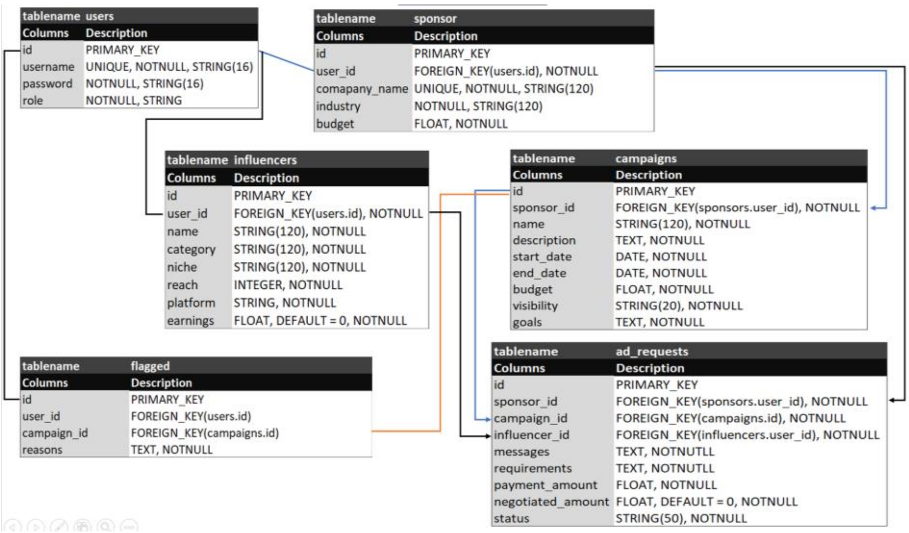
Technologies used

1. Flask: used for building the web application.
2. Flask-SQLAlchemy: extension of Flask, used to handle database connections across the app.
3. Matplotlib: is a comprehensive python library used for plotting data to give us insights.
4. HTML, CSS, Jinja2, BOOTSTRAP: are used to build the frontend of this application.

Database Schema

The following is my data base schema. I have a one-to-one relationship between users and sponsors, users and influencers, users and flagged, and campaigns and flagged tables.

There is a one-to-many relationship between sponsors and campaigns, campaigns and ad_requests, influencers and ad_requests.



tablename	users
Columns	Description
id	PRIMARY_KEY
username	UNIQUE, NOTNULL, STRING(16)
password	NOTNULL, STRING(16)
role	NOTNULL, STRING

tablename	sponsor
Columns	Description
id	PRIMARY_KEY
user_id	FOREIGN_KEY(users.id), NOTNULL
comapany_name	UNIQUE, NOTNULL, STRING(120)
industry	NOTNULL, STRING(120)
budget	FLOAT, NOTNULL

tablename	influencers
Columns	Description
id	PRIMARY_KEY
user_id	FOREIGN_KEY(users.id), NOTNULL
name	STRING(120), NOTNULL
category	STRING(120), NOTNULL
niche	STRING(120), NOTNULL
reach	INTEGER, NOTNULL
platform	STRING, NOTNULL
earnings	FLOAT, DEFAULT = 0, NOTNULL

tablename	campaigns
Columns	Description
id	PRIMARY_KEY
sponsor_id	FOREIGN_KEY(sponsors.user_id), NOTNULL
name	STRING(120), NOTNULL
description	TEXT, NOTNULL
start_date	DATE, NOTNULL
end_date	DATE, NOTNULL
budget	FLOAT, NOTNULL
visibility	STRING(20), NOTNULL
goals	TEXT, NOTNULL

tablename	flagged
Columns	Description
id	PRIMARY_KEY
user_id	FOREIGN_KEY(users.id)
campaign_id	FOREIGN_KEY(campaigns.id)
reasons	TEXT, NOTNULL

tablename	ad_requests
Columns	Description
id	PRIMARY_KEY
sponsor_id	FOREIGN_KEY(sponsors.user_id), NOTNULL
campaign_id	FOREIGN_KEY(campaigns.id), NOTNULL
influencer_id	FOREIGN_KEY(influencers.user_id), NOTNULL
messages	TEXT, NOTNUTLL
requirements	TEXT, NOTNUTLL
payment_amount	FLOAT, NOTNULL
negotiated_amount	FLOAT, DEFAULT = 0, NOTNULL
status	STRING(50), NOTNULL

Wireframe

The following are the routes used in the project.

@app.route('/')
@app.route('/register')
@app.route('/register/sponsors', *methods* = ['GET', 'POST'])
@app.route('/register/influencers', *methods* = ['GET', 'POST'])
@app.route('/login', *methods* = ['GET', 'POST'])
@app.route('/logout')

Admin Routes:

@app.route('/admin/dashboard', *methods* = ['GET', 'POST'])
@app.route('/admin/users', *methods* = ['GET', 'POST'])
@app.route('/admin/campaigns', *methods* = ['GET', 'POST'])

Sponsor Routes

@app.route('/sponsor/dashboard', *methods* = ['GET', 'POST'])
@app.route('/sponsor/edit_profile', *methods* = ['GET', 'POST'])
@app.route('/sponsor/create_campaign', *methods* = ['GET', 'POST'])
@app.route('/sponsor/send_request', *methods* = ['GET', 'POST'])
@app.route('/sponsor/manage_requests', *methods* = ['GET', 'POST'])
@app.route('/sponsor/manage_campaigns', *methods* = ['GET', 'POST'])
@app.route('/sponsor/edit_campaign', *methods* = ['POST'])
@app.route('/sponsor/delete_campaign', *methods* = ['POST'])

Influencer Routes:

@app.route('/influencer/dashboard', *methods* = ['GET'])
@app.route('/influencer/edit_profile', *methods* = ['GET', 'POST'])
@app.route('/influencer/ad_requests', *methods* = ['GET', 'POST'])
@app.route('/influencer/search_campaigns', *methods* = ['GET', 'POST'])

Architecture & Structure

The project code is organized based on its utility in different files. I named my project AdVeri. Here is the consolidate structure of the folder.

- AdVeri
    - instance
        - iescp.db – the database for the project.
    - static
        - images – contains the images that were used for stats.
        - styles.css
    - templates – I have classified admin, influencer and sponsor specific templates in respective folders, and the login, register (including for influencers and sponsors), home pages are present in this folder.
    - app.py – contains the routes for the backend.
    - models.py – contains the database model.
    - requirements.txt – contains all the requirements of the backend required for this project.

Video

The following the link to my project demo video.
[*https://drive.google.com/file/d/1QFrfUddkynHown3K_cc-uhX7tnreuqsq/view?usp=sharing*](https://drive.google.com/file/d/1QFrfUddkynHown3K_cc-uhX7tnreuqsq/view?usp=sharing)