MAD-2 Project Report

IESCP V2

Name: Tanmay Malhotra

Roll number: 23f1002693

Email : 23f1002693@ds.study.iitm.ac.in

About me

I am currently pursuing my diploma level of the BS data science degree. This was my second take at the IESCP problem, this time for MAD 2 project. I have gained hands on experience in skills such as Vue.js, Redis, Celery and Flask via this project.

Project Description

I have developed a Flask based web application for a sponsor coordination platform, named SPOC V2 (short for Sponsor Coordination). In this project, I have implemented CRUD functionalities for campaigns and ad requests using VueJS(CLI) for frontend and Flask for APIs. I have also implemented the functionality of scheduled jobs and reminders using redis and celery.

Tech Stack

- Flask: Backend/API development.
- Flask-Restful: To develop the Restful based API for the app.
- Flask-Security: For Token based authentication.
- Flask-mail: For sending monthly report emails.
- Flask-CORS: To enable CORS for the app.
- VueJS(CLI): Frontend part of the application
- Sqlite3 Flask-SQLAlchemy: Simplified database interaction with ORM
- SQLite: SQL database engine for data storage
- Redis and Celery are used for scheduled jobs/reminders using Google Chat and Mail

Database Schema Design Description:

The database is implemented using SQLite and Flask-SQLAlchemy.

- The 7 tables used in the database are User,Role,UserRoles,SponsorProfile,InfluencerProfile,Campaign,AdRequest
- Roles of Users are stored in the UserRoles table.
- SponsorProfile and Campaign have a one-to-many relationship.

- AdRequest tracks requests by sponsors to influencers, linking to both Campaign and InfluencerProfile.

Constraints

- Primary Keys ensure unique identification of records in each table.
- Foreign Keys establish relationships between tables, maintaining referential integrity.

Reasoning

The structure efficiently manages platform data with relationships between users, roles, campaigns, and ad requests. Foreign keys enforce referential integrity, ensuring that user roles, profiles, and requests are consistently linked.

API Design

CRUD on Campaigns and AdRequests

Description : Campaign and AdRequest Api allow the respective user to perform CRUD operations on Campaigns and AdRequests. It supports HTTP methods like GET, POST, PUT, and DELETE to manage the data in the database.

- API for Influencers and Sponsors to Sign up/Sign in
- API for Sponsors to manage campaign and ad requests, and API for Influencers to manage ad requests.
- API for admin to manage the application, and flag/ delete influencer, sponsor, campaign.

Architecture

The application follows MVC structure

Model(M) is implemented using flask. It enables interaction with the database.

View(V) is handled by vue.js. vue components create the user interface

Controller (C) is handled by flask. All the business logic is implemented using flask

I have implemented all the core functionalities.

Video

link

Kindly paste the link in browser if the above link doesn’t work

https://drive.google.com/file/d/19h2ya3RnbeW8Pl2_u9Ssp5o9CFGXq68A/view?usp=sharing