Modern Application Development 2

Project Report

Submitted by

Name: **Shreya Shambhavi**

Roll Number: **21F1001507**

1. Introduction

I am a Computer Science enthusiast, learning little by little every day. The MAD 2 Project has provided me with a great learning experience and I am ready to build more complex applications.

2. Project Details

IESCP is a dynamic platform for influencers and businesses to connect and flourish together. The core of the application is based on sponsors getting their products and services promoted and advertised by influencers in exchange of remuneration. IESCP acts as a medium where influencers and sponsors can register and login to their personal dashboards. An admin dashboard also exists for the admin to monitor the platform dynamics. Sponsors can create campaigns and ad requests, which can be accepted or rejected by influencers, while the latter can also send ad requests to the sponsors.

3. Frameworks and Technologies Used

- **Flask** - for backend server
- **Flask - Security and Login** - for role and token-based authentication
- **Flask - SQLAlchemy** - for managing relational database (Object Relational Mapper)
- **Flask - Restful** - for API resource
- **Flask - CORS -** for cross origin resource sharing
- **Flask - Excel -** for generating CSV files
- **Flask - Caching** - for caching backend routes
- **SQLite** - for storing database
- **Vue 3 CLI** - for application UI
- **Vue Router 4** - for routing frontend
- **Vuex Store** - for storing login state
- **Axios -** for frontend to backend requests
- **Bootstrap** - for styling UI
- **Redis -** for caching and message broker
- **Celery -** for async and batch jobs
- **Jinja2** - for monthly report HTML template
- **Mailhog** – for daily reminder and monthly report mailing

4. Database Schema Design

The model consists of seven entities: ‘User’, ‘Role’, ‘UsersRoles’, ‘Influencer’, ‘Sponsor’, ‘Campaign’ and ‘AdRequest’.

1. **User**: id, username, password, active, fs_uniquifier
2. **Role**: id, name, description
3. **UsersRoles**: id, user_id, role_id
4. **Influencer**: id, user_id, name, email, category, reach, description, flag_status
5. **Sponsor:** id, user_id, name, email, industry, valuation, description, flag_status
6. **Campaign:** id, sponsor_id, name, start_date, end_date, budget, visibility, target_audience, status, description, flag_status
7. **AdRequest:** id, campaign_id, influencer_id, title, message, requirements, payment, status, owner, work_status

- **Database ER Diagram: Click Here**

5. API Design

APIs for user, influencer, sponsor, campaign and adrequest have been created using Flask-Restful for all the CRUD operations in the application to ensure structures data exchange.

6. Architecture and Features

A. The application root folder consists of backend and frontend folder with the former consisting of the backend routes, the API endpoints, database and all the other instances. The frontend folder consists of all vue and javascript files.

B. Flask – Security takes care of authentication and tokenization for RBAC. Navigation guard has been added at frontend for any unauthorized navigation on the application.

C. The sponsor using IESCP can create campaigns (public and private) and ad requests and can perform CRUD operations on them. They can search for influencers and also receive ad requests from the latter. At the same time, the influencers can search for public campaigns as well as accept or reject the ad requests sent by the sponsors. Both the sponsors and influencers can edit their profile and operate if and only if they are unflagged.

D. The admin can control the mechanism of IESCP; they can monitor the influencer and sponsor statistics as well as flag and further remove influencers, sponsors and campaigns.

E. The structure of IESCP is based on Model-View-Controller (MVC) architecture.

F. Tasks like daily reminder for influencers with pending ad requests along with monthly reports for sponsors to analyze their investment is scheduled.

7. Presentation Video

- **Presentation Video: Click Here**