**Project Details**

**Project Title**: ***Influencer Engagement & Sponsorship Coordination Platform - V2***

**Problem Statement**:  
The project addresses the challenge of managing campaigns between sponsors and influencers in a streamlined manner. Sponsors often face difficulties in keeping track of campaigns, evaluating the performance of influencers, and handling requests. On the other hand, influencers need a platform to manage their active campaigns, submit proposals, and view the status of requests from sponsors. The goal was to create a system that facilitates smooth interactions between sponsors, influencers, and campaigns, while maintaining transparency and performance tracking.

**Approach**:  
We designed and developed a full-stack web application that allows multiple roles (Admin, Sponsor, Influencer) to manage campaigns.

* + **Sponsors**: Can create and view active campaigns, manage new requests, and accept or reject proposals from influencers.
  + **Influencers**: Can view campaigns, submit requests to join a campaign, and manage their participation.
  + **Admin**: Can manage all users, campaigns, and perform backend tasks.  
    The application ensures role-based access to features through user authentication and JWT token-based authorization.

**Frameworks and Libraries Used**

* + **Frontend**: Vue.js, Vuex, Bootstrap for building an interactive, responsive user interface.
  + **Backend**: Flask for handling API requests and managing business logic.
  + **Database**: SQLite for storing campaign, user, and interaction data, using SQLAlchemy ORM.
  + **Authentication**: flask\_security (auth\_required) for secure, token-based user authentication.

**API Resource Endpoints**

1. **Authentication Endpoints**:

* POST /auth/login: Login a user (Admin, Influencer, Sponsor).
* POST /auth/register: Register a new user.

1. **Campaign Management**:

* GET /api/get\_sponsor\_campaign/<int:id>: Get all campaigns for a specific sponsor.
* POST /api/create\_campaign: Create a new campaign.
* POST /api/request\_campaign: Submit a request to join a campaign.

**Request Management**:

* GET /api/get\_requests/<int:sponsor\_id>: Get all requests for a sponsor.
* POST /api/accept\_request: Accept an influencer's request to join a campaign.
* POST /api/reject\_request: Reject a request.

**Instruction to run the application via terminal**

* cd backend
* python main.py

open new terminal

* cd frontend/vfrontend
* npm run serve

open the url link for frontend folder eg: `http://loalhost:8080/`

**ER Diagram**

**A diagram of a computer

Description automatically generated**

**Drive Link to Presentation Video**

https://drive.google.com/file/d/1nA3VW1A9x0Kt-VD7Wq38z\_aF3lWQrBjF/view?usp=drive\_link

This project demonstrated our ability to solve real-world problems by creating a scalable, secure, and user-friendly platform for managing influencer marketing campaigns.Top of Form