Influencer Engagement and Sponsorship Coordination Platform

Student Details

- **Name:** Amreen Parveen
- **Roll Number:** 21f1006805

Video walkthrough

Problem Statement: It's a platform to connect Sponsors and Influencers so that sponsors can get their product/service advertised and influencers can get monetary benefit.

Objective : The objective of this project is to develop a platform that streamlines these tasks by enabling sponsors to create campaigns, interact with influencers, and manage ad requests effectively.

Approach:

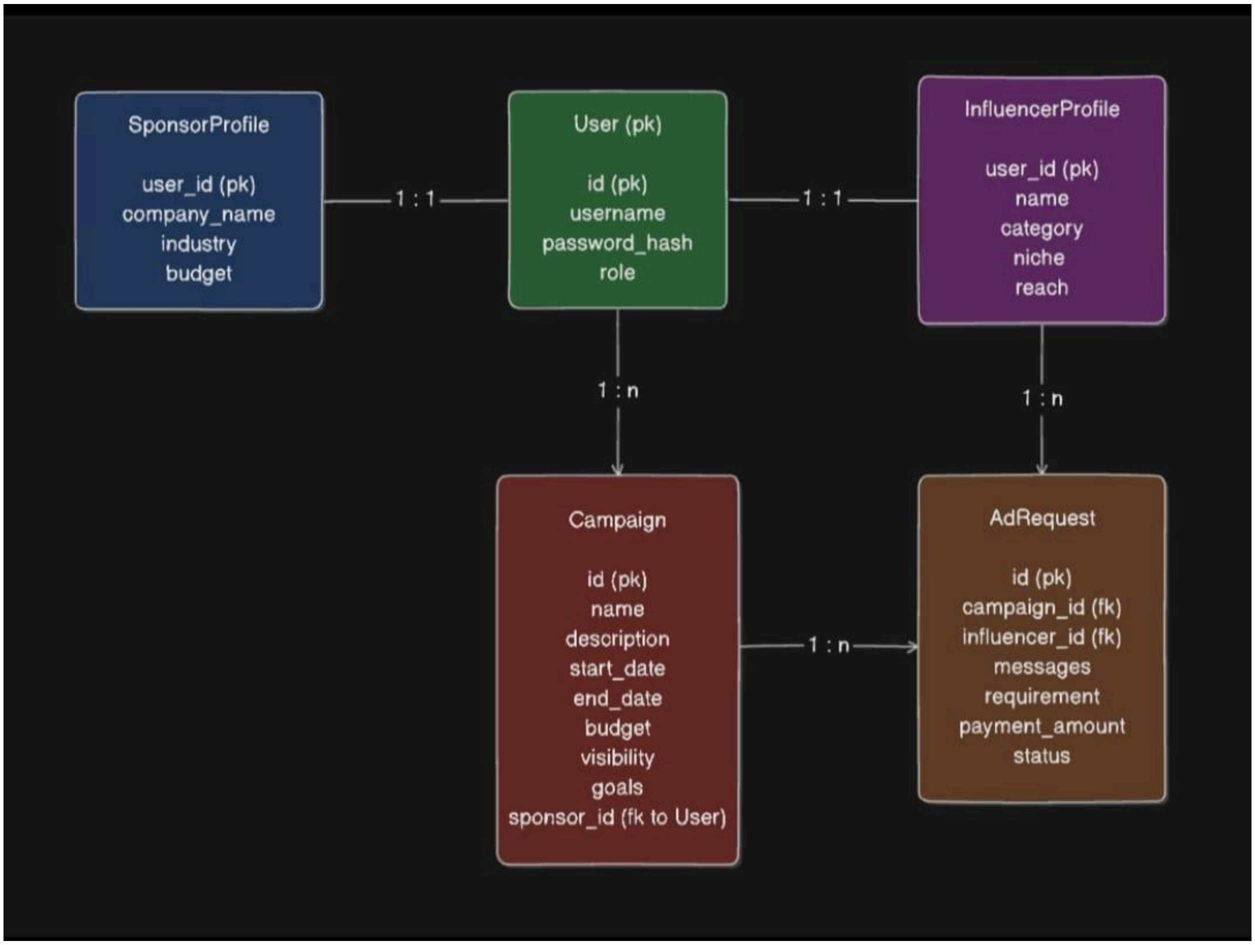
- Design the relational database
- Develop backend using Flask
  - user authentication
  - campaign management
  - ad request handling
  - search functionalities
  - role based control

Frameworks and Libraries Used

- **Backend:**
  - **Flask:** A lightweight web framework for handling HTTP requests, routing, and API development.
  - **SQLite:** A lightweight database used for storing user data, campaigns, and ad requests.
  - **Flask-Login:** For managing user sessions and authentication.
  - **Flask-Bcrypt:** For password hashing and security.
- **Frontend:**
  - **VueJS:** A progressive JavaScript framework used for building the user interface.
  - **Bootstrap:** A CSS framework used for responsive design and styling.
- **Other Libraries:**
  - **SQLAlchemy:** An ORM (Object-Relational Mapper) used for database interactions.
  - **Jinja2:** A templating engine used with Flask for rendering HTML templates.

ER Diagram

Below is the ER diagram representing the database schema for the platform:



Important API Resource Endpoints

1. **User Authentication:**
   - `POST /login`: User login
   - `POST /register`: User registration
2. **Campaign Management:**
   - `GET /sponsor/campaigns`: View all campaigns created by the sponsor
   - `POST /sponsor/campaigns/new`: Create a new campaign
   - `POST /sponsor/campaigns/<int:campaign_id>/update`: Update an existing campaign
   - `POST /sponsor/campaigns/<int:campaign_id>/delete`: Delete a campaign
3. **Ad Request Management:**
   - `POST /sponsor/ad_request/new`: Create a new ad request

  - `POST /sponsor/ad_request/<int:campaign_id>/update`: Update an existing `ad request`
  - `POST /sponsor/ad_request/<int:campaign_id>/delete`: Delete an `ad request`
  - `POST /sponsor/ad_request/respond/<int:ad_request_id>`: Sponsor responds to an ad request (accept, reject, counter)
  - `POST /influencer/ad_request/new`: Create a new ad request
  - `POST /influencer/ad_request/<int:campaign_id>/update`: Update an existing `ad request`
  - `POST /influencer/ad_request/<int:campaign_id>/delete`: Delete an `ad request`
  - `POST /influencer/ad_request/respond/<int:ad_request_id>`: Influencer responds to an ad request (accept, reject, counter)

4. **Search Functionalities:**
   - `GET,POST /influencer/campaigns/search`: Search for public campaigns
   - `GET,POST /sponsor/influencers/search`: Search for influencers based on various criteria
   - `GET /admin/search`: Search for influencers,campaigns or sponsors
5. **User Dashboards:**
   - `GET,POST /sponsor/profile`: View/Update Sponsor Profile
   - `GET,POST /influencer/profile`: View/Update Sponsor Profile
   - `GET,POST /admin/profile`: View/Update Sponsor Profile