

Student Details:

Name: Hiya Syal

Roll Number: 23F3003216

Project Statement: Influencer Engagement and Sponsorship Coordination Platform

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

Approach:

Created a registration form for sponsors and influencers, then created a login form which is common for admin, influencer, and sponsor.

Made the models and defined the relationships, overlaps, and back references.

Added functionality where the admin can view all relevant stats, flag users, reinstate them, and also delete the user from the app.

When the admin flags someone, that user can login but will not be able to participate in the app.

Added functionality where sponsors can create campaigns, search for influencers, and send ad requests for a particular campaign. Sponsors can create multiple campaigns and track each individual campaign.

Sponsors can also edit and delete campaigns and do the same with ad requests.

Added functionality where an influencer will receive ad requests, accept or reject ad requests, and when the influencer marks them as completed, the sponsor can confirm the completion and then make a payment to the influencer. This will all be updated in the database and the front end.

Influencers can update their profile, and so can sponsors. Additionally, the influencer's page will be publicly accessible, but not the sponsor's.

Ensured proper user login authentication using Flask-Login. My app can differentiate all users.

Sponsors can search for relevant influencers based on their niche and followers, and influencers can search for campaigns.

Added functionality where public campaigns will be visible to all influencers, but private campaigns will only be visible to the influencers whose niche matches with the niche of the particular campaign.

Added the total budget and remaining budget option in the app for the sponsor, which will be updated.

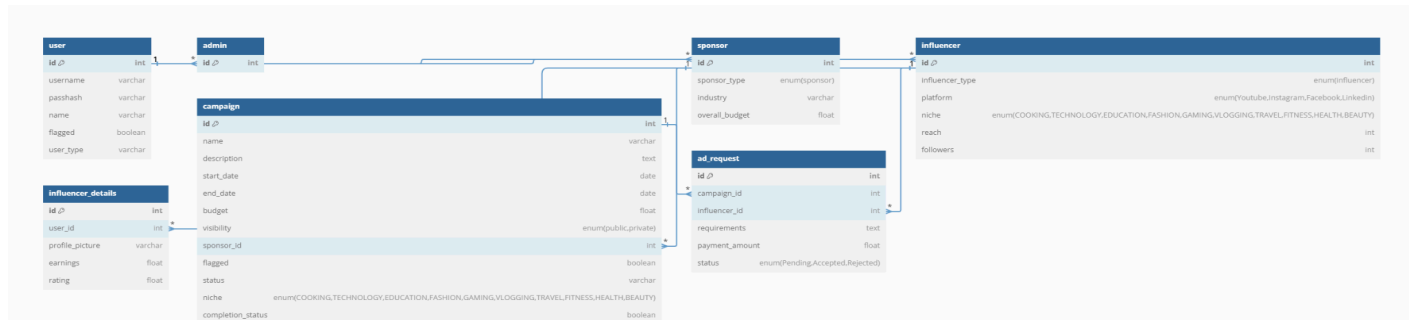
Provided styling and aesthetics to the application by creating a beautiful and responsive frontend.

Implemented a dummy payment portal (just a view taking payment details from sponsors for an ad request, and when clicked on pay, the payment was done).

Frameworks and Libraries Used:

- Flask
- SQLAlchemy
- Jinja2
- WTForms
- Flask-Bootstrap
- Flask-WTF
- Flask-Login
- werkzeug.security (for generate_password_hash and check_password_hash)
- Flask-Migrate

ER Diagram:



1. **User**
 - **PK:** id
 - Attributes: username, passhash, name, flagged, user_type
2. **Admin** (inherits from User)
 - **PK:** id (FK to User.id)
3. **Sponsor** (inherits from User)
 - **PK:** id (FK to User.id)
 - Attributes: sponsor_type, industry, overall_budget
 - **One-to-Many** with Campaign
4. **Influencer** (inherits from User)
 - **PK:** id (FK to User.id)
 - Attributes: influencer_type, platform, niche, reach, followers
 - **One-to-One** with InfluencerDetails
 - **One-to-Many** with AdRequest
5. **InfluencerDetails**
 - **PK:** id
 - **FK:** user_id (to Influencer.id)
 - Attributes: profile_picture, earnings, rating
6. **Campaign**
 - **PK:** id
 - Attributes: name, description, start_date, end_date, budget, visibility, sponsor_id, flagged, status, niche, completion_status
 - **One-to-Many** with AdRequest
7. **AdRequest**
 - **PK:** id
 - Attributes: campaign_id, influencer_id, requirements, payment_amount, status
 - **Many-to-One** with Campaign
 - **Many-to-One** with Influencer

Drive link of the presentation video:

<https://drive.google.com/file/d/1zMqcwvX9dKrjNIIIPRVogByp1U1y0XBnA/view?usp=sharing>