Student details

Name - Shreya Saxena
Roll number - 22f3001013
Email - 22f3001013@ds.study.iitm.ac.in

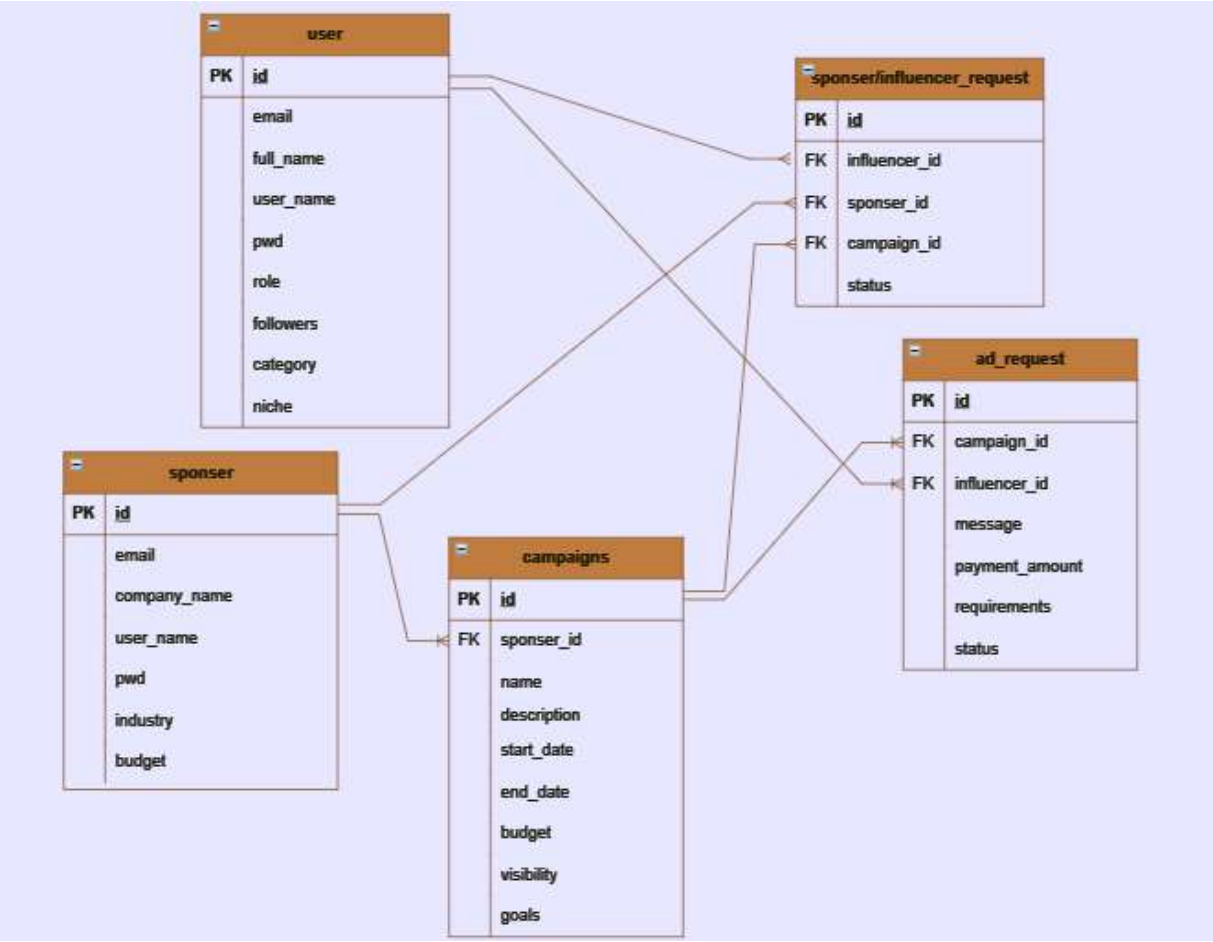
Description

A web application and an API is created for multiuser interaction which allows the sponsors to create and edit their campaigns and the influencers can view the public campaigns and send request for a campaign . The sponsor in response can accept the influencers request and can also send a request to a particular influencer for a private campaign, the influencer can either accept or reject the request. Each user has their own personalised dashboards and can search others based on their username/niche/followers etc.

Frameworks and libraries used

- **Flask framework** was used for developing the application.
- **Jinja2** was used for templating and HTML generation.
- **Bootstrap 5** was used for styling and designing purpose.
- **Flask-SQLAlchemy** and **sqlite3** were used for database operations.
- **Flask-RESTful** was used for implementing the RESTful APIs.
- **Matplotlib** is used for generating the insights on campaigns, sponsors, influencers with the help of histogram plots.

DB Schema Design



- **User table:** id is *PRIMARY KEY* and *AUTO-INCREMENTED.* All the attributes are not null. Default value of type is *general.* This table uses a backref relationship with ad_request table.
- **Sponsor table:** id is *PRIMARY KEY* and *AUTO-INCREMENTED.* . All the attributes are not null. This table uses a backref relationship with campaign table.
- Campaign table : id is *PRIMARY KEY* and *AUTO-INCREMENTED.* . All the attributes are not null. sponsor_id is *FOREIGN KEY* from Sponsor table. Default value of start_date is the system date. Default value of visibility is public.
- Ad_request table: id is *PRIMARY KEY* and *AUTO-INCREMENTED.* . All the attributes are not null. campaign_id is *FOREIGN KEY* from Campaign table and influencer_id is *FOREIGN KEY* from user table. Default value of status id pending.
- Sponser/Influencer_request table: : id is *PRIMARY KEY* and *AUTO-INCREMENTED.* influencer_id is FOREIGN KEY from User table, sponsor_id is FOREIGN KEY from Sponser table, and campaign_id is FOREIGN KEY from Campaign table. Default value of status is pending.

There are multiple sponsors and each sponsor can create multiple campaigns and ad requests. **One-to-Many relationships** exists between User - Ad request table, Sponsor - Campaign table and Campaign - Ad Request table.

API Design

There are 3 resources created for API using the **Resource** class from **Flask-RESTful.** They can handle *GET/POST/PUT/DELETE* or *CRUD* requests and their response is in *JSON* format.

- **InfluencerApi –** Resource for CRUD on User Model
- **SponserApi –** Resource for CRUD on Sponser Model
- **CampaignApi –** Resource for CRUD on Campaign Model

Endpoints are

- **/api/influencer/{influencer_id} –** for creating, reading, updating and deleting the user/influencer.
- **/api/sponser/{sponser_id} -** for creating, reading, updating and deleting the sponser
- **/api/campaigns/{sponser_id} –** for creating and reading the campaigns
- **/api/campaigns/update/{campaign_id} –** for updating a campaign
- **/api/campaigns/delete/{campaign_id} –** for deleting a campaign

Architecture of the application

The root folder consists of app.py and four more folders named backend (which consists of the controller.py file , api_controller.py file and models.py file), instance(consists of the database file), static (consists of css files, images folder, and the stats image folder) and templates(which has all the html files).

Video

https://drive.google.com/file/d/126vWK-SMctwaoMGY94QaLKlyoGnVXz5s/view?usp=drive_link