PROJECT REPORT

Influencer Engagement and Sponsorship Coordination Platform

Modern Application Development - 1

Vaishnavi Saxena

Roll No. - 23f2003009
Email: 23f2003009@ds.study.iitm.ac.in

PROBLEM STATEMENT

The project aims to develop a platform that facilitates mutual benefits—sponsors discover, connect and collaborate with influencers and gain exposure for their products, and influencers receive monetary benefits.

FRAMEWORKS USED

1. **Flask** — backend application code
2. **Flask SQLAlchemy** — ORM for database management
3. **SQLite** — for data storage
4. **Jinja2 Templates** — for HTML template rendering
5. **Bootstrap** — for frontend styling

API DESIGN

The following API endpoints have been created for CRUD of Campaigns

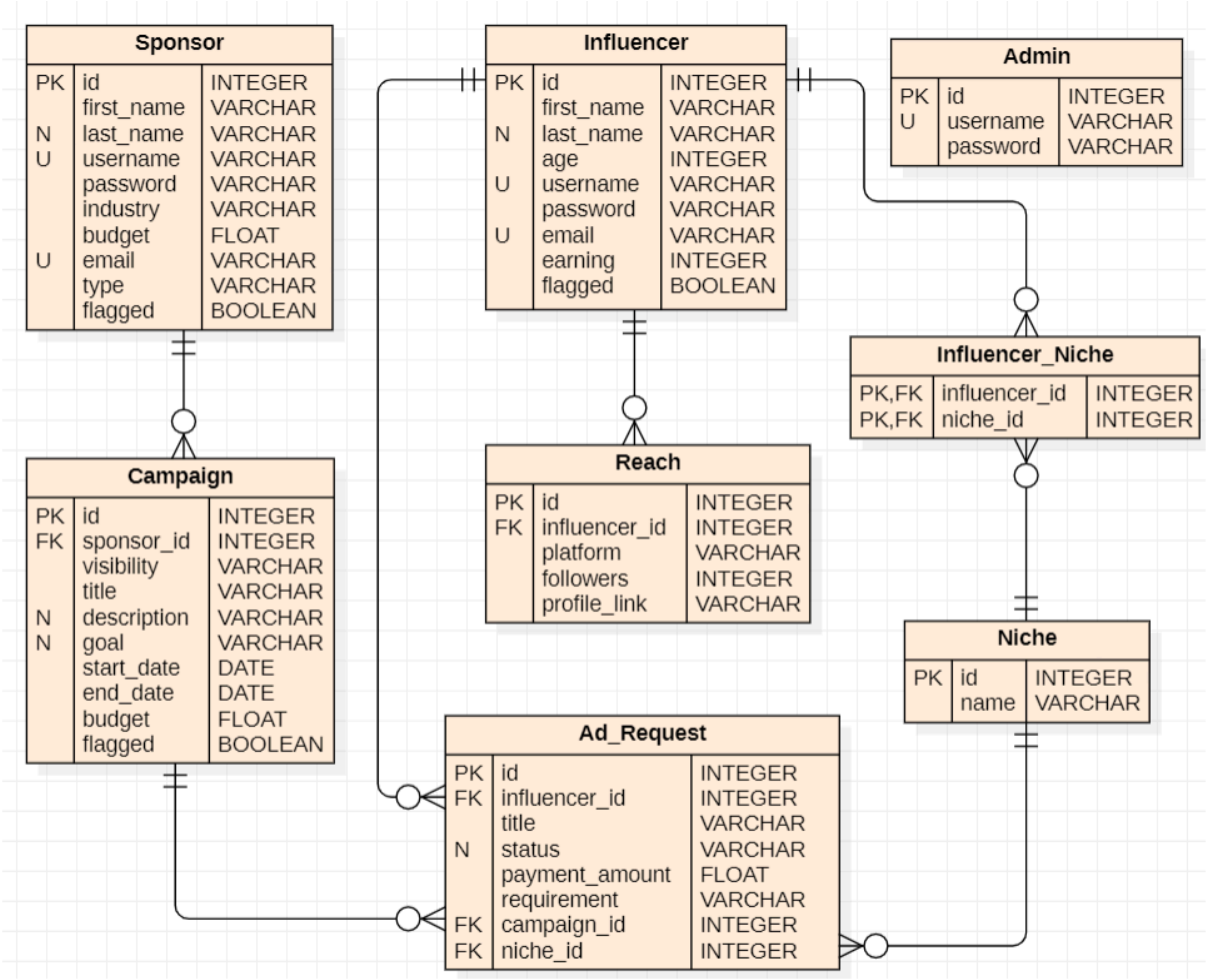
1. **GET** *api/campaign/<int:sponsor_id>*
   - Retrieve all campaigns of the sponsor with given sponsor ID
2. **POST** *api/campaign/<int:sponsor_id>*
   - Creates a new campaign for the sponsor with given sponsor ID
3. **PUT** *api/campaign/update/<int:camp_id>*
   - Updates the campaign with the given campaign ID and responds with a

success message indicating that the campaign has been updated

4. **DELETE** *api/campaign/delete/<int:camp_id>*
   - Deletes the campaign with the given campaign ID and responds with a success message indicating that the campaign has been deleted.
5. **GET** *api/sponsor/<int:sponsor_id>*
   - Retrieving details of the sponsor with the given sponsor ID
6. **GET** *api/influencer/<int:influencer_id>*
   - Retrieving details of the influencer with the given influencer ID

ENTITY-RELATIONSHIP DIAGRAM

All tables have the id attribute as the primary key.



Sponsor		
PK	id	INTEGER
	first_name	VARCHAR
N	last_name	VARCHAR
U	username	VARCHAR
	password	VARCHAR
	industry	VARCHAR
	budget	FLOAT
U	email	VARCHAR
	type	VARCHAR
	flagged	BOOLEAN

Influencer		
PK	id	INTEGER
	first_name	VARCHAR
N	last_name	VARCHAR
	age	INTEGER
U	username	VARCHAR
	password	VARCHAR
U	email	VARCHAR
	earning	INTEGER
	flagged	BOOLEAN

Admin		
PK	id	INTEGER
U	username	VARCHAR
	password	VARCHAR

Influencer_Niche		
PK,FK	influencer_id	INTEGER
PK,FK	niche_id	INTEGER

Campaign		
PK	id	INTEGER
FK	sponsor_id	INTEGER
	visibility	VARCHAR
	title	VARCHAR
N	description	VARCHAR
N	goal	VARCHAR
	start_date	DATE
	end_date	DATE
	budget	FLOAT
	flagged	BOOLEAN

Reach		
PK	id	INTEGER
FK	influencer_id	INTEGER
	platform	VARCHAR
	followers	INTEGER
	profile_link	VARCHAR

Niche		
PK	id	INTEGER
	name	VARCHAR

Ad_Request		
PK	id	INTEGER
FK	influencer_id	INTEGER
	title	VARCHAR
N	status	VARCHAR
	payment_amount	FLOAT
	requirement	VARCHAR
FK	campaign_id	INTEGER
FK	niche_id	INTEGER

PROJECT FOLDER ARCHITECTURE

'IESCP_MAD1_Project' is the project's root folder. It contains a folder for all the code ('IESCP_Code') and a project report pdf file. Within the Code folder, we have —

1. **'app.py'** file that contains the code for initializing the application
2. A folder **'backend'** that contains —
    - **'controllers.py'** : with all the routes and the user defined functions.
    - **'models.py'** : with the schema/model definition of the database.
    - **'api_controllers.py'** : with the API resource classes.
3. A folder **'instance'** that contains —
    - **'IESCP.sqlite3'** : the database file.
4. A folder **' static'** that contains —
    - A folder **'statistics'** : with the images of graphs for admin and users.
    - **'style.css'** : with some css code.
5. A folder **'templates'** that contains — all the HTML pages

PROJECT FEATURES

The flask app opens with the welcome page that navigates the client to the user login page.

1. **User Login —** If the user is already registered, he/she can login using their username and password. The app will check for the existence of the user in the database, if the user is not registered, an appropriate message will be prompted. According to the username and password provided, the user will be redirected to his appropriate dashboard.
2. **User Registration** — a user can register either as an influencer or as a sponsor. The app will check that the username being registered is not already present in the database.
3. **Admin Login** — an admin can not be registered but can login.
4. **Sponsor Functionalities** —
    - from her dashboard, a sponsor can create, edit and delete ongoing as well as scheduled campaigns.
    - within each campaign she can create, edit and delete Ad Requests.
    - she can find and filter relevant influencers according to the niches for her campaigns.

  - she can send ad requests to influencers.
  - she can accept/reject Ads requested by influencers.

5. **Influencer Functionalities —**
   - from her dashboard, an influencer can search for ongoing and scheduled campaigns.
   - she can search for and filter public campaigns according to their niches.
   - she can request to work on particular Ads.
   - she can see the Ad requests assigned to her by sponsors on her dashboard, she can view those requests and can accept or reject them.
6. **Admin Functionality —**
   - the admin can see all the active, scheduled and completed campaigns.
   - admin can search for campaigns and ads.
   - admin has the ability to flag and unflag sponsors, influencers or campaigns. (Ad requests of flagged campaigns can not be assigned to influencers and flagged users can not login into their dashboards.)
   - admin can see the overall statistics of the application viz.
     i. No. of registered sponsors and influencers.
     ii. No. of public and private campaigns.
     iii. No. of flagged and unflagged campaigns, sponsors and influencers.
     iv. No. of campaigns per sponsor.
     v. No. of ads per niche.

PRESENTATION VIDEO LINK

Vaishnavi_DemoVideo_IESCP_MAD1.mp4