

## **PROJECT REPORT:**

**Preeti Maurya**

Roll number : 23f2003521

Student email : [23f2003521@ds.study.iitm.ac.in](mailto:23f2003521@ds.study.iitm.ac.in)

Project Title: Influencer Engagement and Sponsorship Coordination Platform

### **Description:**

It's a platform to connect Sponsors and Influencers so that sponsors can get their product/service advertised and influencers can get monetary benefit. There can be multiple **Users(Sponsors and Influencers)** and **only one Admin**.

The platform have three roles;

1. **Admin** - root access
  - An admin can monitor all the users/campaigns, see all the statistics
  - Ability to flag inappropriate campaigns/users
2. **Sponsors** - a company/individual who wants to advertise their product/service
  - Sponsors will create campaigns, search for influencers and send ad requests for a particular campaign.
  - Sponsors can create multiple campaigns and track each individual campaign.
  - They can accept and negotiate ad requests by influencers for public campaigns.
3. **Influencers** - an individual who has significant social media following
  - An influencer will receive ad requests, accept or reject ad requests, negotiate terms and resend modified ad requests back to sponsors.
  - They can search for ongoing campaigns (which are public), according to category, budget etc. and accept the request.
  - An influencer can update their profile page, which is publicly visible.

### **Technologies used:**

- **Frontend:** HTML, little CSS and Bootstrap have been used for styling and responsiveness of app.
- **Template Engine:** Jinja2 has been used for HTML template generation.
- **Backend:** The application uses Flask, SQL Alchemy to implement core functionalities, in addition, matplotlib for generating graphs.

### **Architecture and features:**

The project has an application folder in root which contains a **templates** folder which has all the template, a **static** folder containing image files, **backend** folder that contains models.py (with all models used) and controllers.py file having all the routes plus one more apicontrollers.py file that handle api requests, an **instance** folder which contains the database file (SQLite) of Application, and a **PDF file** which contains project report.

### **Design:**

#### **1. Sponser Table:**

Parent table to Campaign

Column name	Type	Constraints
id	Integer	primary_key
sponser_company_name	String	Not null
password	String	Not null,unique
Industry	Text	Not null
Budget	Integer	Not null
flagged_status	Integer	Default=0

## 2. Influencer Table: Parent table to Ad\_request

Column name	Type	Constraints
id	Integer	primary_key
influencer_name	String	Not null
password	String	Null,unique
niche	Text	Not null
category	Text	Not null
platform	String	Not null
reach	Integer	Not null
Flagged_status	Integer	Default=0

## 3. Campaign :

Parent table to Ad request and child table to sponser

Column Name	Type	Constraints
id	Integer	Primary key
c_name	String	Not null
visibility	Integer	Default=0
description	Text	
budget	Integer	Not null
start_date	Date	default
end_date	Date	Not null
Campaign_status	String	default

Campaign_flagged_status	Integer	default
no_of_ads	Integer	Default
progress	Integer	default
goal	Text	
Sponsor_id	Integer	Foreign key
niche	Text	

#### 4.Ad request

Child table to Influencer and Campaign

Column Name	Type	Constraints
<b>id</b>	<b>Integer</b>	<b>Primary key</b>
<b>Send_by</b>	<b>Integer</b>	<b>Not null</b>
<b>msg</b>	<b>Text</b>	
<b>requirements</b>	<b>Text</b>	<b>Not null</b>
<b>Payment_amount</b>	<b>Integer</b>	<b>Not null</b>
<b>Ad_request</b>	<b>String</b>	<b>default</b>
<b>nego</b>	<b>Integer</b>	<b>default</b>
<b>Sponsor_id</b>	<b>Integer</b>	<b>Not null</b>
<b>Influencer_id</b>	<b>Integer</b>	<b>Foreign key</b>
<b>Campaign_id</b>	<b>Integer</b>	<b>Foreign key,Not null</b>

[Video link](#)