Modern Application Development -1 Project Report

IITM BS Degree Program

Name: Satyaki Goswami **Roll Number:** 21F2001257 **Email:** SATYAKI GOSWAMI

About Myself: I am a dedicated student who has recently started delving into the world of WebDev. After this project, I would like to explore more of this field.

1. Description

SyncIt is an Influencer Engagement and Sponsorship Coordination platform that helps to connect Sponsors and Influencers in order for the Influencers to to enjoy monetary benefits in exchange of advertising goods/product/services. Built in accordance with the guidelines for MAD - 1 Project for the May - 2024 term, this app provides separate registration pages for Influencers and Sponsors, while providing separate login pages for Influencers, Sponsors, and the Admin. Sponsors can create campaigns(public and private), and ad-request(s) for campaigns, which can be accepted or rejected by the Influencers. At the same time, Influencers can send ad-request for campaigns of their liking which can be accepted or rejected by the Sponsors. The mechanism is based on the database consisting of models of Influencers (inf_info), Sponsors (spon_info), Campaign (campaign), and Ad-Request (adreq).

2. Technologies Used

- **Flask :** This Python web framework has been used for backend development.
- **Flask-SQLAlchemy :** This library provides an object relational mapper.
- **SQLite3 :** Provides a lightweight and simple database storage engine
- **HTML :** It has been used to define the content and structure of the web pages
- **CSS :** Used for rudimentary styling and design
- **Jinja2 :** This has been used as a templating language
- **Chart.js :** This JavaScript library has been used to visualise data

3. Database Schema Design

- **Influencer Table (inf_info)**
  inf_id (Primary Key; integer), email (string; not null), full_name (string; not null), user_name (string; unique; not null), category (string; not null), niche (string; not null), no_of_followers (integer; not null), password (string; not null), inf_flag_status (string, default: “Unflagged”)
- **Sponsor Table (spon_info)**
  spon_id (Primary Key; integer), spon_type (string; not null), email (string; not null), spon_full_name (string; not null), spon_user_name (string; unique; not null), industry

(string; not null), evaluation (integer; not null), spon_password (string; not null), spon_flag_status (string; default: “Unflagged)

- **Campaign Table (campaign)**
  camp_id (Primary Key; integer), spon_id (Foreign Key; integer; not null), name (string; not null), description (text; not null), sdate (date; not null), edate (date; not null), budget (integer; not null), Visibility (string; not null), goals (text; not null), completion_status (string; default: “Active”), camp_flag_status (string; default: “Unflagged”)
- **Ad-Request Table (adreq)**
  req_id (Primary Key; integer), camp_id (Foreign Key; integer; not null), inf_id (Foreign Key; integer; not null), messages (text; default:”DEFAULT TEMPLATE”), requirements (text; not null), payment (integer; not null), req_status (string; default: “PENDING”), sender (string; default: “WAIT”)

LINK TO THE ER DIAGRAM

4. Architecture and Features

- SyncIt follows a Model-View-Controller (MVC) architecture, with “models.py”, in the backend folder, defining the database operations (Model), templates, which include the html templates, helping to view and manage the frontend (View), while “app.py” and “controllers.py” in the root folder taking care of routing and business logic (Controller).
- The Sponsor can create campaigns with the facility to update or delete them. At the same time, they can create multiple ad requests for each campaign, which could also be updated or deleted. They can search for Influencers with filtration features. Apart from this they can receive ad requests from Influencers as well which they can accept or reject.
- Influencers can search for campaigns (with filtering features) and view the details of each public campaign (private campaigns created by sponsors are not visible through search). They can also send ad requests to the Sponsors for a campaign of their choice which can be accepted or rejected by the Sponsors (updatable and deletable). Influencers can also accept or reject ad requests created by Sponsors.
- The Admin can monitor all the activities taking place on SyncIT. They can flag Sponsors, barring them from creating campaigns or being displayed through the Influencer’s search bar. When a campaign is flagged, the corresponding Sponsor won’t be able to create ad requests for it. Similarly, if an Influencer is flagged, they won’t be shown in the Sponsor’s search bar and they won’t be able to send an ad request. The Admin can also keep track of Influencer and Sponsor statistics.

5. Presentation Video

LINK TO THE PRESENTATION VIDEO