Modern Application Development I

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.

Frameworks to be used

These are the mandatory frameworks on which the project has to be built.

- Flask for application code
- Jinja2 templates + Bootstrap for HTML generation and styling
- SQLite for data storage

Note: All demos should be possible on your local machine.

Roles

The platform will have **three** roles;

- **Admin - root access**
  - An admin can monitor all the users/campaigns, see all the statistics
  - Ability to flag inappropriate campaigns/users
- **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 ad requests by influencers for public campaigns.
  - Each Sponsor may have;
    - Company Name / Individual Name
    - Industry
    - Budget

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.
Each Influencer profile may have;

Name
Category
Niche
Reach (can be calculated by number of followers / activity etc.)

Terminologies

Ad request : A contract between campaign and influencer, stating the requirements of the particular advertisement (E.g. show Samsung s23 in 3 videos for 10 seconds each), the amount to be paid etc.
Ad request may have:

campaign_id (Foreign Key to Campaign table)
influencer_id (Foreign Key to Influencer/user table)
messages
requirements
payment_amount
status (Pending, Accepted, Rejected)

Campaign : A container for ads requests for a particular goal (E.g. advertisement for Samsung s23). It can have multiple Ad requests, a campaign description, budget, ability to set public or private
Campaigns may have:

name
description
start_date
end_date
budget
visibility (public, private)
goals

Application Wireframe

IESCP_wireframe.png

Note: The wireframe is provided only to get the flow of the application and what should appear when a specific user navigates from one page to another. It is NOT mandatory to exactly replicate the views given in the wireframe. Students can work on their own frontend idea.

Core Functionalities

1 Admin login and user login

A login/register form with fields like username, password etc. for sponsor, influencer and admin login

You can create separate forms for each type of user

You can either use a proper login framework, or just use a simple HTML form with username and password (we are not concerned with how secure the login or the app is)

The app must have a suitable model to store and differentiate all the types of user of the app.

2. Admin Dashboard - for the Admin

The application must have an admin dashboard which display all the relevant statistics of the application, e.g. active users, campaigns (public/private), ad requests and their status, flagged sponsors/influencers etc.

Students can decide what more statistics to be shown apart from the ones given above

3. Campaign Management - for the sponsors

Create a new campaign and categorize it into various niches.

Update an existing campaign - e.g. start_date, end_date, budget and/or other fields

Delete an existing campaign

4. Ad request Management - for the sponsors

Create ad requests based on the goals on the campaign

Edit an existing ad request - e.g. influencer_id, requirements, payment_amount, status

Delete an existing ad request.

5. Search for influencers, public campaigns

The sponsors should be able to search for relevant influencers based on their niche, reach, followers etc.

The Influencers should be able to search for public campaigns based on their niche, relevance etc.

6. Take action on a particular ad request - for the Influencers

Ability to view all the ad requests from all the campaigns

Ability to accept/reject a particular ad request

Ability to negotiate the “payment_amount” for a particular ad

Recommended Functionalities

API resources created to interact with the users, ad requests and/or campaigns. (Please note: you can choose which API resources to create from the given ones, It is NOT mandatory to create API resources for CRUD of all the components)

APIs can either be created by returning JSON from a controller or using flask extension like flask_restful

External APIs/libraries for creating charts, e.g. ChartJS

Implementing frontend validation on all the form fields using HTML5 form validation or JavaScript

Implementing backend validation within the controllers of your app.