Project Report

Student Details

Name: Aarjav Gupta

Roll No. : 22f1000936

Institution: IITM

Course: BS Data Science

Date: 20/09/2024

Project Details

Project Title

Sponsify: A Platform Connecting Sponsors and Influencers

Problem Statement

The objective of this project is to create a web-based platform called Sponsify, which allows sponsors to create and manage advertising campaigns and influencers to find and accept ad requests. The platform must handle the following:

1. Sponsors should be able to create, view, and manage campaigns, including flagging inappropriate campaigns.
2. Influencers should be able to browse public campaigns, accept or reject ad requests, and manage their profiles.
3. Admins should monitor flagged campaigns and manage the overall platform integrity.
4. An authentication system will distinguish between different roles: Admin, Sponsor, and Influencer.

Problem Approach

The project approach involves breaking down the problem into multiple modules:

- Authentication and Session Management: Used a session-based authentication where users' roles (admin, sponsor, influencer) are stored in the session after logging in.
- Role-based Dashboards: Created separate dashboard views for each role with different functionalities (e.g., sponsors manage campaigns, influencers manage ad requests).
- Ad Request Management: Built a system for sending and managing ad requests between sponsors and influencers.
- Flagging Mechanism: Added functionality for sponsors to flag campaigns, visible to admins for moderation.

Frameworks and Libraries Used

1. Flask: The web framework used to develop the backend of the application. It was chosen due to its simplicity, flexibility, and the ability to integrate easily with Python libraries.
2. SQLAlchemy: An ORM (Object Relational Mapper) used to handle database operations, providing a clean abstraction over SQL queries.
3. Jinja2: The templating engine used in Flask to render HTML pages dynamically based on the user's role (admin, sponsor, influencer).
4. Bootstrap: A front-end framework that enabled rapid prototyping of the web pages with responsive designs.
5. FontAwesome: Used for icons throughout the platform to enhance the user experience.
6. SQLite: Chosen as the database for development due to its simplicity, ease of setup, and integration with Flask.

ER Diagram and Database Structure

Tables:

1. User: Stores basic information about users, including their role (Admin, Sponsor, Influencer).
2. Campaign: Represents the advertising campaigns created by sponsors. Includes fields such as name, description, start_date, end_date, budget, and is_flagged.
3. AdRequest: Tracks the interactions between campaigns and influencers. Includes fields for status (Pending, Accepted, Rejected), campaign_id, and influencer_id.
4. InfluencerProfile: Stores additional details specific to influencers, such as category, niche, and reach.
5. SponsorProfile: Stores additional details specific to sponsors, including company name, industry, and budget.

Relationships between Tables:

- User - Campaign: A one-to-many relationship where each sponsor (user with the role "Sponsor") can create multiple campaigns.
- Campaign - AdRequest: A one-to-many relationship where each campaign can have multiple ad requests.
- User - AdRequest: A many-to-many relationship facilitated by the AdRequest table, where influencers can interact with multiple campaigns, and a campaign can have multiple influencers.

Video demonstration Link :
https://drive.google.com/file/d/1pbSFZhaNXAdz2k0kL3Cfc4oUnqJ0TjSR/view?usp=sharing