Author

Name: Shrinidhi Kumar
Roll Number: 23f1002153
Student Email: 23f1002153@ds.study.iitm.ac.in
About Me: I'm a data science enthusiast who's starting to explore the world of web development. I love digging into data and finding patterns, but I'm also curious about how to build applications that people can actually use. This project was a great opportunity to blend both interests.

Description

For this project, I created a web application that helps sponsors and influencers collaborate more easily. The app includes features like logging in, managing campaigns, and an admin dashboard that shows important stats. My focus was on making the app functional and easy to use, while also ensuring that data is handled smoothly and securely in the background.

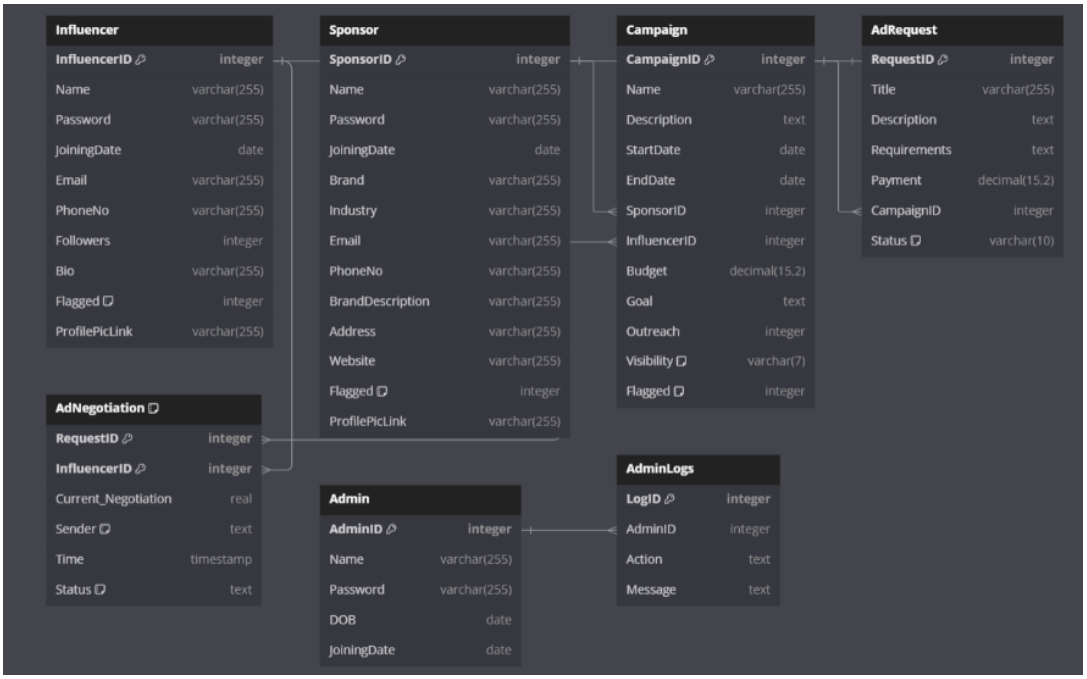
Technologies Used

- **Python**: The backbone of the project, handling all the logic and data processing.
- **Flask**: A simple yet powerful web framework that made routing and user management a breeze.
- **Flask-SQLAlchemy**: This extension was perfect for managing the database without getting lost in SQL code.
- **SQLite**: Chosen for its simplicity and reliability, SQLite stored all the important data like users and campaigns.
- **HTML/CSS**: These were used to build and style the frontend, keeping the design clean and user-friendly.
- **Matplotlib**: A trusty library for generating the charts that display key statistics on the admin dashboard.
- **Jinja2**: Made it easy to dynamically render content in HTML, adding flexibility to the templates.

Purpose: These technologies were selected because they work well together and allowed me to focus on both the web development and data visualization aspects of the project.

DB Schema Design

The database schema was designed to reflect the relationships between users, campaigns, and ad requests. Each table is structured to ensure data integrity and efficient querying. The detailed schema is illustrated in the ER diagram provided:



Error Handling and Data Validation

Throughout the project, I implemented error handling and data validation to ensure that the application runs smoothly and securely. This includes validating user input, handling database errors gracefully, and providing clear feedback to users when something goes wrong.

Architecture and Features

The project follows the MVC (Model-View-Controller) pattern, making it easier to manage and scale:

- **Controllers**: Handle all the application logic and user requests, located in the `application` directory, ending with *_controllers.py
- **Models**: Define the database structure and relationships using SQLAlchemy, found in `models.py`.
- **Templates**: The HTML files live in the `Templates` directory, and they're rendered dynamically using Jinja2.
- **Static**: All the static files like CSS and images are stored in the `static` directory.

Features:

- **User Authentication**: Admins, sponsors, and influencers can log in, and the system controls what each can access.
- **Campaign Management**: Sponsors can create, edit, and manage campaigns, while influencers can view and join them.
- **AdRequest Management**: Sponsors can create AdRequests on campaigns. They can also view, update or delete (CRUD Operations) on the AdRequests. Influencers have the ability to view AdRequests, negotiate on the price, and accept/reject the request.
- **Dashboard**: The admin dashboard shows useful stats, like top sponsors and influencers, through visual charts created with Matplotlib.
- **Additional Features**: I also implemented cascading deletes to keep the database clean when related records are removed.

Video

https://drive.google.com/file/d/1LEJGC6VO2tjVq4gR8paKXLLQMaqha7es/view?usp=sharing