PROJECT REPORT

GARVIT MAN SINGH
22f3002384@ds.study.iitm.ac.in
Phone number: 916396635684

I am a Diploma year student in Data Science and Programming Online Degree, passionate about machine learning, web development and eager to create impactful digital solutions through the before mentioned fields.

Description

Spin is an Influencer Engagement and Sponsorship Coordination Platform designed to connect sponsors with influencers. Sponsors can advertise their products or services, while influencers benefit monetarily. The web application allows users to manage their profiles, engage with content, and explore new connections. It provides functionalities for users to create, and view profiles, along with handling sponsorship requests and tracking engagements.

Technologies used

- **Flask**: Used for developing the core application logic.
- **Jinja2**: Utilized for templating and generating HTML.
- **Bootstrap**: Employed for styling and ensuring responsiveness across devices.
- **SQLite**: Chosen for data storage and management.
- **Flask-Login**: Used for managing user sessions and authentication.
- **Blueprint:** Used for routing different endpoints into separate files for better modularization.
- Various Python modules such as `os`, werkzeug, `datetime`, and `sys` were also used for various functionalities.

Database Schema Design

- **Sponsor**: This model represents a sponsor within the platform. It includes attributes for basic details such as name, username, email, and industry. It also tracks the number of ongoing and successful campaigns, provides a phone number, and includes an optional picture. Sponsors can have multiple campaigns and ad requests associated with them.

- **Influencer**: This model represents an influencer who participates in campaigns. It stores personal details like name, username, email, niche, phone number, date of birth, and gender. It also tracks the number of ongoing and successful campaigns, and includes an optional profile picture. Influencers are linked to various ad requests and campaigns.

- **Campaign**: This model represents a marketing campaign created by a sponsor and assigned to an influencer. It includes details such as the campaign name, description, start and end dates, budget, and visibility. Campaigns also track the number of posts and have an optional picture. Each campaign is associated with a sponsor and an influencer, and can have multiple ad requests.

- **AdRequest**: This model represents a request made by a sponsor for an influencer to participate in a campaign. It includes information such as the message, payment details, and number of posts requested. Each ad request is linked to a specific campaign, sponsor, and influencer, and records the date of the request.

The project models several key relationships between entities. **Sponsors** and **Influencers** both participate in **Campaigns**, creating a one-to-many relationship between them. **Campaigns** are linked to **AdRequests**, establishing another one-to-many relationship. **AdRequests** connect sponsors with influencers for specific campaigns. The relationships ensure that each sponsor can have multiple campaigns and ad requests, each influencer can participate in multiple campaigns and receive multiple ad requests, and each campaign can generate multiple ad requests.

.py files for endpoints

- **auth.py** - contains endpoints for Sign in for users.
- **profiles.py** - contains endpoints for user's profiles and dashboards.
- **camp_req.py** - contains endpoints for activities relating to campaigns and ad requests.
- **user.py** - contains endpoints for admin and home page activities.

Architecture and Features

The project utilizes Flask's structure with separate folders for better organization:

- **Root Folder**: Contains app.py for the main app, requirements.txt , virtual environment folder mad1project , instance folder which contains database named garage.db. Its also app folder which contains all the main files :
  - **static** : For images, CSS, and icons used in website.
  - **init.py** : used for all the initializations for app.
  - **models.py** : Defines database models.
  - **extensions.py** : Contains extensions for database variable.
  - **Controllers** : This folders contains all the .py files for routing along with their respective templates :
    - **Auth** : contains auth.py and templates folder for all authorization purposes.
    - **Profiles** : contains profiles.py and templates folder for routing related to profiles , dashboards and their functions.
    - **camp_req** : contains camp_req.py and templates folder for routing campaigns and ad requests and their functions .
    - **User** : contains user.py and templates for all endpoints relating to admin and home page.

Additional features include:

- Flash messages for success or failure notifications.
- Blueprints for modularisation.
- werkzeug for uploading images.
- os for constructing file paths.

- datetime for handling date and time .

Video

https://drive.google.com/file/d/1pCXWqgRj97qgsp9JtDIOr5Ojipt55g7J/view?usp=sharing