MAD 1 Project Report - Brand Bridge

1. Student Details:

Name: Manav Madhukar Dalvi
Roll no.: 21f1006451
Course: Modern Application Development Project
Institute: Indian Institute of Technology, Madras (I ITM)

2. Project Details:

2.1 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. [Demonstration Video]

2.2 Approach:

The project began with the goal of creating a platform where sponsors could post campaigns, and influencers could accept ad requests, negotiate terms, and earn monetary benefits. In the initial phase, I focused on developing a single-user platform that allowed sponsors to post and manage campaigns. Once this basic functionality was established, I extended the platform to include influencers, enabling them to interact with the posted campaigns, accept or reject ad requests, and negotiate terms. In the final phase, I implemented administrative functionality to manage the platform effectively. This included the ability to view, flag, and manage campaigns, sponsors, and influencers. The final touches involved refining the features and ensuring that the platform was both functional and user-friendly, completing the overall development process.

3. Frameworks and Libraries Used:

For the development of the Brand Bridge platform, several frameworks and libraries were utilized to ensure a robust and efficient application.

Flask - I used Flask as the core framework for building the web application. It facilitated the creation of routes, handling HTTP requests, and managing user sessions and authentication.

Jinja2 Templates - To generate dynamic HTML pages, I utilized the Jinja2 templating engine. This allowed me to create reusable templates and ensure a consistent design throughout the platform.

Bootstrap - Bootstrap was employed for responsive design and styling. It provided pre-designed components and CSS, making the user interface more appealing and user-friendly.

SQLite - For data storage, I chose SQLite due to its lightweight nature and ease of setup. It handled all the data related to users, campaigns, and ad requests.

Flask-SQLAlchemy - Flask-SQLAlchemy was used as the ORM (Object-Relational Mapping) tool. It simplified database interactions and allowed me to define database models for Sponsors, Campaigns, Influencers, and Ad Requests in Python.

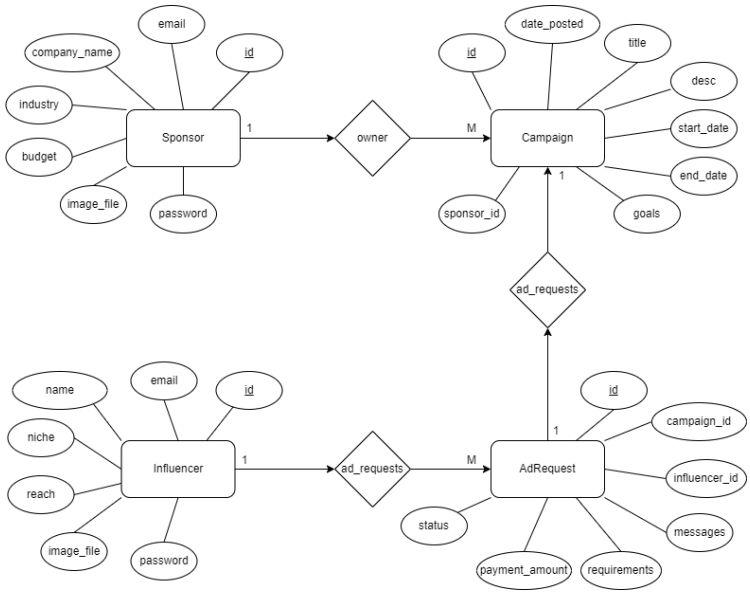
Flask-Bcrypt - To ensure secure storage of user passwords, I integrated Flask-Bcrypt. It provided password hashing utilities, enhancing the security of user authentication.

Flask-Login - Flask-Login was used to manage user sessions and authentication. It enabled the implementation of login and logout functionality, ensuring that users could securely access their accounts and manage their campaigns and ad requests.

4. Project Structure and Database Schema:

```
D:\Projects\IESCP>tree /f
Folder PATH listing for volume Data
Volume serial number is 160D-E670
D:.
    requirements.txt
    run.py

├──BrandBridge
│       forms.py
│       models.py
│       routes.py
│       __init__.py
│
│   ├──instance
│   │       site.db
│   │
│   ├──static
│   │   │   main.css
│   │   │
│   │   ├──images
│   │   │       Brand_icon.png
│   │   │       Brand_Icon_Final.png
│   │   │       brand_icon_test.png
│   │   │
│   │   └──profile_pics
│   │           default.jpg
│   │           default_company.png
│   │
│   └──templates
│           about.html
│           account.html
│           admin_dashboard.html
│           campaign.html
│           home.html
│           influencer_profile.html
│           layout.html
│           login.html
│           new_campaign.html
│           register.html
│           register_influencer.html
│           request_ad.html
│           search_results.html
│           sponsor_campaigns.html
│
└──instance
        site.db
```



5. API Endpoints:

In this project, the following are some of the main API resources and endpoints that were implemented using flask routes to facilitate interactions within the platform:

User Management: `**/login**` authenticates users and manages session initiation, `**/logout**` ends user sessions and `**/account**`: allows users to view and update their personal account information.

Campaign Management: `**/campaigns**` enables sponsors to create, view, update, and delete campaigns.

Ad Request Management: `**/ad_requests**` handles creation, viewing, and status updates of ad requests.

These endpoints provide comprehensive functionality for managing users, campaigns, and ad requests, ensuring efficient operation of the platform.

6. Conclusion:

The **Influencer Engagement and Sponsorship Coordination Platform - Brand Bridge** successfully integrates sponsors and influencers, facilitating seamless campaign management and ad request handling. The project was developed using Flask, Jinja2 templates, Bootstrap, and SQLite, with a clear focus on providing a user-friendly interface and efficient backend operations.For a detailed demonstration of the project, please refer to the presentation video. The complete source code can be accessed via the GitHub repository.