**MAD I**

**Project Report**

**Author**

Name: Harini Suryanarayanan

Roll No.: 23f1000747

Email: 23f1000747@ds.study.iitm.ac.in

**Description**

The project requires developing a platform for connecting sponsors and influencers, implementing user roles and logins, managing campaigns and ad requests, and providing search functionalities. Key tasks include creating a dashboards for all users, integrating data visualization for user interactions.

**Technologies used**

**HTML, CSS, JavaScript**: Frontend technologies for structure, styling, and interactivity.

**Flask**: Web framework for building the application.

**Flask-SQLAlchemy**: ORM for database interactions.

**SQLite3**: Database engine for storing data.

**Flask-Migrate**: Database migrations management, mostly to keep the models in the database updated incase of changes.

**Werkzeug.Security**: Security features for password hashing.

**OS**: Access environment variables and interact with the operating system as well as saving files onto system.

**DOTENV**: Load environment variables from a .env file.

**Datetime**: Manage dates and times in the application.

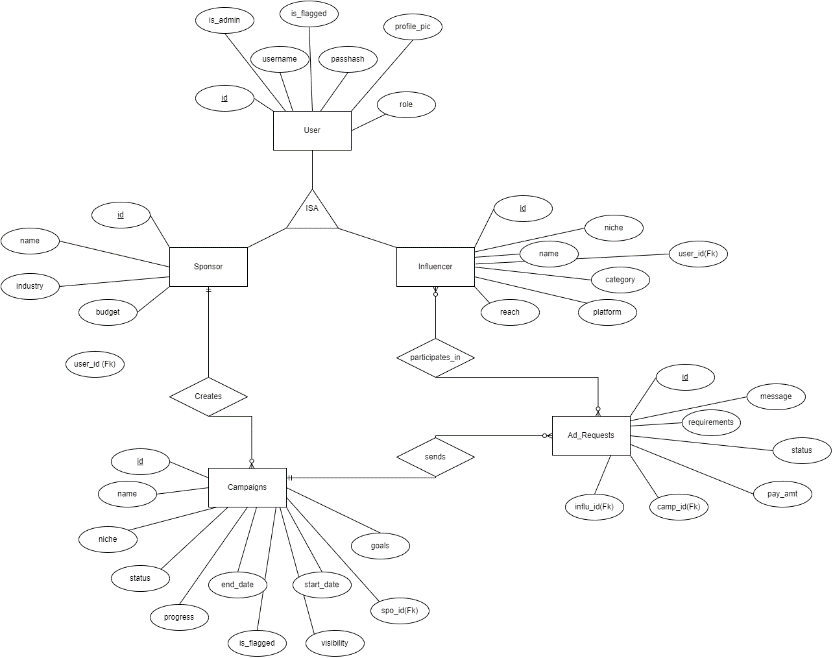
**Imghdr**: Determines the type of image contained in a file or byte stream.

**Collections**: Specialized data structures like named tuples, deque, and default dictionaries.

**Functools**: wraps decorator preserves the original function’s metadata when it is decorated.

**DB Schema Design**

ER Diagram of database:



**Architecture and Features**

All the html files are stored under templates folder, under which each user (sponsor, influencer, The project is organized with a clear structure. app.py creates and runs the Flask app, while config.py handles configurations and environment variables from .env. models.py defines the database models, and routes.py manages endpoints and logic. HTML files are stored in the templates folder, with separate subfolders for sponsors, influencers, and admins, organizing operations by user type. Dependencies are in the venv folder, listed in requirements.txt, ensuring a consistent and isolated environment.

**Implemented Features**

1. **User Roles and Logins**: Separate login/register forms for Admin, Sponsor, and Influencer roles, with appropriate user models.
2. **Admin Dashboard**: Displays statistics on active users, campaigns, ad requests, and flagged users, with root access for monitoring, flagging and unflagging.
3. **Campaign Management**: Sponsors can create, update, categorize, and delete campaigns.
4. **Ad Request Management**: Sponsors create, edit, and delete ad requests, detailing requirements, payment amounts, and status.
5. **Search Functionalities**: Sponsors search influencers by niche, reach, and followers; Influencers search public campaigns by niche and relevance.
6. **Ad Request Actions**: Influencers view, accept/reject ad requests. Sponsors can also accept/reject requests from influencers.
7. **Chart Integration**: Utilized ChartJS for data visualization.
8. **Backend Validations**: Implemented backend validations for form submissions.

**Video**

<<Link to your online video of not more than 3 minutes length>>