Name : Thendral R
Roll No. : 22F2001643
Email : 22f2001643@ds.study.iitm.ac.in
Organisation : Indian Institute of Technology, Madras (IITM)

Project Statement : Influencer Engagement & Sponsorship Coordination Platform - V2
Description : This is a web-based application designed to streamline the interaction between influencers and sponsor's campaigns which provides facilities to give requests and receive under the supervision of the admin.

Technologies Used:
Flask : Flask is a lightweight and flexible Python web framework that enables the rapid development of web applications with minimal setup.
Flask_cors : enables Cross-Origin Resource Sharing (CORS) in Flask applications, allowing them to handle requests from different origins securely.
Flask_mail : simplifies sending emails from Flask apps.
Flask_caching : adds caching support to Flask applications to improve performance.
Flask_jwt_extended : provides JWT (JSON Web Token) support for authentication and authorization in Flask applications.
SQLite3:lightweight, serverless SQL database engine integrated with Python
VueJS :progressive JavaScript framework for building user interfaces and single-page applications.
Redis: an in-memory data structure store used as a database, cache, and message broker.
Celery:an asynchronous task queue/job queue based on distributed message passing for executing tasks in the background.

1. User Table (`user`)

`id` (INTEGER, PRIMARY KEY, AUTOINCREMENT): Unique identifier for each user.
`username` (TEXT, UNIQUE, NOT NULL): The user's username, which must be unique.
`password` (TEXT, NOT NULL): The hashed password for user authentication.
`role` (TEXT, NOT NULL): The role of the user (e.g., 'influencer', 'sponsor', 'admin').
`platform` (TEXT): The social media platform associated with the influencer (e.g., Instagram, YouTube).
`industry` (TEXT): The industry in which the sponsor operates (e.g., fashion, technology).
`ratings` (INTEGER): A rating score for the influencer, which could be based on performance or reviews.
`earnings` (REAL): The total earnings an influencer has made through campaigns.
`flag` (BOOLEAN): Indicates whether the user has been flagged for violations.
`email` (VARCHAR): The email address of the user.
`followers` (INTEGER): The number of followers the influencer has on their platform.
`is_flagged` (BOOLEAN): A secondary flag status indicating specific issues or violations.
`last_login` (TIMESTAMP): Records the timestamp of the user's last login.

2. Campaign Table (`campaign`)

`id` (INTEGER, PRIMARY KEY, AUTOINCREMENT): Unique identifier for each campaign.
`sponsor_id` (INTEGER, NOT NULL): The ID of the sponsor who created the campaign (foreign key referencing `user.id`).
`title` (TEXT, NOT NULL): The title or name of the campaign.
`description` (TEXT): A brief description of the campaign.
`image_url` (TEXT): A URL to the campaign's image.
`niche` (TEXT): The specific niche or category the campaign belongs to (e.g., fashion, tech).
`start_date` (TEXT): The start date of the campaign.
`end_date` (TEXT): The end date of the campaign.

`budget` (REAL): The budget allocated for the campaign.
`status` (TEXT, NOT NULL): The current status of the campaign (e.g., 'active', 'completed').
`is_flagged` (BOOLEAN): Indicates whether the campaign has been flagged for any issues.

3. Influencer Request Table (`influencer_request`)

`id` (INTEGER, PRIMARY KEY, AUTOINCREMENT): Unique identifier for each request.
`influencer_id` (INTEGER, NOT NULL): The ID of the influencer making the request (foreign key referencing `user.id`).
`campaign_id` (INTEGER, NOT NULL): The ID of the campaign the influencer is requesting to join (foreign key referencing `campaign.id`).
`status` (TEXT, NOT NULL): The current status of the request (e.g., 'pending', 'approved', 'rejected').

4. Influencer Campaign Table (`influencer_campaign`):

`id` (INTEGER, PRIMARY KEY, AUTOINCREMENT): Unique identifier for each influencer-campaign relationship.
`influencer_id` (INTEGER): The ID of the influencer (foreign key referencing `user.id`).
`campaign_id` (INTEGER): The ID of the campaign (foreign key referencing `campaign.id`).
`status` (TEXT, NOT NULL): The status of the influencer's participation in the campaign (e.g., 'active', 'completed').

5. Message Table (`message`)

`id` (INTEGER, PRIMARY KEY, AUTOINCREMENT): Unique identifier for each message.
`influencer_id` (INTEGER, NOT NULL): The ID of the influencer sending or receiving the message (foreign key referencing `user.id`).
`sponsor_id` (INTEGER, NOT NULL): The ID of the sponsor sending or receiving the message (foreign key referencing `user.id`).
`message` (TEXT, NOT NULL): The content of the message.
`timestamp` (TIMESTAMP, DEFAULT CURRENT_TIMESTAMP): The time the message was sent.

API Design Principles:

Security: The API uses JWT for secure authentication and role-based access control to protect sensitive endpoints.
Modularity: Different functionalities (authentication, messaging, campaign management) are encapsulated in specific functions and routes, making the API modular and easier to maintain.
Performance Optimization: Caching with Redis and asynchronous task handling with Celery helps optimize the API's performance and responsiveness

Architecture and Features:

IESCP

```
app => app.py
          tasks.py
          Instance => users.db
Iescp-frontend => dist
                     public
                     src => assets
                            components
                            router
                            main.js
                            App.vue
```

Video Link :
https://drive.google.com/file/d/1c6rnZ4L1iTeWqwLopmUMh6RfMtVxDyr0/view?usp=sharing