INFLUENCER ENGAGEMENT AND SPONSORSHIP COORDINATION PLATFORM-V2

INFLUENCIA:

Author:

Name: Manoj Prathapa

Roll no: 23f1001473

Email: 23f1001473@ds.study.iitm.ac.in

About : I am a student studying in 3rd year of B.tech in CMRIT and i am currently pursuing diploma in programming in IITM and i hope to complete my BS Degree with IITM.

1.DESCRIPTION: The web application is designed to manage and facilitate interactions between users, influencers, sponsors, and administrators. It integrates functionalities for user authentication, role-based access, and data handling using various technologies and services. This report provides an overview of the application's features, architecture, implementation details, and future enhancements.

2. Features:

- User Authentication:

- Users can register, log in, and log out of the system. Passwords are securely hashed using bcrypt before being stored in the database.

- JSON Web Tokens (JWT) are used for secure authentication and authorization.

- **Role-Based Access:**

- The application supports different roles including Admins, Influencers, and Sponsors. Each role has distinct access rights and functionalities.

- Redirection based on user roles is implemented to direct users to their respective home pages upon login.

- Data Caching:

- Redis is used to cache frequently accessed data, improving performance and reducing load times.

- Key functionalities include setting, getting, and checking the existence of cached data.

- Campaign and Ad Management:

- Admins can manage users and oversee campaigns.

- Sponsors can create campaigns, manage budgets, and view campaign statuses.

- Influencers can view and respond to ad requests.

Architecture

The application follows a client-server architecture with separate components for the frontend and backend:

- Frontend:

- Built using Vue.js, the frontend provides a responsive and interactive user interface.

- Utilizes `vue-router` for managing different views and navigation.

- The application's layout and user interactions are managed through Vue components and templates.

- Backend:

- Caching: Utilizes Redis for caching data.

- Authentication: Implements password hashing and JWT for secure user authentication

- Database Management: SQLLite-based relational database with tables for users, influencers, sponsors, campaigns, and ad requests.

Frameworks and Libraries Used

- **Backend**: Flask (Python), SQLite for the database
- **Frontend**: Vue.js, Bootstrap for styling
- **Task Scheduling**: Redis and Celery
- **Authentication**: JSON Web Tokens (JWT)
- **Other Libraries**: Axios (API calls), Flask-SQLAlchemy (ORM), Flask-RESTful

- **Database:**

Entities:

1. **User**: user_id, username, password, role, bio
2. **Sponsor**: sponsor_id, user_id, company, sector, sponsor_budget
3. **Influencer**: influencer_id, user_id, specialization, audience_size
4. **Campaign**: campaign_id, sponsor_id, title, description, start_date, end_date, visibility, budget
5. **Ad Request**: request_id, campaign_id, influencer_id, requirements, payment_amount, status

Frontend Implementation:

- **Framework**: Vue.js with Bootstrap for styling.
- **Key Components**:
  - **Ad Request Form**: Sponsors can create ad requests by selecting a campaign and influencer from dynamically fetched lists, specifying requirements, and submitting data.
  - **Error Handling**: Messages displayed for missing authorization tokens or API failures.
  - **Token Management**: Uses localStorage for storing and sending authorization tokens.
  - **Navigation**: The application includes navigation and routing features, redirecting users based on roles and actions.

Backend Implementation:

- **Framework**: Flask with a modular setup.
- **Database**: SQLite for data storage; schema includes tables for Users, Sponsors, Campaigns, Influencers, Ad Requests, etc.

API Resource Endpoints

1. **Campaigns**:
   - GET /api/campaigns-list: Fetch all campaigns.
   - POST /api/campaigns: Create a new campaign.
2. **Ad Requests**:
   - POST /sponsor/add_adRequest_data: Add a new ad request.
   - GET /api/requests: View all requests.
3. **Influencers**:
   - GET /api/creators: Fetch list of influencers.
4. **Authentication**:
   - POST /auth/login: Login endpoint for JWT-based authentication.

Project demo video-
https://drive.google.com/file/d/1nAESNVG5qYW2rXp54IYxp8Pz4k_ge1me/view?usp=sharing