Sparked

by Sarthak Saini [23f2005347]

Introduction

Sparked is a platform that simplifies influencer marketing by connecting brands with relevant influencers. It facilitates discussions, price negotiations, and campaign discovery in one centralized location.

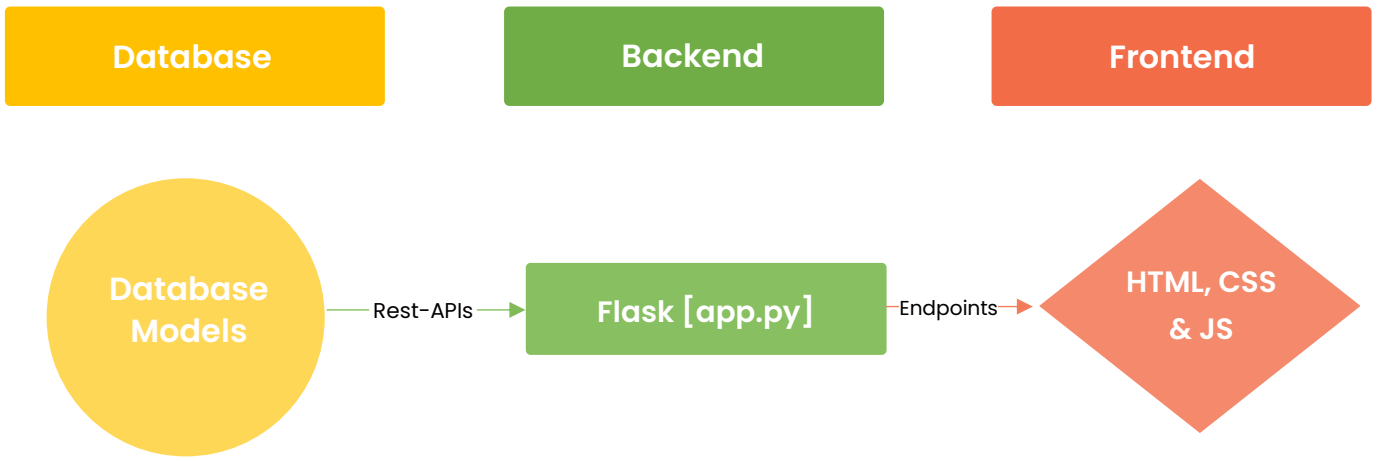
The application is moderated by an admin who oversees all accounts and activities.

Technologies Used

Flask HTML CSS JavaScript SQLite Rest-API

System Architecture

The main.py file is the central hub of the application, managing crucial CRUD operations via REST APIs. It seamlessly integrates app routes to render HTML templates at designated endpoints, ensuring clear code separation. This method enhances both comprehensibility and efficiency, similar to the structured approach seen in frameworks.



Database Management

The database, powered by SQLite, serves as the foundation. The models, created using Flask-SQLAlchemy, are carefully designed with constraints to ensure data integrity and consistency. Below is a concise code snippet demonstrating the structure of the Influencer model.

```python
class login(db.Model):
    user_id = db.Column(db.Integer, primary_key = True, autoincrement=True)
    username = db.Column(db.String(30), unique = True, nullable = False)
    password = db.Column(db.String(20), nullable = False)
    acc_type = db.Column(db.String(20), nullable = False)
    data_id  = db.Column(db.Integer)
    last_logged_in = db.Column(db.String(255), default=datetime.now().strftime("%d/%m/%y"))
    warnings = db.Column(db.String(30))

    def __repr__(self):
        return f"{self.user_id}-{self.username} {self.password} {self.acc_type}"
```

API Management

CRUD operations on the database are executed through APIs using Flask-Restful, implementing GET, POST, PUT, and DELETE functionalities at specific endpoints. Flask-Restful integrates these operations seamlessly, enabling efficient communication between the application and the database. This ensures a consistent and scalable way to interact with the database, enhancing maintainability of the system.

Multi-User Functionalities and Features

Influencer

- Receive Ad Requests
- View All Campaigns
- Search for Sponsors
- Negotiate Payment
- Make their Custom Profile Page

Sponsor

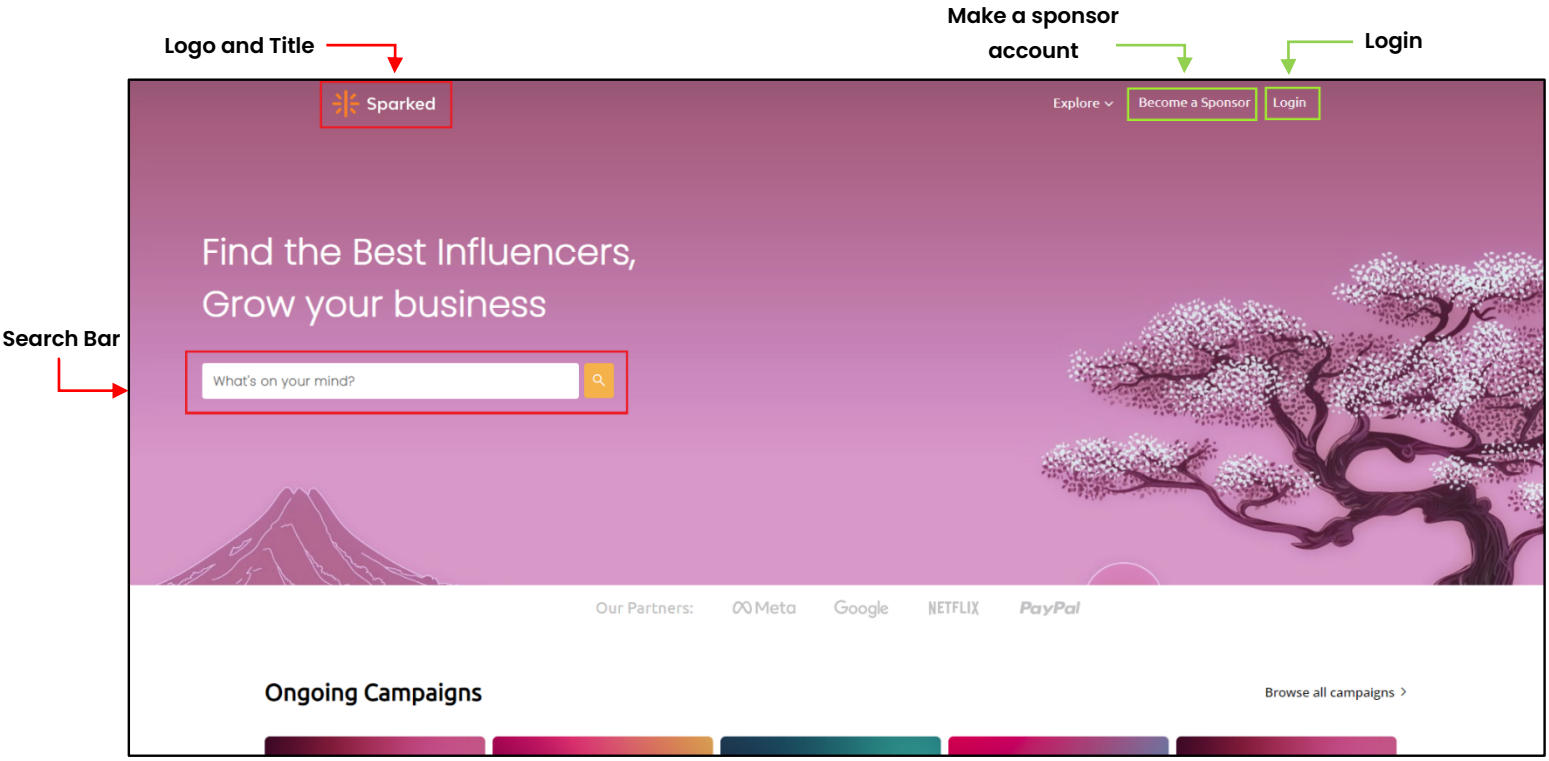
- Create Campaigns
- Search for Influencers
- Accept Ad Requests
- Negotiate Price
- Make their Custom Profile Page

Admin

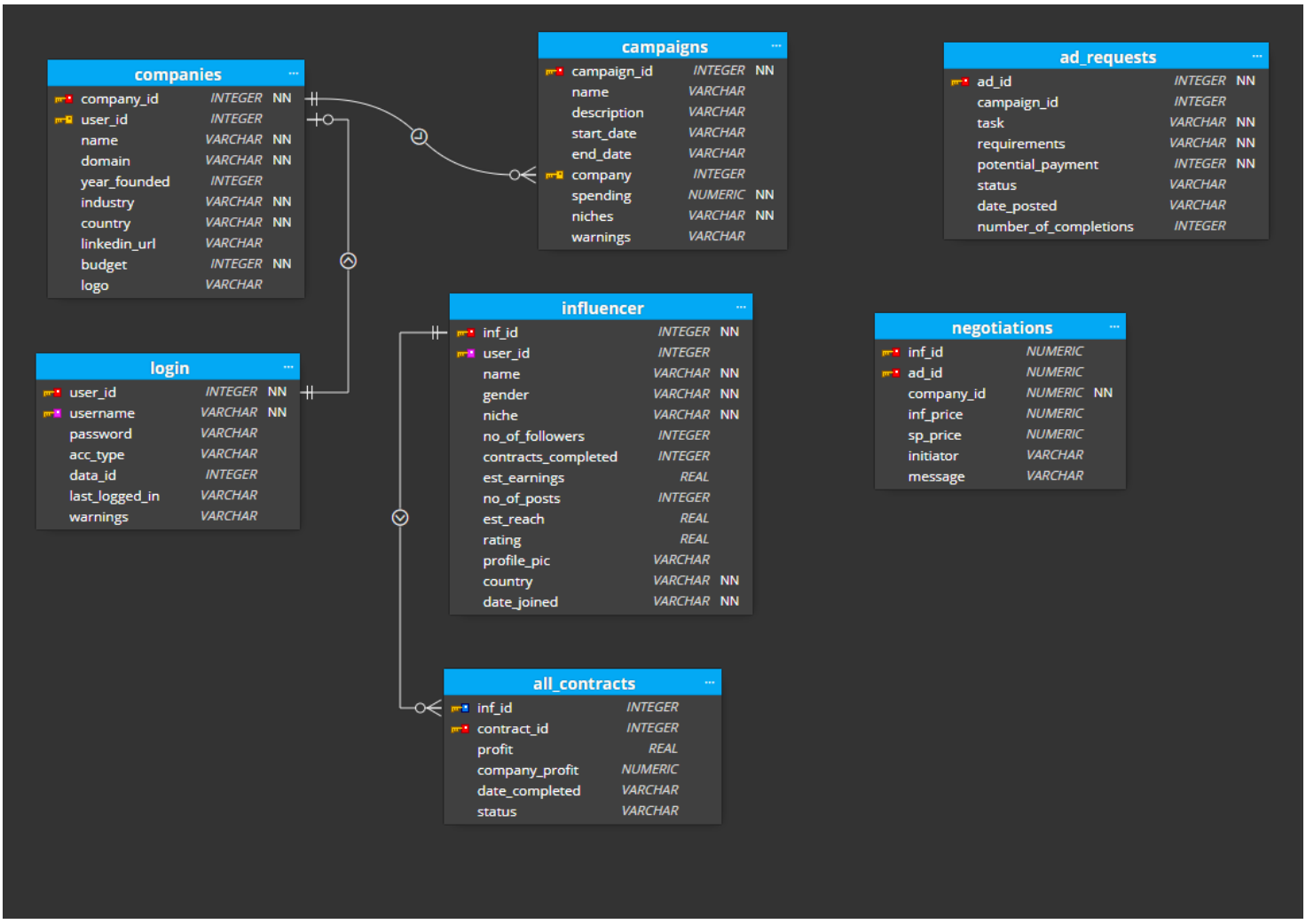
- View all Statistics
- Delete/Flag Campaigns
- Flag Users
- Ban Users

User Interface

The web application boasts an intuitive interface that caters to the unique needs of influencers, sponsors, and admins. Designed with a focus on user experience (UX), the interface prioritizes a calming atmosphere, creating a positive and engaging experience for everyone who interacts with the platform. This approach aims to boost engagement and satisfaction across all user roles, ensuring a smooth and enjoyable overall experience.



Database Schema



Conclusion and Extra Information

I had a lot of fun while creating this project. The excitement to implement new functionalities and features and tackling errors was the best. This was my first major project and I am really satisfied to have completed a functional web app.

- There are around 2000+ influencers and 500 sponsors loaded into the app with their respective campaigns and contracts.
- A single admin moderates the platform. The sign-in process for the admin involves entering a specific username and password on the login page which redirects to the admin page.
- **How to run the app? -** In the terminal type [python -m app] this runs the app as a module.
- **Presentation Video Link:** https://drive.google.com/file/d/1zTqf1c3IsGqpA848AfrFCOPTAN0t8ndY/view?usp=sharing