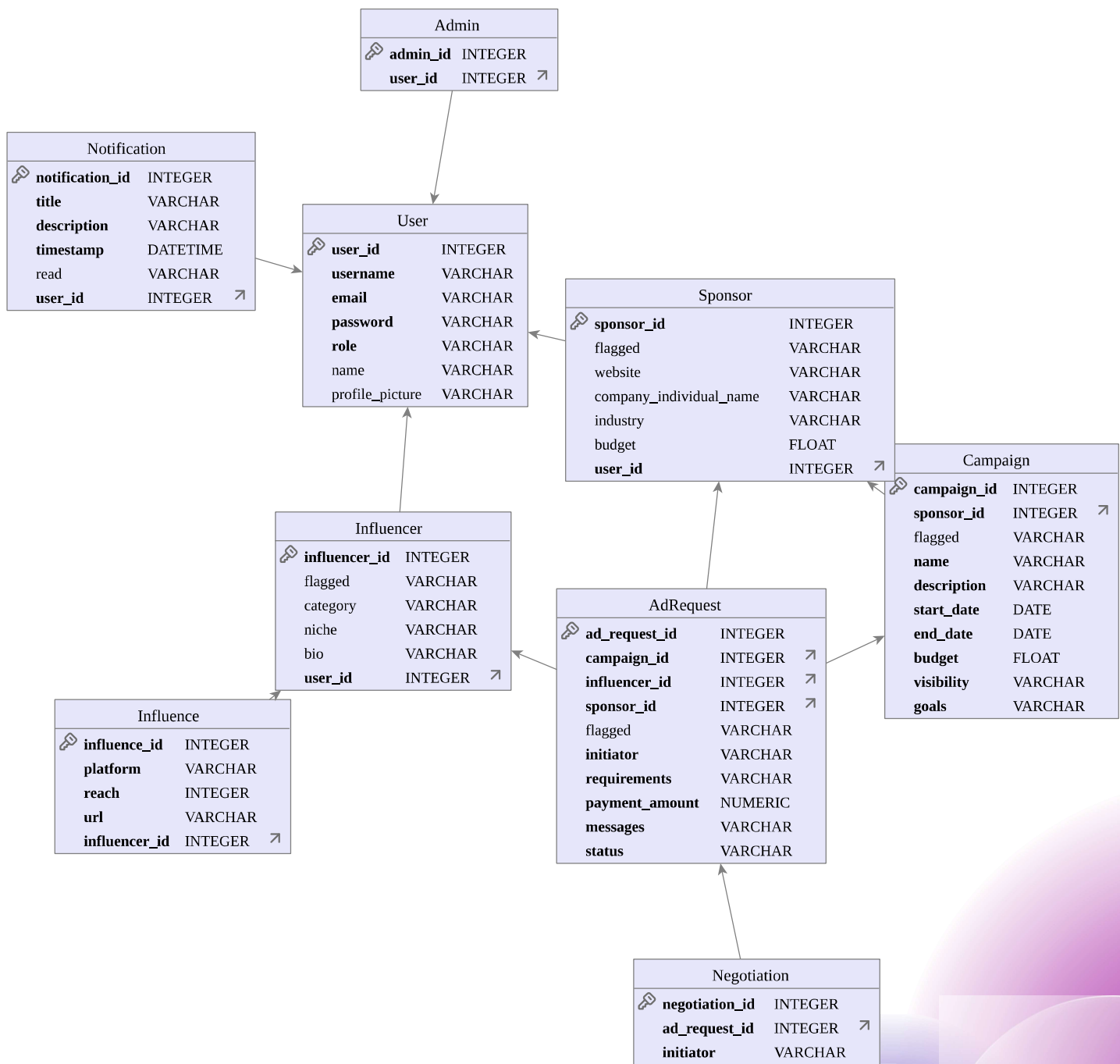# TECHNOLOGIES USED

- **Flask**: A lightweight Python web framework.
- **Jinja2**: A templating engine for generating HTML.
- **Flask_SQLAlchemy**: Adds SQLAlchemy support for database management in Flask.
- **Flask_Bcrypt**: Provides bcrypt hashing for secure passwords.
- **Flask_RESTful**: Simplifies building REST APIs in Flask.
- **Flask_WTF**: Integrates Flask with WTForms for form handling.
- **Requests**: A library for making HTTP requests in Python.
- **Email-Validators**: Validates email addresses for correctness.
- **SQLite**: A lightweight, disk-based database.
- **HTML**: Markup language for creating web pages.
- **CSS**: Stylesheet language for designing web pages.
- **JavaScript**: Programming language for interactive web content.
- **Bootstrap**: Framework for responsive, mobile-first web design.
- **ChartJS**: Library for creating interactive charts.

The database schema supports the Influencer Engagement and Sponsorship Coordination Platform with tables for users, influencers, sponsors, ad requests, campaigns, notifications, and negotiations. It ensures seamless coordination by linking these tables, enabling efficient campaign management, ad handling, and activity monitoring. This structure supports streamlined data management and workflows.

Admin

🔑 admin_id	INTEGER	
user_id	INTEGER	↗

Notification

🔑 notification_id	INTEGER	
title	VARCHAR	
description	VARCHAR	
timestamp	DATETIME	
read	VARCHAR	
user_id	INTEGER	↗

User

🔑 user_id	INTEGER	
username	VARCHAR	
email	VARCHAR	
password	VARCHAR	
role	VARCHAR	
name	VARCHAR	
profile_picture	VARCHAR	

Sponsor

🔑 sponsor_id	INTEGER	
flagged	VARCHAR	
website	VARCHAR	
company_individual_name	VARCHAR	
industry	VARCHAR	
budget	FLOAT	
user_id	INTEGER	↗

Campaign

🔑 campaign_id	INTEGER	
sponsor_id	INTEGER	↗
flagged	VARCHAR	
name	VARCHAR	
description	VARCHAR	
start_date	DATE	
end_date	DATE	
budget	FLOAT	
visibility	VARCHAR	
goals	VARCHAR	

Influencer

🔑 influencer_id	INTEGER	
flagged	VARCHAR	
category	VARCHAR	
niche	VARCHAR	
bio	VARCHAR	
user_id	INTEGER	↗

AdRequest

🔑 ad_request_id	INTEGER	
campaign_id	INTEGER	↗
influencer_id	INTEGER	↗
sponsor_id	INTEGER	↗
flagged	VARCHAR	
initiator	VARCHAR	
requirements	VARCHAR	
payment_amount	NUMERIC	
messages	VARCHAR	
status	VARCHAR	

Influence

🔑 influence_id	INTEGER	
platform	VARCHAR	
reach	INTEGER	
url	VARCHAR	
influencer_id	INTEGER	↗

Negotiation

🔑 negotiation_id	INTEGER	
ad_request_id	INTEGER	↗
initiator	VARCHAR	

API Resource ✕

Campaign Endpoints
- **GET** /api/campaign: Retrieve all campaigns or filter by name, sponsor_name, industry, budget, flagged.
- **GET** /api/campaign/string:campaign_id: Retrieve a specific campaign by ID or name.
- **POST** /api/campaign: Create a new campaign.
- **PUT** /api/campaign/string:campaign_id: Update an existing campaign by ID.
- **DELETE** /api/campaign/string:campaign_id: Delete a specific campaign by ID.

Ad Request Endpoints
- **GET** /api/ad-request/int:ad_request_id: Retrieve all ad requests or a specific one by ad_request_id.
- **POST** /api/ad-request: Create a new ad request with campaign_id, influencer_id, sponsor_id, initiator, requirements, payment_amount, messages.
- **PUT** /api/ad-request/int:ad_request_id: Update an existing ad request by ID.
- **DELETE** /api/ad-request/int:ad_request_id: Delete a specific ad request by ID.

Negotiation Endpoints
- **GET** /api/negotiation: Retrieve all negotiations or a specific one by negotiation_id.
- **POST** /api/negotiation: Create a new negotiation with ad_request_id, initiator, message, status.
- **PUT** /api/negotiation/int:negotiation_id: Update an existing negotiation by ID.
- **DELETE** /api/negotiation/int:negotiation_id: Cancel a specific negotiation by ID.

API Resource

Influencer Endpoints
- **GET** /api/search_influencer/string:username_or_name: Retrieve all influencers or a specific influencer by username or name.
- **GET** /api/search_influencerid/int:influencer_id: Retrieve all influencers or filter by name, sponsor_name, industry, budget, flagged or a specific influencer by influencer_id.
- **POST** /api/search_influencerid/int:influencer_id: Update an influencer's profile by ID.
- **PUT** /api/search_influencerid/int:influencer_id: Add or Update an influencer's social media accounts by ID.
- **DELETE** /api/search_influencerid/int:influencer_id: Delete a specific social media accounts by influence/platform ID.

Sponsor Endpoints
- **GET** /api/search_sponsor/int:sponsor_id: Retrieve all sponsors or a specific sponsor by ID.
- **PUT** /api/search_sponsor/int:sponsor_id: Update a sponsor's profile by ID.
- **DELETE** /api/search_sponsor/int:sponsor_id: Delete sponsor's profile picture by ID.

Presentation Video

Presentation Video of My Project is available here.

https://drive.google.com/file/d/1Pltm3ym-F4tpNf3RlVb6R8OouYoCI-Ez/view