# **Influencer Platform**

This is a web platform where businesses and influencers can collaborate seamlessly. The platform allows users to register, post tasks, and manage their accounts.

## **Features**

* **User Authentication**: Register, log in, and manage profiles.
* **Task Management**: Post and view tasks,update tasks and apply for task.
* **Responsive Frontend**: Clean and intuitive UI for easy interaction.
* **API Integration**: Backend API to handle authentication, task operations, user management and task application management.

## **Technologies Used**

### **Backend:**

* **Django**: Web framework for backend development.
* **Django Rest Framework (DRF)**: For creating RESTful APIs.
* **PostgreSQL**: Relational database for storing data.

### **Frontend:**

* **Vue.js**: Reactive frontend framework.
* **Axios**: For making HTTP requests.

### **Deployment:**

#### Backend

* **Render**: Hosting the backend.
* **Gunicorn**: WSGI HTTP Server for the backend.
* **Whitenoise**: Static file serving in production.

#### Frontend

* **Vercel:** Hosting the frontend

## **Prerequisites**

### **Software Required:**

1. **Python**: Version >= 3.7
2. **Node.js**: Version >= 14.x
3. **PostgreSQL**: Installed and running on your machine.
4. **Vue CLI**: Installed globally for frontend development.

### 

### **Libraries and Plugins:**

Make sure you have the following Python and Node.js dependencies:

#### **Python Dependencies:**

* django
* djangorestframework
* django-cors-headers
* gunicorn
* whitenoise
* psycopg2

Install all backend dependencies via:

pip install -r requirements.txt

#### 

#### **Node.js Dependencies:**

* vue
* axios
* vue-router

Install frontend dependencies via:

npm install

## **Installation Steps**

### **Backend Setup**

1. Clone the repository:  
     
   git clone https://github.com/mengchenghui123/influencer-platform.git

cd influencer-platform

1. Set up a virtual environment and activate it:  
   python -m venv venv

source venv/bin/activate # For Mac/Linux

venv\Scripts\activate # For Windows

1. Install dependencies:  
   pip install -r requirements.txt
2. Create a .env file in the backend directory with the following:  
     
   SECRET\_KEY=django-insecure-=h0&b5u7tr0q)b%savimqab^p&=b\_q%skzmqi9u@0w=&\*err8%

DATABASE\_URL=postgresql://postgres.wxefcldfldcsysukdcve:mengchenghui@aws-0-us-east-1.pooler.supabase.com:6543/postgres

1. Run migrations:  
     
   python manage.py migrate
2. Start the backend server:  
     
   python manage.py runserver

### **Frontend Setup**

1. Navigate to the frontend directory:  
     
   cd influencer-platform/frontend
2. Install dependencies:  
     
   npm install
3. Start the frontend development server:  
     
   npm run serve
4. Open your browser and visit:<http://localhost:8080>

## **Database Setup**

This project uses **Supabase** for database management. Follow these steps to connect:

### **1. Supabase Connection Information**

* Login to [Supabase](https://supabase.com) with GitHub account.
* Use the provided credentials to connect:
  + **Database Host:** aws-0-us-east-1.pooler.supabase.com
  + **Database Name:** influencer\_platform
  + **Username:** postgres
  + **Password:** mengchenghui

### **2. Configure Environment Variables**

Add the following to your .env file:

DATABASE\_URL=postgresql://postgres.wxefcldfldcsysukdcve:mengchenghui@aws-0-us-east-1.pooler.supabase.com:6543/postgres

### **3. Apply Migrations**

Run the following commands to set up the database schema:

python manage.py makemigrations

python manage.py migrate

## **Deployment**

**Backend Deployment (Render):**

1. Add a render.yaml file in the root directory.
2. Configure your Render project for backend.
3. Push your code to a GitHub repository connected to Render.

### **Running in Production:**

Ensure the following:

* Static files are collected using python manage.py collectstatic.
* Database URL is configured for PostgreSQL in production.

### **Frontend Deployment (Vercel)**

1. Deploy the frontend by connecting the frontend directory to Vercel.

## **Notes**

* **Static Files**: Managed using Whitenoise for production.
* **Environment Variables**: Ensure all sensitive information is stored in environment variables.

## **Usage Guide**

This section provides a detailed guide for the platform's three user roles: **Merchants**, **Influencers**, and **Administrators**.

### **Merchant Guide**

Merchants can post tasks and manage them throughout their lifecycle.

1. **Register as a Merchant**:
   * Select "Merchant" during registration to gain access to merchant-specific features.
2. **Log In**:
   * Use the credentials created during registration to log into your account.
3. **Post a Task**:
   * Navigate to the "Post Task" section to create a new task.
   * Fill in all required fields, including title, description, budget, and deadline.
4. **View and Manage Tasks**:
   * Access the "My Posted Tasks" section to view all tasks you’ve posted.
   * For available tasks, you can update or delete them.
   * For tasks in progress, you can mark them as completed.

### **Influencer Guide**

Influencers can apply for tasks posted by merchants and track their application status.

1. **Register as an Influencer**:
   * Choose "Influencer" during registration to access influencer-specific features.
2. **Log In**:
   * Log in using the credentials created during registration.
3. **Browse Tasks**:
   * Go to the "All Tasks" page to view tasks posted by merchants.
   * Filter tasks by status (available, in progress, completed) or search by task title or merchant name.
4. **Apply for Tasks**:
   * Click "Apply" on available tasks to submit your application.
   * The system prevents duplicate applications for the same task.
5. **Manage Applications**:
   * In "My Applications," view the status of all your task applications.
   * Cancel an application if you no longer wish to be considered.

### **Admin Guide**

Administrators oversee platform activity, manage user accounts, and control task applications.

1. **Admin Login**:
   * Log in with admin credentials provided during the initial setup.
2. **Manage Users**:
   * Access the user management section in the admin panel to manage merchant and influencer accounts.
   * Suspend or delete accounts as necessary.
3. **Manage Tasks and Applications**:
   * In the admin panel, view all tasks and applications.
   * Approve or reject applications from influencers on behalf of the merchant.
   * When an application is approved, the task status changes to "in progress," and other applications for the task are automatically rejected.

## **API Endpoints**

The platform provides the following key API endpoints:

### **Authentication:**

* /api/login/: User login with token authentication.
* /api/register/: User registration.
* /api/logout/: Logout and blacklist the token.

### **User Profile:**

* /api/me/: Retrieve the current user's profile.
* /api/update-profile/: Update the user's profile.

### **Task Management:**

* /api/tasks/: Retrieve all tasks.
* /api/tasks/create/: Create a new task (merchant-only).
* /api/tasks/<task\_id>/: Get details of a specific task.
* /api/tasks/<task\_id>/update/: Update a task (merchant-only).
* /api/tasks/<task\_id>/delete/: Delete a task (merchant-only).
* /api/tasks/<task\_id>/mark\_completed/: Mark a task as completed (merchant-only).

### **Task Applications:**

* /api/tasks/<task\_id>/apply/: Apply for a task (influencer-only).
* /api/my-applied-tasks/: View an influencer’s applied tasks.
* /api/my-posted-tasks/: View a merchant’s posted tasks.
* /api/task-applications/<application\_id>/: Cancel a task application (influencer-only).