## Project Overview

This Django project is web scraping, data processing, data visualization and a MySQL-backed Django application.

Project link: https://github.com/shown440/scraping\_project

## Prerequisites

Before you begin, ensure you have the following installed:

* Python 3.12+ (recommended)
* Git
* MySQL server (or another supported database)
* Virtual environment tool (venv, virtualenv)
* System dependencies for mysqlclient (on Ubuntu/Debian):
* sudo apt-get update  
  sudo apt-get install default-libmysqlclient-dev build-essential
* sudo apt install python3.12 python3.12-venv python3.12-dev python3-pip -y

## Installation Steps

1. **Clone the repository**

* git clone git@github.com:shown440/scraping\_project.git  
  cd scraping\_project

1. **Create a virtual environment and activate it**

* python3.12 -m venv sebpo\_scrap\_env\_312  
  source sebpo\_scrap\_env\_312/bin/activate

1. **Install Python dependencies**

* pip install --upgrade pip  
  pip install -r requirements.txt

## Configuration

1. **Environment variables**

Already inserted inside the code, because It’s anon production project.

## Database Setup and Migrations

1. **Create the database** in MySQL:

* 1. Login inside mysql database
* 2. CREATE DATABASE sebpo\_scrap\_db;
* 3. GRANT ALL PRIVILEGES ON sebpo\_scrap\_db.\* TO 'ffwc'@'localhost';
* 4. FLUSH PRIVILEGES;
* 5. Exit
* 6. Restore dumped mysql db: mysql -u db\_username -p sebpo\_scrap\_db < sebpo\_scrap\_db.sql

1. **Apply migrations**:

* 1. python3.12 manage.py makemigrations
* 1. python3.12 manage.py migrate

1. **Create a superuser** (for admin access):

* python3.12 manage.py createsuperuser

## Running the Development Server

Start Django’s built-in development server:

Python3.12 manage.py runserver then it will run is: 0.0.0.0:8006

Visit http://127.0.0.1:8006/ in your browser.

## Running Scheduled and Custom Commands

* **Daily scraping task**:
* python3.12 manage.py fetch\_cia\_data
* **Daily scraping task** via cron:
* 1. goto project folder
* 2. go to: /accessories/bash\_script\_and\_crontab/bash\_script/fetch\_cia\_data.sh
* 3. find: “/var/www/prod/scraping\_project” then replace with your project path
* 4. find: “/var/www/Project\_Environments/python\_environments/sebpo\_scrap\_env\_312
* ” then replace with your python environment path
* 5. then provide permission update cia\_data.sh: sudo chmod 777 /var/www/prod/scraping\_project/accessories/bash\_script/fetch\_cia\_data.sh
* 6. now write in terminal: sudo crontab -e
  + Then: 20 15 \* \* \* /var/www/prod/scraping\_project/accessories/bash\_script/fetch\_cia\_data.sh
  + Then: save and exit

## Static Files (Production)

1. Collect static assets:

* python3.12 manage.py collectstatic

1. Configure your web server (e.g., Nginx) to serve the static/ directory.

## Deployment

For deploying to production, consider using:

* **Gunicorn + Nginx**

**Gunicorn service:** Goto the path: ../scraping\_project/accountable\_services/ubuntu\_server/scrap\_webservice\_port.service

1. find: “/var/www/Project\_Environments/python\_environments/sebpo\_scrap\_env\_312” then replace with your python environment path
2. Create Command: sudo nano scrap\_webservice\_port.service

And paste updated command inside here from: ../scraping\_project/accountable\_services/ubuntu\_server/scrap\_webservice\_port.service

1. Enable Command: sudo systemctl enable scrap\_webservice\_port.service
2. Start Command: sudo systemctl start scrap\_webservice\_port.service
3. Restart Command: sudo systemctl scrap\_webservice\_port.service
4. Monitoring Command: sudo systemctl status scrap\_webservice\_port.service

**Nginx configuration process:** Goto the path: ../scraping\_project/accountable\_services/ubuntu\_server/scrap\_ng\_port

1. find: “/var/www/prod/scraping\_project” then replace with your python project path
2. find: “scrap.sebpo.com” then replace with your own domain
3. then copy the whole command and goto: sudo nano /etc/nginx/sites-available/scrap\_ng\_port and paste and save it
4. Link the Configuration File: sudo ln -s /etc/nginx/sites-available/ scrap\_ng\_port /etc/nginx/sites-enabled/
5. sudo nginx -t
6. sudo systemctl restart nginx
7. If need to https the we can use certbot to do that:
   1. Then install: pip install certbot certbot-nginx
   2. Renew https domain: sudo certbot –nginx
   3. sudo systemctl restart nginx

## Contributing

1. Fork the repository
2. Create a feature branch (git checkout -b feature/XYZ)
3. Commit your changes (git commit -m "Add XYZ feature")
4. Push to the branch (git push origin feature/XYZ)
5. Open a Pull Request