Technical Installation Manual

Ctrl Alt Defeat

July 30, 2023

Contents

1 Introduction

The following is a Technical Installation Manual for COS301 capstone project, Domain Pulse - A Sentiment Analysis Platform. Domain Pulse is a system designed to fetch, analyse, aggregate, and present results of sentiment analysis on publically available data pertaining to businesses, products, places, and more! Sources of data include (but are not limited to) Google reviews, TripAdvisor and Youtube.

This installation guide details how to setup and install the system locally. The system uses a Django backend (consisting of 5 independently deployable projects), an Angular frontend as well as PostgreSQL and MongoDB databases. The installation of these components and the packages used within in them, shall be explained within this manual.

Please note that the production application is a web application that is deployed to a private virtual machine acting as a web server. For security purposes, this guide will not provide details or keys for connecting to the deployment server - instead this details local installation of the application.

Once again, this is not the recommended approach for using the application. Instead, is it recommended that one interacts with the hosted web application available at: www.domainpulse.app

1.1 Hosted Web App Approach - Recommended

In order to interact with the web app hosted at www.domainpulse.app, no configuration or installation is required. The only prerequisites are: a modern web browser (Brave, Firefox, or Google Chrome recommended) and a stable Internet connection. This is the intended and preferred approach of using the application.

1.2 Overview of local installation process

The process of installing the application locally consists of the following stages, each of which is to be described in detail later in the manual. To reiterate, this is not the recommended approach as there is a substantial number of prerequisites to install and configure.

- Installation and acquisition of prerequisites
- Downloading and configuring application
- Installing project specific dependencies
- Running the application

2 Prerequisites - General Software

The following software are prerequisites for the complete installation and set-up of Domain Pulse: Python 3.8, Pipenv (any 2022 or post 2022 version), Node.js (v18 later), Angular (v14), MongoDB (Community Edition, v6 or later), PostgreSQL 12. Of course an operating system of either Linux (Ubuntu) or Windows is required, as well as a modern web browser.

Note: The installation of the project specific dependencies and packages is detailed later in the manual.

2.1 Operating System

The system may be installed locally on either a Windows or Linux (Ubuntu) operating system. The following versions are most suitable

- Linux (Ubuntu): v22 or later
- Windows: v10 or later

Note that earlier or different versions of these operating systems may successfully install and run the application, however the provided versions are deemed to be most suitable and confirmed to work.

2.2 Web Browser

Any modern web browser should support the appropriate rendering of the application. The following browsers have been confirmed as suitable:

- Google Chrome - Latest version
- Mozilla Firefox - Latest version
- Brave - Latest version

2.3 Python 3.8

2.3.1 Linux

The following resources commands be used for installing Python 3.8 on Linux:

```
sudo apt-get update
sudo add-apt-repository ppa:deadsnakes/ppa -y
sudo apt-get update
sudo apt install python3.8
```

Ensure that the correct version of python is installed by running the following command: `python -version`

2.3.2 Windows

The following resources may be used for installing Python 3.8 on Windows:
https://www.python.org/downloads/release/python-380/

2.4 Pipenv (any 2022 or post 2022 version)

Once Python is installed, pip will be installed along with in. To install pipenv, enter into the terminal:
`pip install pipenv`

2.5 Node.js v18 or later

Node.js is needed for the installation and use of Angular and can be installed following the instructions on the following resource:
https://nodejs.dev/en/learn/how-to-install-nodejs/

2.6 Angular v14

Once Node.js is installed, Angular can be installed by entering the following command into the terminal:
`npm install -g @angular/cli`

2.7 MongoDB (Community Edition v6 or later)

The following resources may be used for installing MongoDB

- Windows: https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-windows/
- Ubuntu: https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-ubuntu/
- MacOS: https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-os-x/

2.8 PostgreSQL 12

The following resource can be used for installing PostgreSQL 12

- https://www.postgresql.org/download/

Important note: Ensure that, regardless of what installer is used, PostgreSQL 12 specifically is installed.
Once installed, the following needs to be completed to ensure that the database is ready for use:

Linux

- Log in to psql CLI using `sudo -u postgres psql`
- Create the database using `CREATE DATABASE profiles;`

Windows

Note: Ensure that the PostgreSQL bin directory is added to the PATH environment variable.

- Log in to psql CLI using `psql -U postgres`
- Create the database using `CREATE DATABASE profiles;`

3 Deployment and Running

Once all previous steps have been followed and completed, and PostgreSQL and MongoDB are running, the application may be run locally. To do so, the depolyment of the backend and frontend is done separately.

3.1 Backend

In terminal, navigate to the 'backend' directory. To deploy the backend, issue the following commands:
`pipenv shell`
Next: `cd scripts`
Finally: `./run-projects.sh`

3.2 Frontend

In terminal, navigate to the 'frontend' directory. To deploy the frontend, issue the following commands:

4 Prerequisites - Hardware

The following are the minimum recommended specifications

4.1 PostgreSQL 12

The following resource can be used for installing PostgreSQL 12

- https://www.postgresql.org/download/

Important note: Ensure that, regardless of what installer is used, PostgreSQL 12 specifically is installed.
Once installed, the following needs to be completed to ensure that the database is ready for use:

Linux

- Log in to psql CLI using `sudo -u postgres psql`
- Create the database using `CREATE DATABASE profiles;`

Windows

Note: Ensure that the PostgreSQL bin directory is added to the PATH environment variable.

- Log in to psql CLI using `psql -U postgres`
- Create the database using `CREATE DATABASE profiles;`

required to install and run Domain Pulse locally

- CPU: Intel i5 11th Gen or Ryzen equivalent recommended
- RAM: 8GB minimum, 16GB recommended
- Storage: At least 20GB free space recommended (but database size dependent on use of application)

5 Installation

5.1 Cloning the repo

The repo may be cloned locally from Github using the following command:
`git clone https://github.com/COS301-SE-2023/Domain-Pulse-A-Sentiment-Analysis-Platform.git`

5.2 Configuring the application

5.3 Installing dependencies

5.3.1 Backend

In terminal, navigate to the 'backend' directory. To install all necessary dependencies at the appropriate versions, issue the following command:
`pipenv install`
This will install all the necessary dependencies into a local virtual environment.

Note: Some of the dependencies are fairly sizable. Depending on Internet speeds, this process may take a few minutes.

The full list of dependencies that will be installed can be found in the the file backend/Pipfile. Furthemore, to see the specific versions you may issue the CLI command:
`pipenv graph`
The dependencies are as follows:

- coverage==7.2.6
- django-cors-headers==4.2.0
- djangorestframework-simplejwt==5.2.2
- mock==5.1.0
- nltk==3.8.1
- psycopg2-binary==2.9.6
- pymongo==4.4.0
- python-dotenv==1.0.0
- transformers==4.30.2
- vaderSentiment==3.3.2

Note that each dependencies may have its own sub-dependencies. These too will be installed via `pipenv install`

5.3.2 Frontend

To install the dependencies required for the frontend, navigate to th 'frontend' directory and issue the following command to install all the necessary dependencies at the correct version:
`npm ci`
A list of the dependencies that will be installed (along with their appropriate versions) can be found in the file `package.json`

5.4 Configuring the application for local deployment

Domain Pulse is a web application deployed to a private virtual machine provded by our industry client. Consequently, the default configuration of the application is set to interact with this server via SSH. For security reasons, this guide details how to configure the application for local deployment only, and no SSH or API credential will be provided.

5.4.1 Acquiring API Keys

In order for the project to successfully interact with the necessary external services, one needs to acquire API keys for the following services. Note that each service may require some setup of its own.

- Outscraper: https://outscraper.com/
- Youtube Data API v3: https://blog.hubspot.com/website/how-to-get-youtube-api-key
- Azure Blob Storage: https://learn.microsoft.com/en-us/azure/storage/blobs/storage-quickstart-blobs-portal

5.4.2 Setting up enviroment variables

To establish connections to local database, you will need to create an environment variable file called `.postgresql.env` (in the root of the project directory). The file needs to have the following contents:

```
USE_TUNNEL=False
SQL_DATABASE_HOST=localhost
SQL_DATABASE_NAME=[Your PostgreSQL database name]
SQL_DATABASE_USER=[Your PostgreSQL server username]
SQL_DATABASE_PASS=[Your PostgreSQL server password]
SQL_DATABASE_PORT=[The port on which your PostgreSQL server is running]
MONGO_HOST=localhost
MONGO_PORT=[The port on which your MongoDB server is running]
MONGO_DB_NAME=[Your MongoDB database name]
```

Similarly, you will need to create a file named `.env` which will contain a number of other details. These include some details regarding service ports and external service API keys.

```
ENVIRONMENT=local
DJANGO_DOMAINS_PORT=8000
DJANGO_ENGINE_PORT=8001
DJANGO_PROFILES_PORT=8002
DJANGO_SOURCECONNECTOR_PORT=8003
DJANGO_WAREHOUSE_PORT=8004
MONGO_PORT = [The port on which your MongoDB server is running]
OUTSCRAPER_API_KEY = [The API key obtained from Outscraper]
TRIPADVISOR_API_KEY = [The API key obtained from Outscraper]
GOOGLE_REVIEWS_API_KEY = [The API key obtained from Outscraper]
TRUSTPILOT_API_KEY = [The API key obtained from Outscraper]
YOUTUBE_API_KEY = [The Youtube Data API v3 key obtained]
SECRET_KEY = [Private key string to be used for encryption]
AZURE_SAS = [API key associated with Azure Blob Storage container]
```

6 Deployment and Running

Once all previous steps have been followed and completed, and PostgreSQL and MongoDB are running, the application may be run locally. To do so, the depolyment of the backend and frontend is done separately.

6.1 Backend

In terminal, navigate to the 'backend' directory. To deploy the backend, issue the following commands:
`pipenv shell`

Next: `cd scripts`

Finally: `./run-projects.sh`

6.2 Frontend

In terminal, navigate to the 'frontend' directory. To deploy the frontend, issue the following commands: