# Setup documentation search engine

# Local – Linux/MacOS

1. Make sure you have python 3.9 and pip 21 installed.
2. This project uses a MySQL database, if you want to connect to a local MySQL database, make sure you have MySQL server installed ( <https://dev.mysql.com/doc/mysql-installation-excerpt/5.7/en/> )
3. A SQL script is provided in the repository with the database articles used for testing.
4. Clone the Git repository to your local system.
5. Make sure you have virtualenv installed:
   1. sudo pip3 install virtualenv
6. Create a virtual environment with the name of the project: virtualenv coldcaseenv
7. Activate the virtual environment: source coldcaseenv/bin/activate
8. Go inside the project folder, and copy the .env.example file, and make the name of the new file “.env”: cp .env.example .env
9. Fill this file with your own environment variables. The variables that need to be filled are already there.
10. Install the required packages: pip install -r requirements.txt
11. Now, migrate the database tables into the database with the following command: python manage.py migrate
12. It is time to start the server, which you can do with the following command: python manage.py runserver

# Local- windows

The setup for Windows is very similar to the one on Linux. These are the differences:

Installing Python and pip:

* As a windows user, you can get Python, including pip, from here: <https://www.python.org/downloads/>
* Next, to install a virtual environment tool: pip3 install virtualenvwrapper-win
* Then, to start the virual environment, this is the command: mkvirtualenv coldcaseenv
* The environment is activated automatically, so, now we can go on with step 8

# AWS:

To install this project on an AWS server, the steps are very similar to the ones for the local machine. This is what needs to be done.

* First, create an amazon EC2 instance, with SSH access to yourself, and http inbound requests for everyone.
* SSH into this instance, with the keys you got during the setup of this instance.
* Install the necessary packages: sudo apt-get install python3-pip apache2 libapache2-mod-wsgi-py3
* This project uses a MySQL database, if you want to connect to a local MySQL database, make sure you have MySQL server installed ( <https://dev.mysql.com/doc/mysql-installation-excerpt/5.7/en/> )
* A SQL script is provided in the repository with the database articles used for testing.
* Clone the Git repository to your local system.
* Make sure you have virtualenv installed:

sudo pip3 install virtualenv

* Create a virtual environment with the name of the project: virtualenv coldcaseenv
* Activate the virtual environment: source coldcaseenv/bin/activate
* Go inside the project folder, and copy the .env.example file, and make the name of the new file “.env”: cp .env.example .env
* Fill this file with your own environment variables. The variables that need to be filled are already there.
* Install the required packages: pip install -r requirements.txt
* Now, migrate the database tables into the database with the following command: python manage.py migrate
* Get out of the local instance with the following command: deactivate
* Go to the root directory of the EC2 instance with: cd
* Go to the directory where the Apache configuration file is located, and edit it: sudo vi /etc/apache2/sites-available/000-default.conf
* Replace the file with the following config:

<VirtualHost \*:80>

**ServerAdmin** webmaster@localhost.com

**DocumentRoot** /home/ubuntu/django/coldcasefind/ColdCaseFind

**ErrorLog** ${APACHE\_LOG\_DIR}/error.log

**CustomLog** ${APACHE\_LOG\_DIR}/access.log combined

<Directory /home/ubuntu/django/coldcasefind/ColdCaseFind/ColdCaseFind>

<Files wsgi.py>

**Require** all granted

</Files>

</Directory>

WSGIDaemonProcess ColdCaseFind python-path=/home/ubuntu/django/coldcasefind/ColdCaseFind python-home=/home/ubuntu/django/coldcasefindenv

WSGIProcessGroup ColdCaseFind

WSGIScriptAlias / /home/ubuntu/django/coldcasefind/ColdCaseFind/ColdCaseFind/wsgi.py

</VirtualHost>

* Edit this config in order to make the routes correct for your configuration.
* Restart Apache with: sudo service apache2 restart
* Done!

This deployment guide is inspired by: <https://medium.com/saarthi-ai/ec2apachedjango-838e3f6014ab>