Disaster Vulnerability and Damage Assessment tool Environment Setup Deployment instructions.

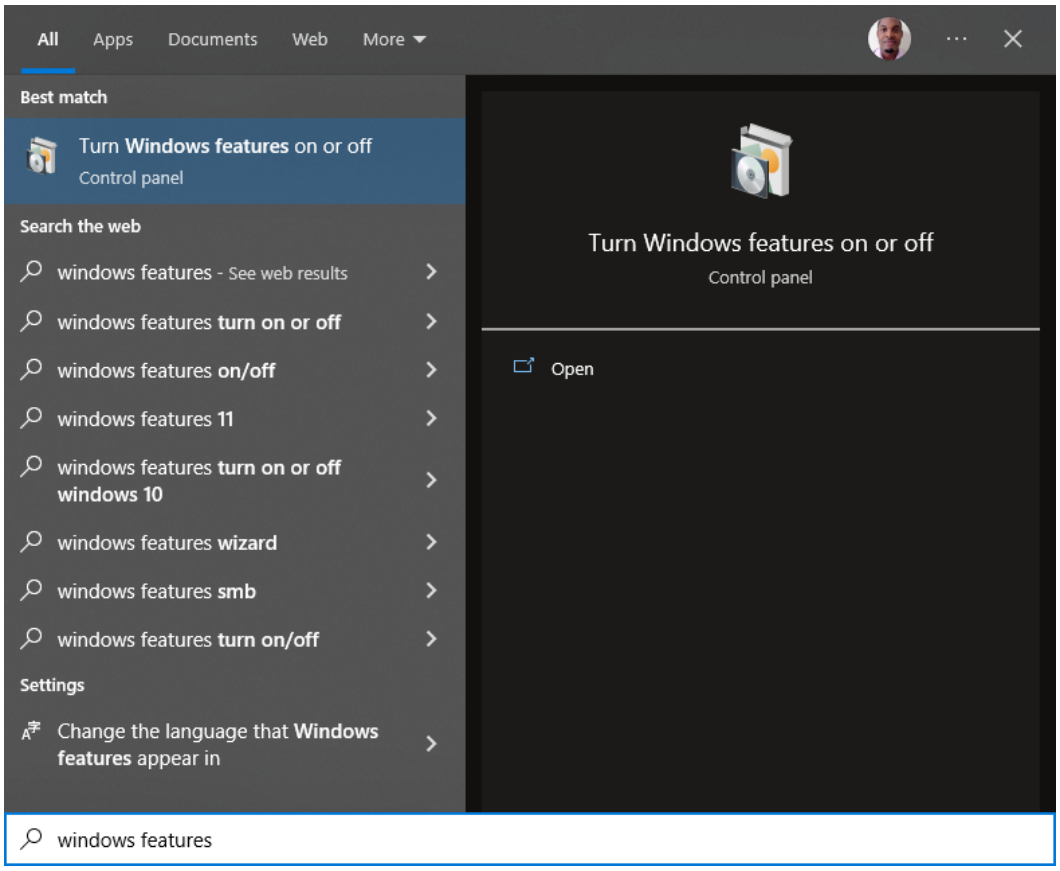
Summary

These deployment instructions are for the windows environment. It uses IIS as the webserver. It includes a guide to install the software prerequisites; a guide to setup the database and then to deploy the site to the IIS web server.

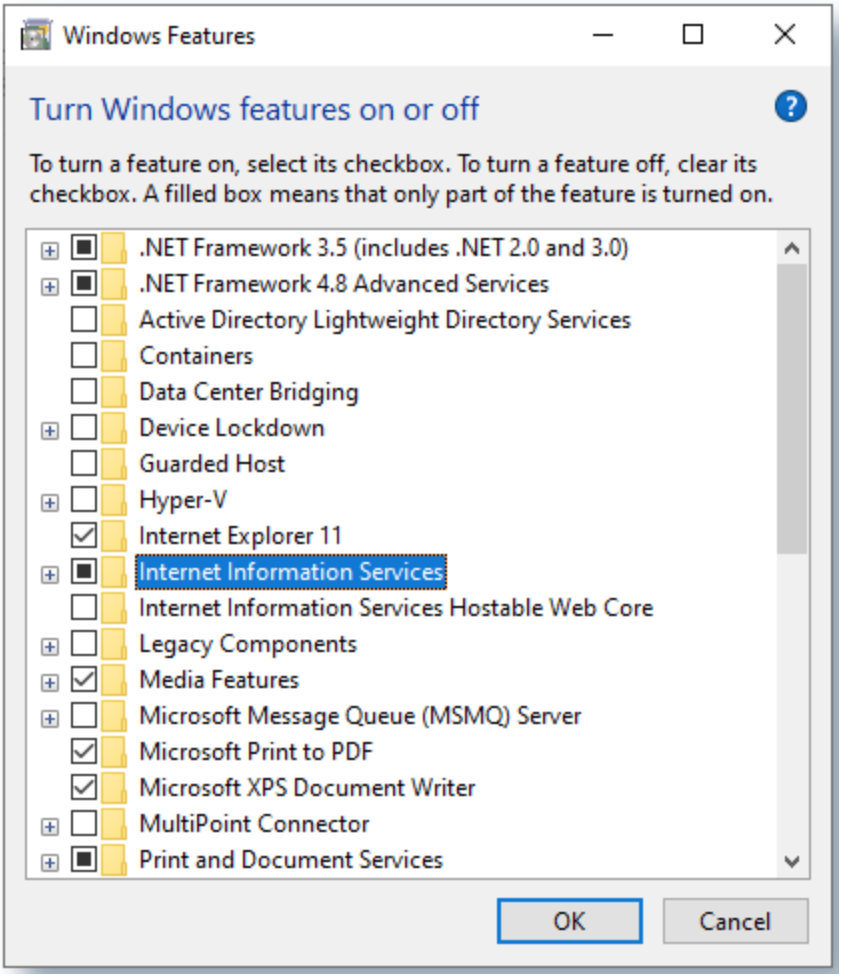
Software Prerequisites

Internet Information Services (IIS):

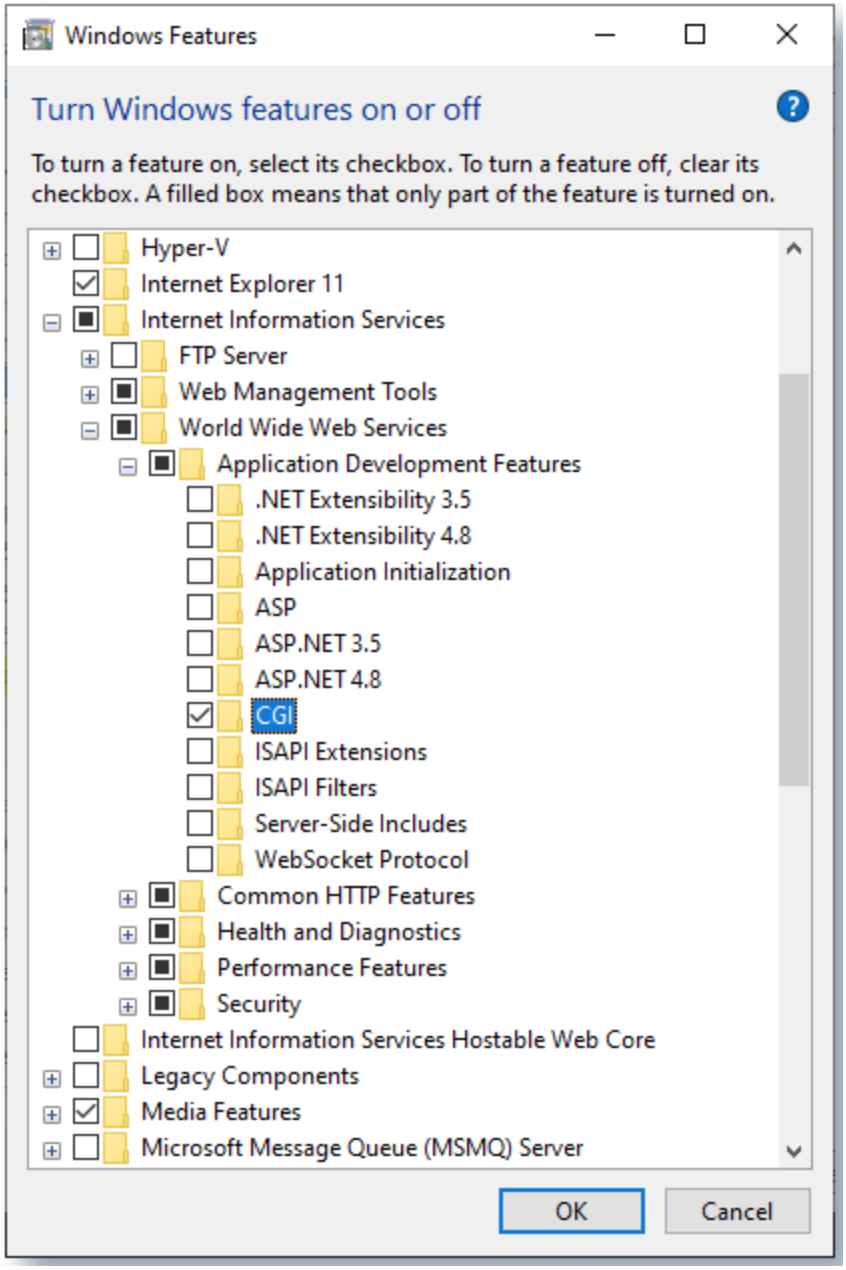
This is the webserver and must be enabled to be used. Search for windows features by clicking the windows button and typing in “Windows features”. Select the option “Turn Windows features on or off”



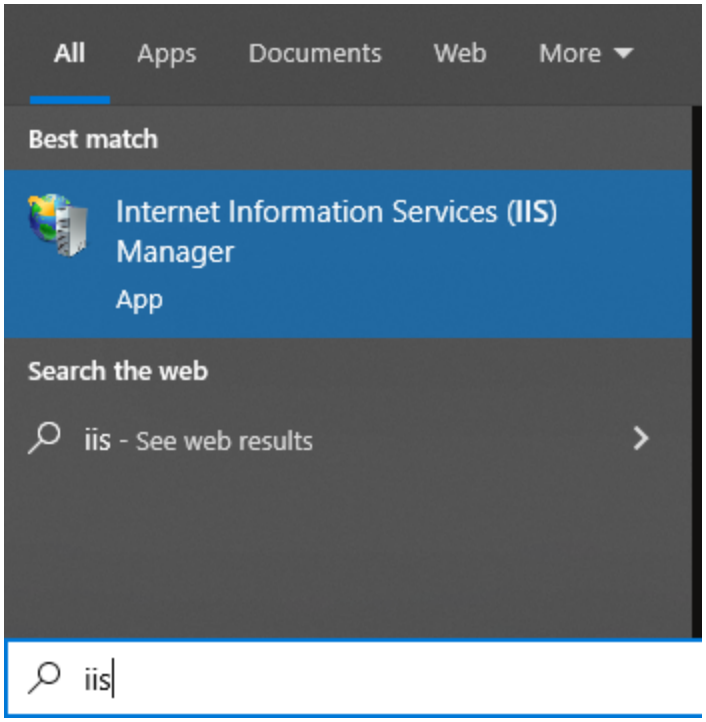
Next navigate to the Internet Information Services option and select it.



The Django framework that we used to implement the tool uses CGI to run on the Web Server. Therefore CGI must be enabled. Do this by navigating to World Wide Web Services -> Application Development Features -> CGI. Select okay to save and enable the changes.

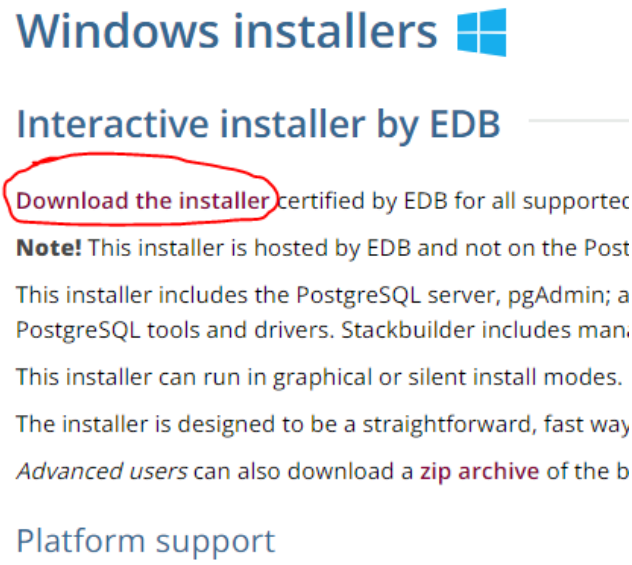


Once this is done you can open the IIS manager by searching for IIS in the start menu.



Python: The web application is implemented using the Django python framework. Therefore python needs to be installed on the system. Once installed additional python libraries will be installed. Python can be installed by going to the https://www.python.org/downloads/ page. From there you could download python and follow the prompt for installation.

PostgreSql: The Database used for the solution is PostgreSql. To install it go to https://www.postgresql.org/download/windows/ .



Install the latest postgresql by downloading the exe file and following the prompt.

Python libraries

Install the python libraries using the following instructions (Note this references the attached requirements.txt file). Open your command line and change the directory to that of the requirements.txt. Enter and execute the following command.

```
pip3 install -r requirements.txt
```

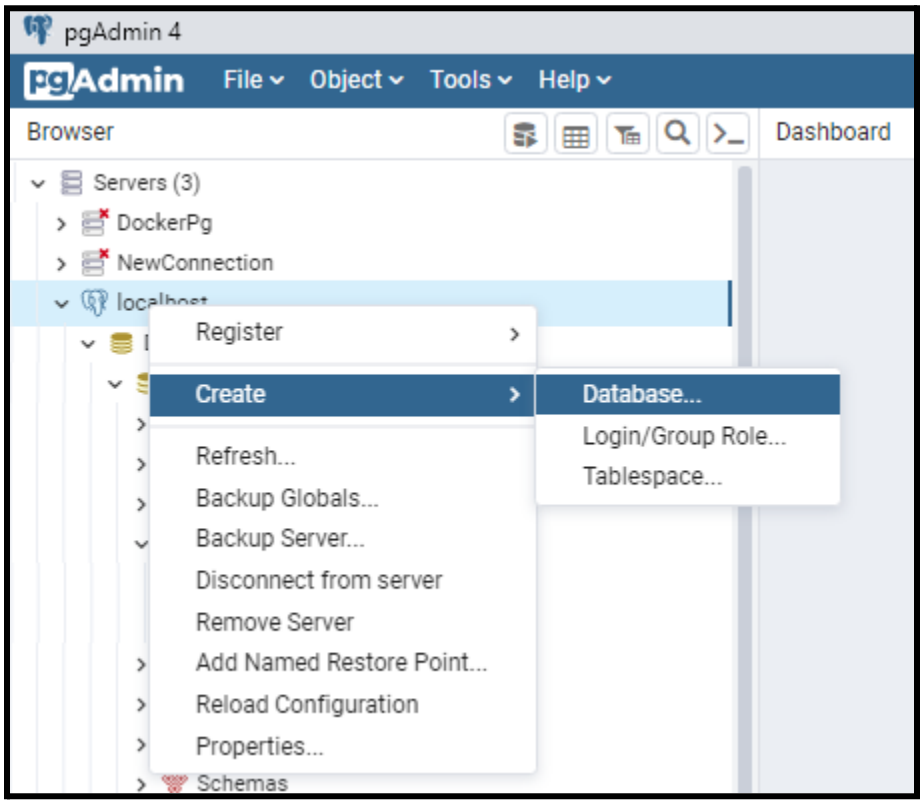
or (Depending on how python is installed)

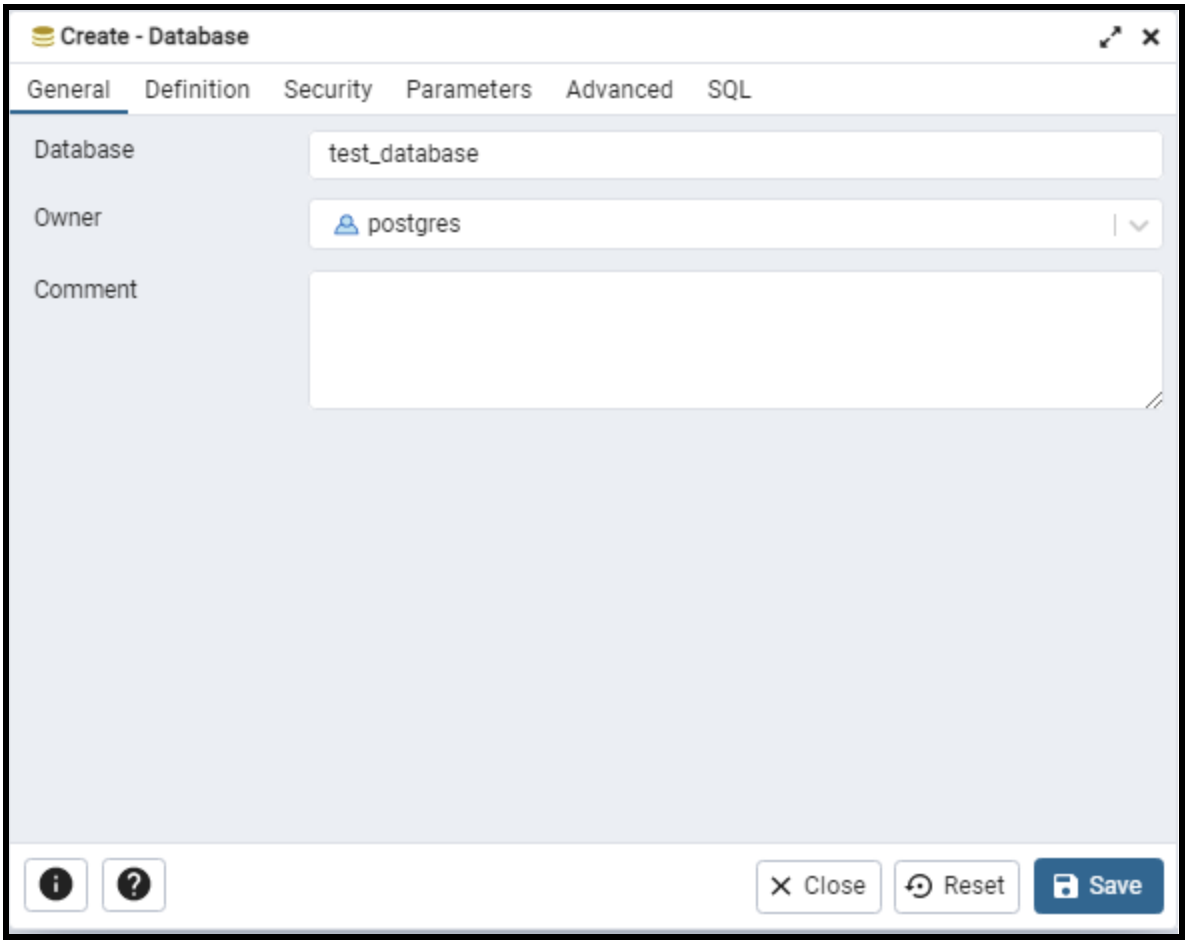
```
pip install -r requirements.txt
```

PostgreSql setup

Create a Database to be used.

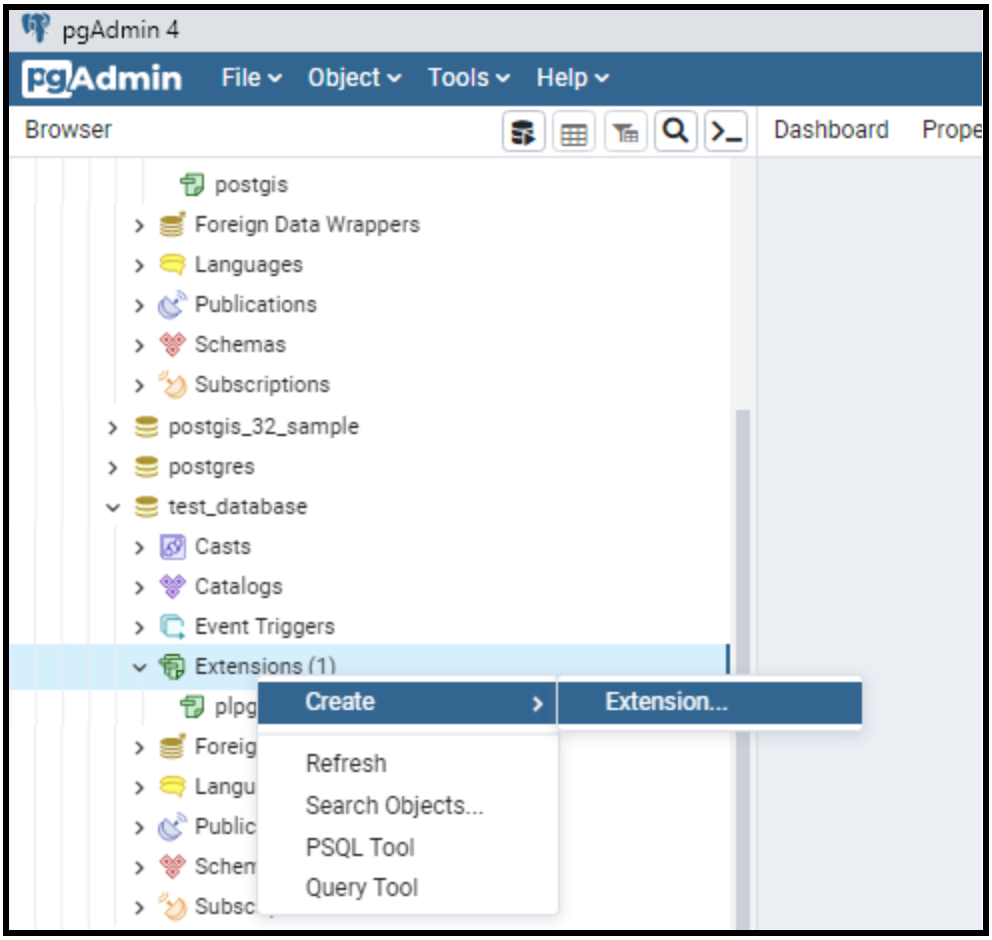
(This can be done by using pgAdmin to connect to the database. pgadmin.org/download/)

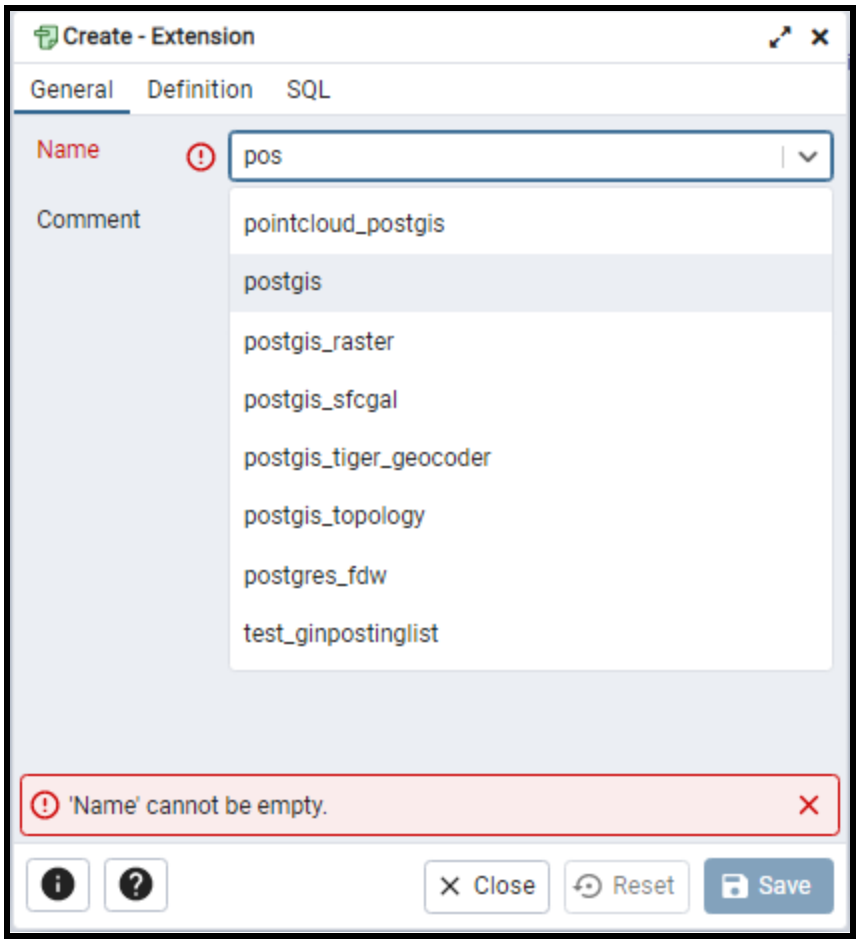




Add PostGIS extension to PostgreSql Database

The postGIS extension allows for spatial queries and calculations and must be installed on the machine before being enabled on the required database. PostGIS can be installed by following the steps found at https://postgis.net/install/. Once installed then the PostGIS extension should be enabled on your database. This can be done using the command line and the instructions found at https://postgis.net/install/ or alternatively it can be enabled using PgAdmin and by following the steps in the images below.


pgAdmin 4
pgAdmin
File
Object
Tools
Help
Browser
Dashboard
Prope
postgis
Foreign Data Wrappers
Languages
Publications
Schemas
Subscriptions
postgis_32_sample
postgres
test_database
Casts
Catalogs
Event Triggers
Extensions (1)
plpg
Create
Extension...
Refresh
Search Objects...
PSQL Tool
Query Tool



Restore Attached database to created database.

This can be done by running the command `psql -U db_user db_name < dump_name.sql`

example: `psql -U db_user test_database < dvdatDb.sql`

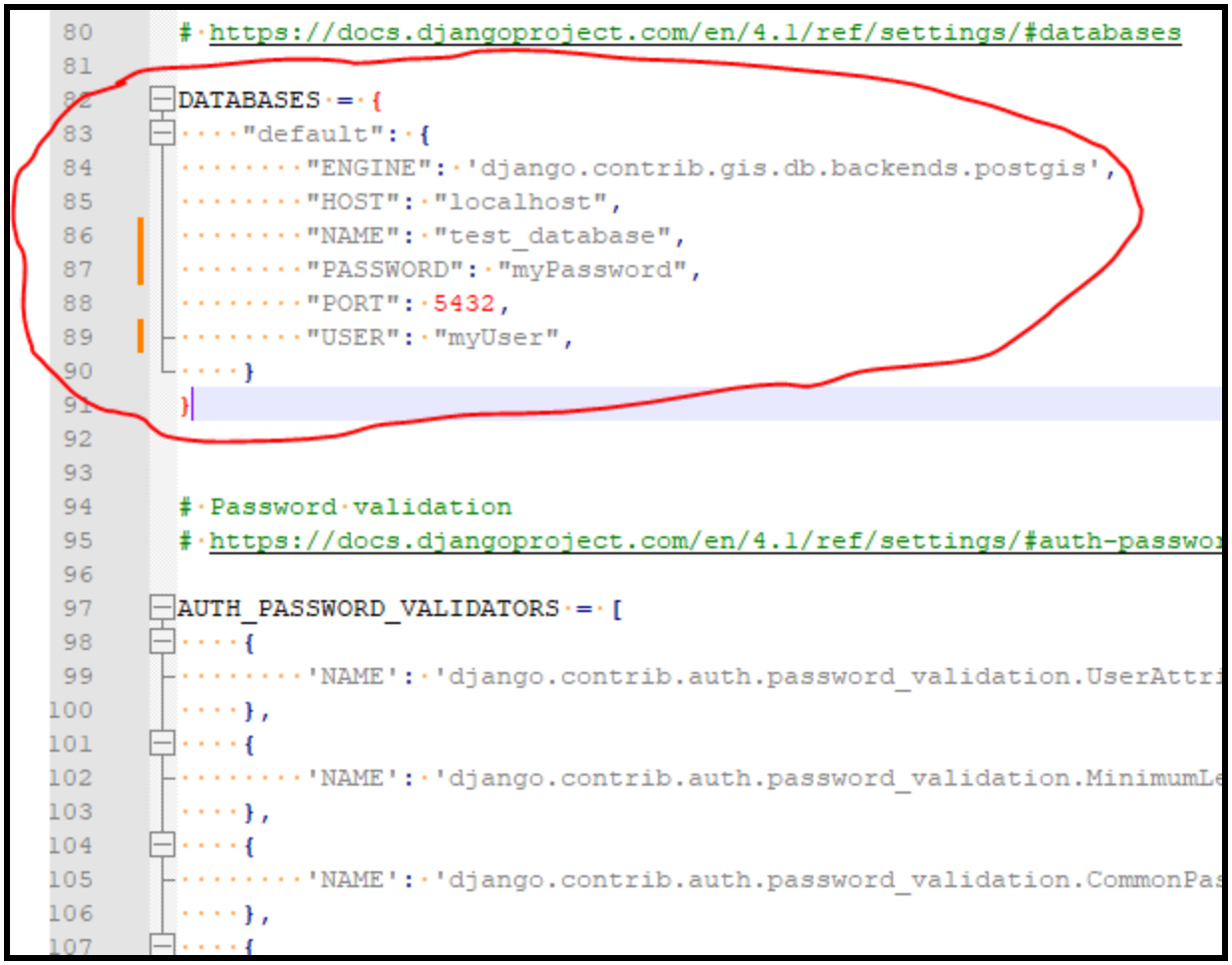
Note: (Your postgresql bin folder should be added to the path environment variable for this to work as shown.)

Change Settings file to point to your database with the correct credentials

Settings file location: **hotMap\hotMap\settings.py**

The options that should be changed are:

- HOST: Database server ip address or location
- NAME: The Database name
- PASSWORD: The password for the user credentials connecting to the database
- USER: The username for the user credentials connecting to the database



```
# https://docs.djangoproject.com/en/4.1/ref/settings/#databases

DATABASES = {
    "default": {
        "ENGINE": 'django.contrib.gis.db.backends.postgis',
        "HOST": "localhost",
        "NAME": "test_database",
        "PASSWORD": "myPassword",
        "PORT": 5432,
        "USER": "myUser",
    }
}


# Password validation
# https://docs.djangoproject.com/en/4.1/ref/settings/#auth-passwo

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttri
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLe
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPas
    },
    {
```

Create Admin User

Once you have changed the settings to connect to your database then you should create a superuser. This is the admin user that will be used to log into the system and can be used to create other users. Use the command line to navigate to your directory with manage.py (two levels up from settings.py). Run the following command and follow the prompt.

```
python manage.py createsuperuser --username=joe --email=joe@example.com
```

(Replace joe and joe@example.com with your desired user).

Deployment instructions.

Open DeploytoIIS.bat file using notepad. Change “SET djangoSitePath“ text near the top to point to the location of the website on your machine.

SET djangoSitePath=C:\xxxx\xxxx\hotMap

Open a command line window and execute deploytoIIS.bat This will deploy the application to your IIS server.