

- [summary](#)
 - [DatBase](#)
 - [sqlit is already configured in the django project](#)
 - [MySQL Create database and configure](#)
 - [Use PostgreSQL](#)
 - [Make migrations](#)
- [USE axios To handle JavaScript API](#)
- [to use docker](#)

summary

- create requirements File `pip freeze > requirements.txt`

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

- install python

```
python -m pip install --upgrade pip
python -m pip --version
```

- install django `pip install django`

- create env `python -m venv env`

- activate venv

- command

```
cd env
Scripts\activate
```

`deactivate`

- In PowerShell :

```
cd venv
Scripts\Activate.ps1
```

- install requirements ``

DatBase

sqlite is already configured in the django project

MySQL Create db and configure

```
- use Xampp server
-Creat Dat base in php my admin

- add database config in app setting file``      DATABASES = {
'default': {                                'ENGINE': 'django.db.backends.mysql',
'NAME': 'cs50_summary',                      'USER': 'root',                      'PASSWORD':
'',                                           'HOST': 'localhost',                      'PORT': '3306',
}      }      ``

- install mysqlclient

``      pip install mysqlclient      ``
```

Use PostgreSQL

-pip install postgres

- for local host
- [Download](#)
- after install App
- open sql Shell `sh psql -U postgres CREATE DATABASE postgres 5432 <!-- #port--> Username: postgres Password :*****`

```
sh CREATE DATABASE cs50_summary; CREATE USER postgres WITH
PASSWORD '123456'; GRANT ALL PRIVILEGES ON DATABASE postgres TO
postgres; check db work
```

```
psql -h localhost -U postgres postgres
```

- update setting.py `ALLOWED_HOSTS = ['localhost', '127.0.0.1']` -option:add
pgpass.conf `` localhost:5432:postgres:your_db_user:postgres

Make makemigrations

- offerc choise db make imgratian

```
python manage.py makemigrations
```

```
python manage.py migrate
```

- run app `python manage.py runserver`
- createsuperuser

```
python manage.py createsuperuser
```

- usee Sass

```
sass --watch style.scss styles.css
```

USE axios To handel JavaScript API

```
``  
npm install axios  
or use it in page`<script  
src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>`  
``
```

to use docker

- creat Dockerfile

```
# Use an official Python runtime as a parent image
FROM python:3.12-slim

# Set environment variables
ENV PYTHONDONTWRITEBYTECODE 1
ENV PYTHONUNBUFFERED 1

# Set the working directory
WORKDIR /app

# Install dependencies
COPY requirements.txt /app/
RUN pip install --upgrade pip
RUN pip install -r requirements.txt

# Copy the rest of the application code
COPY . /app/

# Expose the port the app runs on
EXPOSE 8000

# Command to run the application
CMD ["gunicorn", "--workers", "3", "--bind", "0.0.0.0:8000",
"your_project_name.wsgi:application"]
```

- create Docker Compose

```
version: '3.8'

services:
  web:
    build: .
    command: gunicorn your_project_name.wsgi:application --bind 0.0.0.0:8000
    volumes:
      - ./app
    ports:
      - "8000:8000"
    env_file:
      - .env
    depends_on:
      - db

  db:
    image: postgres:15
    volumes:
      - postgres_data:/var/lib/postgresql/data/
    environment:
      POSTGRES_DB: your_db_name
      POSTGRES_USER: your_db_user
```

```
POSTGRES_PASSWORD: your_db_password
POSTGRES_HOST_AUTH_METHOD: trust
```

```
volumes:
  postgres_data:
```

- edit setting.py

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': os.getenv('POSTGRES_DB'),
        'USER': os.getenv('POSTGRES_USER'),
        'PASSWORD': os.getenv('POSTGRES_PASSWORD'),
        'HOST': 'db',
        'PORT': 5432,
    }
}
```

- creat .env file

```
DEBUG=1
SECRET_KEY=your_secret_key
POSTGRES_DB=your_db_name
POSTGRES_USER=your_db_user
POSTGRES_PASSWORD=your_db_password
```

- start project

```
docker-compose up --build
```

- start imgration

```
docker-compose run web python manage.py migrate
```

- to stop docker

```
docker-compose down
```

