

Exploring Django with Docker

Student: Erdana Seitzhan

Date of Submission: 11.oct.2024

Table of content

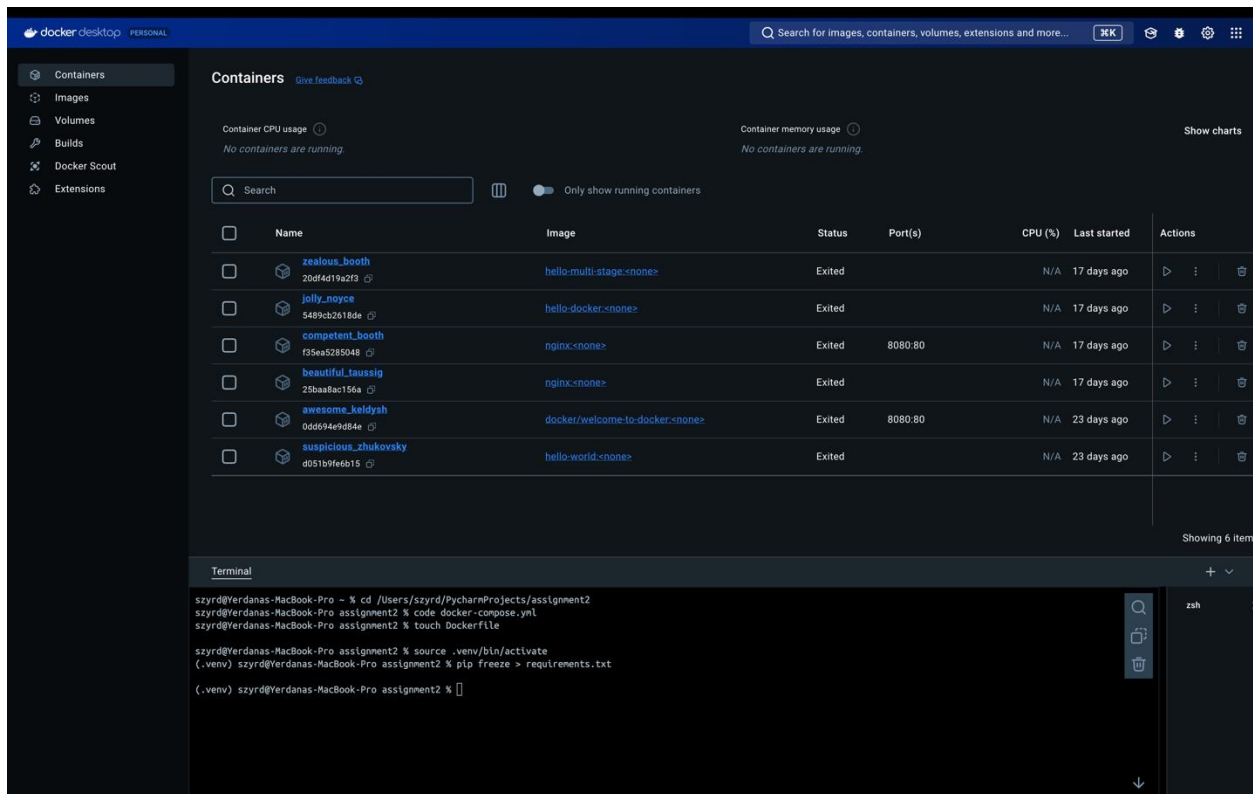
1. Introduction
2. Docker Compose
3. Docker Networking and Volumes
4. Django Application setup
5. Conclusion
6. References

Introduction

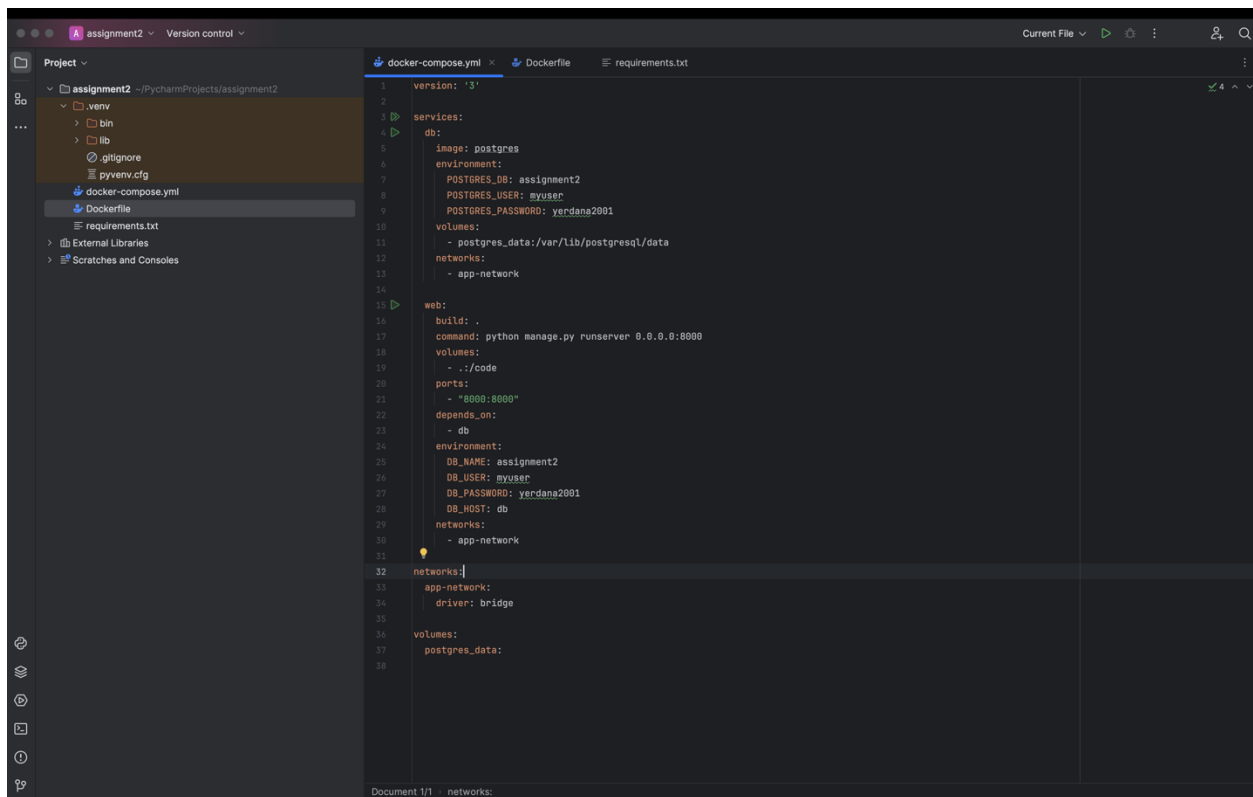
The objective of this assignment was to explore the integration of Django within a Dockerized environment using Docker Compose. The task involved configuring services such as a Django web server and a PostgreSQL database, setting up Docker networking, and managing data persistence with volumes.

Docker Compose

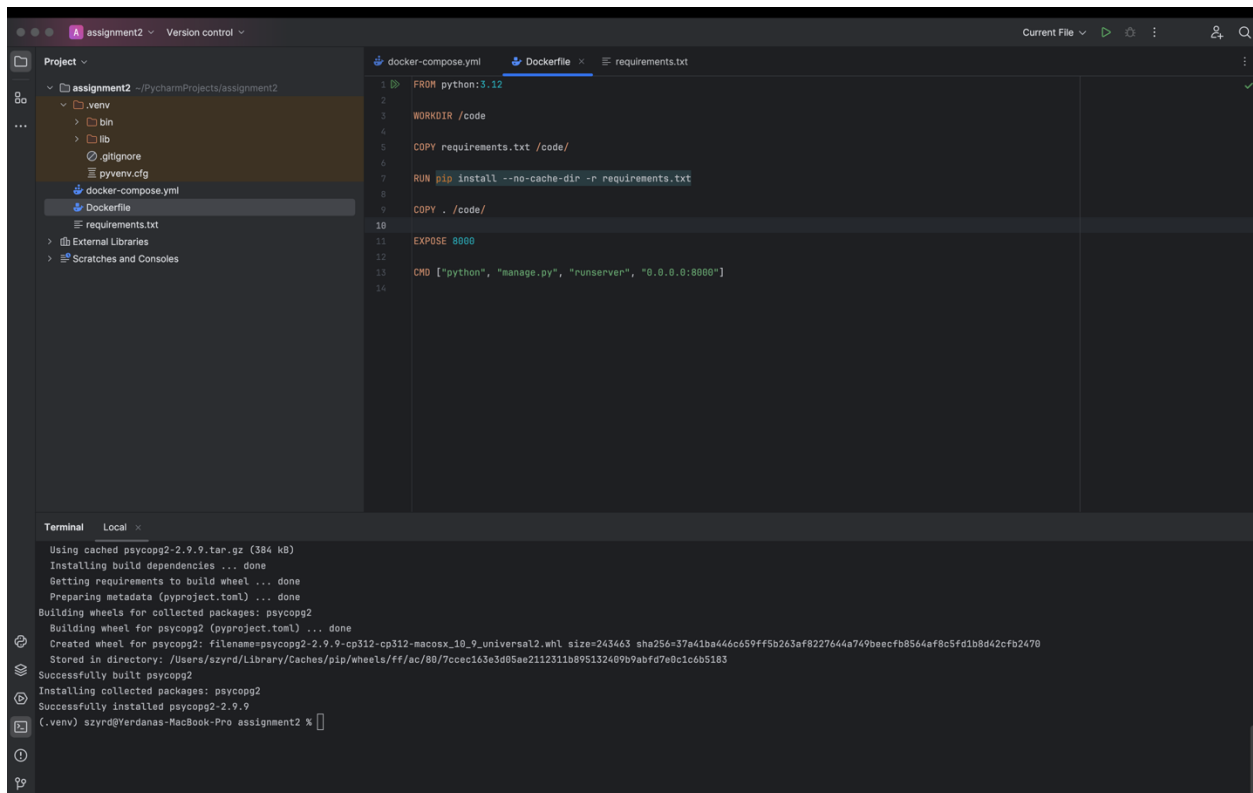
Docker Compose is a tool that allows us to define and run multi-container Docker applications. In this project, I created a docker-compose.yml file that sets up two services: a Django web server and a PostgreSQL database.



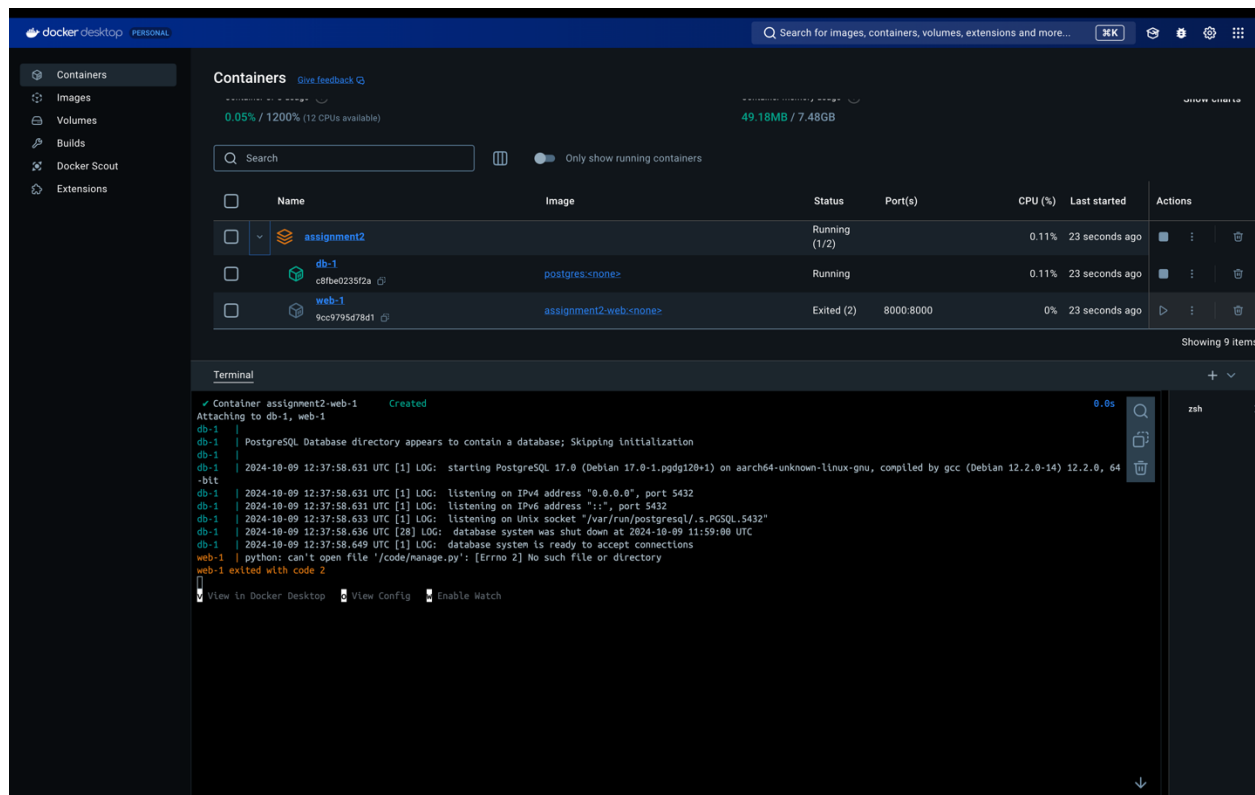
The docker-compose.yml file includes two services: a Django web server and a PostgreSQL database. Environment variables such as DB_NAME, DB_USER, and DB_PASSWORD were defined to configure the database connection.



And I also create DockerFile



To build and run the containers, I used the command: `docker-compose up --build`



Docker Networking and Volume

Networking

A custom network was set up in the docker-compose.yml file to allow seamless communication between the Django web service and the PostgreSQL database. Docker automatically handles the internal network, allowing services to communicate by name. The db service, representing PostgreSQL, was accessible by the Django web server through the hostname db, simplifying the connection setup.

Volumes

Two volumes were defined in the Docker Compose file. One volume was used to persist PostgreSQL data, ensuring the database information is retained across container restarts. The other volume was created for Django's static files and uploaded media, allowing file persistence beyond the lifecycle of individual containers.

```
networks:|
  app-network:
    driver: bridge

volumes:
  postgres_data:
```

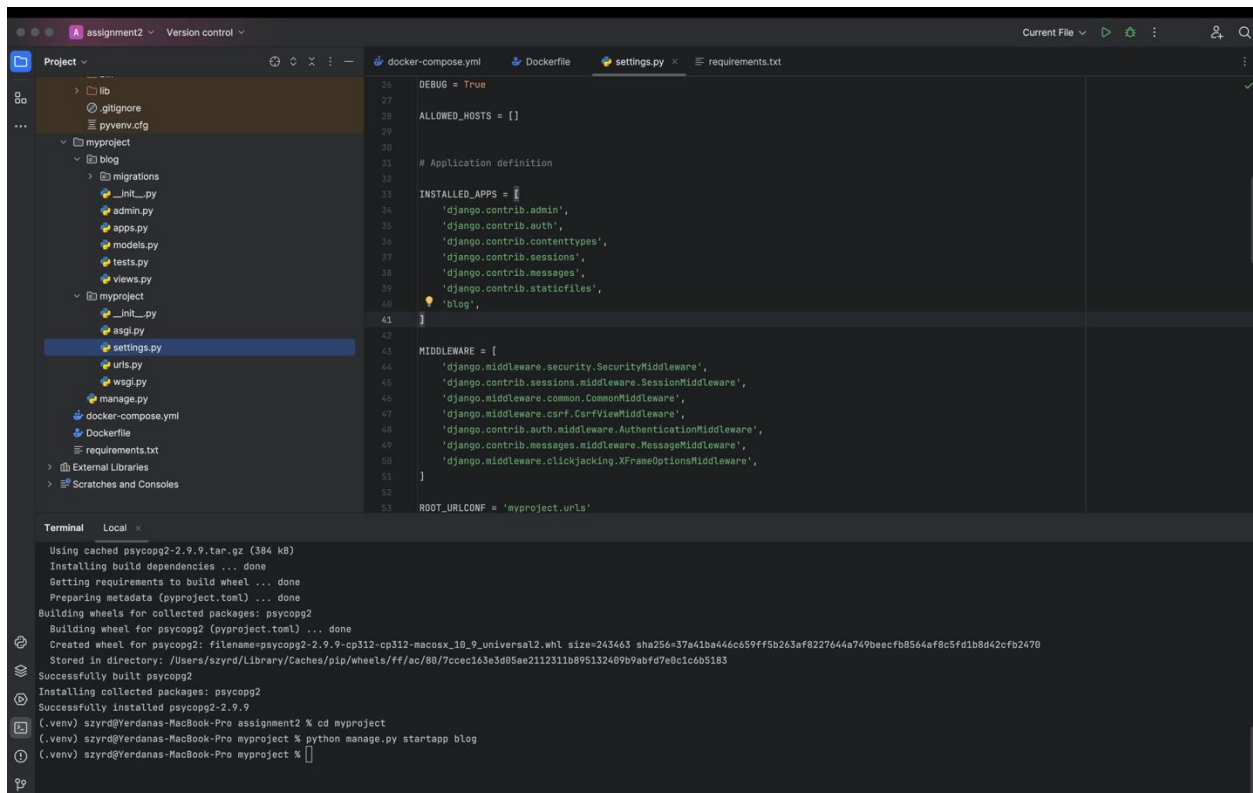
Django Application Setup

The Django project was initialized using the following terminal command:

`django-admin startproject myproject`

I then created a simple app called blog using: `python manage.py startapp blog`

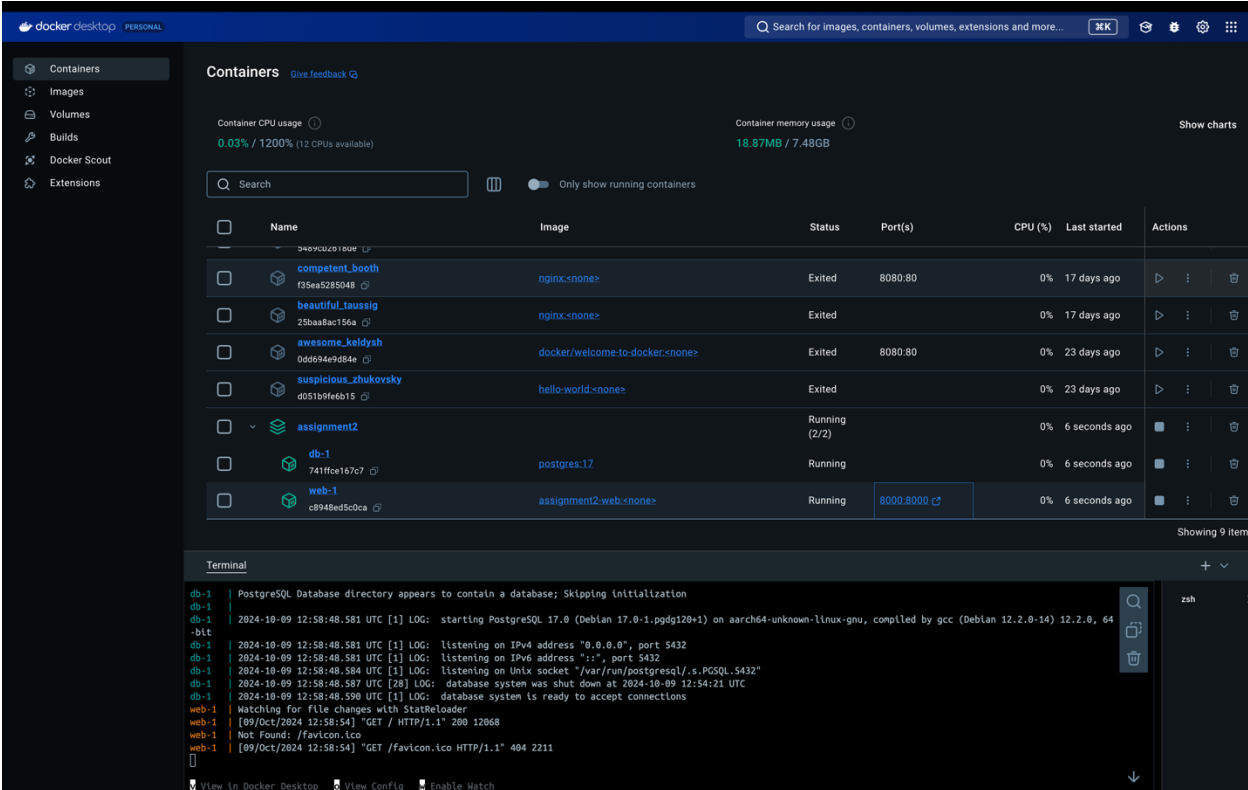
The PostgreSQL database was configured by modifying the DATABASES setting in the settings.py file.



The screenshot shows an IDE with the following components:

- Project Explorer:** Shows a project structure with folders like 'blog' and 'myproject', and files like 'settings.py', 'urls.py', and 'manage.py'.
- Editor:** Displays the 'settings.py' file. The 'INSTALLED_APPS' list includes 'django.contrib.admin', 'django.contrib.auth', 'django.contrib.contenttypes', 'django.contrib.sessions', 'django.contrib.messages', 'django.contrib.staticfiles', and 'blog'. The 'MIDDLEWARE' list includes 'django.middleware.security.SecurityMiddleware', 'django.contrib.sessions.middleware.SessionMiddleware', 'django.middleware.common.CommonMiddleware', 'django.middleware.csrf.CsrfViewMiddleware', 'django.contrib.auth.middleware.AuthenticationMiddleware', 'django.contrib.messages.middleware.MessageMiddleware', and 'django.middleware.clickjacking.XFrameOptionsMiddleware'. The 'ROOT_URLCONF' is set to 'myproject.urls'.
- Terminal:** Shows the output of the command `python manage.py startapp blog`. It indicates that the package 'psycopg2' was successfully installed and that the application was created.

Then using command `docker-compose up --build` to successful to build up and run both web and database.



I ran migrations using the following command to set up the database schema:

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

docker desktopPERSONAL

Search for images, containers, volumes, extensions and more...

Containers

Images

Volumes

Builds

Docker Scout

Extensions

Containers

Give feedback

Search

Only show running containers

	Name	Image	Status	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	assignment2		Running (2/2)		0.8%	35 seconds ago	
<input type="checkbox"/>	db-1	postgres:17	Running		0.04%	35 seconds ago	
<input type="checkbox"/>	web-1	assignment2-web:latest	Running	8000:8000	0.76%	35 seconds ago	

Showing 9 items

Terminal

```
db-1 | 2024-10-09 12:59:51.085 UTC [1] LOG: database system is shut down
db-1 exited with code 0
(.venv) szryd@Verdanas-MacBook-Pro assignment2 % docker-compose exec web python manage.py migrate

WARN[0000] /Users/szyrd/PycharmProjects/assignment2/docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
service "web" is not running
(.venv) szryd@Verdanas-MacBook-Pro assignment2 % docker-compose exec web python manage.py migrate

WARN[0000] /Users/szyrd/PycharmProjects/assignment2/docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion

Operations to perform:
Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
(.venv) szryd@Verdanas-MacBook-Pro assignment2 %
```

Install postgresql mac...

Everything You Need...

Installing psycopg2...

Install PostgreSQL on...

Welcome

Thread: shared_preli...

Assignment 2, Web ap...

The install worked su...

localhost:8000

Учебник по веб-а... weavesilk Documents and E... Guitar Tabs with R... Maps Build Software Ap... 7 Bootstrap Forms... CSSO's Web Progr... Email Services - E... DaVinci AI Vector Icons and S... Android Developm...



The install worked successfully! Congratulations!

View [release notes](#) for Django 5.1

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

django



Django Documentation
Topics, references, & how-to's



Tutorial: A Polling App
Get started with Django



Django Community
Connect, get help, or contribute

Conclusion

Working on this assignment showed me how useful Docker can be for managing web apps. Setting up Django with PostgreSQL using Docker Compose made things easier, especially when it came to handling multiple services. Docker networking and volumes helped ensure everything communicated well and kept the data safe. Overall, Docker made the whole process smoother and more efficient, and it's a great tool for making development more consistent and scalable.

References

Django tutorial documentation: <https://docs.djangoproject.com/en/5.1/>

Docker documentation (guid): <https://docs.docker.com/guides/>

PostgreSQL documentation: <https://www.postgresql.org/docs/>

Running Django with PostgreSQL in Docker: <https://medium.com/@jonas.granlund/running-django-with-postgresql-in-docker-a-step-by-step-guide-f6ab3bf05f44>