

web 2

Assignment 2: Web Application Development with Docker and Django

Timur Kazdayev

October 13, 2024

Table of Contents

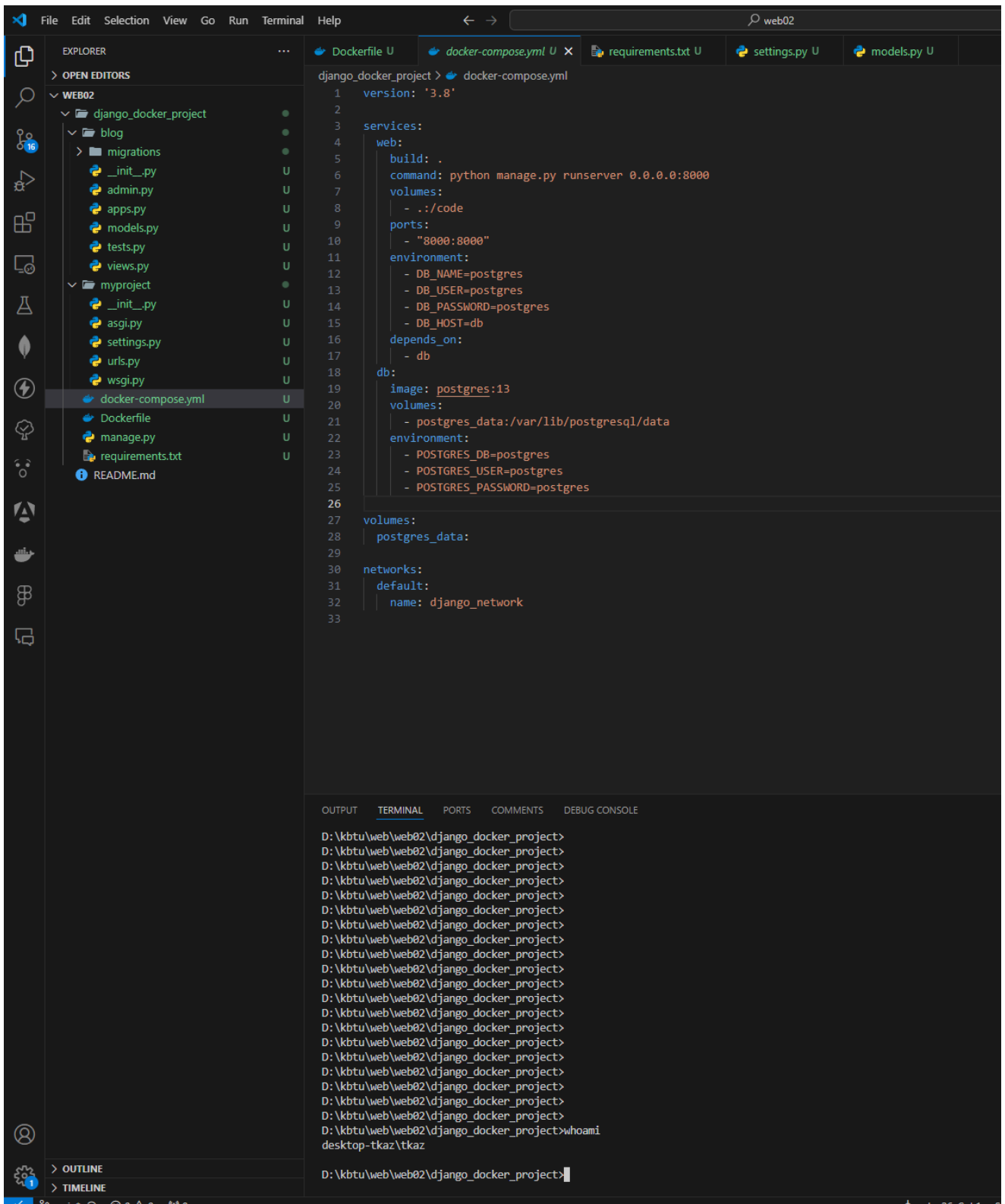
1. [Introduction](#)
2. [Docker Compose](#)
3. [Docker Networking and Volumes](#)
4. [Django Application Setup](#)
5. [Conclusion](#)
6. [References](#)

Introduction

In this assignment, I worked with Docker and Django to create a web application. The main goal was to learn how to use Docker Compose to run Django and a PostgreSQL database in containers.

Docker Compose

I created a `docker-compose.yml` file to set up and run the application. Here's what I did:



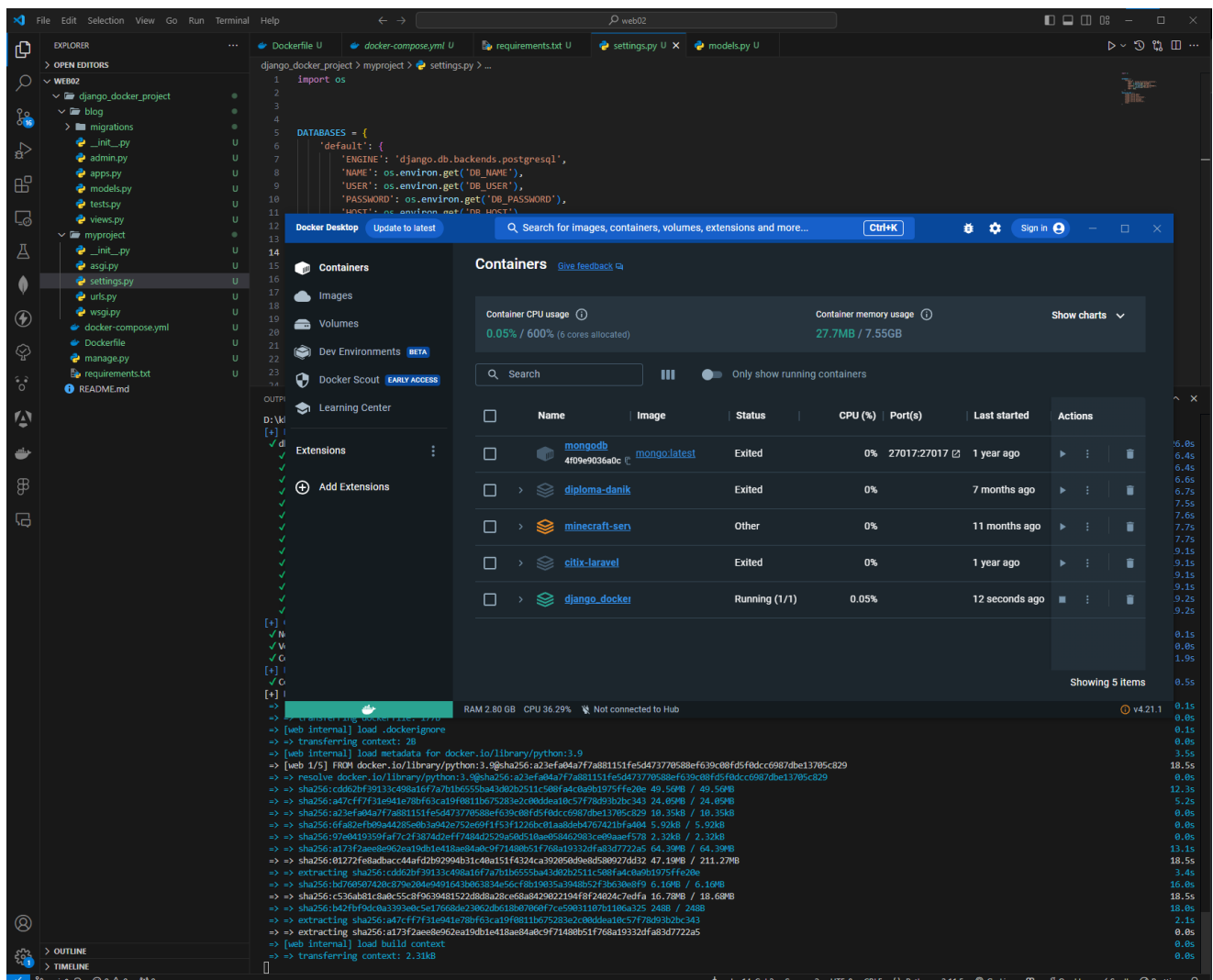
In this file, I described two services:

1. `web` for the Django application
2. `db` for the PostgreSQL database

To run everything, I used the command:

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

Here's what I got:



Docker Networking and Volumes

Docker Compose automatically created a network for the containers. This allowed the Django application to communicate with the database.

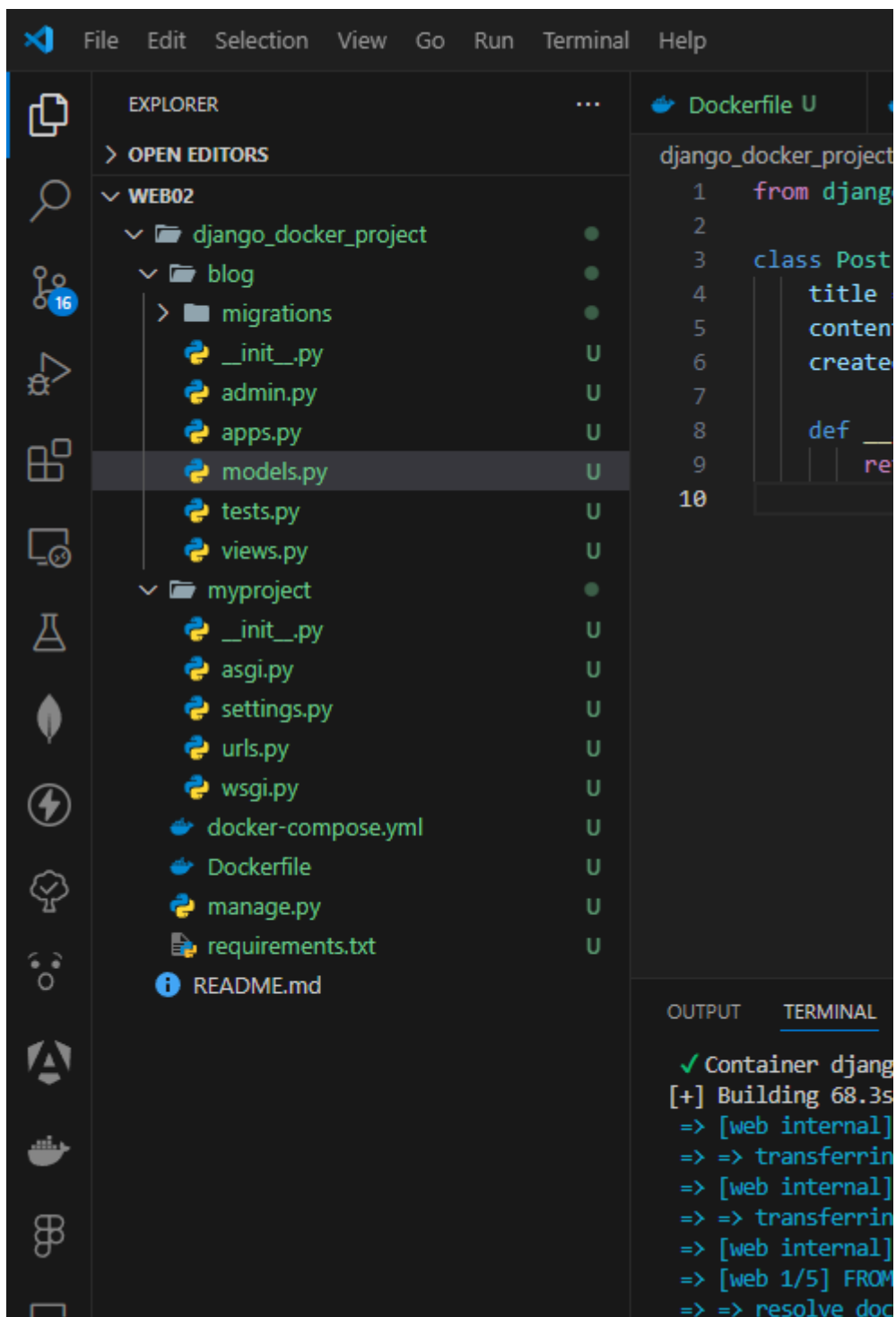
I also used a volume to save the database data:

```
volumes:
  postgres_data:
```

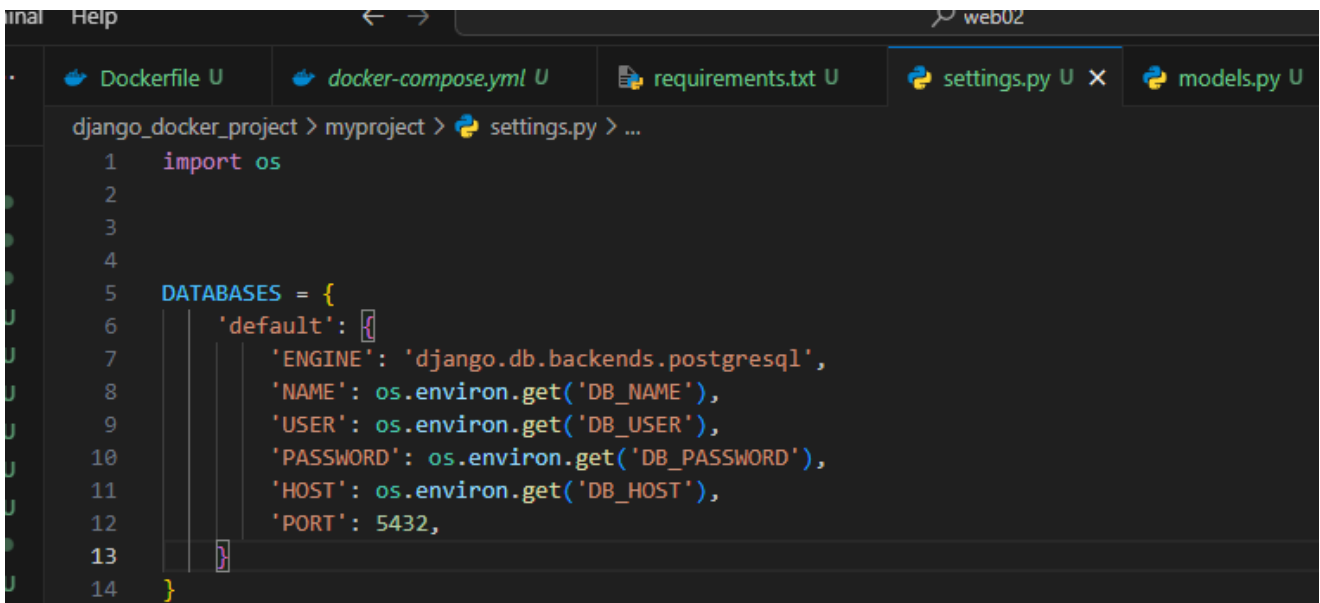
This is needed so that the data doesn't disappear when the containers are stopped.

Django Application Setup

Here's what my project structure looks like:

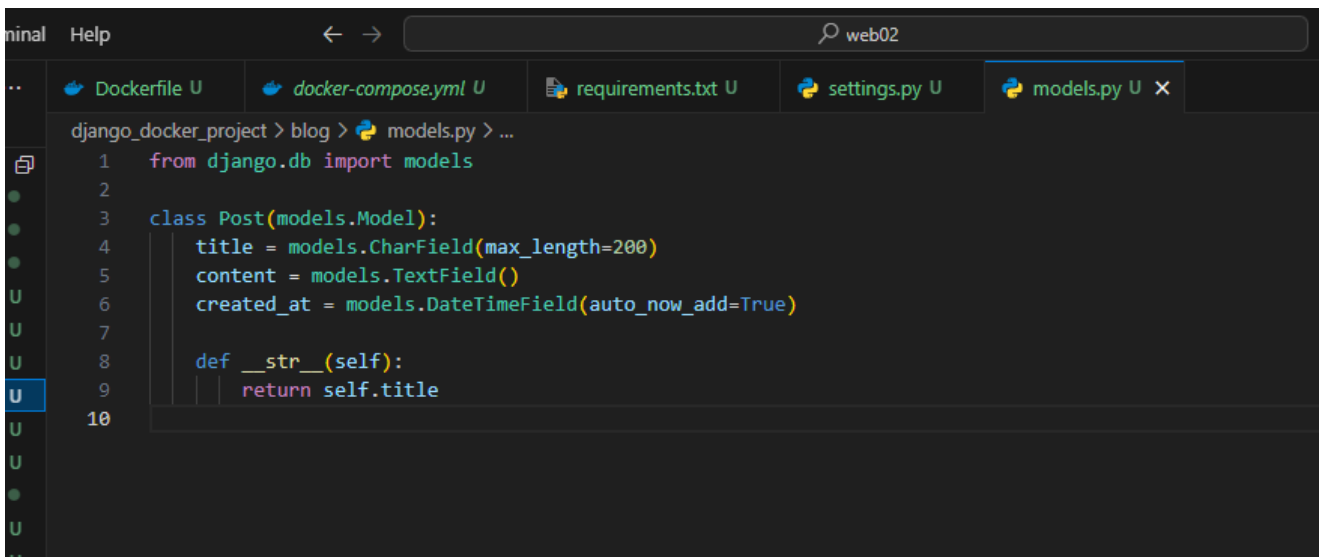


I configured Django to work with PostgreSQL:



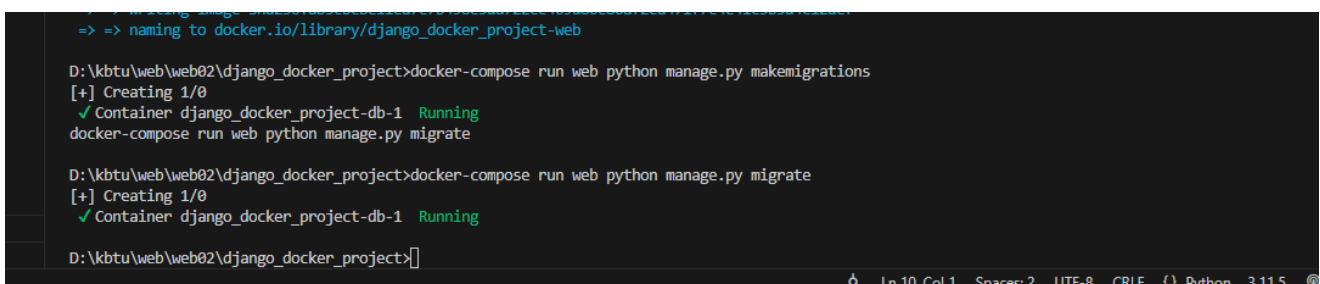
```
django_docker_project > myproject > settings.py > ...
1  import os
2
3
4
5  DATABASES = {
6      'default': {
7          'ENGINE': 'django.db.backends.postgresql',
8          'NAME': os.environ.get('DB_NAME'),
9          'USER': os.environ.get('DB_USER'),
10         'PASSWORD': os.environ.get('DB_PASSWORD'),
11         'HOST': os.environ.get('DB_HOST'),
12         'PORT': 5432,
13     }
14 }
```

And created a simple model for the blog:



```
django_docker_project > blog > models.py > ...
1  from django.db import models
2
3  class Post(models.Model):
4      title = models.CharField(max_length=200)
5      content = models.TextField()
6      created_at = models.DateTimeField(auto_now_add=True)
7
8      def __str__(self):
9          return self.title
10
```

After setting up, I ran migrations and created an admin. Here's what the Django admin looks like:



```
=> => naming to docker.io/library/django_docker_project-web

D:\kbtu\web\web02\django_docker_project>docker-compose run web python manage.py makemigrations
[+] Creating 1/0
✓ Container django_docker_project-db-1 Running
docker-compose run web python manage.py migrate

D:\kbtu\web\web02\django_docker_project>docker-compose run web python manage.py migrate
[+] Creating 1/0
✓ Container django_docker_project-db-1 Running

D:\kbtu\web\web02\django_docker_project>
```

Conclusion

In this assignment, I learned:

- How to create a `docker-compose.yml` file

- How to set up services for a web application and database
- How to use volumes to save data
- How to configure Django to work with Docker and PostgreSQL

It was challenging, especially setting up the database, but overall Docker made the development process easier.

Reference

- My GitHub Repository: <https://github.com/retsaftu/web02>