**Assignment 1: Containerizing Microblog with Docker**

For this assignment, I chose to containerize a Flask web app called ***Microblog***. It’s from Miguel Grinberg’s Flask Mega-Tutorial. I wanted to make sure the application would run smoothly in a container, while also following good Docker practices like using a lightweight image, avoiding root access, and setting things up in a clean and secure way.

**GitHub Repo**: <https://github.com/TwinkleM97/my-microblog-app>

* **Step 1: Created the Dockerfile**
* I started with the python:3.11-slim image because it’s lightweight and secure. I added a few environment variables to keep the output clean and made sure Flask would run correctly.
* **Step 2: Installed dependencies**
* I installed system libraries like gcc and libpq-dev, which are needed to build some Python packages. Then I used **pip** to install everything from **requirements.txt**.
* **Step 3: Copied my code and boot script**
* I added the application files (like app/, migrations/, config.py, and microblog.py) to the image, and included a boot.sh script that runs the app with **Gunicorn**. I also made sure the script had the right permissions.
* **Step 4: Added a non-root user**
* For security reasons, I created a user named flaskuser and gave it ownership of the app directory, so the app doesn’t run as root.
* **Step 5: Set up logging with a bind mount**
* To keep logs persistent, I used a bind mount that maps a local logs/ folder to /app/logs inside the container.
* Once the image was built, I ran the container with below command:

**docker run -d -p 5000:5000 \**

**--mount type=bind,source="$(pwd)/logs",target=/app/logs \**

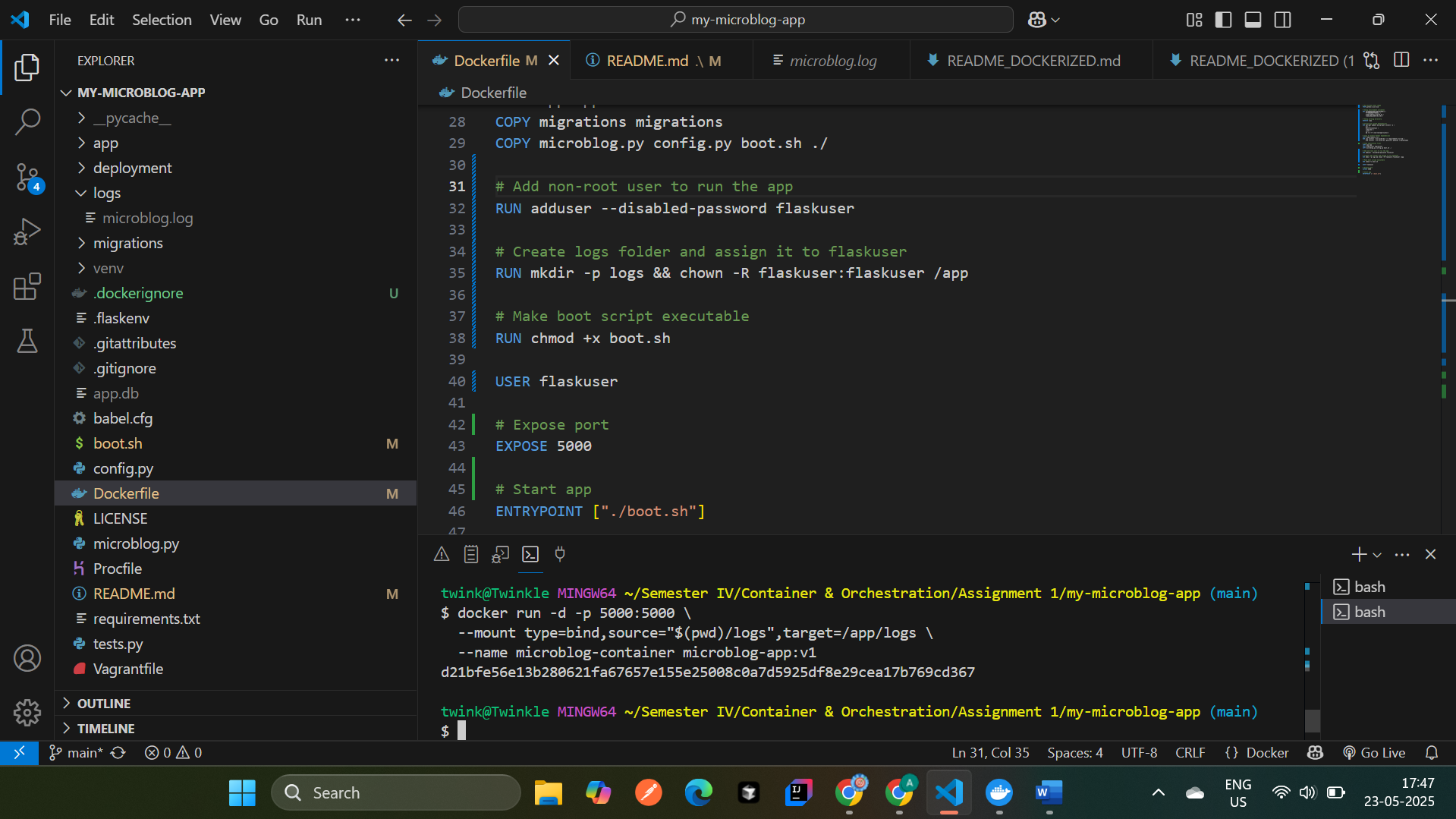
**--name microblog-container microblog-app:v1**

* **What Challenges I faced:**
* At first, the app kept crashing with a **PermissionError** when trying to write to the **logs** directory. I realized it was because the **flaskuser** didn’t have access. I fixed it by creating the ***logs/ folder*** manually and setting up the bind mount properly.
* **Conclusion:**
* This project really helped me understand how to structure Docker images effectively and securely. Setting up volumes, dealing with permissions, and getting everything running inside a container was super satisfying in the end.

**Screenshot 1: Docker Image Built Successfully Using docker build Command**A screenshot of a computer

AI-generated content may be incorrect.

### **Screenshot 2: Running the Container with Bind Mounts Using docker run**



**Screenshot 3: Docker Desktop - Container Running**A screenshot of a computer

AI-generated content may be incorrect.

### **Screenshot 4: Web Application Accessible in Browser**

A screenshot of a computer

AI-generated content may be incorrect.