**GitHub Integration via MobaXterm and Local Deployment with Docker**

**Objective**

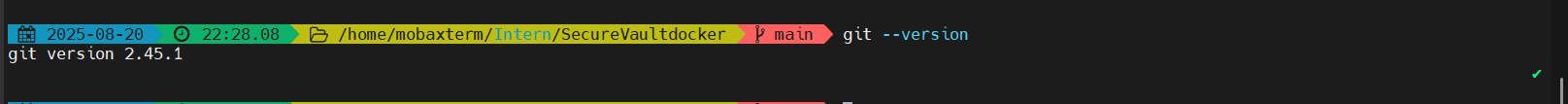
The objective was to use MobaXterm as the terminal environment on Windows to:

* Configure Git and authenticate with GitHub.
* Clone an existing repository, make changes, and push updates back to GitHub.
* Build and run the project locally using Docker.
* Document the workflow with narrative and screenshots, including an inventory of local Docker images.

**1. Preparing the Environment in MobaXterm**

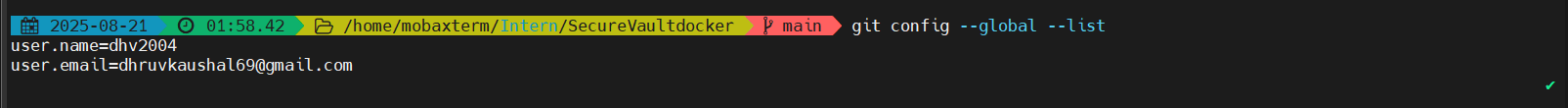
I started by verifying the availability of required tools from within MobaXterm:

* Confirmed Git installation using git --version.
* Verified Docker CLI connectivity using docker version. If the Docker daemon wasn’t reachable, I ensured Docker Desktop was running and, if needed, switched to the default context using docker context use default.

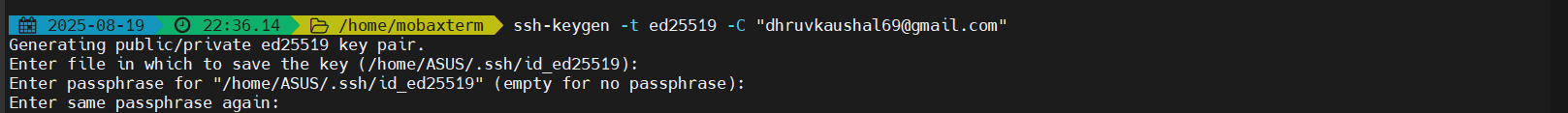


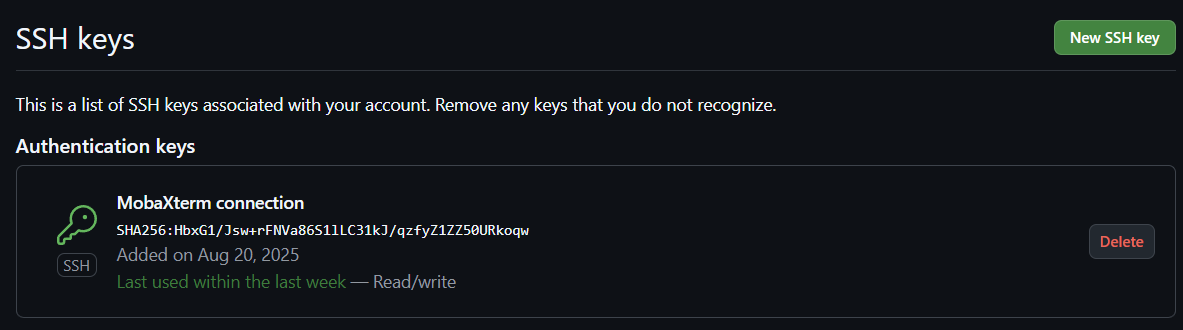
To enable authenticated pushes, I configured Git identity:

* Set user.name and user.email globally so commits are properly attributed.

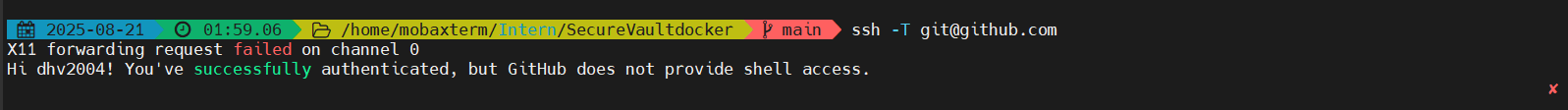


* Generated an SSH key (ed25519) and added the public key to my GitHub account under SSH and GPG keys.





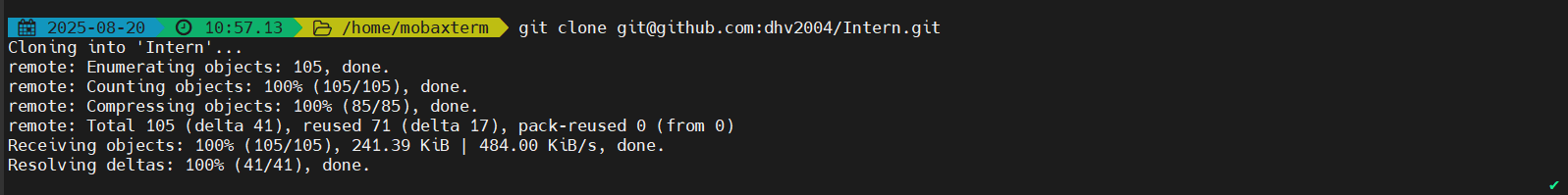
* Validated the SSH setup using ssh -T [git@github.com](mailto:git@github.com) to confirm GitHub recognized the key.



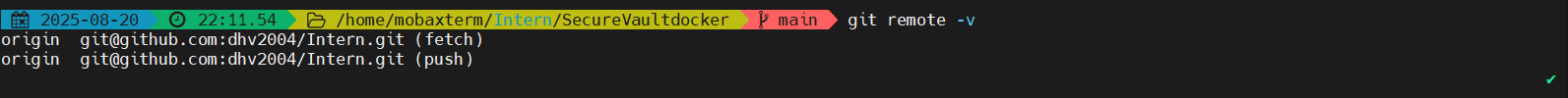
**2. Cloning the Repository with MobaXterm**

Using SSH for reliability, I cloned the parent repository (not a subfolder) to ensure full history and proper Git operations:

* git clone [git@github.com:dhv2004/Intern.git](mailto:git@github.com:dhv2004/Intern.git)

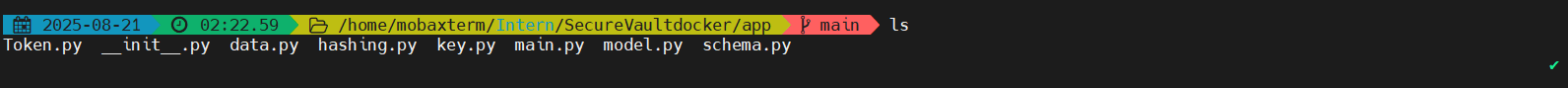


* Verified the remote origin with git remote -v .



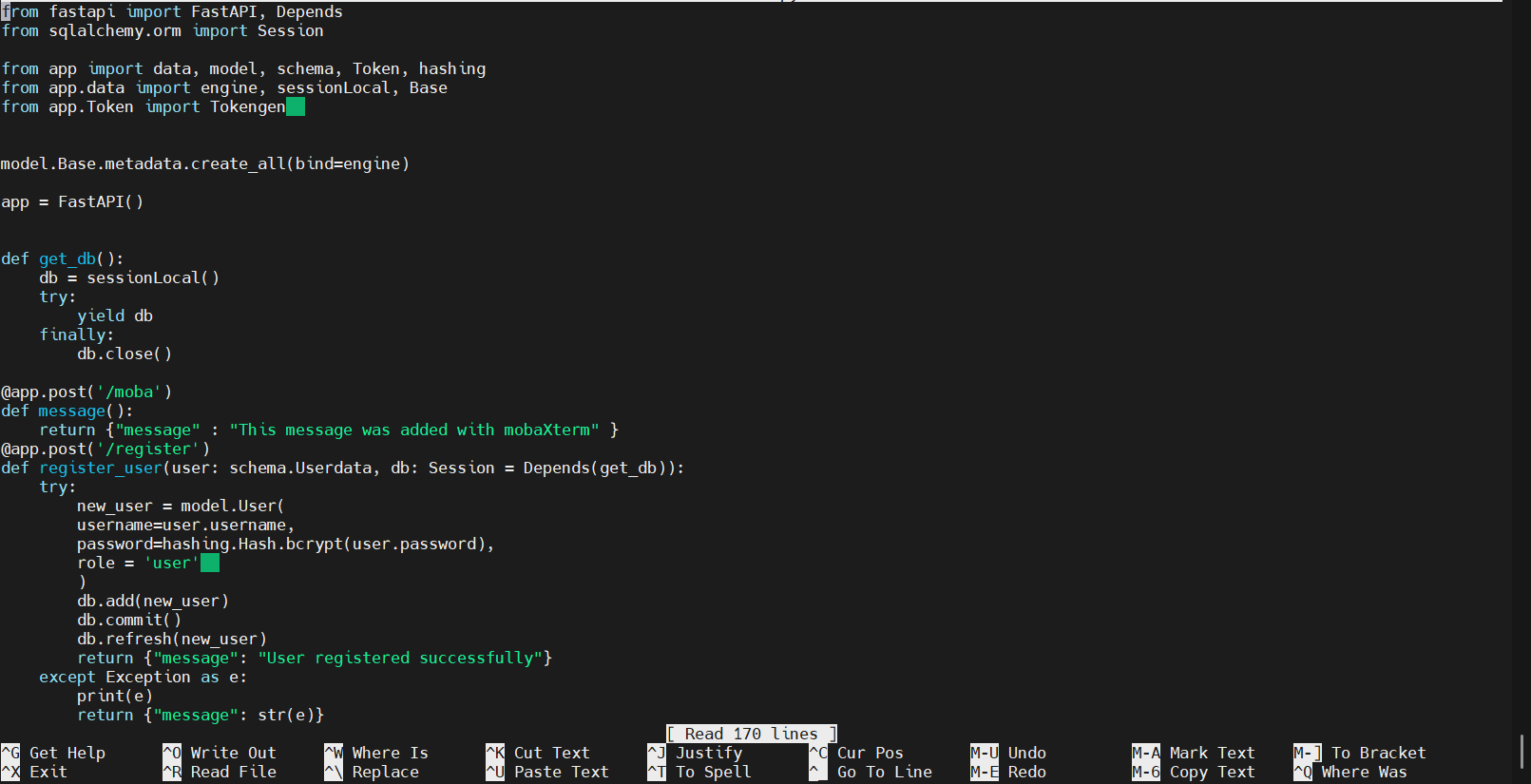
I then navigated into the project path:

* cd Intern/SecureVaultdocker/app
* Confirmed the expected file layout with ls and validated the target file (main.py).



**3. Making and Tracking Changes**

I used a terminal editor (nano) in MobaXterm to modify main.py, saving the changes locally. Before committing, I reviewed the working tree state with git status.



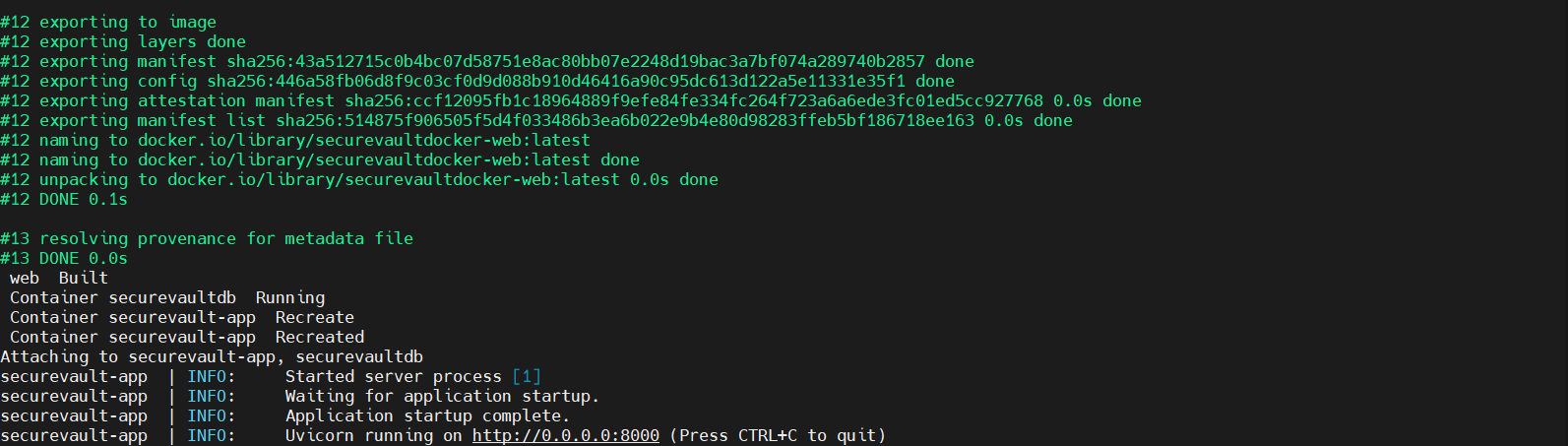
I staged the updated file and committed with a descriptive message:

* git add SecureVaultdocker/app/main.py
* git commit -m "Updating with help of moba ."
* git push

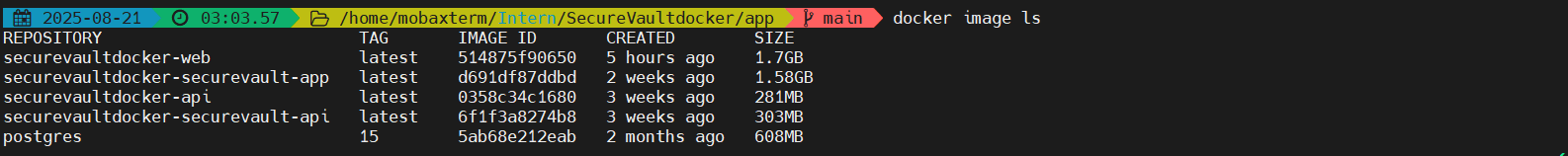


**5. Local Deployment with Docker**

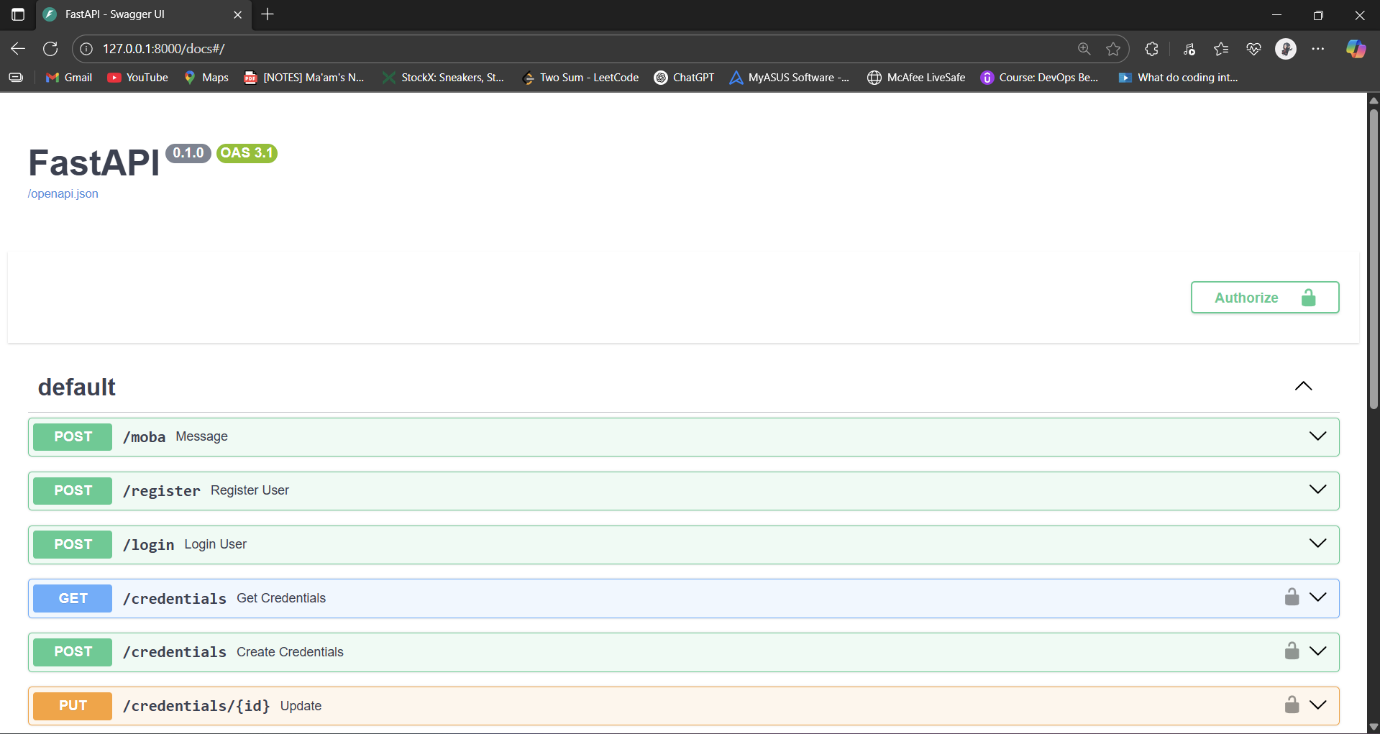
* To run the project locally, I used Docker. With Docker Desktop running, I navigated to the directory containing the Dockerfile and docker-compose.yml.
* For projects requiring Postgres, I ensured a Postgres instance was available Compose.



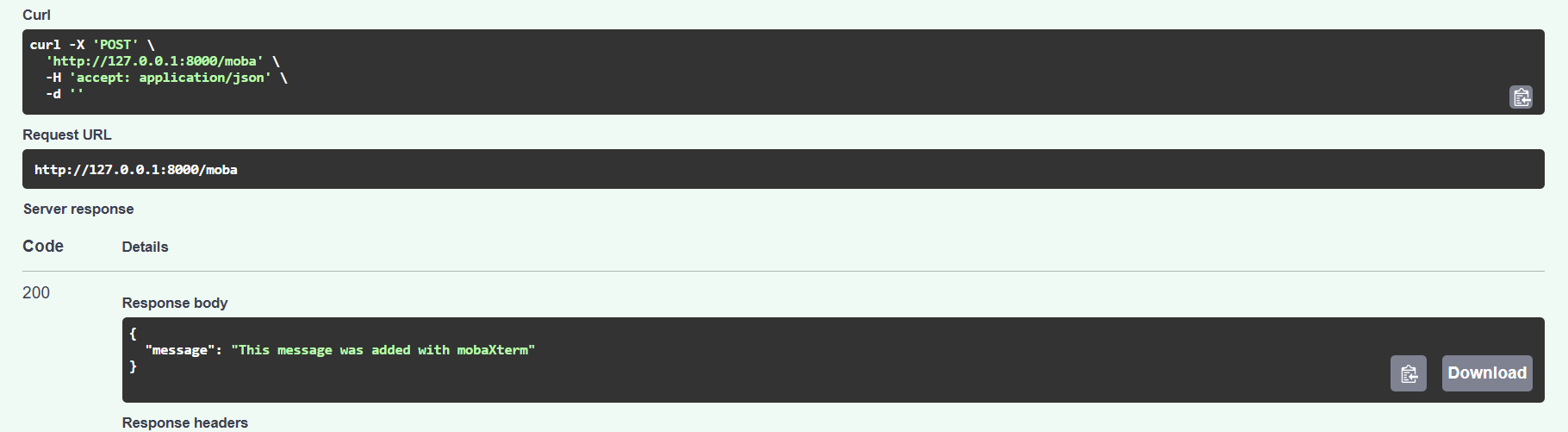
* Docker Images used



* Securevault running locally using docker



* Changes made been shown on the browser



**6. Results**

* Repository successfully cloned, updated, and pushed using MobaXterm.
* Application deployed locally with Docker.
* Verified functionality via browser.