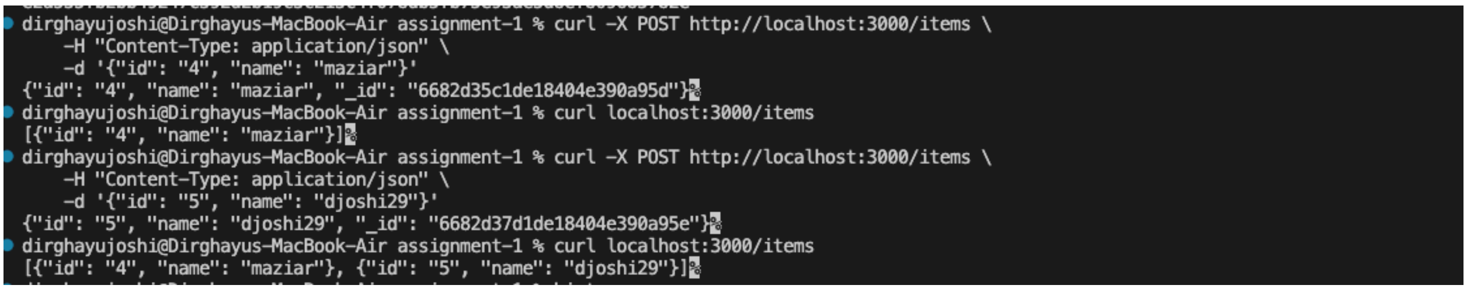
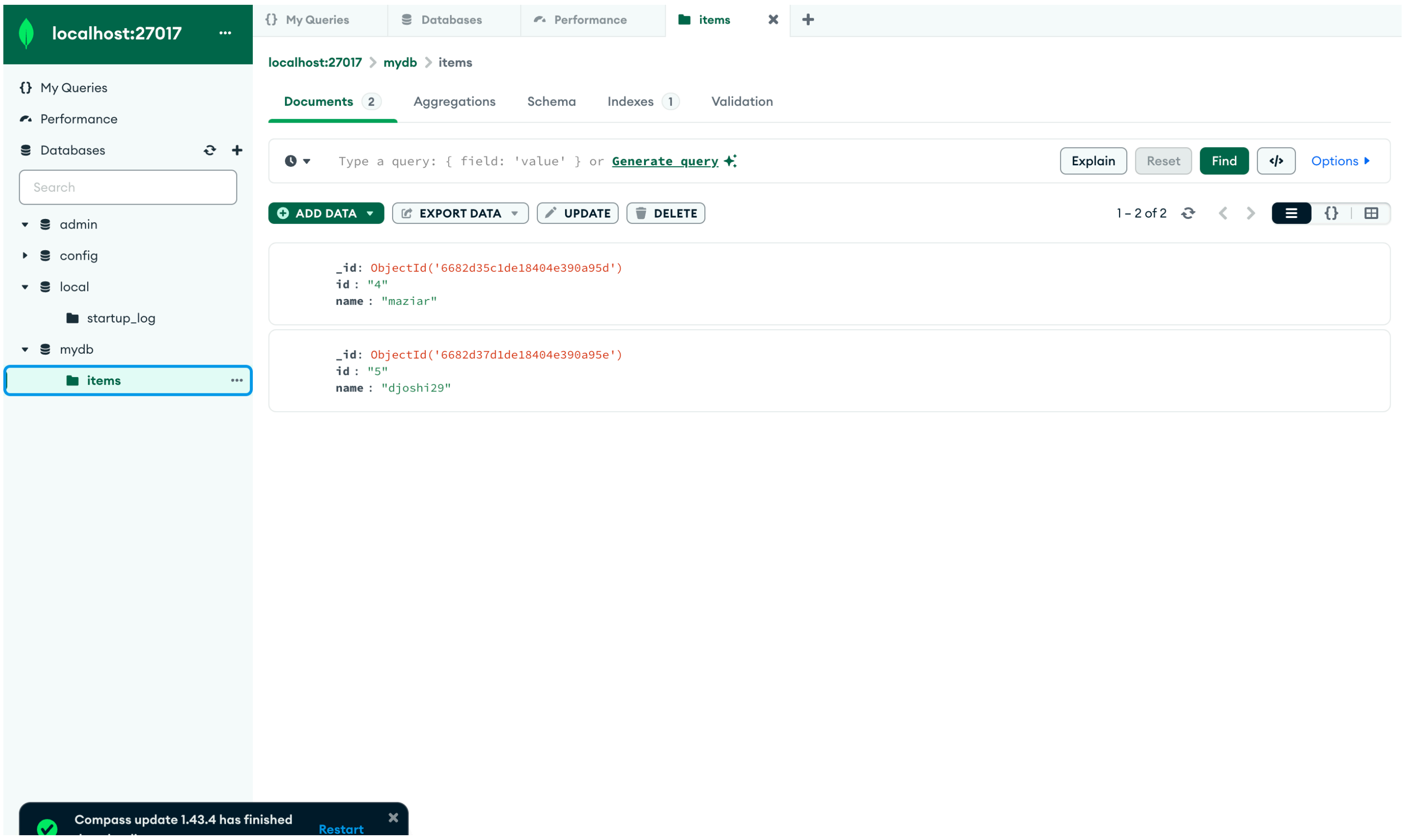
Source Links:

- **Github -** https://github.com/dirghayu101/clo835-assignment1
- **Docker -** https://hub.docker.com/r/dirghayu101/clo835-assignment-1

Screenshots:



```
dirghayujoshi@Dirghayus-MacBook-Air assignment-1 % curl -X POST http://localhost:3000/items \
    -H "Content-Type: application/json" \
    -d '{"id": "4", "name": "maziar"}'
{"id": "4", "name": "maziar", "_id": "6682d35c1de18404e390a95d"}%
dirghayujoshi@Dirghayus-MacBook-Air assignment-1 % curl localhost:3000/items
[{"id": "4", "name": "maziar"}]%
dirghayujoshi@Dirghayus-MacBook-Air assignment-1 % curl -X POST http://localhost:3000/items \
    -H "Content-Type: application/json" \
    -d '{"id": "5", "name": "djoshi29"}'
{"id": "5", "name": "djoshi29", "_id": "6682d37d1de18404e390a95e"}%
dirghayujoshi@Dirghayus-MacBook-Air assignment-1 % curl localhost:3000/items
[{"id": "4", "name": "maziar"}, {"id": "5", "name": "djoshi29"}]%
```



```
_id: ObjectId('6682d35c1de18404e390a95d')
id : "4"
name : "maziar"

_id: ObjectId('6682d37d1de18404e390a95e')
id : "5"
name : "djoshi29"
```

Writing Dockerfile for Python:

- bson dependency will be installed with pymongo (one of its dependency).
- Why did I use python:3.12-slim? It was the latest version which had a slim version, and this basic web server image should work with minimal requirements.
- Instead of multi-stage build, we could have directly created an alpine python image with dependencies and all of that installed, why did we do it in two stages?
- To understand this, you need to know about Build tools. Build tools are software and libraries required to compile and install certain dependencies, especially those that need to be built from source. These tools are often not needed at runtime and can significantly increase the size of the Docker image.
- If we use multi stage then there won't be any build tools in the final image.

Building Image:

- Considerations: networking with mongo, should communicate with mongodb at mongodb://mongodb:27017
- You can use docker-compose to achieve networking between mongodb and python web server container.
- A network device driver controls the networking hardware. This driver allows your operating system to interface with the network card, facilitating communication between your computer and networks, like the internet or a local network.
- URI is used to refer to any string that identifies a resource like mongodb://mongodb:27017, URL is used commonly with web addresses.

Why not docker-compose instead of bridge networking?

- docker-compose will increase the complexity in simple projects and decrease the complexity in complex project. This use case is the former.

Commands Used:

1. <u>Importing images:</u>
   ```
   docker pull python:3.12-slim
   ```

2. <u>Building image using dockerfile:</u>
   ```
   docker build -t clo835-assignment-1 .
   docker ps
   docker ps -a
   ```

3. <u>Working with Network:</u>
   ```
   docker network create -d bridge db-bridge
   ```

4. <u>Running images:</u>

docker run --name clo835-assignment-1 -d -p 127.0.0.1:3000:3000 --network db-bridge clo835-assignment-1
docker run --name clo835-assignment-1-mongo -d -p 127.0.0.1:27017:27017 --network db-bridge mongo:latest

5. <u>Others (very useful)</u>
   docker system prune -a   #-> Removes all stopped container, images and networks (not in use)
   docker logs clo835-assignment-1 # -> Logs generated, helped in debugging mongodb issue
   docker inspect clo835-assignment-1   # -> Metadata of the container specified.

6. <u>Others (common)</u>
   docker ps -a
   docker network ls
   docker rm <name || id>
   docker network prune
   git init && git add . && git commit -m "<something>" && git remote add origin <repoLink> && git push -u origin main

7. <u>Pushing image to docker hub:</u>
   docker tag <iamgeID> dirghayu101/<imageName>
   docker push dirghayu101/<imageName>:latest

Mistakes I made:

1. Custom named mongo container, use the name "mongodb", only this name will be resolved to the running container because it's hard coded in the source. -> Container in the same network use container names as hostnames for communicating.

<u>References:</u>

- https://docs.docker.com/build/building/multi-stage/ -> multistage official docs
- https://docs.docker.com/reference/dockerfile/#copy -> dockerfile reference
- https://docs.docker.com/reference/cli/docker/image/build/ -> docker build CLI reference
- https://stackoverflow.com/questions/45142528/what-is-a-dangling-image-and-what-is-an-unused-image -> Dangling image.
- https://chatgpt.com/share/f698c2eb-8d47-4060-8eae-9c8e8226526c