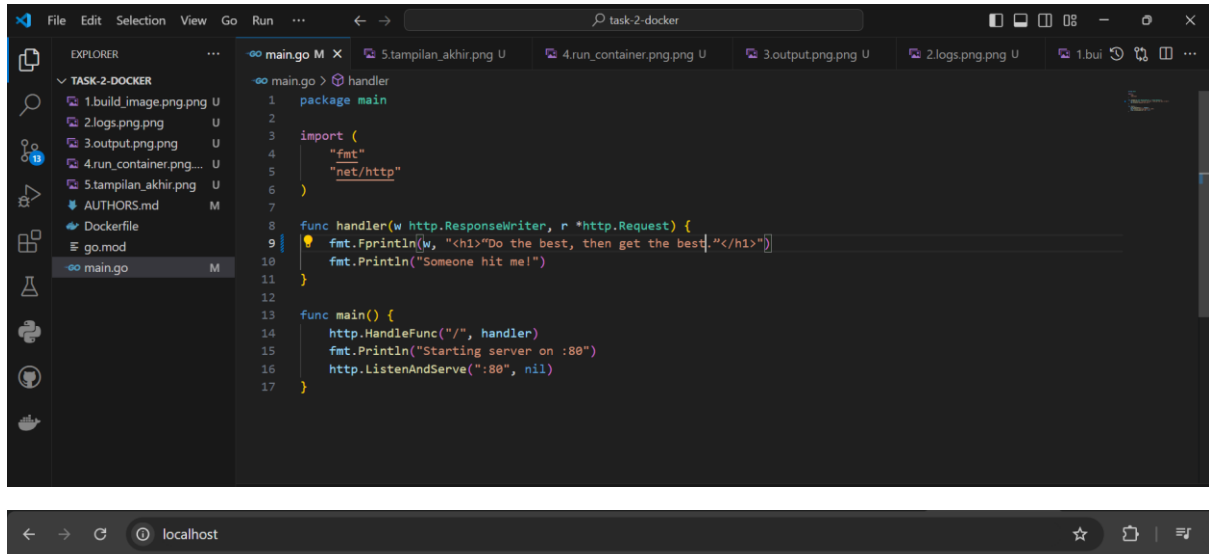
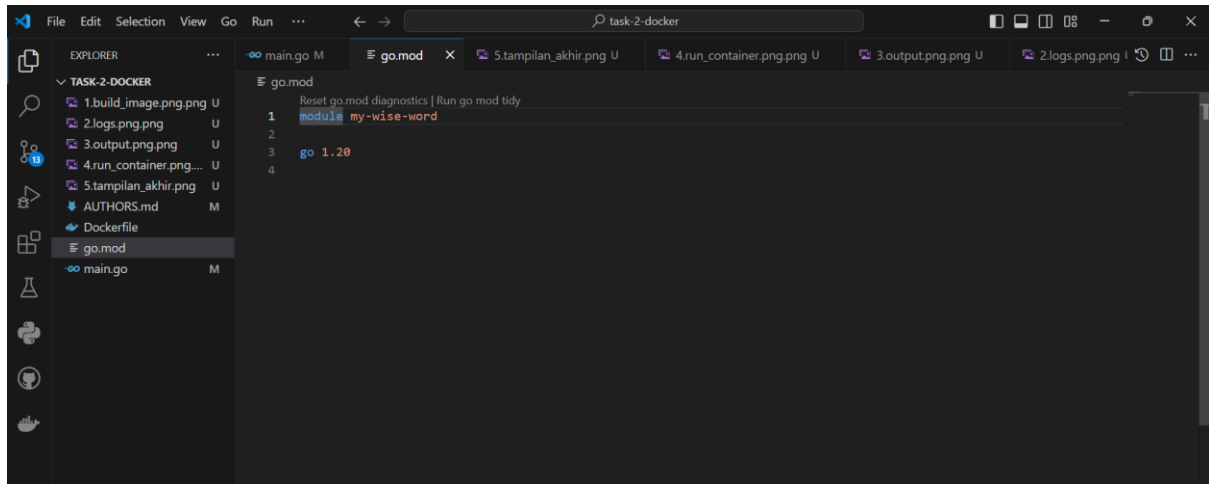


Task: Build Dockerfile to Docker Image

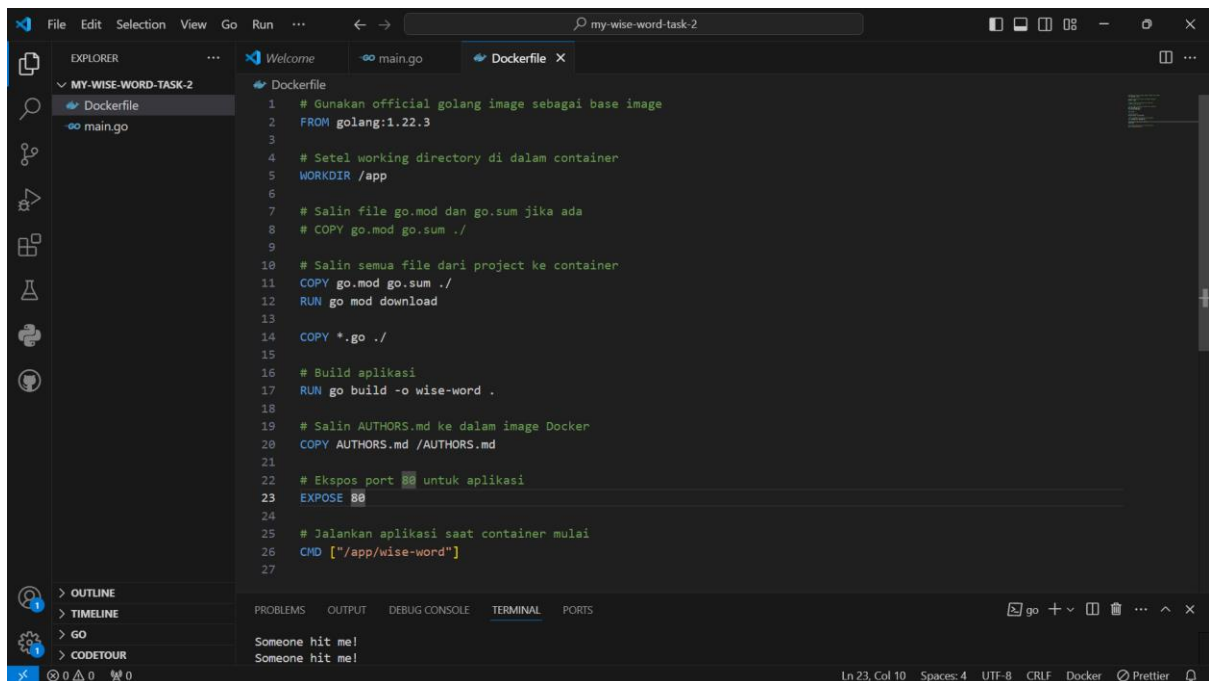
1. Developed a basic Golang project to handle HTTP requests.



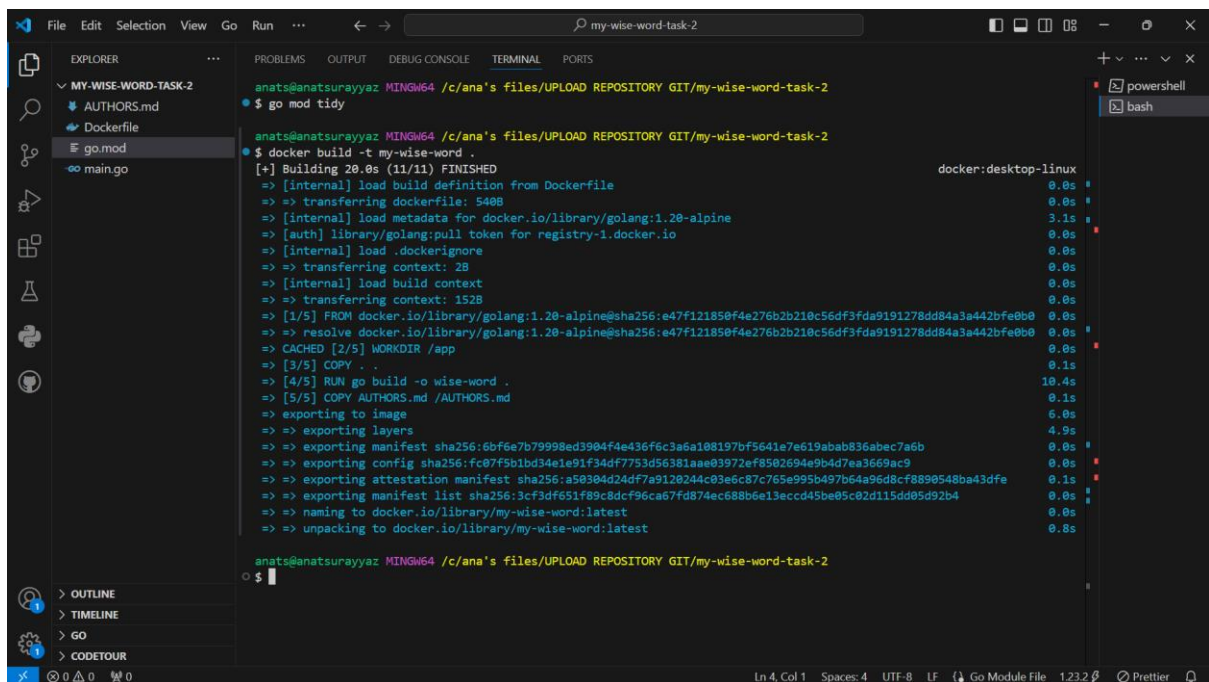
2. Introduced a new file, "AUTHORS.md," to the Golang project, listing my GitHub username, "@dilanisa26."



3. Created a Dockerfile to build and run the Golang project.



4. Built the Docker image using the following command: `docker build -t my-wise-word .`



```
anats@anatsurayyaz MINGW64 /c/ana's files/UPLOAD REPOSITORY GIT/my-wise-word-task-2
$ go mod tidy

anats@anatsurayyaz MINGW64 /c/ana's files/UPLOAD REPOSITORY GIT/my-wise-word-task-2
$ docker build -t my-wise-word .
[+] Building 20.0s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 540B
=> [internal] load metadata for docker.io/library/golang:1.20-alpine
=> [auth] library/golang:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 152B
=> [1/5] FROM docker.io/library/golang:1.20-alpine@sha256:e47f121850f4e276b2b210c56df3fda9191278dd84a3a442bfe0b0
=> => resolve docker.io/library/golang:1.20-alpine@sha256:e47f121850f4e276b2b210c56df3fda9191278dd84a3a442bfe0b0
=> CACHED [2/5] WORKDIR /app
=> [3/5] COPY . .
=> [4/5] RUN go build -o wise-word .
=> [5/5] COPY AUTHORS.md /AUTHORS.md
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:6bfe7b79998ed3904f4e436f6c3a5a188197bf5641e7e619abab836abec7a6b
=> => exporting config sha256:fc07f5b1bd34e1e91f34df7753d56381aae03972ef8502694e9b4d7ea3669ac9
=> => exporting attestation manifest sha256:a58304d24df7a9120244c03e6c87c765e995b497b64a96d8cf8890548ba43dfe
=> => exporting manifest list sha256:3cf3df651f89c8dcf96ca67fd874ec688b6e1eccc45be05c02d115dd05d92b4
=> => naming to docker.io/library/my-wise-word:latest
=> => unpacking to docker.io/library/my-wise-word:latest

anats@anatsurayyaz MINGW64 /c/ana's files/UPLOAD REPOSITORY GIT/my-wise-word-task-2
$
```

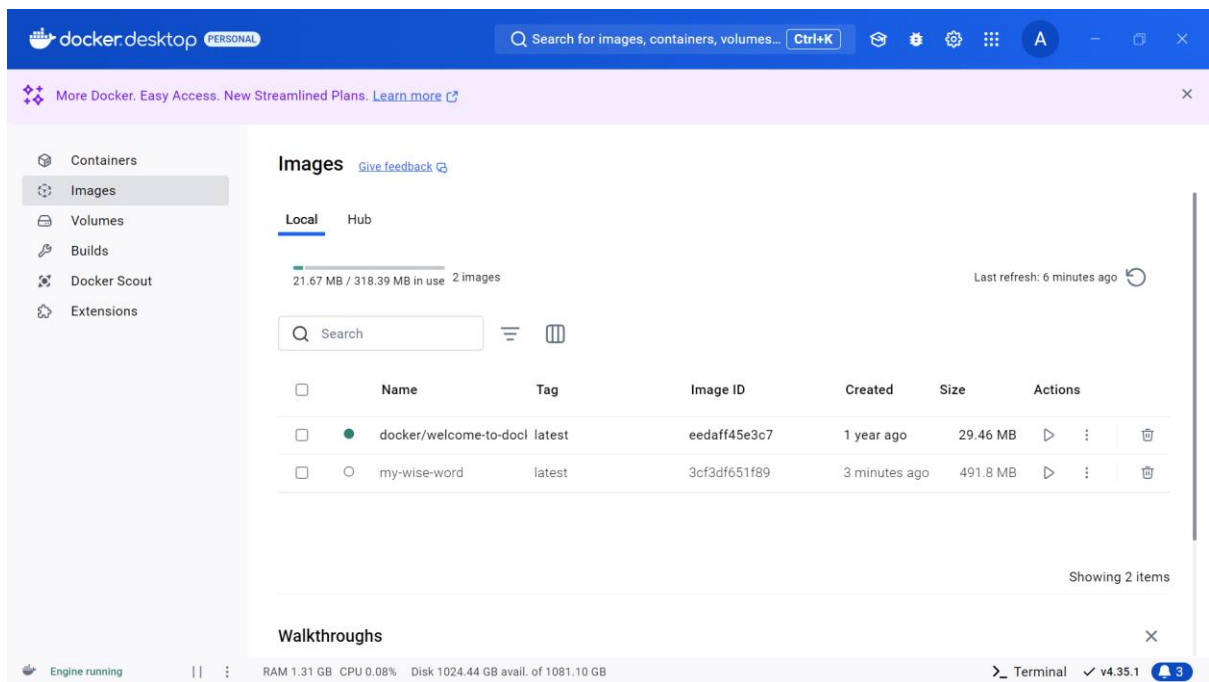
1. Verified the image was successfully added by listing Docker images: `docker image ls`



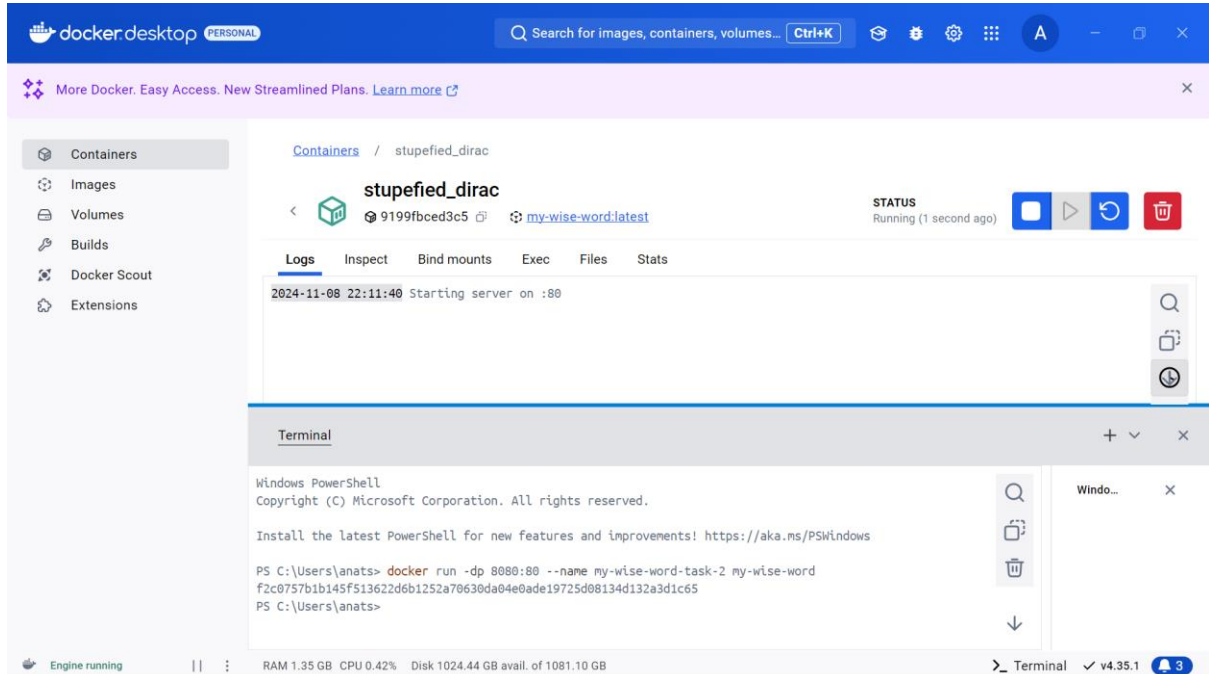
```
anats@anatsurayyaz MINGW64 /c/ana's files/UPLOAD REPOSITORY GIT/my-wise-word-task-2
$ docker image ls
REPOSITORY          TAG       IMAGE ID       CREATED        SIZE
my-wise-word        latest    3cf3df651f89   About a minute ago   492MB
docker/welcome-to-docker latest    eedaff45e3c7   12 months ago      29.5MB

anats@anatsurayyaz MINGW64 /c/ana's files/UPLOAD REPOSITORY GIT/my-wise-word-task-2
$
```

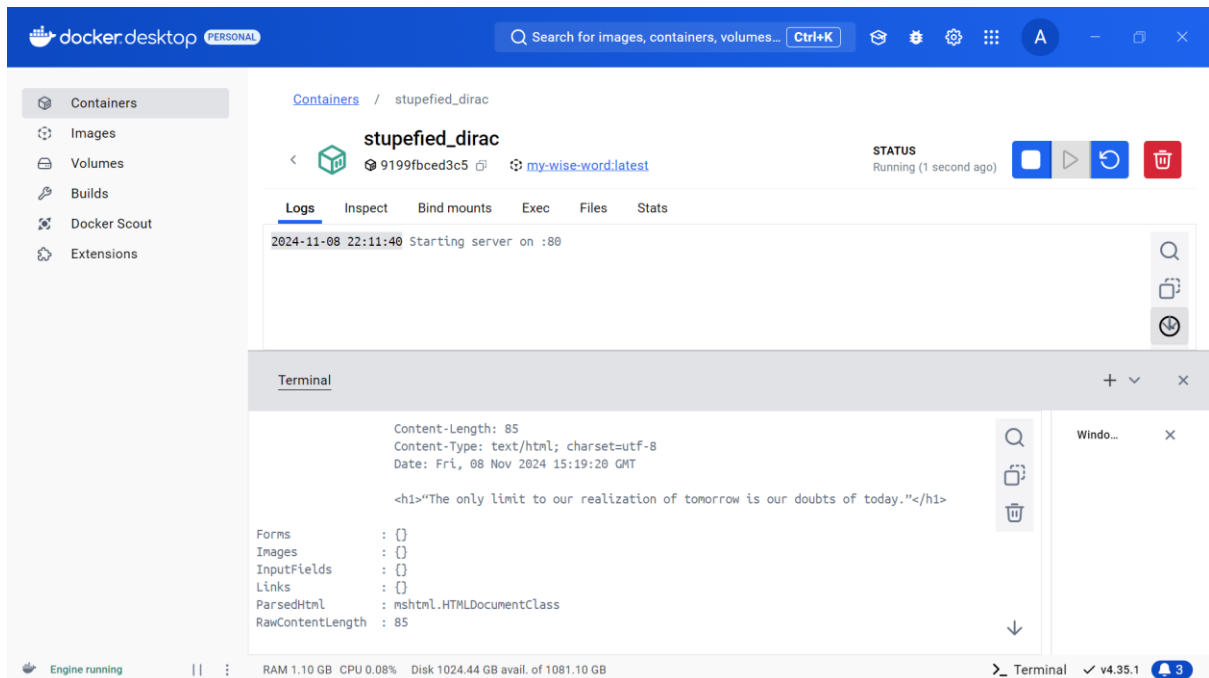
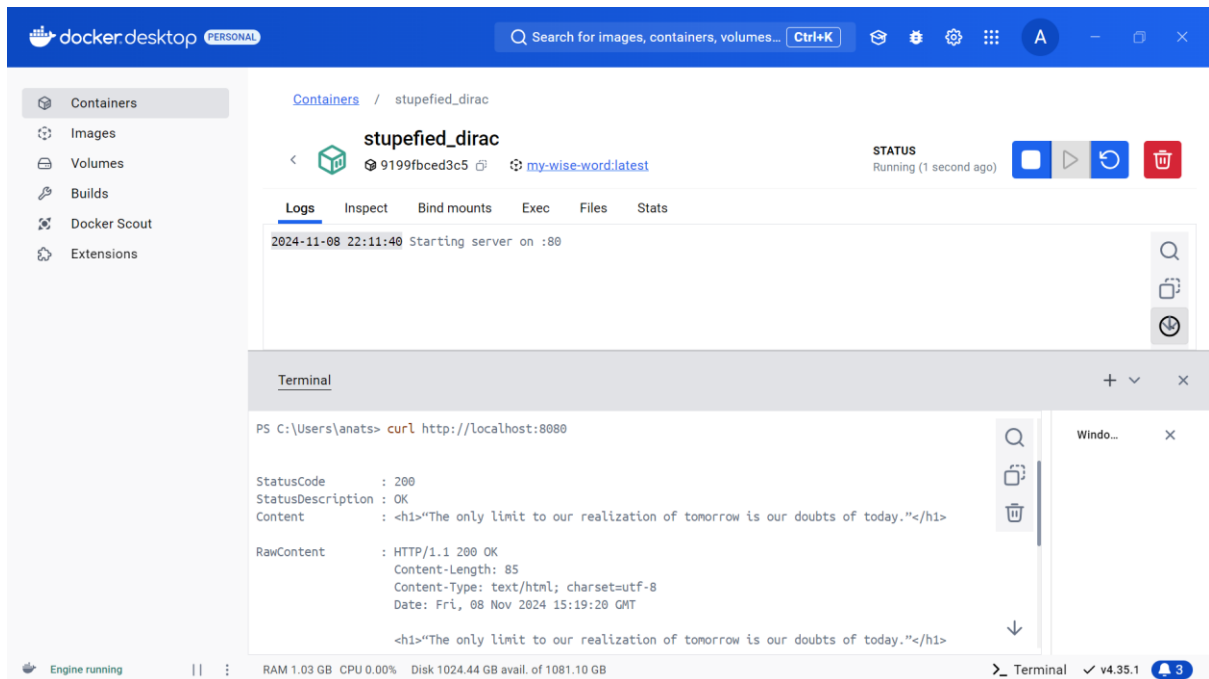
The image successfully added



2. Ran a new container with the name “my-wise-word-task-2” using the command:
`docker run -dp 8080:80 --name my-wise-word-task-2 my-wise-word`



- Exposed the application on host port 8080 and verified it with: curl http://localhost:8080



4. Displayed the logs of the "my-wise-word-task-2" container to check for activity and confirm the application is running as expected.

Terminal

```
PS C:\Users\anats> docker logs my-wise-word-task-2
Starting server on :80
Someone hit me!
```

