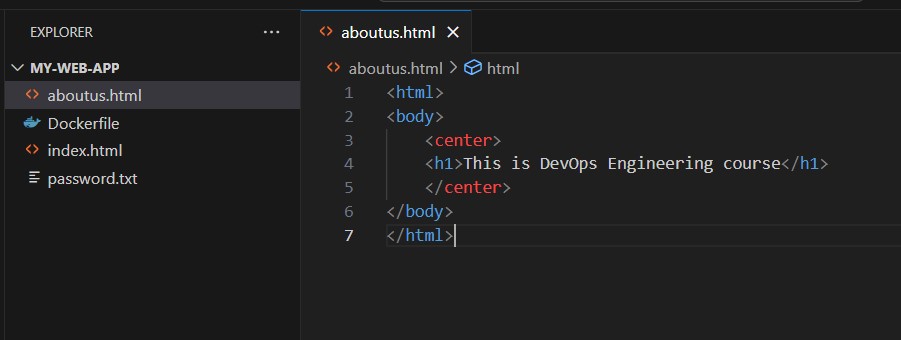
# EXPERIMENT-2

**AIM : .dockerignore in Docker**

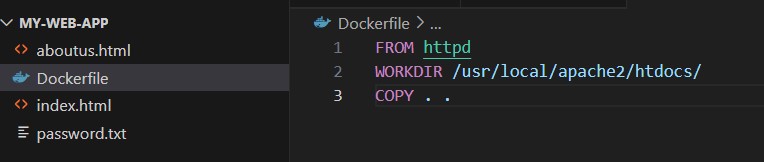
**Create a Sample Project:**

* Create a simple project directory with various types of files, including source code, configuration files, and build artifacts.



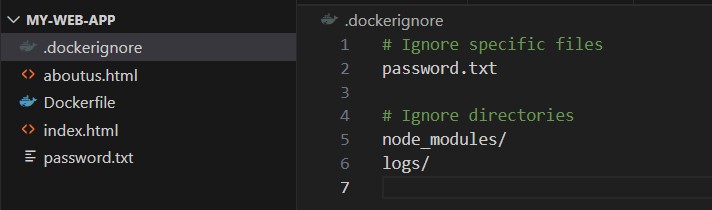
**Create a Dockerfile:**

* Write a Dockerfile that specifies the build steps for your application, including any necessary dependencies.



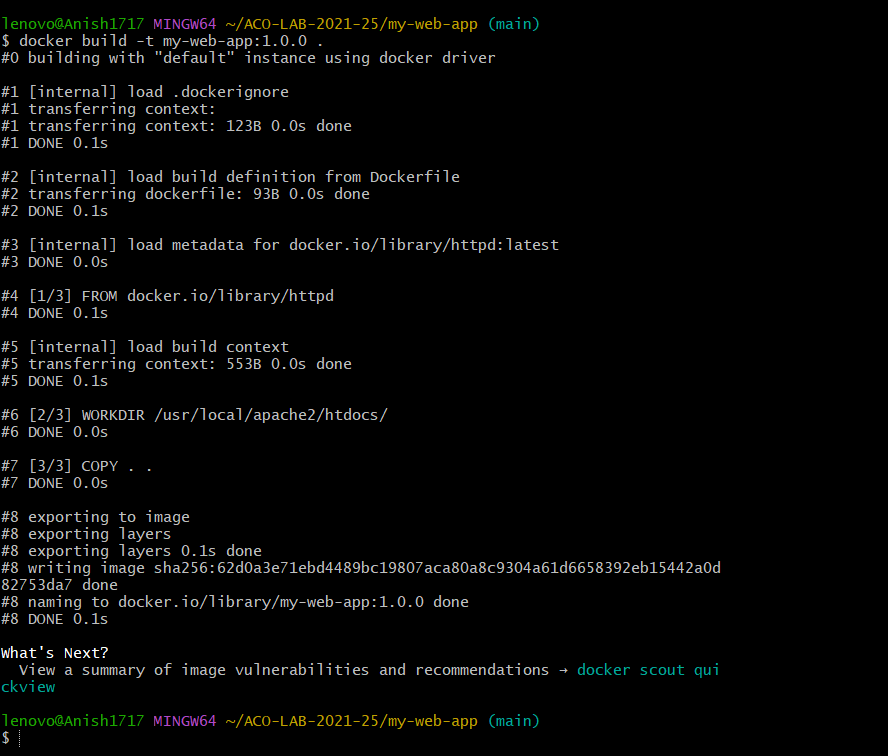
**Create a .dockerignore File:**

* Create a .dockerignore file in the root directory of your project.
* Add specific files or directories to the .dockerignore file that you do not want to be included in the Docker image during the build process.



**Build the Docker Image:**

* Use the docker build command to build the Docker image from the Dockerfile.



* Observe the build context and check which files are being included in the Docker image.

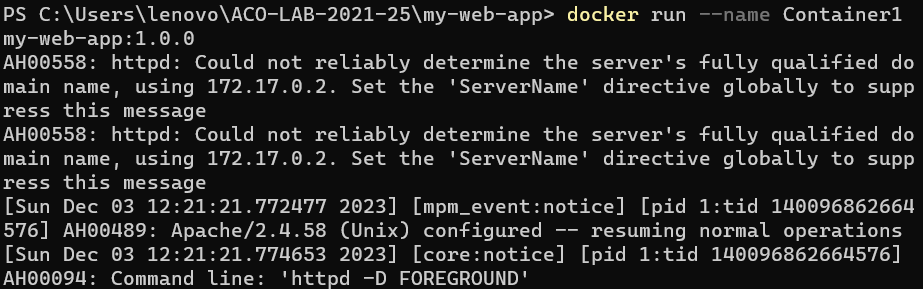


**Modify .dockerignore and Rebuild:**

* Modify the .dockerignore file to exclude different types of files or directories from the Docker image.
* Rebuild the Docker image and observe the changes in the build context.

**Test the Docker Image:**

* Run the Docker image as a container using the docker run command.
* Verify that the container works as expected, despite certain files being excluded from the Docker image.

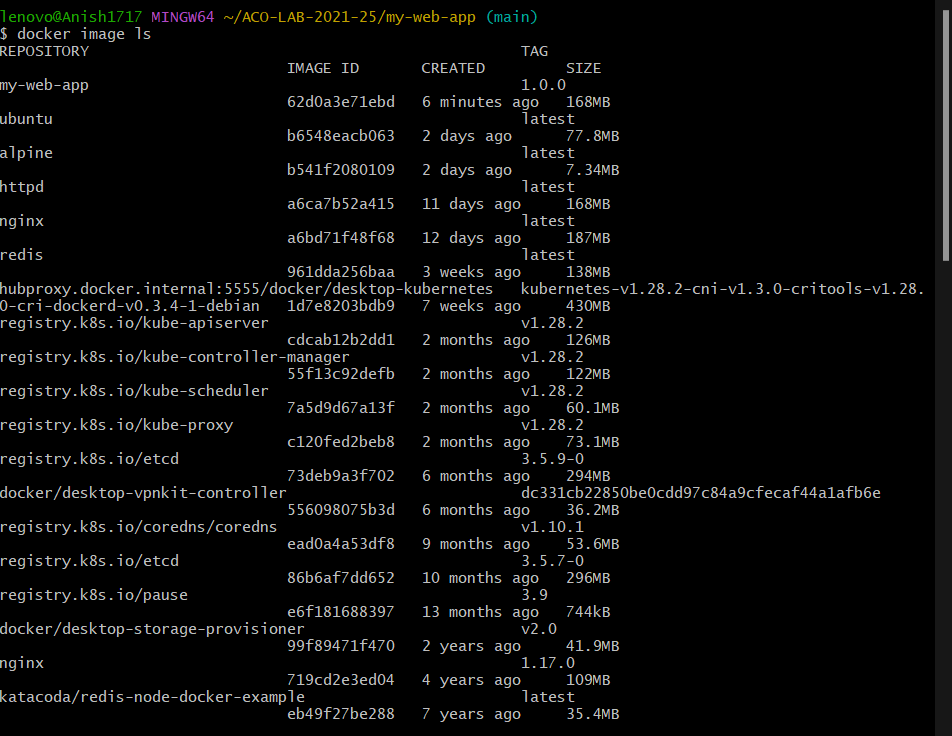


**Optimize the .dockerignore File:**

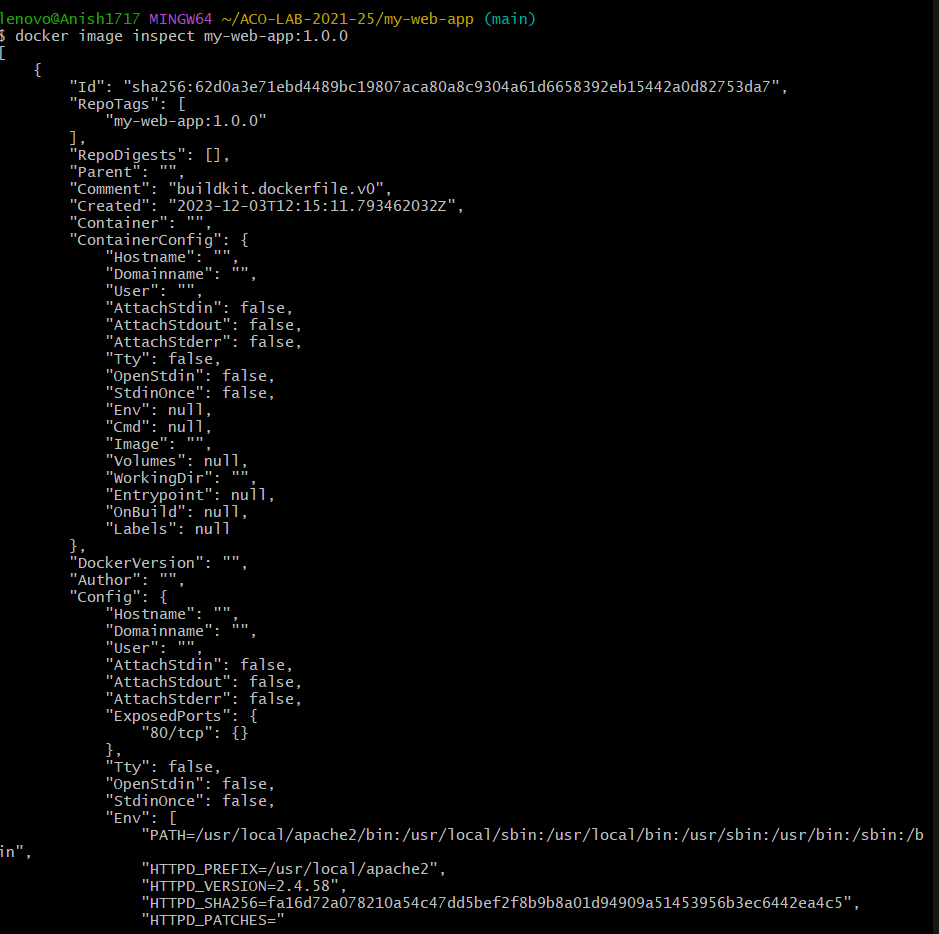
* Experiment with different patterns and wildcards in the .dockerignore file to optimize the build context and reduce the image size.

**Check Image Size and Contents:**

* Use the docker image ls and docker image inspect commands to check the size of the Docker image and its contents.

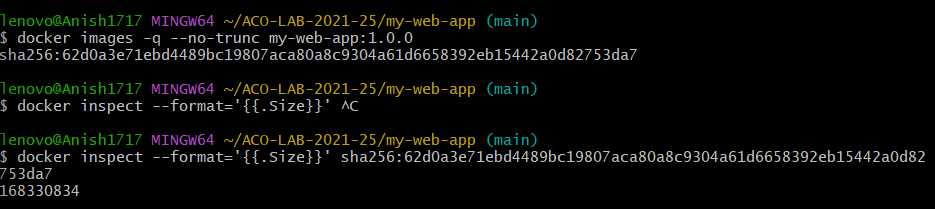








* Command that list the hash id of the image being created and then list the size of the image using its hash id :



**Documentation and Best Practices:**

* Document your findings and the best practices for using the .dockerignore file in your projects.

Through this exercise, you'll gain a better understanding of how to use the

.dockerignore file effectively to optimize Docker image builds and reduce image sizes. Adjust the exercise based on your specific project requirements.