1. **Create your HTML Application:**

Create a directory for your HTML application.

Inside this directory, create an index.html file and any other assets (CSS, JavaScript, images) your application requires.

For example, create a directory named html\_app and inside it, create index.html with some content:

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Simple HTML App</title>

</head>

<body>

<h1>Hello, Dockerized HTML App!</h1>

</body>

</html>

2. **Create a Dockerfile:**

In the same directory as your HTML files, create a file named Dockerfile. This file will contain instructions for building your Docker image.

Open Dockerfile and add the following content:

**Dockerfile**(use Notepad++ and select "All files" under save as type)

# Use a base image with a web server

FROM nginx:alpine

# Copy the HTML files into the web server directory

COPY . /usr/share/nginx/html

This Dockerfile instructs Docker to use the nginx:alpine image as a base and copy the contents of the current directory into the web server directory (/usr/share/nginx/html).

3. **Build the Docker Image:**

Open a terminal (PowerShell as Administrator) and navigate to the directory containing your Dockerfile and HTML files.

Run the following command to build your Docker image:

**docker build -t my-html-app .**

This command will build a Docker image named my-html-app using the Dockerfile in the current directory (.).

4. **Run the Docker Container:**

After the image is built successfully, you can run a container based on that image using the following command:

**docker run -d -p 8080:80 my-html-app**

This command will start a container in detached mode (-d), map port 8080 on your host machine to port 80 in the container (-p 8080:80), and use the my-html-app image.

5. **Access your HTML Application:**

Open a web browser and navigate to **http://localhost:8080** to see your HTML application running inside the Docker container.

you have containerized your simple HTML application using Docker on Windows. You can stop the container by running **docker stop <container\_id>** and remove it with **docker rm <container\_id>,** where <container\_id> is the ID of your container obtained from docker ps command.