ADVANCED PYTHON PROGRAMMING

NAME:E.AGNES RACHEL

REG.NO:22MID0181

DATE:06/11/2025

DOCKER

Prerequisites:

- Install docker desktop from https://www.docker.com/products/docker-desktop/

To verify docker installation:

Open command prompt ->docker –version

Step 1: Create Your Project Folder

1. Create a new folder on your desktop called my-docker-website
2. Inside this folder, create another folder called html

```
Desktop/
└── my-docker-website/
    ├── Dockerfile
    └── html/
        ├── index.html
        └── styles.css
```

It should be like this

Inside html folder create 3 new text documents and name as index.html,styles.css,Dockerfile.

Index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>My Simple Docker Website</title>
   <link rel="stylesheet" href="styles.css">
</head>
<body>
   <header>
      <h1>My Awesome Site</h1>
   </header>
   <main>
      <section class="hero">
         <h2>Welcome Aboard! 🚀 </h2>
         <p>This is a simple website deployed using Docker and Nginx.</p>
         <a href="#about" class="button">Learn More</a>
```

```
    </section>
    <section id="about" class="content-box">
      <h3>About This Project</h3>
      <p>Dockerizing static content makes deployment clean and repeatable. No matter where you run this container, the website looks and functions exactly the same.</p>
    </section>
  </main>
  <footer>
    <p>&copy; 2025 Simple Web Deployment</p>
  </footer>
</body>
</html>
```

Styles.css

```
/* Basic Reset */
body {
  font-family: 'Arial', sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f4f4f9;
  color: #333;
  line-height: 1.6;
}

/* Header Styling */
header {
    background-color: #007bff;
    color: white;
    padding: 20px 0;
    text-align: center;
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}

header h1 {
    margin: 0;
    font-size: 2.5em;
}

/* Main Content Area */
main {
    max-width: 900px;
    margin: 40px auto;
    padding: 0 20px;
}
```

```css
/* Hero Section */
.hero {
   background-color: #ffffff;
   padding: 40px;
   border-radius: 8px;
   text-align: center;
   margin-bottom: 40px;
   border-left: 5px solid #007bff;
   box-shadow: 0 4px 8px rgba(0, 0, 0, 0.05);
}

.hero h2 {
   color: #007bff;
   font-size: 2em;
   margin-top: 0;
}

/* Button Styling */
.button {
   display: inline-block;
   background-color: #28a745;
```

```
    color: white;
    padding: 10px 20px;
    text-decoration: none;
    border-radius: 5px;
    margin-top: 15px;
    transition: background-color 0.3s ease;
}

.button:hover {
    background-color: #218838;
}

/* Content Box Section */
.content-box {
    background-color: #fff;
    padding: 30px;
    border-radius: 8px;
    margin-top: 20px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.05);
}

.content-box h3 {
    border-bottom: 2px solid #ccc;
    padding-bottom: 10px;
    color: #555;
}

/* Footer Styling */
footer {
    text-align: center;
    padding: 20px 0;
    background-color: #333;
    color: white;
    margin-top: 40px;
}
```

Dockerfile:

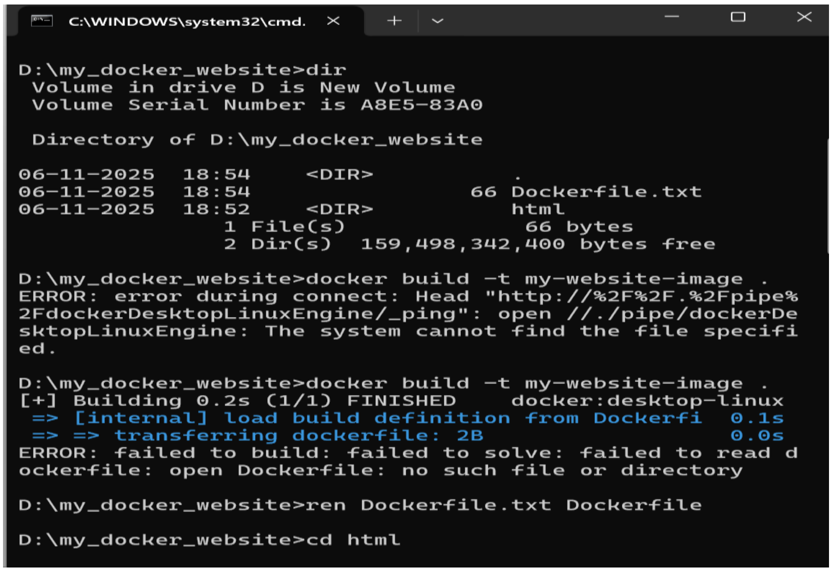
```
FROM nginx:alpine

COPY html/ /usr/share/nginx/html

EXPOSE 80
```

Step 2: Open Command Line and Build Your Docker Image

1. Open Command Prompt/Terminal

2. Navigate to your project folder:

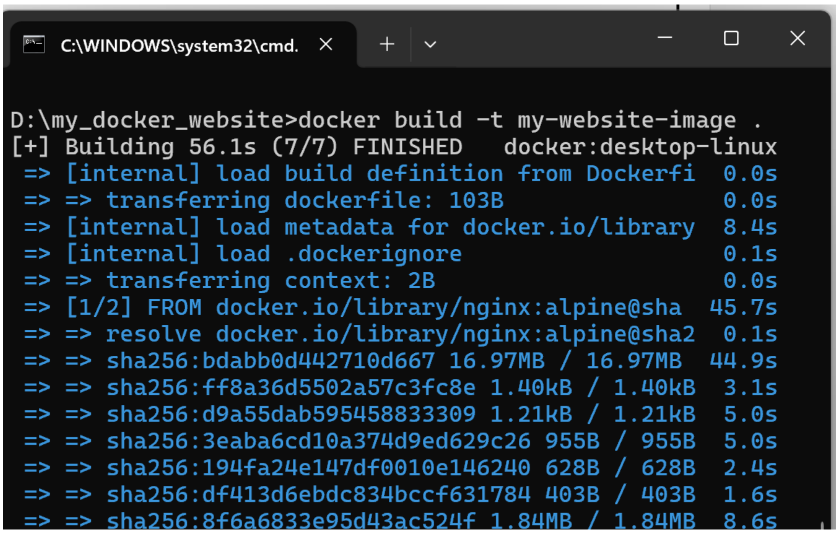
cd Desktop/my-docker-website



```
D:\my_docker_website>dir
 Volume in drive D is New Volume
 Volume Serial Number is A8E5-83A0

 Directory of D:\my_docker_website

06-11-2025  18:54    <DIR>          .
06-11-2025  18:54                66 Dockerfile.txt
06-11-2025  18:52    <DIR>          html
               1 File(s)             66 bytes
               2 Dir(s)  159,498,342,400 bytes free

D:\my_docker_website>docker build -t my-website-image .
ERROR: error during connect: Head "http://%2F%2F.%2Fpipe%
2FdockerDesktopLinuxEngine/_ping": open //./pipe/dockerDe
sktopLinuxEngine: The system cannot find the file specifi
ed.

D:\my_docker_website>docker build -t my-website-image .
[+] Building 0.2s (1/1) FINISHED    docker:desktop-linux
 => [internal] load build definition from Dockerfi  0.1s
 => => transferring dockerfile: 2B                  0.0s
ERROR: failed to build: failed to solve: failed to read d
ockerfile: open Dockerfile: no such file or directory

D:\my_docker_website>ren Dockerfile.txt Dockerfile

D:\my_docker_website>cd html
```

Step 3: Build Your Docker Image

docker build -t my-website-image .

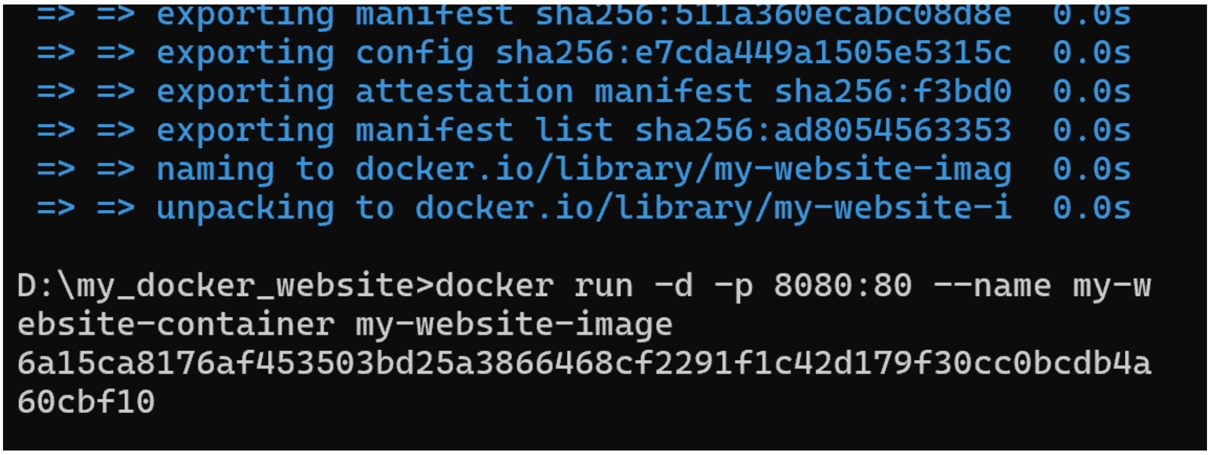


```
D:\my_docker_website>docker build -t my-website-image .
[+] Building 56.1s (7/7) FINISHED    docker:desktop-linux
 => [internal] load build definition from Dockerfi  0.0s
 => => transferring dockerfile: 103B                0.0s
 => [internal] load metadata for docker.io/library  8.4s
 => [internal] load .dockerignore                   0.1s
 => => transferring context: 2B                     0.0s
 => [1/2] FROM docker.io/library/nginx:alpine@sha  45.7s
 => => resolve docker.io/library/nginx:alpine@sha2  0.1s
 => => sha256:bdabb0d442710d667 16.97MB / 16.97MB  44.9s
 => => sha256:ff8a36d5502a57c3fc8e 1.40kB / 1.40kB  3.1s
 => => sha256:d9a55dab595458833309 1.21kB / 1.21kB  5.0s
 => => sha256:3eaba6cd10a374d9ed629c26 955B / 955B  5.0s
 => => sha256:194fa24e147df0010e146240 628B / 628B  2.4s
 => => sha256:df413d6ebdc834bccf631784 403B / 403B  1.6s
 => => sha256:8f6a6833e95d43ac524f 1.84MB / 1.84MB  8.6s
```

Step 4: Run Your Container

After the build completes, run:

docker run -d -p 8080:80 --name my-website-container my-website-image

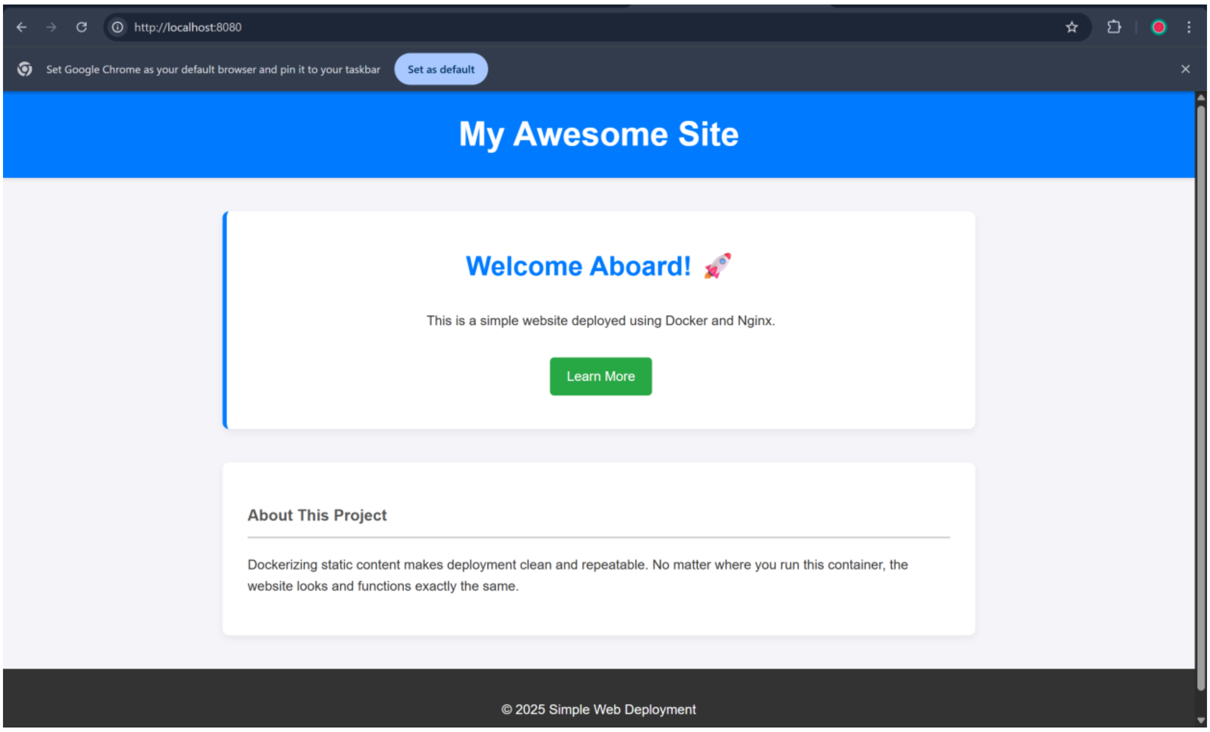


```
 => => exporting manifest sha256:511a360ecabc08d8e  0.0s
 => => exporting config sha256:e7cda449a1505e5315c  0.0s
 => => exporting attestation manifest sha256:f3bd0  0.0s
 => => exporting manifest list sha256:ad8054563353  0.0s
 => => naming to docker.io/library/my-website-imag  0.0s
 => => unpacking to docker.io/library/my-website-i  0.0s

D:\my_docker_website>docker run -d -p 8080:80 --name my-w
ebsite-container my-website-image
6a15ca8176af453503bd25a3866468cf2291f1c42d179f30cc0bcdb4a
60cbf10
```

Step 5: Test Your Website

Open your web browser and go to:

http://localhost:8080



http://localhost:8080
Set Google Chrome as your default browser and pin it to your taskbar
Set as default

My Awesome Site

Welcome Aboard! 🚀

This is a simple website deployed using Docker and Nginx.

Learn More

About This Project

Dockerizing static content makes deployment clean and repeatable. No matter where you run this container, the website looks and functions exactly the same.

© 2025 Simple Web Deployment