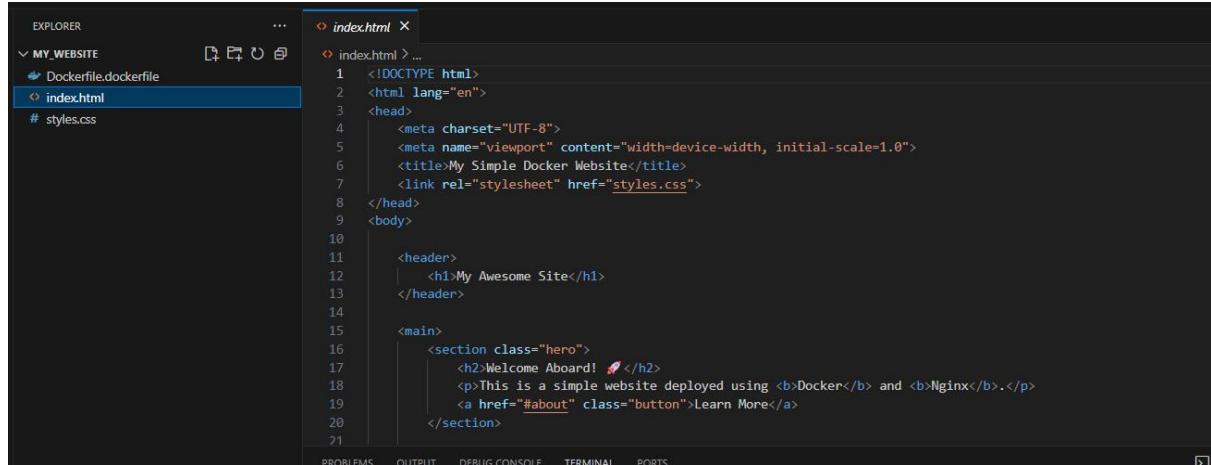


Name:S.Jewel Reddy

Reg No:22MID0161

Docker

Index.html



The screenshot shows a code editor interface with the following details:

- EXPLORER** sidebar: MY_WEBSITE folder containing Dockerfile.dockerfile and index.html.
- index.html** tab is active in the main editor area.
- Content of 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 <b>Docker</b> and <b>Nginx</b>. <a href="#about" class="button">Learn More</a></p>
        </section>
    </main>
</body>

```

- Bottom navigation bar:** PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS.

Styles.css

```
EXPLORER          ...
MY WEBSITE
  Dockerfile.dockerfile
  index.html
# styles.css

# styles.css > body
body {
    font-family: 'Arial', sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f4f4f9;
    color: #333;
    line-height: 1.6;
}

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 {
    max-width: 900px;
    margin: 40px auto;
    padding: 0 20px;
}

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

> OUTLINE
> TIMELINE
⊗ 0 △ 0
```

Dockerfile

```
# 1. Use a lightweight Nginx image
FROM nginx:alpine

# 2. Copy website files into the Nginx web root
COPY html/ /usr/share/nginx/html

# 3. Expose port 80
EXPOSE 80
```

Commands:

docker login

docker build -t my-website-image .

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

docker tag my-website-image your-username/my-website-image:latest

docker push your-username/my-website-image:latest

<http://localhost:8080>

```
Run 'docker run --help' for more information
PS C:\Users\sadaliOneDrive\Documents\my_website> docker build -t my-website-image .
>> docker run -d -p 8080:80 --name my-website-container my-website-image
>> docker build -t my-website-image .
>>
[+] Building 21.9s (7/7) FINISHED
-> [internal] load build definition from Dockerfile
-> => transferring dockerfile: 213B
-> [internal] load metadata for docker.io/library/nginx:alpine
-> [internal] load .dockerignore
-> => transferring context: 2B
-> [internal] load build context
-> => transferring context: 2.60kB
=> [1/2] FROM docker.io/library/nginx:alpine@sha256:b3c656d55d7ad751196f21b7fd2e8d4da9cb430e32f646adc92441b72f82b14
=> => resolve docker.io/library/nginx:alpine@sha256:b3c656d55d7ad751196f21b7fd2e8d4da9cb430e32f646adc92441b72f82b14
=> => sha256:df413d6ebdc834bccf63178455d406c4d25e2c2d38d21ab79ee5494b18e5624 0B / 403B
=> => sha256:3eaba6cd10a374d9ed629c26d76a5258e20ddfa09fccef511c98aa620dcf3fae4 955B / 955B
```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\sadal\OneDrive\Documents\my_website> docker run -d -p 8080:80 my-website-image
--> exporting config sha256:e1a02dc5ca5386f5b0c2cd1240736fdb0e59ae0e8ad4e04c4f754abed5c3061a
--> exporting attestation manifest sha256:0015917c39dd0e65f179d991cdbf5e1abb6aae38929c97f8a80925a23c79c72
--> exporting manifest list sha256:d4d85555f6c2acd9757a77f0c54219d50485433989e5168c845c1466aa26da0
--> naming to docker.io/library/my-website-image:latest
--> unpacking to docker.io/library/my-website-image:latest
PS C:\Users\sadal\OneDrive\Documents\my_website> docker run -d -p 8080:80 my-website-image
-->
5f33437b1e9ad2bc0b5ba45c45f394f79d398935ce51190243b130c296aaa2
docker: Error response from daemon: ports are not available: exposing port TCP 0.0.0.0:8080 -> 127.0.0.1:0: listen tcp 0.0.0.0:8080: bind: Only one usage of each socket address (protocol/net work address/port) is normally permitted.

Run 'docker run --help' for more information    docker run -d -p 8080:80 my-website-image
--> C:\Users\sadal\OneDrive\Documents\my_website> docker run -d -p 8081:80 my-website-image
fae8be31f23c7362429330b873c14dd5e20a5bcba074e54cb2ed57d620
docker: Error response from daemon: ports are not available: exposing port TCP 0.0.0.0:8080 -> 127.0.0.1:0: listen tcp 0.0.0.0:8080: bind: Only one usage of each socket address (protocol/net work address/port) is normally permitted.

Run 'docker run --help' for more information
PS C:\Users\sadal\OneDrive\Documents\my_website> docker run -d -p 8081:80 my-website-image
-->
e68264b7eab47dffccf84chc1c182be5b0a7b21b652c39c066732eb2017a9007
PS C:\Users\sadal\OneDrive\Documents\my_website> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e68264b7eab my-website-image "/docker-entrypoint..." About a minute ago Up About a minute 0.0.0.0:8081->80/tcp, [::]:8081->80/tcp quizzical_visvesvaraya
PS C:\Users\sadal\OneDrive\Documents\my_website>

```

Ln 8, Col 1 Spaces: 4 UTF-8 CRLF Dockerfile Signed out ENG 11:32

Containers Give feedback

Container CPU usage: 0.00% / 1200% (12 CPUs available)

Container memory usage: 10.17MB / 3.58GB

	Name	Container ID	Image	Port(s)	CPU (%)	Actions
<input type="checkbox"/>	musing_elion	5f33437b1e9a	my-website	8080:80	0%	
<input type="checkbox"/>	my-website-con	8326f08d13ae	docker-deb	8082:80	0% 1	

Walkthroughs

- Multi-container applications** 8 mins
- Containerize your application** 3 mins

[View more in the Learning center](#)

localhost:8081

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.