DEVOPS PROJECT

Deploying Web App Using Docker & GitHub

Overview:

This project demonstrates how to containerize and deploy a simple Typing Speed Test web application using Docker and GitHub. The steps below detail the entire process, from setting up the application to deploying it via Docker.

Prerequisites

Git Installed: Check with `git –version`

Docker Installed: Verify with `docker --version`

GitHub Account: Ensure you have a GitHub account and know how to create repositories.

Text Editor/IDE: Preferably VS Code or similar.

Step 1: Create the Web App

1. Set Up the Project

Open VS Code or any text editor.

Create a folder named TypingSpeedApp

Inside the folder**, create the following three files**:

- `index.html`
- `style.css`
- `script.js`

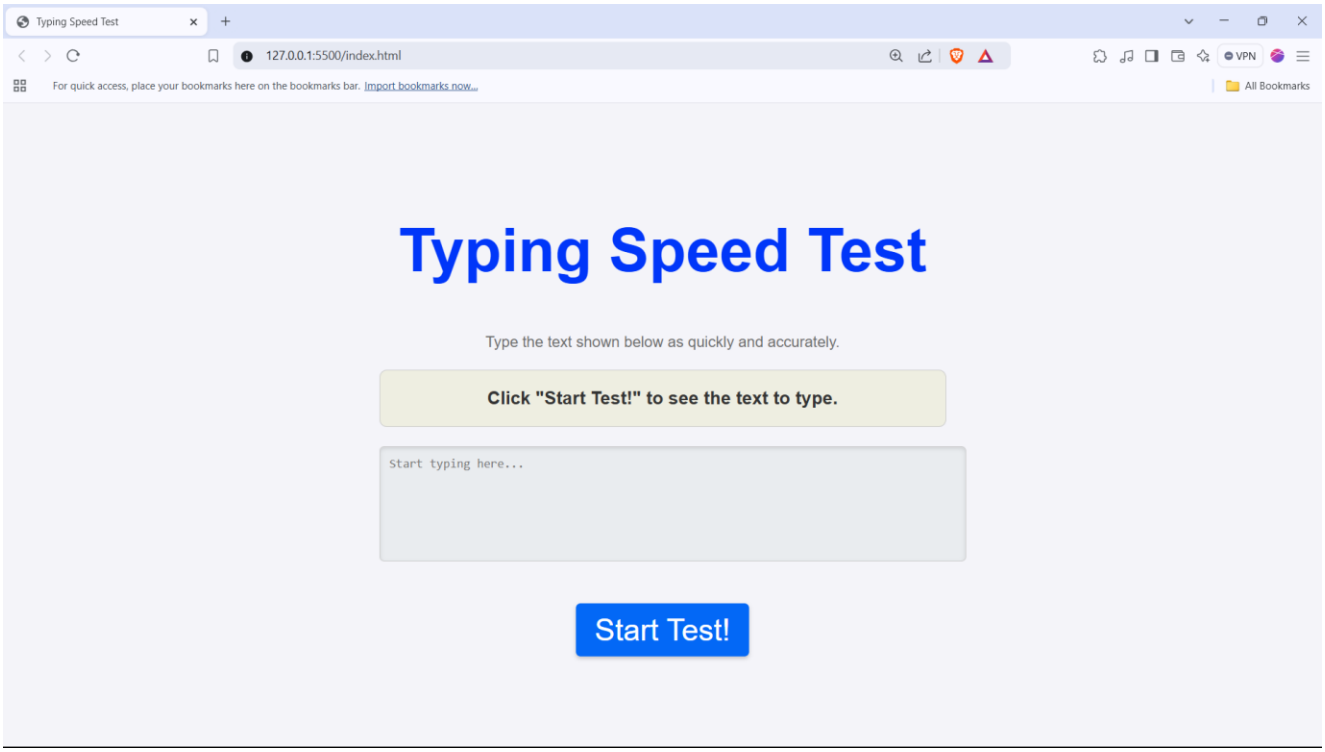
Copy the respective content into each file from my GitHub Repository:

https://github.com/abhiram0a/TypingSpeedApp

2. Test the App Locally

Open index.html in your browser or use a **Live Server extension in VS Code**.

Ensure the app works correctly.



Step 2: Set Up a GitHub Repository

1. Initialize Git Repository

Open a terminal in VS Code.

Navigate to the project folder:

```
cd TypingSpeedApp
```

Initialize a Git repository:

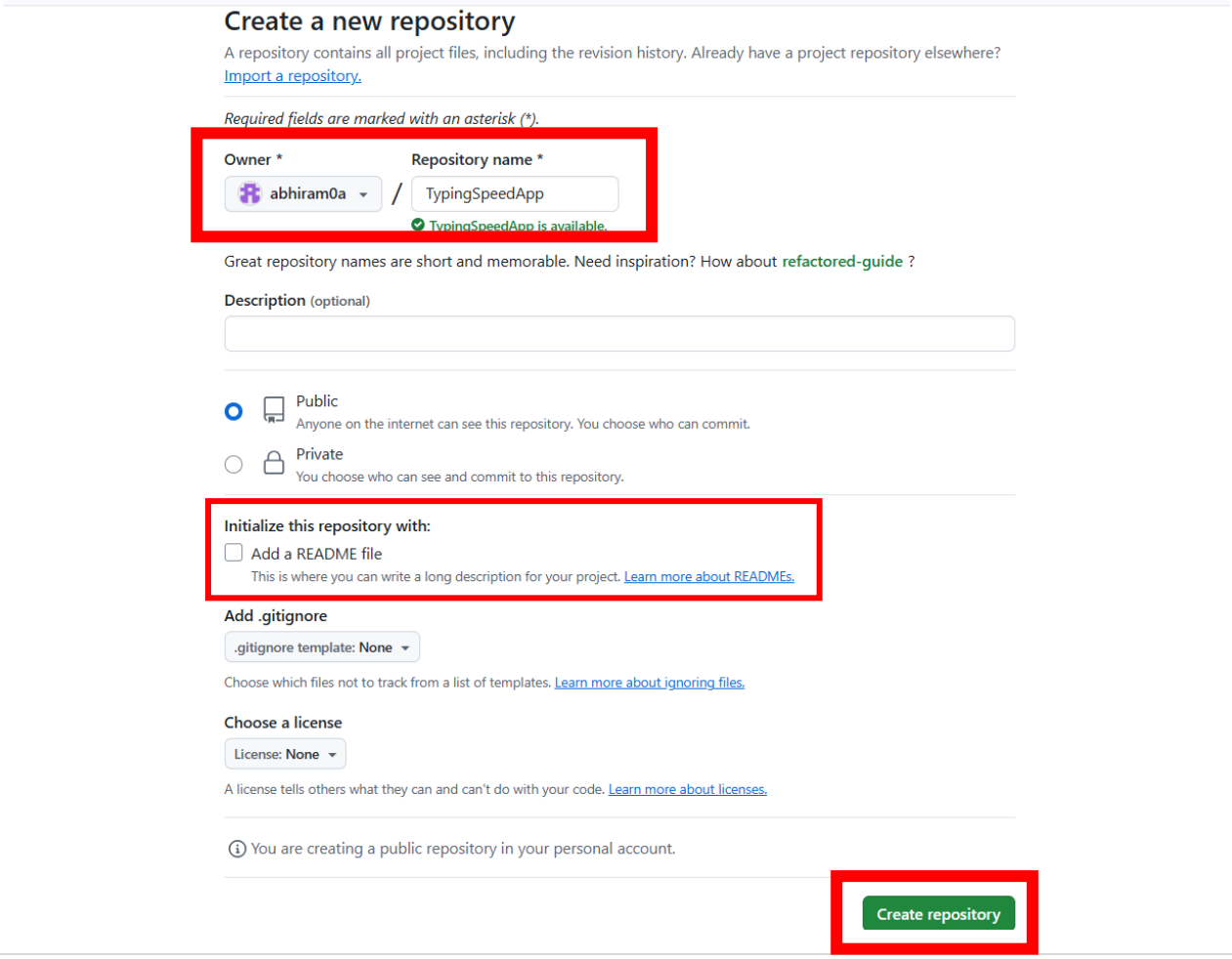
```
git init
```

2. Create a GitHub Repository

Go to your GitHub account.

Create a new repository named TypingSpeedApp

Do Not initialize with a README (since Git is already initialized locally).



3. Push Code to GitHub

Open terminal in VS Code

Add the remote URL (**after replacing placeholder** with your **GitHub username**):

```
git remote add origin https://github.com/<your-username>/TypingSpeedApp.git
```

Add and **Commit** files:

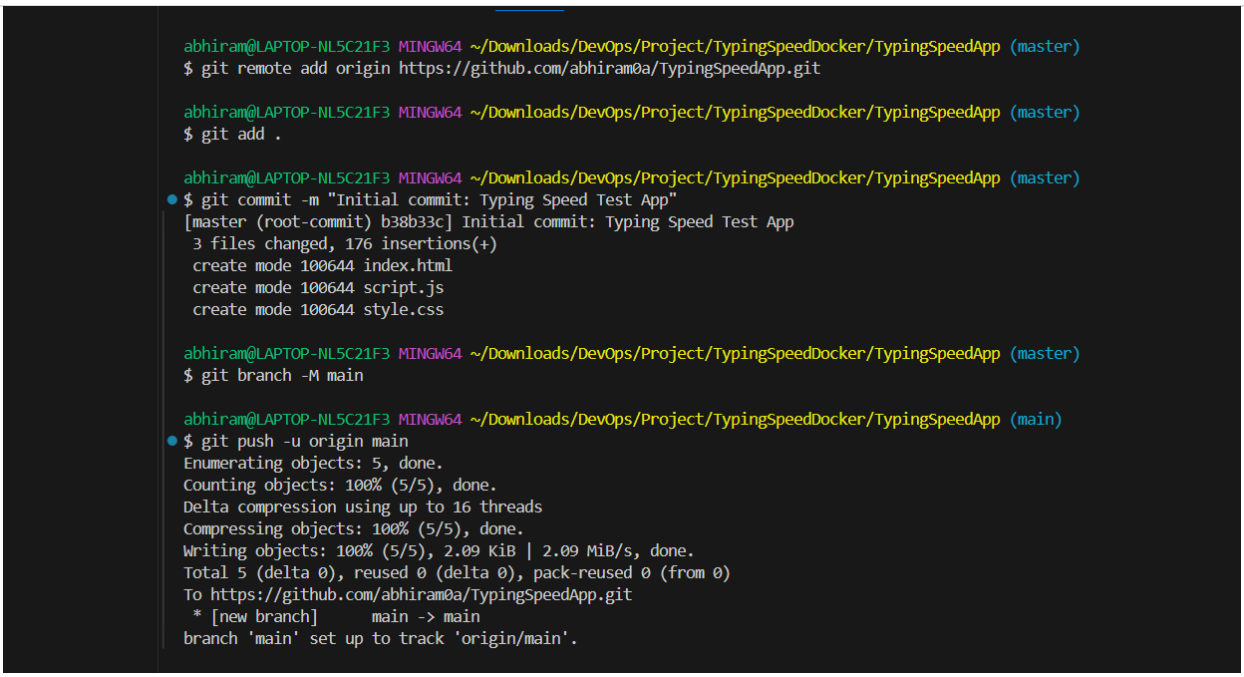
```
git add .
```

```
git commit -m "Initial commit: Typing Speed Test App"
```

Push to GitHub:

```
git branch -M main
```

```
git push -u origin main
```



Step 3: Write a Dockerfile

Create a file named Dockerfile in the project folder

Add the following content & **Save**:

```
FROM nginx:latest

COPY index.html /usr/share/nginx/html/index.html

COPY style.css /usr/share/nginx/html/style.css

COPY script.js /usr/share/nginx/html/script.js

EXPOSE 80
```

Step 4: Build and Run the Docker Container

Ensure that Docker Desktop is running

Build the **Docker Image**:

```
docker build -t typing-speed-test .
```

Run the Container:

```
docker run -d -p 8080:80 --name typing-speed-app typing-speed-test
```

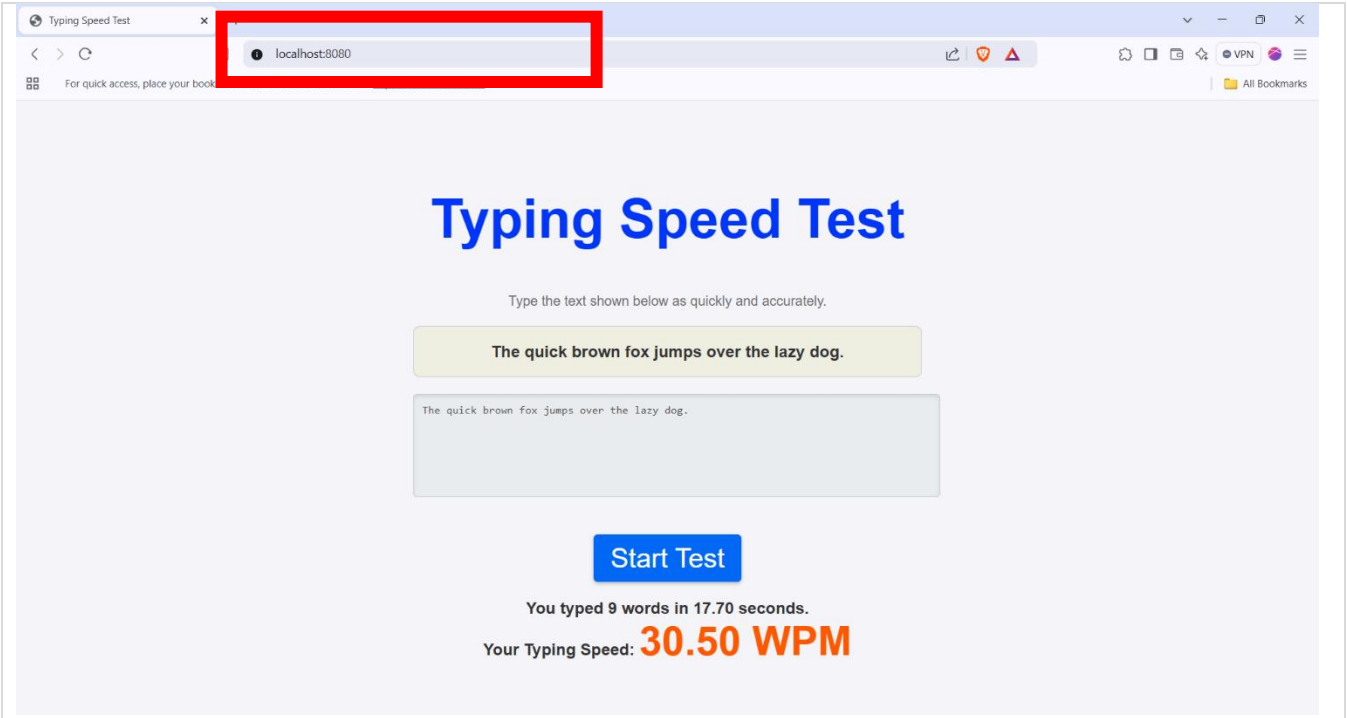
```
abhiram@LAPTOP-NL5C21F3 MINGW64 ~/Downloads/DevOps/Project/TypingSpeedDocker/TypingSpeedApp (main)
$ docker build -t typing-speed-test .
[+] Building 65.3s (9/9) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 347B
 => [internal] load metadata for docker.io/library/nginx:latest
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/4] FROM docker.io/library/nginx:latest@sha256:0a399eb16751829e1af26fea27b20c3ec28d7ab1fb72182879dcae1cca21206a
 => => resolve docker.io/library/nginx:latest@sha256:0a399eb16751829e1af26fea27b20c3ec28d7ab1fb72182879dcae1cca21206a
 => => sha256:9bea9f2796e236cb18c2b3ad561ff29f655d1001f9ec7247a0bc5e08d25652a1 8.58kB / 8.58kB
 => => sha256:207b812743af01771d6c94b3ebc59543bf3c6fefe5d4d9ebfa6e5e5558801d34 43.84MB / 43.84MB
 => => sha256:2426c815287ed75a3a33dd28512eba4f0f783946844209ccf3fa8990817a4eb9 2.29kB / 2.29kB
 => => sha256:af302e5c37e9dc1dbe2eadc8f5059d82a914066b541b0d1a6daa91d0cc55057d 28.21MB / 28.21MB
 => => sha256:841e383b441eda86c164d8e833da080b951c951d612853cac2fd44eed0d72226 627B / 627B
 => => sha256:0a399eb16751829e1af26fea27b20c3ec28d7ab1fb72182879dcae1cca21206a 10.27kB / 10.27kB
 => => sha256:0256c04a8d84ab5ae434b607055df8fdfe0278dba43f3e603a861dc7e26da395 957B / 957B
 => => sha256:38e992d287c563d332ed8c9bdefec32a84f50f94cc39bb4729a3e73cefde88eb 406B / 406B
 => => sha256:9e9aab598f58e86706b73c23e0ff33fc4eb60a5e69424147f9cb7c02e853323b 1.21kB / 1.21kB
 => => sha256:4de87b37f4ad0b499be3db4cc6831db3d8158867f12dabc0bf42046b56784d3b 1.40kB / 1.40kB
 => => extracting sha256:af302e5c37e9dc1dbe2eadc8f5059d82a914066b541b0d1a6daa91d0cc55057d
 => => extracting sha256:207b812743af01771d6c94b3ebc59543bf3c6fefe5d4d9ebfa6e5e5558801d34
 => => extracting sha256:841e383b441eda86c164d8e833da080b951c951d612853cac2fd44eed0d72226
 => => extracting sha256:0256c04a8d84ab5ae434b607055df8fdfe0278dba43f3e603a861dc7e26da395
 => => extracting sha256:38e992d287c563d332ed8c9bdefec32a84f50f94cc39bb4729a3e73cefde88eb
 => => extracting sha256:9e9aab598f58e86706b73c23e0ff33fc4eb60a5e69424147f9cb7c02e853323b
 => => extracting sha256:4de87b37f4ad0b499be3db4cc6831db3d8158867f12dabc0bf42046b56784d3b
 => [internal] load build context
 => => transferring context: 4.70kB
 => [2/4] COPY index.html /usr/share/nginx/html/index.html
 => [3/4] COPY style.css /usr/share/nginx/html/style.css
 => [4/4] COPY script.js /usr/share/nginx/html/script.js
 => exporting to image
 => => exporting layers
 => => writing image sha256:169017d902f6d51fd28965511557d24e91378fb53f345f720fdae86353805b31
 => => naming to docker.io/library/typing-speed-test

abhiram@LAPTOP-NL5C21F3 MINGW64 ~/Downloads/DevOps/Project/TypingSpeedDocker/TypingSpeedApp (main)
$ docker run -d -p 8080:80 --name typing-speed-app typing-speed-test
ee9e14a6da5af6479ebae678008792f1ddab21bb8decca1b70b9a0dc0a62dd1f
```

Test the App

Open browser and **search**:

```
http://localhost:8080
```



Step 5: Push the Dockerfile to GitHub

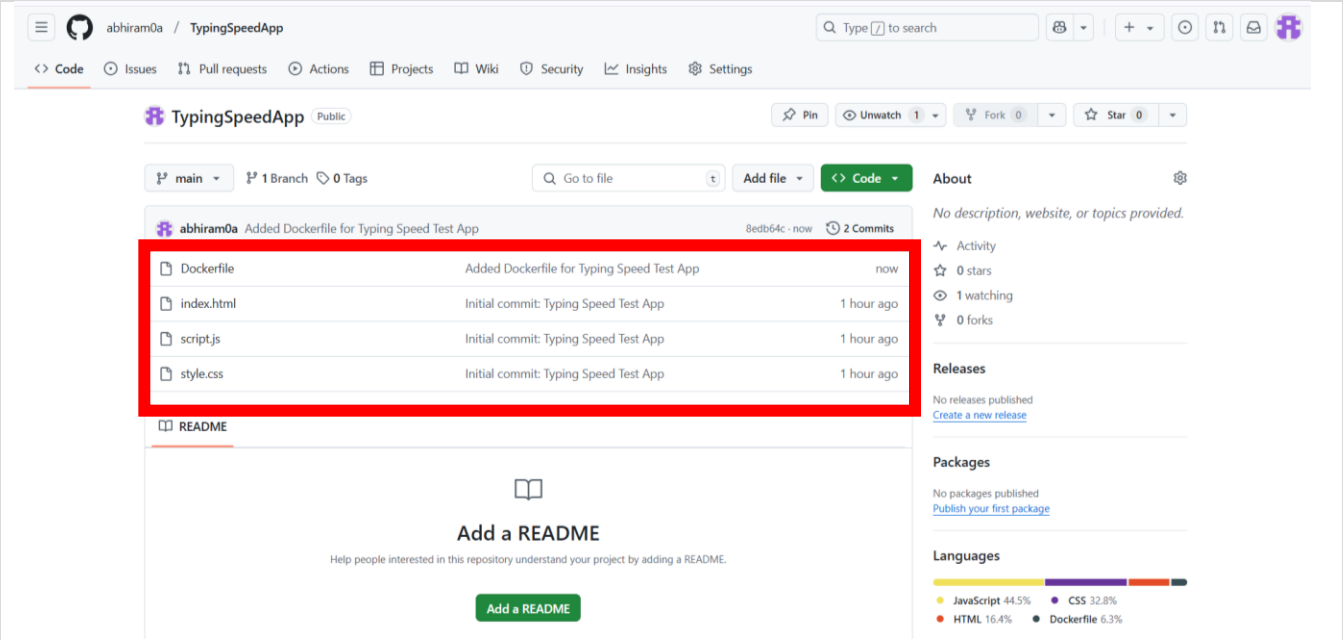
Add the Dockerfile to the repository:

```
git add Dockerfile
```

```
git commit -m "Added Dockerfile for Typing Speed Test App"
```

```
git push origin main
```

Check the GitHub repository



Stop the Container (Optional)

If you no longer need the container running, stop it with this command:

```
docker stop typing-speed-app
```

Conclusion

This project demonstrated how to containerize and deploy a Typing Speed Test web application using Docker and GitHub. By following these steps, you learned to set up version control with Git, write a Dockerfile, build and run Docker containers, and push your project to GitHub.