DEVOPS PROJECT

CI/CD Pipeline for Web App using GitHub Actions and Docker

Overview:

This project demonstrates how to set up a CI/CD pipeline for a web application using GitHub Actions and Docker. The pipeline automates building, testing, and deploying the application to Docker Hub, ensuring seamless updates and deployment.

Prerequisites

Before starting, ensure you have the following installed and set up:

- **Git**
- **GitHub Account**
- **Docker Desktop**
- **VS Code**

Step 1: Set Up the Project File

Open VS Code and create a new folder named **bmi-calculator**

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

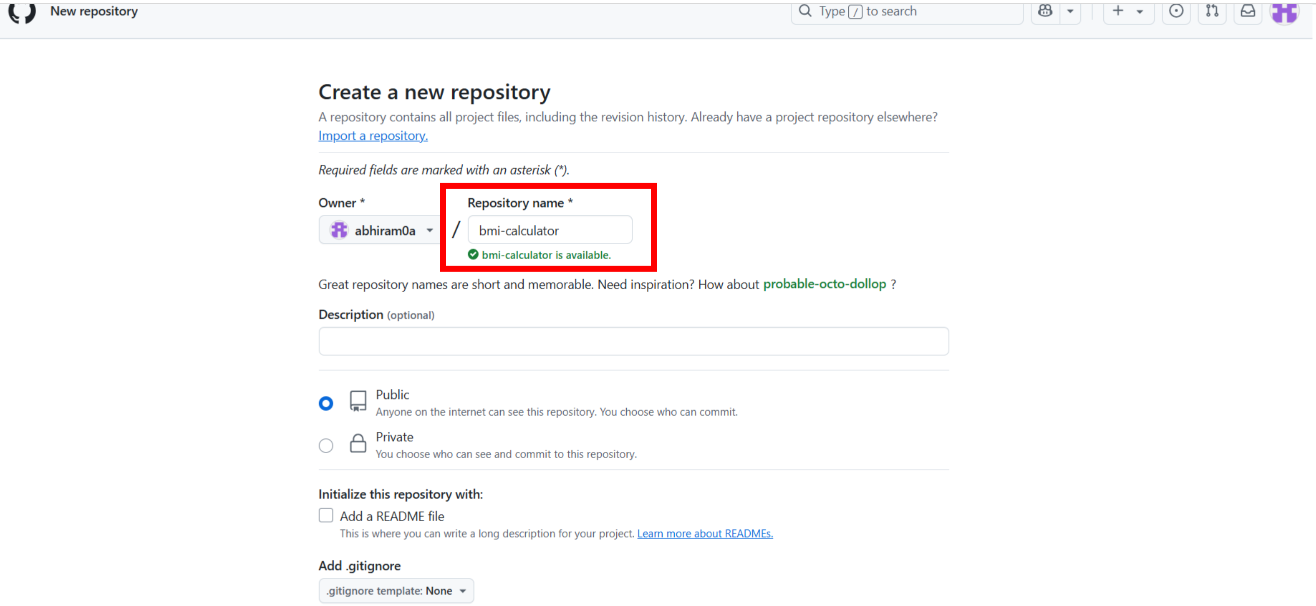
- `index.html`
- `style.css`
- `script.js`

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

https://github.com/abhiram0a/bmi-calculator_CI-CD-Pipeline

Step 2: Set Up GitHub Reposito

Create new repository on GitHub named **bmi-calculator**


New repository

Type / to search

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner *: abhiram0a / Repository name *: bmi-calculator

bmi-calculator is available.

Great repository names are short and memorable. Need inspiration? How about probable-octo-dollop ?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:

□ Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Open Terminal in VS Code

Initialize Git in the local project folder:

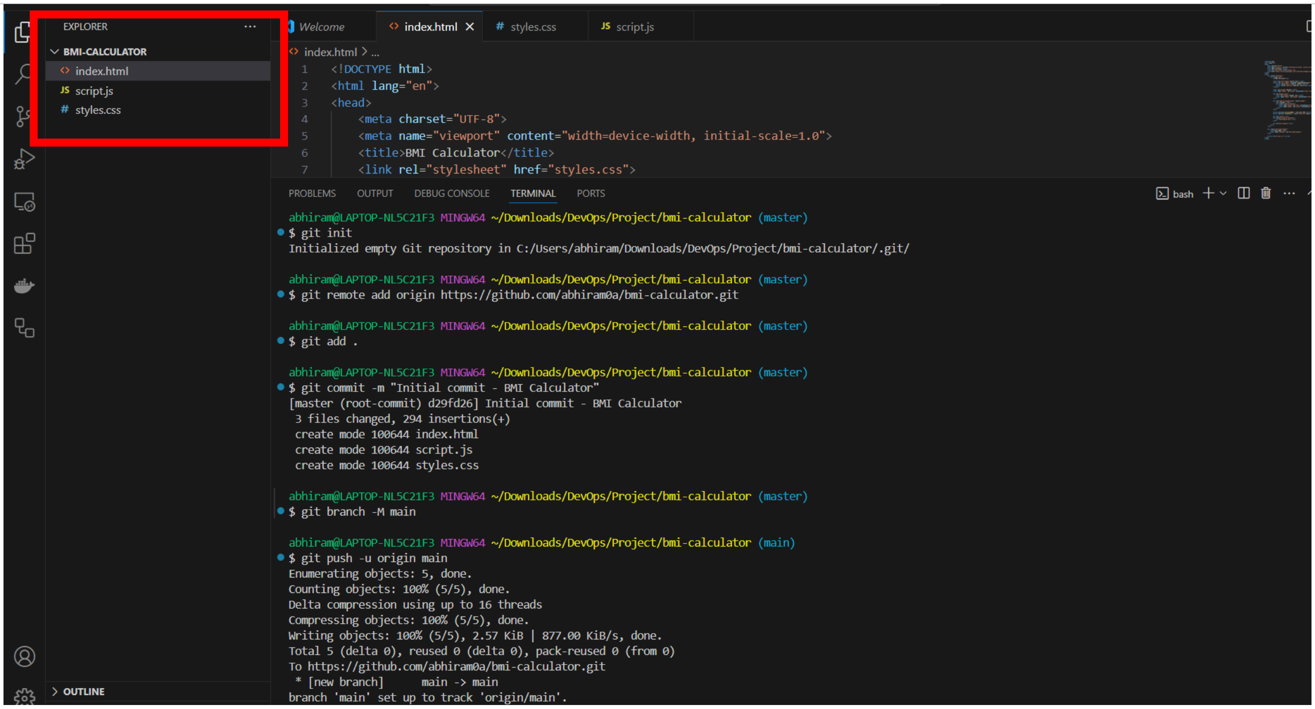
```
git init
```

Add the remote URL (Replace the placeholder)

```
git remote add origin https://github.com/your-username/bmi-calculator.git
```

Add and **push** your project files to GitHub Repository:

```
git add .

git commit -m "Initial commit - BMI Calculator"

git branch -M main

git push -u origin main
```



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>BMI Calculator</title>
    <link rel="stylesheet" href="styles.css">
```

```
abhiram@LAPTOP-NL5C21F3 MINGW64 ~/Downloads/DevOps/Project/bmi-calculator (master)
$ git init
Initialized empty Git repository in C:/Users/abhiram/Downloads/DevOps/Project/bmi-calculator/.git/

abhiram@LAPTOP-NL5C21F3 MINGW64 ~/Downloads/DevOps/Project/bmi-calculator (master)
$ git remote add origin https://github.com/abhiram0a/bmi-calculator.git

abhiram@LAPTOP-NL5C21F3 MINGW64 ~/Downloads/DevOps/Project/bmi-calculator (master)
$ git add .

abhiram@LAPTOP-NL5C21F3 MINGW64 ~/Downloads/DevOps/Project/bmi-calculator (master)
$ git commit -m "Initial commit - BMI Calculator"
[master (root-commit) d29fd26] Initial commit - BMI Calculator
 3 files changed, 294 insertions(+)
 create mode 100644 index.html
 create mode 100644 script.js
 create mode 100644 styles.css

abhiram@LAPTOP-NL5C21F3 MINGW64 ~/Downloads/DevOps/Project/bmi-calculator (master)
$ git branch -M main

abhiram@LAPTOP-NL5C21F3 MINGW64 ~/Downloads/DevOps/Project/bmi-calculator (main)
$ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 2.57 KiB | 877.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/abhiram0a/bmi-calculator.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Step 3: Containerize the App Using Docker

Ensure Docker Desktop is running.

Create a Dockerfile

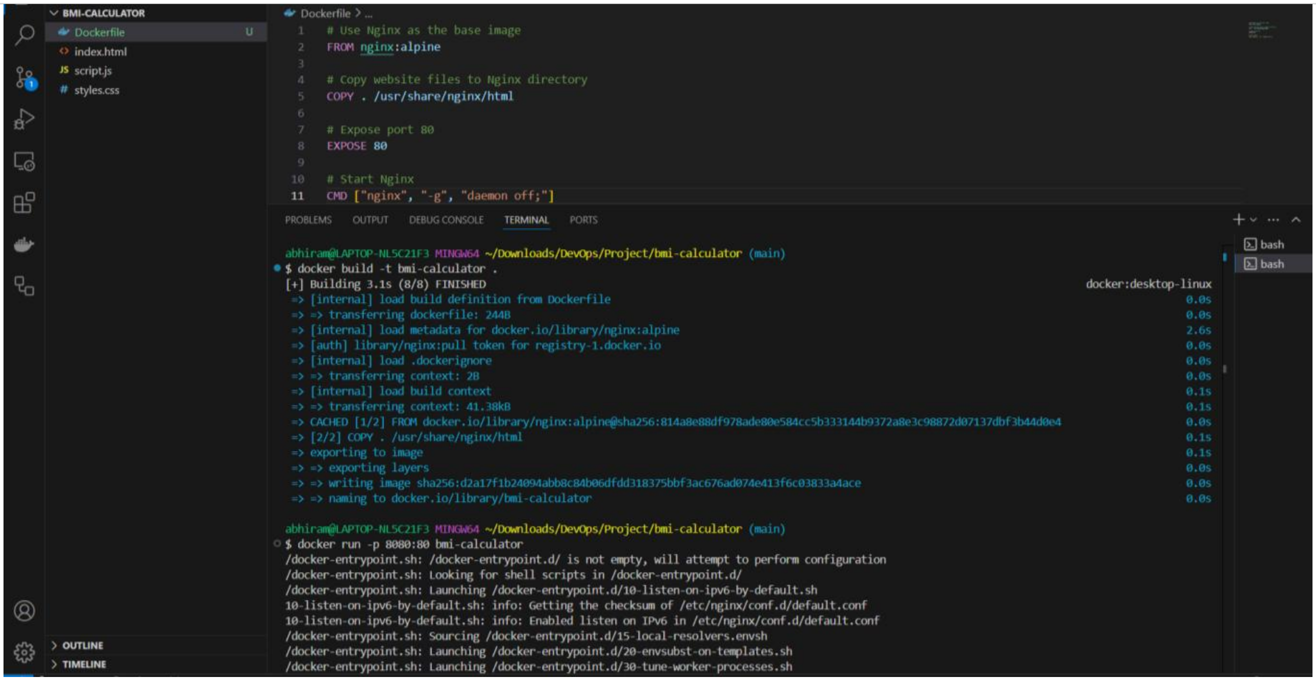
Create file named Dockerfile in the root of your **project folder**

Add the following code in it:

```
# Use Nginx as the base image

FROM nginx:alpine


# Copy website files to Nginx directory

COPY . /usr/share/nginx/html


# Expose port 80

EXPOSE 80


# Start Nginx

CMD ["nginx", "-g", "daemon off;"]
```

Test the Docker image locally:

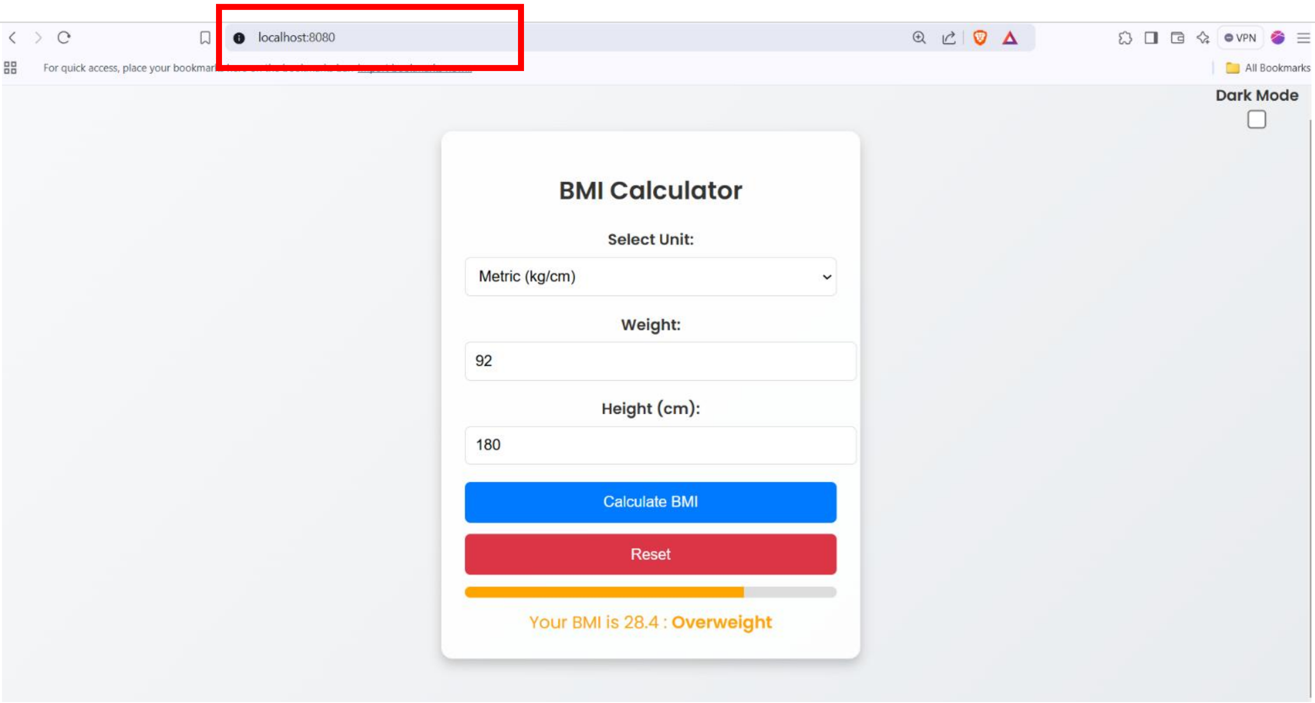
```
docker build -t bmi-calculator .

docker run -p 8080:80 bmi-calculator
```



```
# Use Nginx as the base image
FROM nginx:alpine

# Copy website files to Nginx directory
COPY . /usr/share/nginx/html

# Expose port 80
EXPOSE 80

# Start Nginx
CMD ["nginx", "-g", "daemon off;"]
```

```
abhiram@LAPTOP-NL5C21F3 MINGW64 ~/Downloads/DevOps/Project/bmi-calculator (main)
$ docker build -t bmi-calculator .
[+] Building 3.1s (8/8) FINISHED                                                              docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                          0.0s
 => => transferring dockerfile: 244B                                                                          0.0s
 => [internal] load metadata for docker.io/library/nginx:alpine                                               2.6s
 => [auth] library/nginx:pull token for registry-1.docker.io                                                  0.0s
 => [internal] load .dockerignore                                                                             0.0s
 => => transferring context: 2B                                                                               0.0s
 => [internal] load build context                                                                             0.1s
 => => transferring context: 41.38kB                                                                          0.1s
 => CACHED [1/2] FROM docker.io/library/nginx:alpine@sha256:814a8e88df978ade80e584cc5b333144b9372a8e3c98872d07137dbf3b44d0e4  0.0s
 => [2/2] COPY . /usr/share/nginx/html                                                                        0.1s
 => exporting to image                                                                                        0.1s
 => => exporting layers                                                                                       0.0s
 => => writing image sha256:d2a17f1b24094abb8c84b06dfdd318375bbf3ac676ad074e413f6c03833a4ace                  0.0s
 => => naming to docker.io/library/bmi-calculator                                                             0.0s

abhiram@LAPTOP-NL5C21F3 MINGW64 ~/Downloads/DevOps/Project/bmi-calculator (main)
$ docker run -p 8080:80 bmi-calculator
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
```

Open http://localhost:8080/ in your browser to verify it’s running.



Select Unit: Metric (kg/cm)

Weight: 92

Height (cm): 180

Your BMI is 28.4 : **Overweight**

Stop the container using `Ctrl + C`

Push the Dockerfile to GitHub:

```
git add Dockerfile
```

```
git commit -m "Added Dockerfile"
```

```
git push origin main
```

Step 4: Set Up Docker Hub

Create a Docker Hub account if you don’t have already:

https://hub.docker.com/

Log in to Docker from the **terminal:**

```
docker login
```

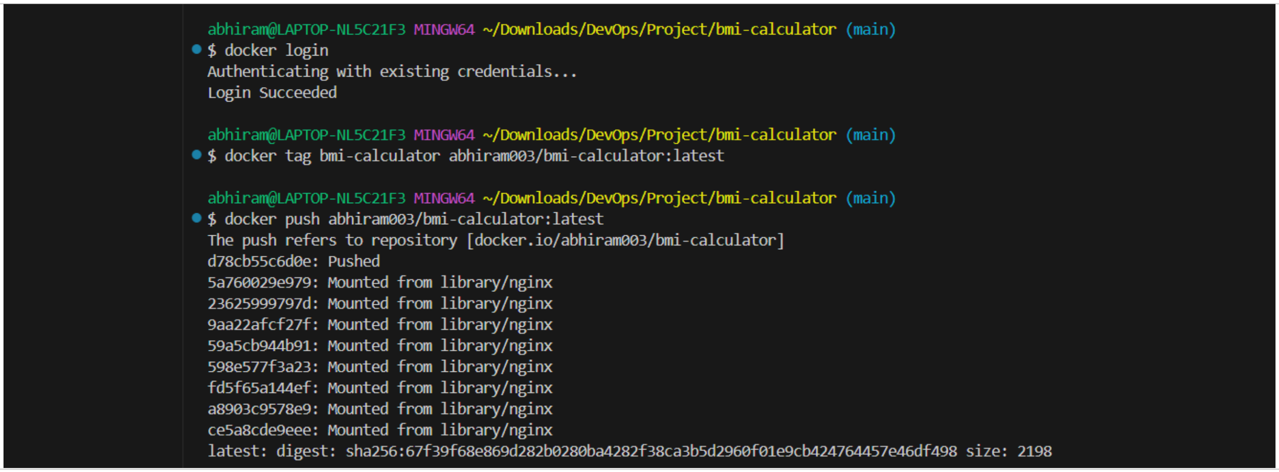
If logging in for the first time, **complete the authentication** in your browser

Tag the Docker image: *(Replace placeholders)*

```
docker tag web-app your-dockerhub-username/bmi-calculator:latest
```

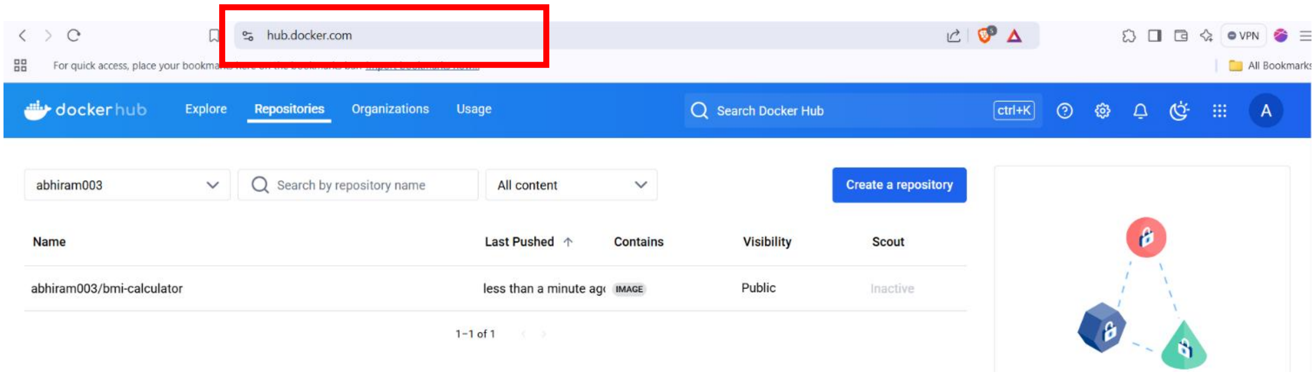
Push the image to Docker Hub: *(Replace placeholders)*

```
docker push your-dockerhub-username/bmi-calculator:latest
```



```
abhiram@LAPTOP-NL5C21F3 MINGW64 ~/Downloads/DevOps/Project/bmi-calculator (main)
$ docker login
Authenticating with existing credentials...
Login Succeeded

abhiram@LAPTOP-NL5C21F3 MINGW64 ~/Downloads/DevOps/Project/bmi-calculator (main)
$ docker tag bmi-calculator abhiram003/bmi-calculator:latest

abhiram@LAPTOP-NL5C21F3 MINGW64 ~/Downloads/DevOps/Project/bmi-calculator (main)
$ docker push abhiram003/bmi-calculator:latest
The push refers to repository [docker.io/abhiram003/bmi-calculator]
d78cb55c6d0e: Pushed
5a760029e979: Mounted from library/nginx
23625999797d: Mounted from library/nginx
9aa22afcf27f: Mounted from library/nginx
59a5cb944b91: Mounted from library/nginx
598e577f3a23: Mounted from library/nginx
fd5f65a144ef: Mounted from library/nginx
a8903c9578e9: Mounted from library/nginx
ce5a8cde9eee: Mounted from library/nginx
latest: digest: sha256:67f39f68e869d282b0280ba4282f38ca3b5d2960f01e9cb424764457e46df498 size: 2198
```

Your Docker image is now available on Docker Hub!



Name	Last Pushed ↑	Contains	Visibility	Scout
abhiram003/bmi-calculator	less than a minute ag	IMAGE	Public	Inactive

1–1 of 1

Step 5: Set Up GitHub Actions for CI/CD

1. Create a GitHub Actions workflow:

In the root of your project (`/bmi-calculator`), **create a directory** `.github/workflows/`

Inside, **create a file** named `docker-ci.yml`

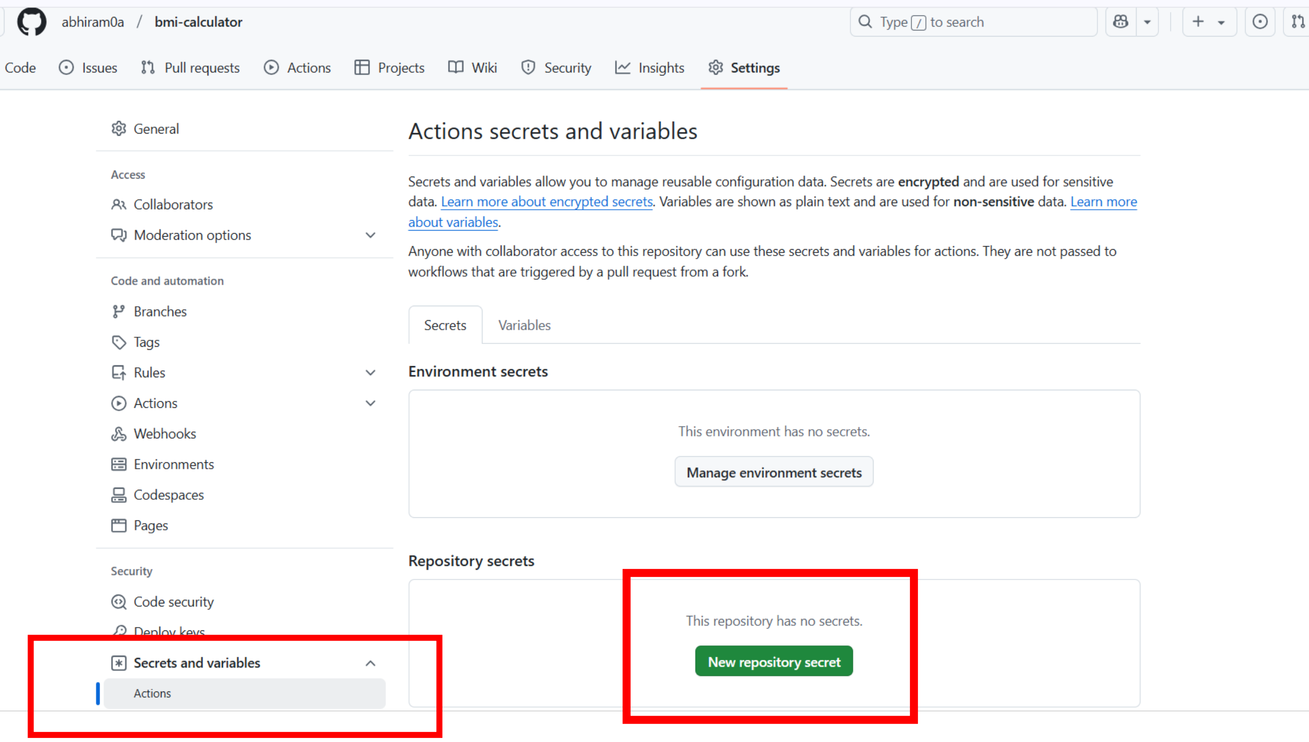
2. Add the following YAML configuration *(Replace placeholder at last line)*

```
name: CI/CD Pipeline

on:
  push:
    branches:
      - main
  pull_request:
    branches:
      - main

jobs:
  build-and-deploy:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v3

      - name: Set up Docker Buildx
        uses: docker/setup-buildx-action@v2

      - name: Log in to Docker Hub
        uses: docker/login-action@v2
        with:
          username: ${{ secrets.DOCKER_USERNAME }}
          password: ${{ secrets.DOCKER_PASSWORD }}

      - name: Build and push Docker image
        uses: docker/build-push-action@v4
        with:
          context: .
          file: ./Dockerfile
          push: true
          tags: your-dockerhub-username/bmi-calculator:latest
```

3. Add GitHub Secrets for Docker Hub Login:

Go to **GitHub Repository** > **Settings** > **Secrets and Variables** > **Actions**

Click New repository secret



Actions secrets and variables

Secrets and variables allow you to manage reusable configuration data. Secrets are **encrypted** and are used for sensitive data. Learn more about encrypted secrets. Variables are shown as plain text and are used for **non-sensitive** data. Learn more about variables.

Anyone with collaborator access to this repository can use these secrets and variables for actions. They are not passed to workflows that are triggered by a pull request from a fork.

Environment secrets

This environment has no secrets.

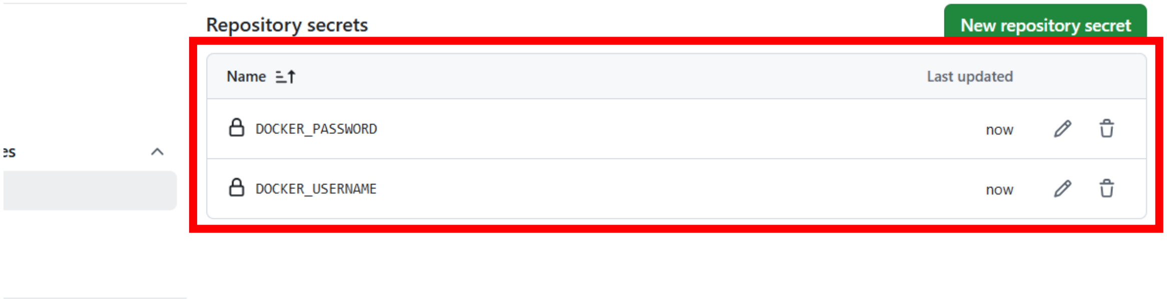
Repository secrets

This repository has no secrets.

Add two secrets:

DOCKER_USERNAME: `Your Docker Hub username`

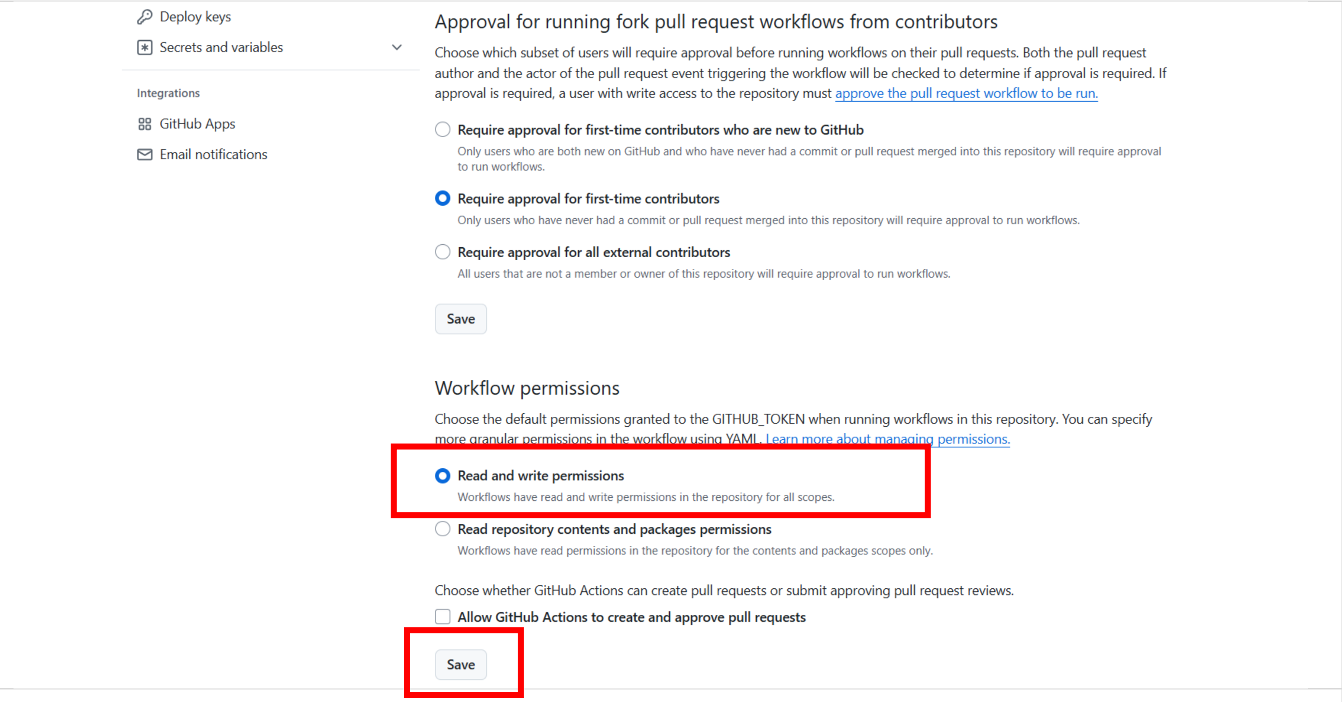
DOCKER_PASSWORD: `Your Docker Hub password`



Repository secrets

Name	Last updated
DOCKER_PASSWORD	now
DOCKER_USERNAME	now

Go to Repository **Settings** > **Actions** > **General** and **Enable Read and Write permissions**



Approval for running fork pull request workflows from contributors

Choose which subset of users will require approval before running workflows on their pull requests. Both the pull request author and the actor of the pull request event triggering the workflow will be checked to determine if approval is required. If approval is required, a user with write access to the repository must approve the pull request workflow to be run.

○ Require approval for first-time contributors who are new to GitHub
Only users who are both new on GitHub and who have never had a commit or pull request merged into this repository will require approval to run workflows.

● Require approval for first-time contributors
Only users who have never had a commit or pull request merged into this repository will require approval to run workflows.

○ Require approval for all external contributors
All users that are not a member or owner of this repository will require approval to run workflows.

Workflow permissions

Choose the default permissions granted to the GITHUB_TOKEN when running workflows in this repository. You can specify more granular permissions in the workflow using YAML. Learn more about managing permissions.

● Read and write permissions
Workflows have read and write permissions in the repository for all scopes.

○ Read repository contents and packages permissions
Workflows have read permissions in the repository for the contents and packages scopes only.

Choose whether GitHub Actions can create pull requests or submit approving pull request reviews.

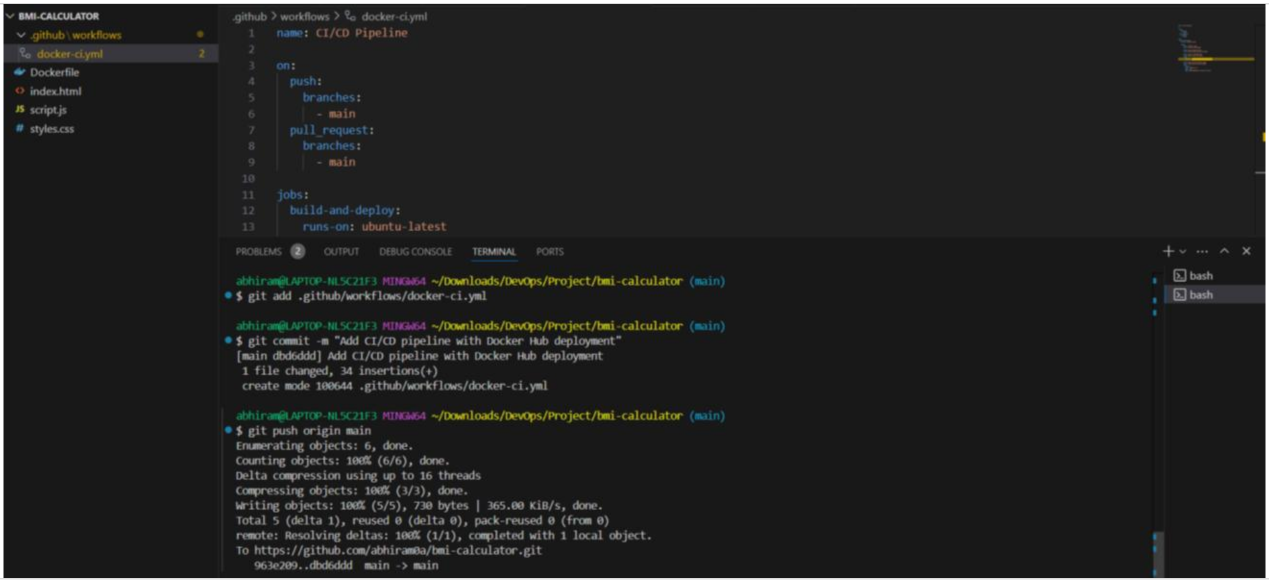
□ Allow GitHub Actions to create and approve pull requests

4. Commit and push the workflow file:

```
git add .github/workflows/docker-ci.yml
```

```
git commit -m "Add CI/CD pipeline with Docker Hub deployment"
```

```
git push origin main
```



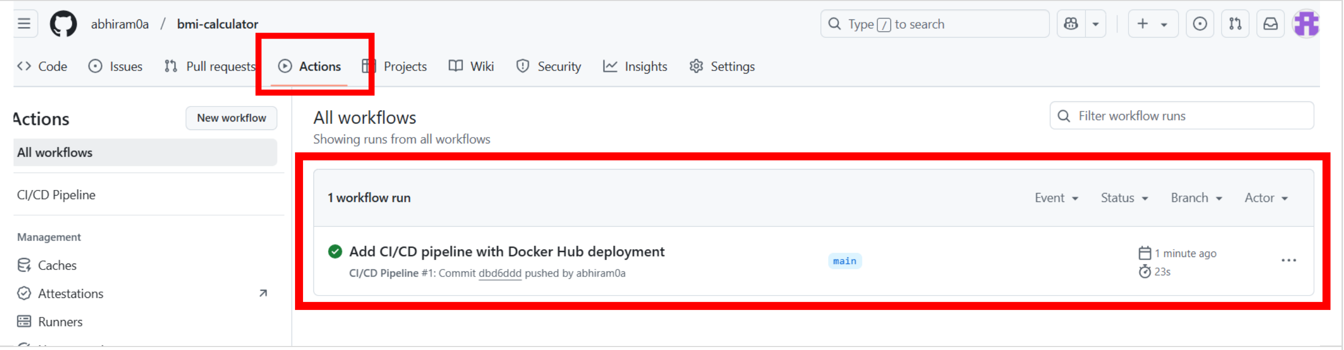
```
name: CI/CD Pipeline

on:
  push:
    branches:
      - main
  pull_request:
    branches:
      - main

jobs:
  build-and-deploy:
    runs-on: ubuntu-latest
```

```
abhiram@LAPTOP-NL5C21F3 MINGW64 ~/Downloads/DevOps/Project/bmi-calculator (main)
$ git add .github/workflows/docker-ci.yml

abhiram@LAPTOP-NL5C21F3 MINGW64 ~/Downloads/DevOps/Project/bmi-calculator (main)
$ git commit -m "Add CI/CD pipeline with Docker Hub deployment"
[main dbd6ddd] Add CI/CD pipeline with Docker Hub deployment
 1 file changed, 34 insertions(+)
 create mode 100644 .github/workflows/docker-ci.yml

abhiram@LAPTOP-NL5C21F3 MINGW64 ~/Downloads/DevOps/Project/bmi-calculator (main)
$ git push origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 730 bytes | 365.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/abhiram0a/bmi-calculator.git
   963e209..dbd6ddd  main -> main
```

Step 6: Monitor the GitHub Actions Workflow

Go to GitHub Actions in **your repository**.

Check if the workflow is **running**.

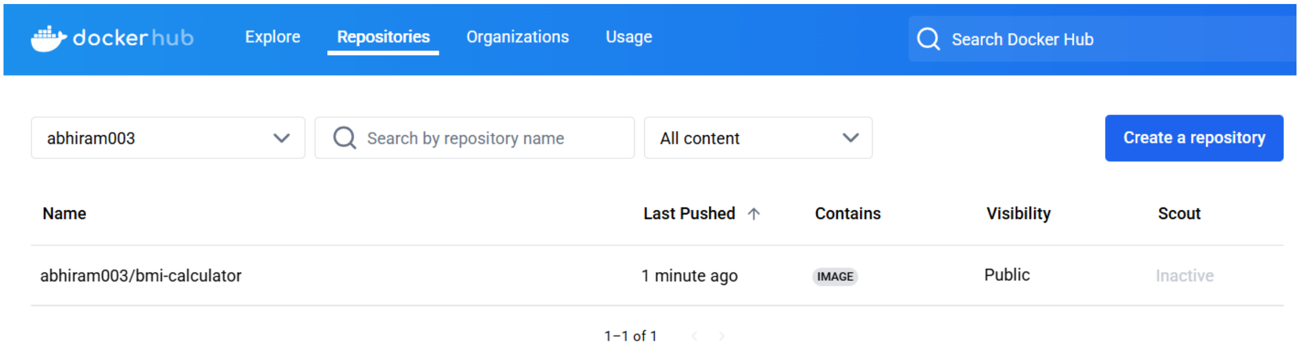


All workflows
Showing runs from all workflows

1 workflow run

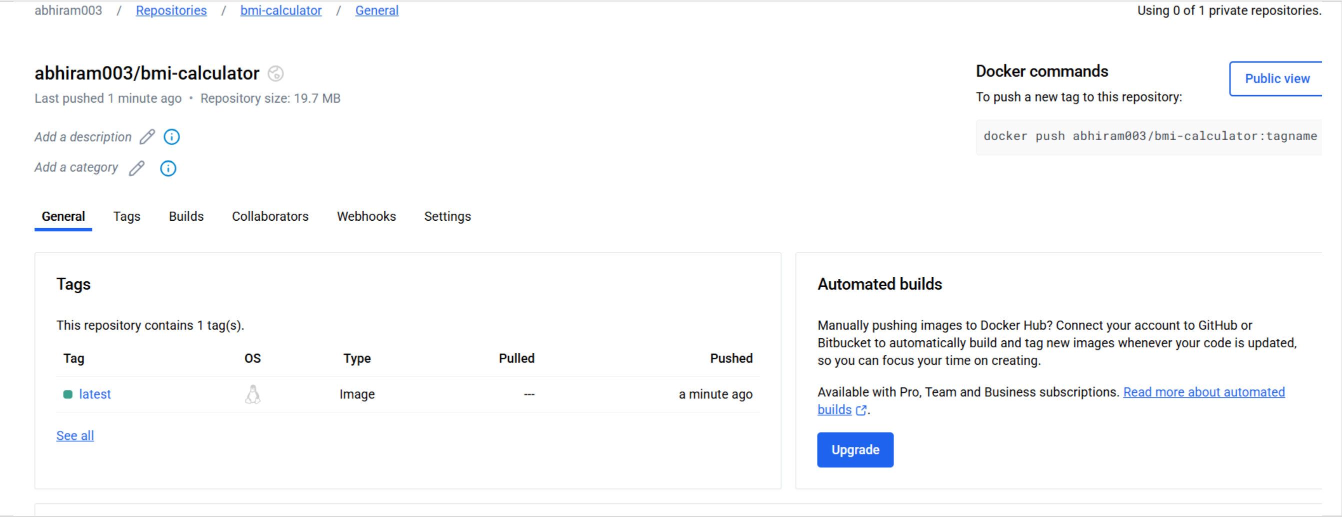
Add CI/CD pipeline with Docker Hub deployment
CI/CD Pipeline #1: Commit dbd6ddd pushed by abhiram0a
main
1 minute ago
23s

Once completed, the Docker image will be **pushed to** Docker Hub.



Name	Last Pushed ↑	Contains	Visibility	Scout
abhiram003/bmi-calculator	1 minute ago	IMAGE	Public	Inactive

1–1 of 1



abhiram003 / Repositories / bmi-calculator / General

Using 0 of 1 private repositories.

abhiram003/bmi-calculator

Last pushed 1 minute ago • Repository size: 19.7 MB

Add a description

Add a category

Docker commands

To push a new tag to this repository:

```
docker push abhiram003/bmi-calculator:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
🟩 latest		Image	---	a minute ago

See all

Automated builds

Manually pushing images to Docker Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. Read more about automated builds.

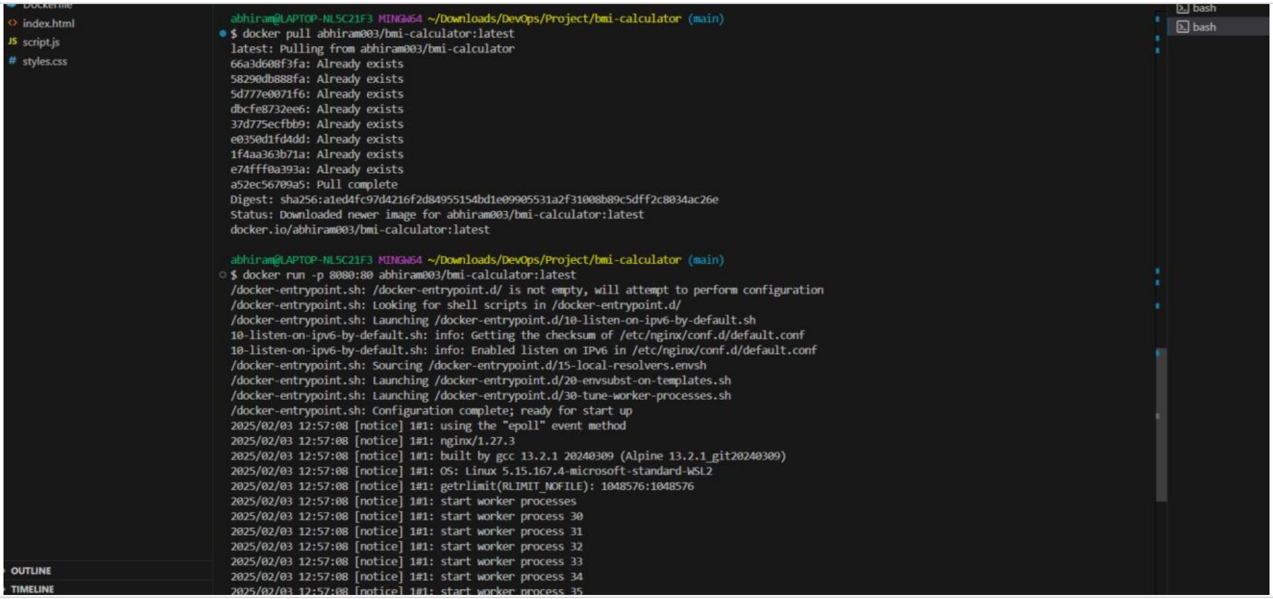
Step 7: Deploy the Docker Image

Pull the **latest image** from Docker Hub: *(Replace placeholders)*

```
docker pull your-dockerhub-username/bmi-calculator:latest
```

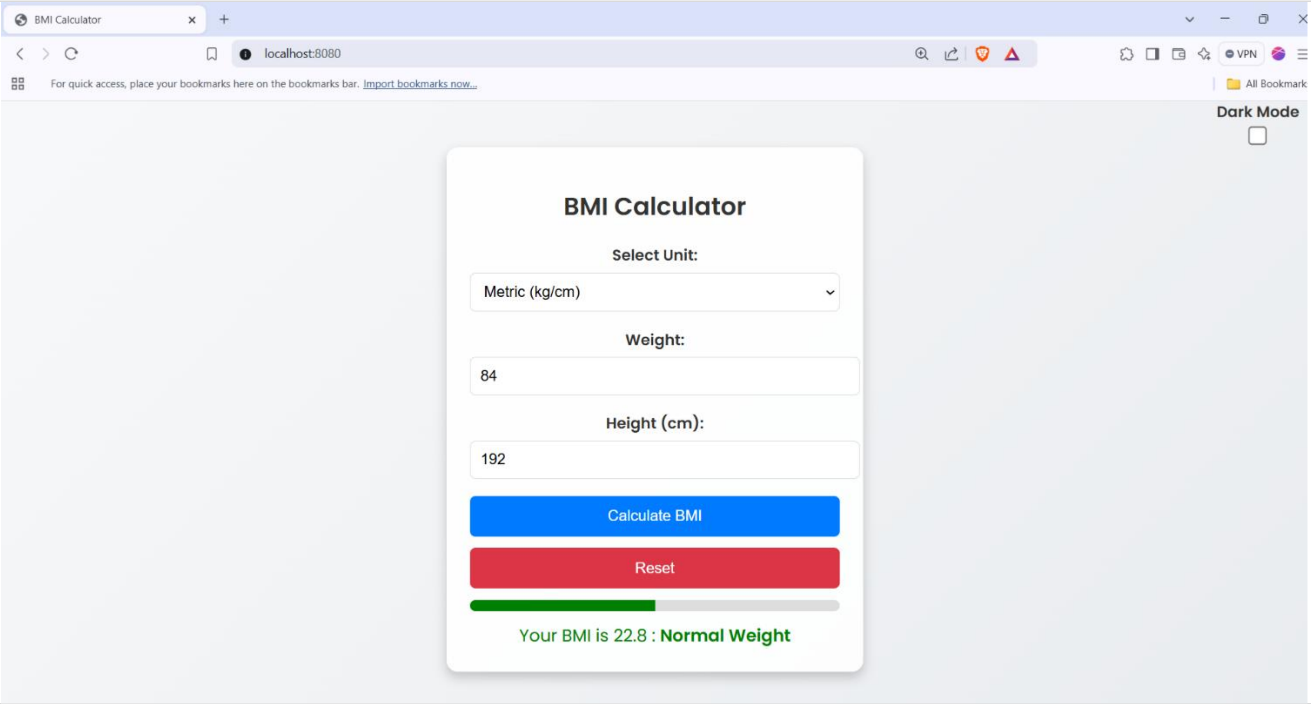
Run the container: *(Replace placeholders)*

```
docker run -p 8080:80 your-dockerhub-username/bmi-calculator:latest
```



```
abhiram@LAPTOP-NL5C21F3 MINGW64 ~/Downloads/DevOps/Project/bmi-calculator (main)
$ docker pull abhiram003/bmi-calculator:latest
latest: Pulling from abhiram003/bmi-calculator
66a3d608f3fa: Already exists
58290db888fa: Already exists
5d777e0071f6: Already exists
dbcfe8732ee6: Already exists
37d775ecfbb9: Already exists
e0350d1fd4dd: Already exists
1f4aa363b71a: Already exists
e74fff0a393a: Already exists
a52ec56709a5: Pull complete
Digest: sha256:a1ed4fc97d4216f2d84955154bd1e09905531a2f31008b89c5dff2c8034ac26e
Status: Downloaded newer image for abhiram003/bmi-calculator:latest
docker.io/abhiram003/bmi-calculator:latest

abhiram@LAPTOP-NL5C21F3 MINGW64 ~/Downloads/DevOps/Project/bmi-calculator (main)
$ docker run -p 8080:80 abhiram003/bmi-calculator:latest
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/02/03 12:57:08 [notice] 1#1: using the "epoll" event method
2025/02/03 12:57:08 [notice] 1#1: nginx/1.27.3
2025/02/03 12:57:08 [notice] 1#1: built by gcc 13.2.1 20240309 (Alpine 13.2.1_git20240309)
2025/02/03 12:57:08 [notice] 1#1: OS: Linux 5.15.167.4-microsoft-standard-WSL2
2025/02/03 12:57:08 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/02/03 12:57:08 [notice] 1#1: start worker processes
2025/02/03 12:57:08 [notice] 1#1: start worker process 30
2025/02/03 12:57:08 [notice] 1#1: start worker process 31
2025/02/03 12:57:08 [notice] 1#1: start worker process 32
2025/02/03 12:57:08 [notice] 1#1: start worker process 33
2025/02/03 12:57:08 [notice] 1#1: start worker process 34
2025/02/03 12:57:08 [notice] 1#1: start worker process 35
```

Open http://localhost:8080/ in **your browser** to verify the deployment



For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

Select Unit: Metric (kg/cm)

Weight: 84

Height (cm): 192

Your BMI is 22.8 : **Normal Weight**

Conclusion:

By following these steps, you have successfully:

- **Containerized your web app** using Docker.
- **Set up a CI/CD pipeline** using GitHub Actions.
- **Automated deployment** to Docker Hub.

This project showcases essential **DevOps best practices** and provides valuable experience in **CI/CD, containerization, and automation**.