**Github Repository**

|  |  |
| --- | --- |
| **https://github.com/Prashanth6782/GHA\_HV.git** | Contains all code and also the Github Action Script .. Python app is as well there |
| **https://github.com/Prashanth6782/GHA\_HV.git** | **Document1.docx (Test Screenshots)** |

Problem Statement GITHUB actions

**2. GitHub Actions CI/CD Pipeline Flask App**

**Objective:**

Implement a CI/CD workflow using GitHub Actions for a Python application.

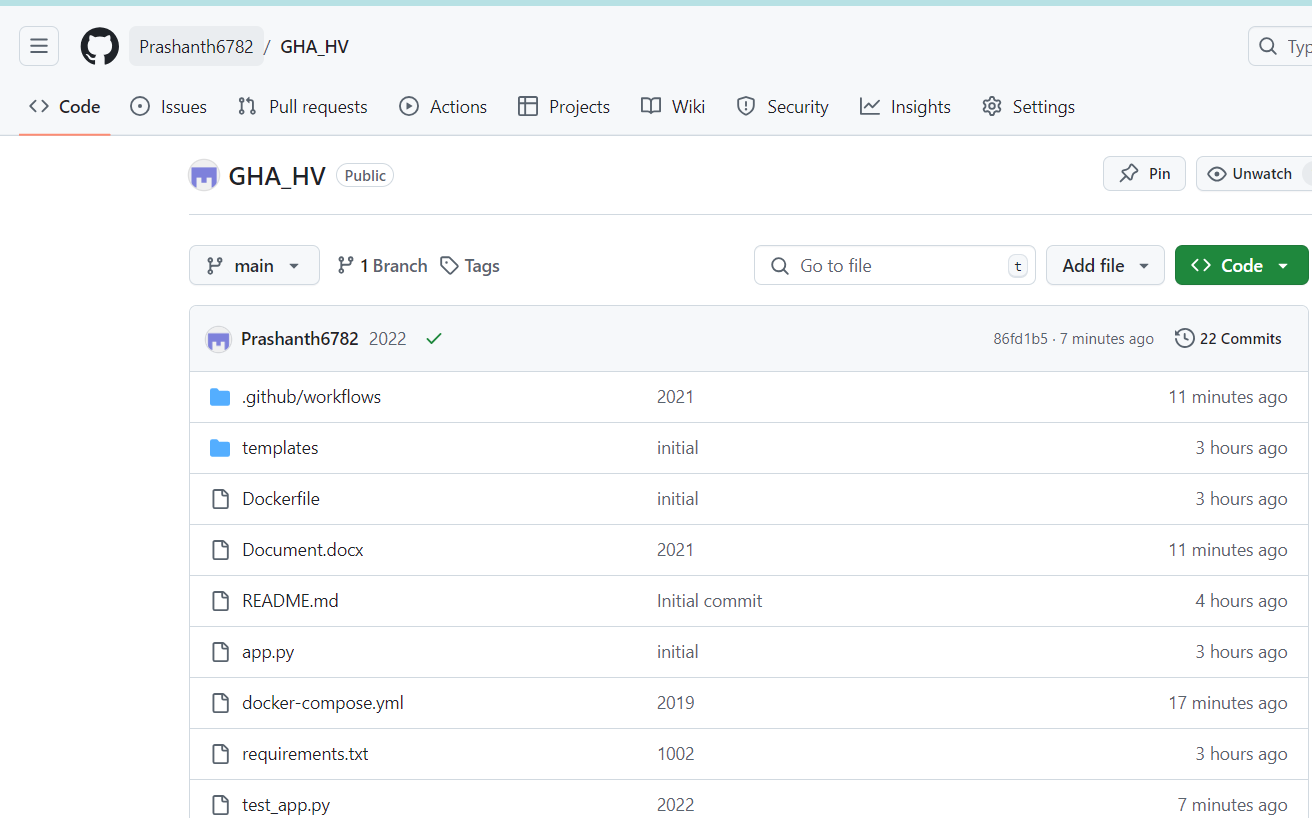
**Requirements:**

1. Setup:

   - Use a provided Python application repository on GitHub (provide a link to a sample Python application repository).

   - Ensure the repository has a main branch and a staging branch.

***Created a main branch***



2. GitHub Actions Workflow:

   - Create a .github/workflows directory in your repository.

   - Inside the directory, create a YAML file to define the workflow.

***main.yml is created already and is available in the github path***

3. Workflow Steps:

     - Define a workflow that performs the following jobs:

     - Install Dependencies: Install all necessary dependencies for the Python application using pip.

     - Run Tests: Execute the test suite using a framework like pytest.

     - Build: If tests pass, prepare the application for deployment.

     - Deploy to Staging: Deploy the application to a staging environment when changes are pushed to the staging branch.

     - Deploy to Production: Deploy the application to production when a release is tagged.

4. Environment Secrets:

   - Use GitHub Secrets to store sensitive information required for deployments (e.g., deployment keys, API tokens).

5. Documentation:

   - Update the README.md file with instructions on how the GitHub Actions workflow works and how to configure the necessary secrets.

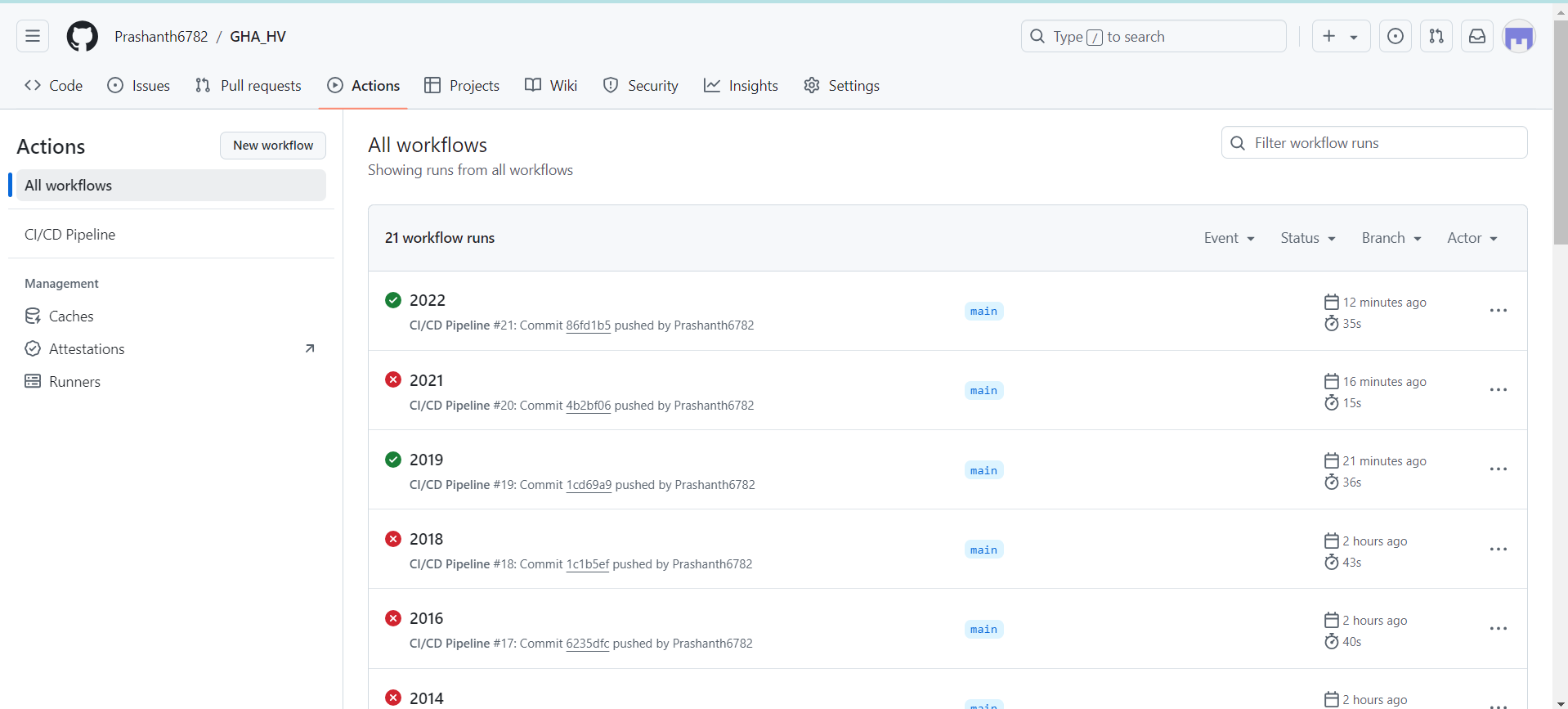
6. Submission:

   - Provide the URL to the GitHub repository with the workflow file and updated README.md.

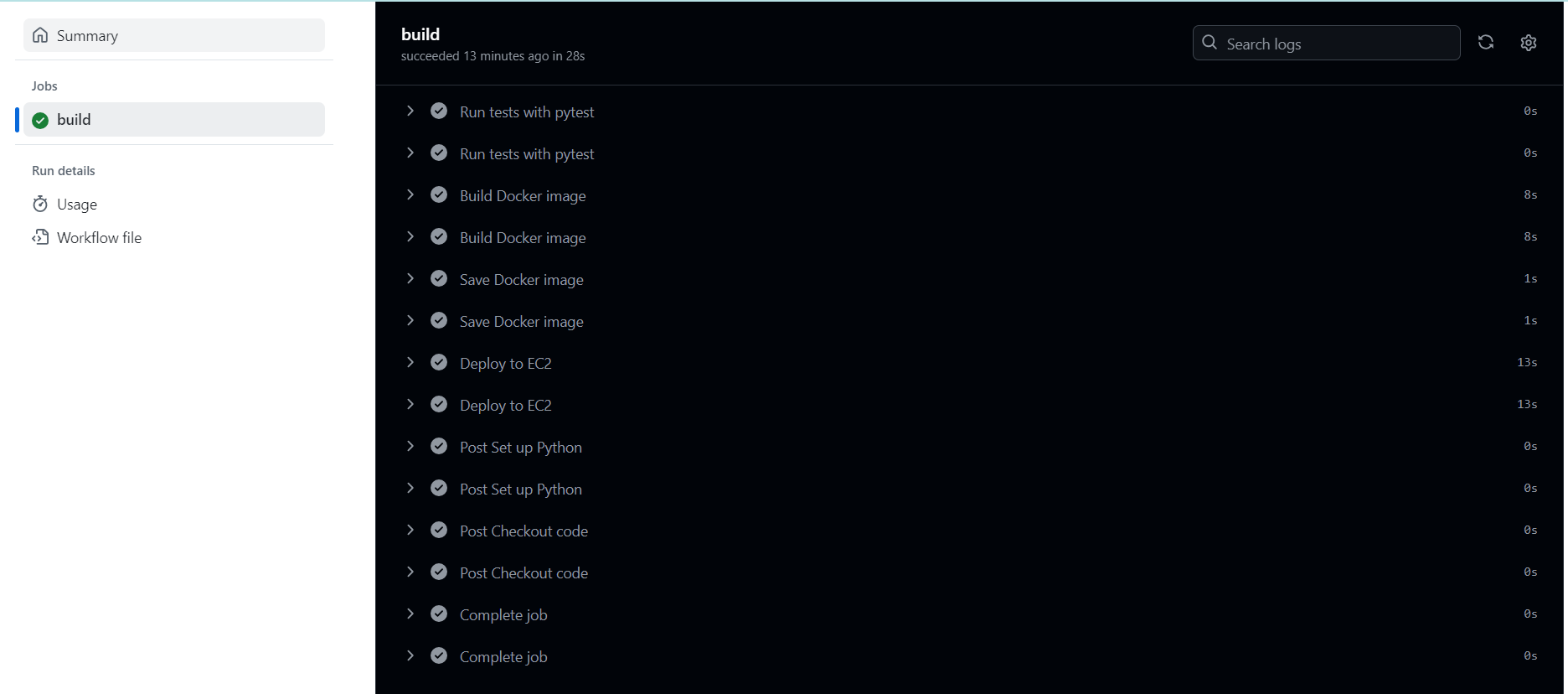
   - Include screenshots of the GitHub Actions workflow runs showing successful execution of all steps.

***The Python/flask app was built and created as a docker image. Further the docker image is pushed to the EC2 instance where it is deployed.***

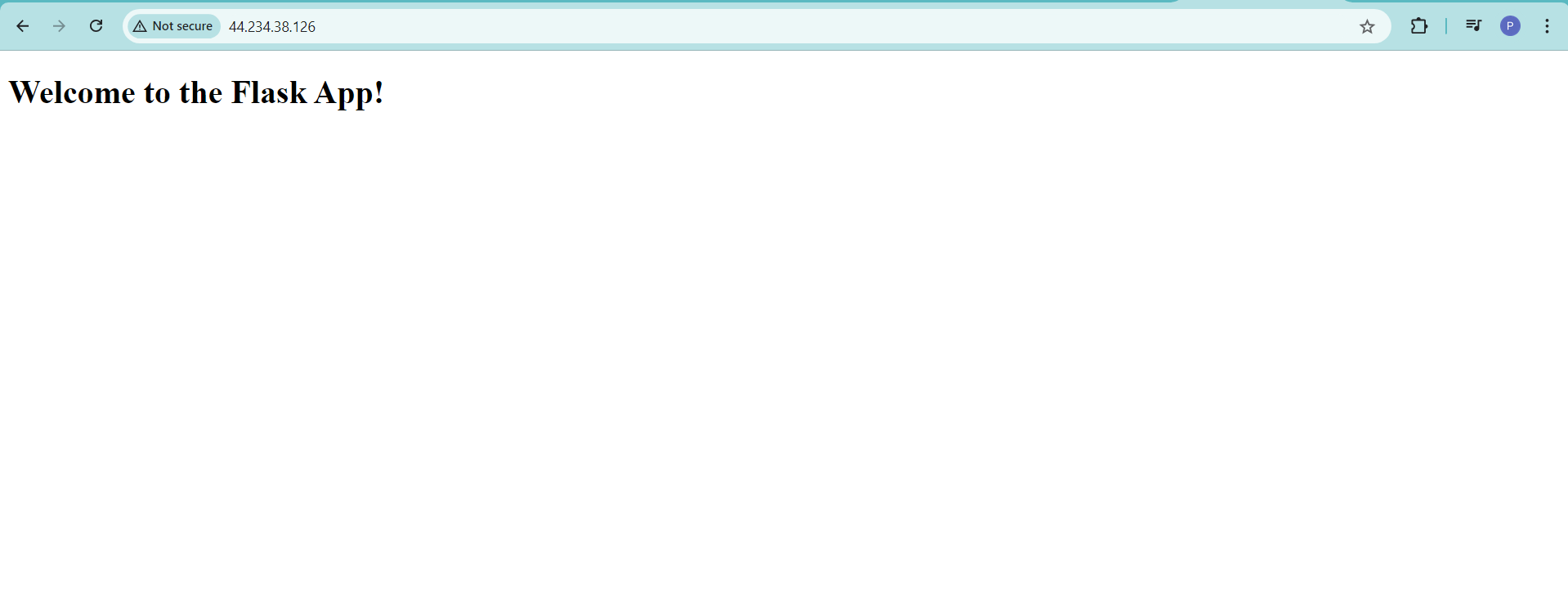
***Please see the screenshots for the workflow which is run***

***2022 workflow is executed successfully when I push the code 🡪 As expected***

***To check whether the application ran successfully in GITHUB action***

******

***Finally im using the browser to view the flask application***

******

***\*\*\*\*Testing is also completed with the PYTEST\*\*\*\*\*\*\*\*\*\*\****