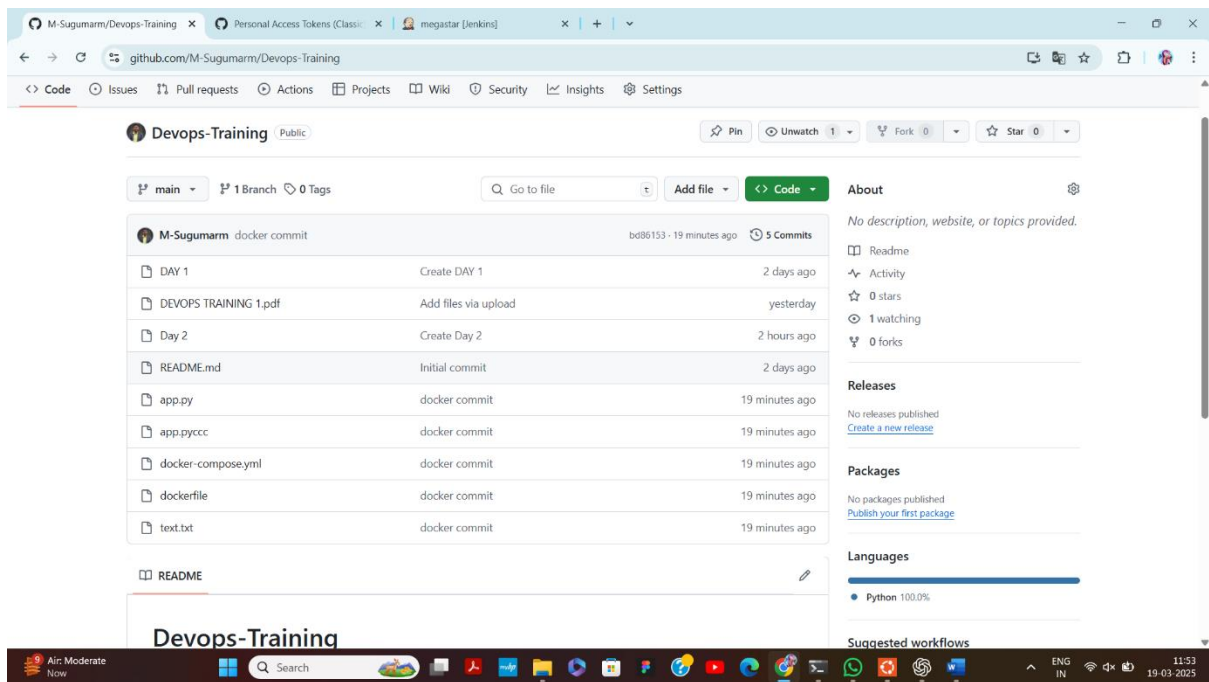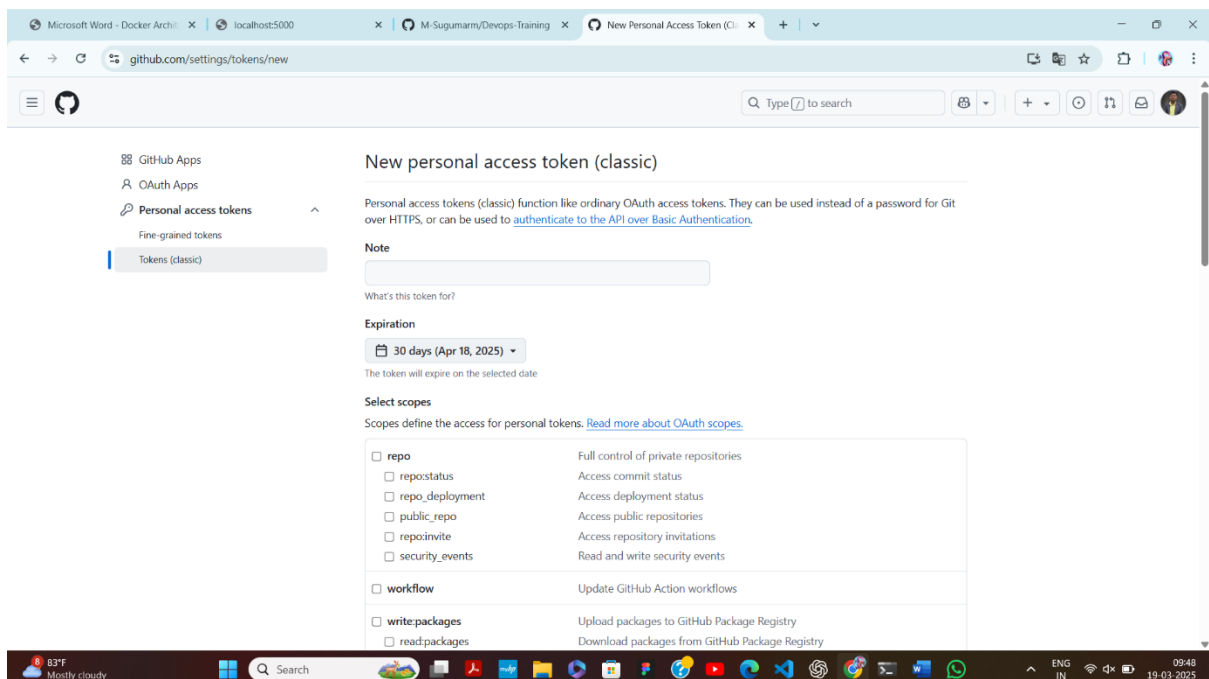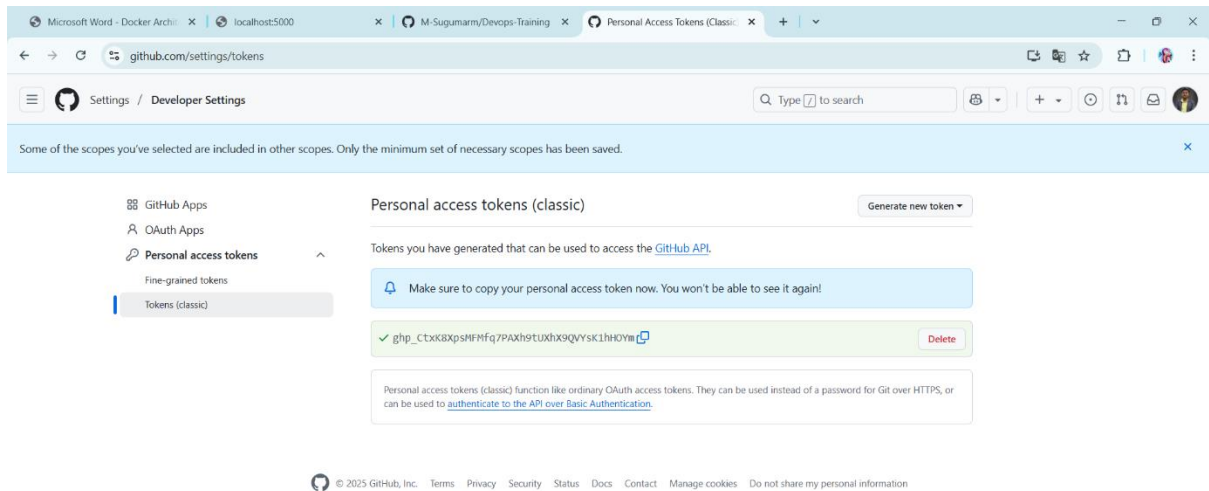# DEVOPS TRAINING

DAY 2

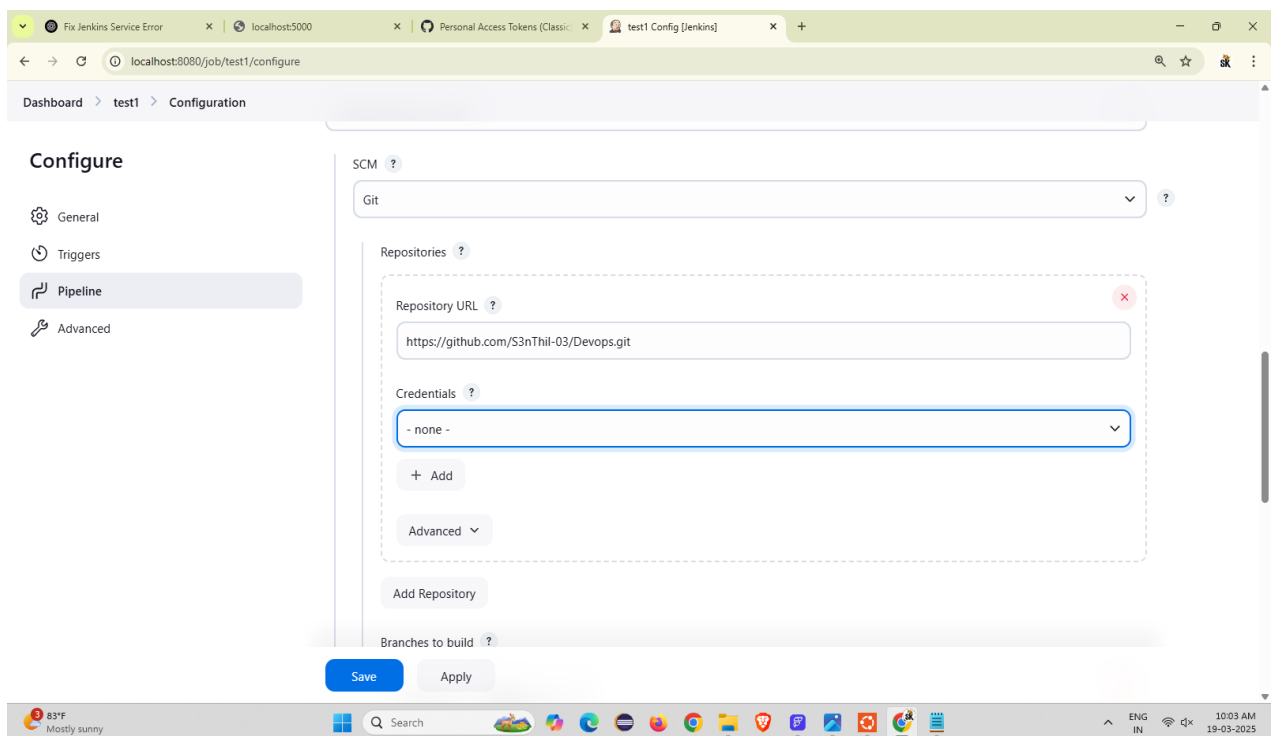## Step 1: create repository in github
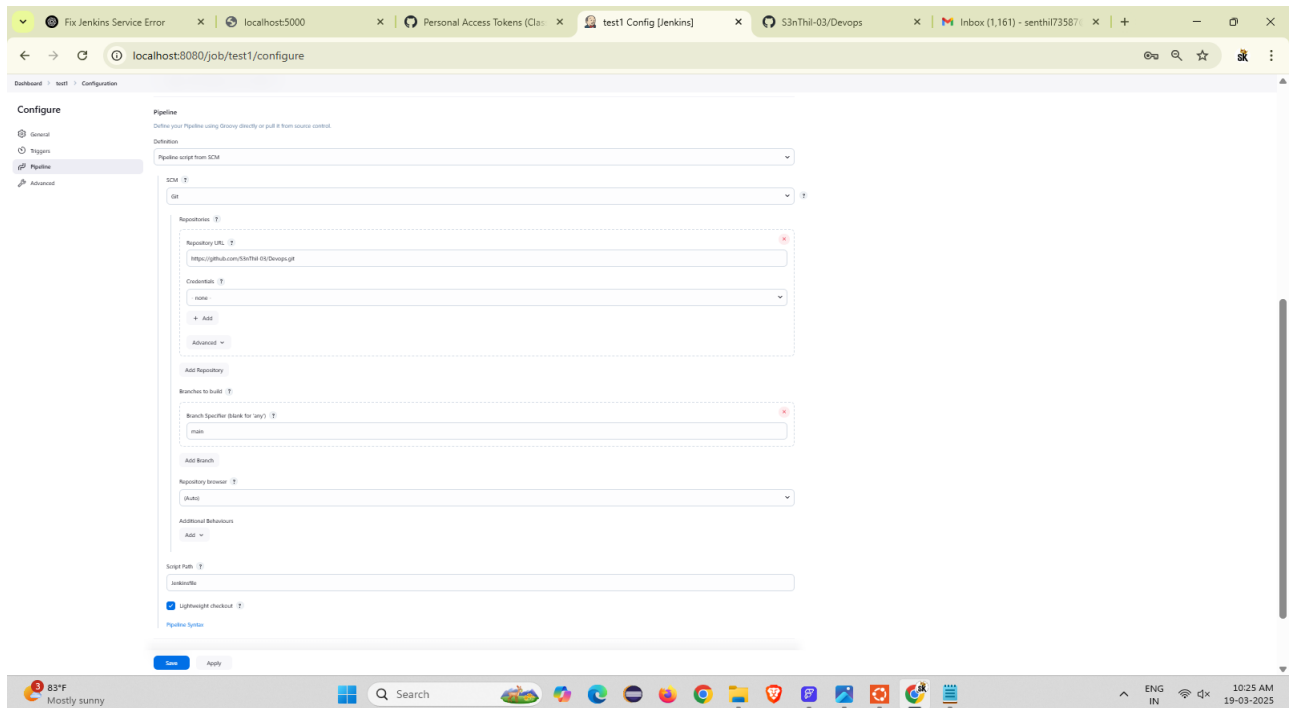

Step 2: go to developer settings

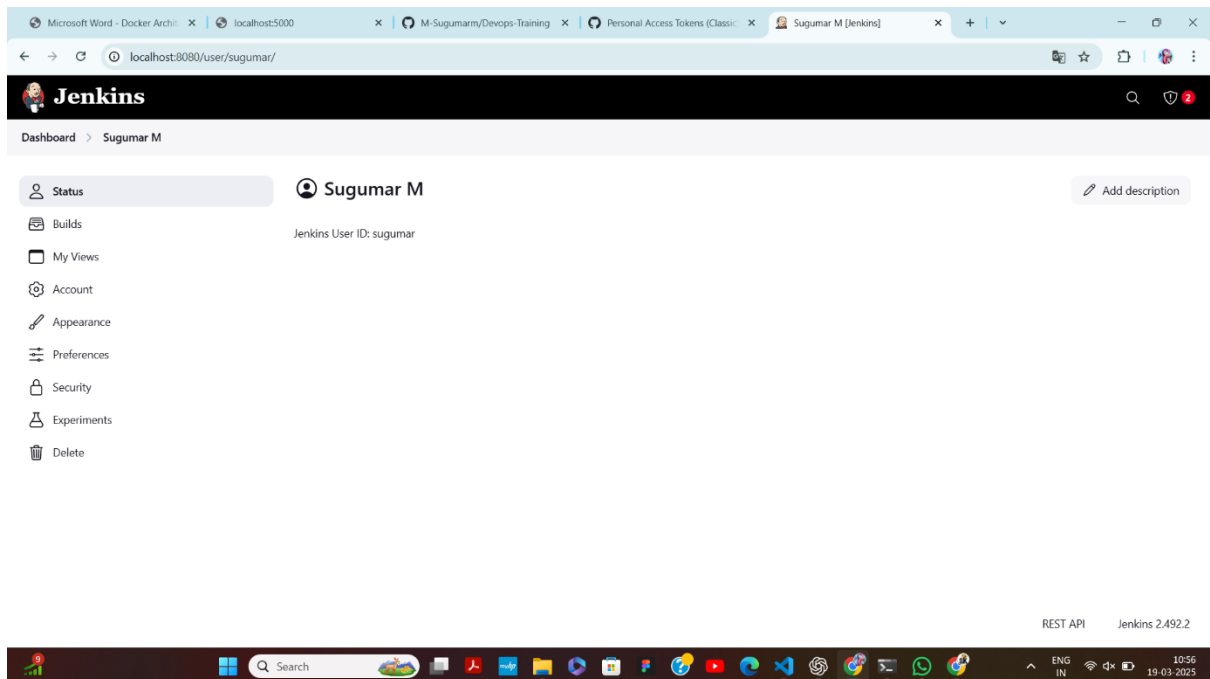## Step 3: generate and copy the token (classic)

Step 4: open Jenkins and create new item and select pipeline in that go to configuration add github repository url into it

Step 5: in Jenkins configure save it

Step 6: verify the status page

Step 7: clone the git repository

```
student@CTS-6:~/docker-python-app$ ls
app.py  docker-compose.yml  dockerfile  requirements.txt
student@CTS-6:~/docker-python-app$ cat docker-compose.yml
version: '3.8'
services:
  web:
    build: .
    ports:
      - "5000:5000"
    volumes:
    - .:/app
    restart: always
student@CTS-6:~/docker-python-app$ sudo systemctl enable jenkins
[sudo] password for student:
Synchronizing state of jenkins.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins
student@CTS-6:~/docker-python-app$ sudo systemctl start jenkins
student@CTS-6:~/docker-python-app$ cat dockerfile
FROM python:3.11
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir flask
COPY . .
EXPOSE 5000
CMD ["python", "app.py"]
student@CTS-6:~/docker-python-app$ ls
app.py  docker-compose.yml  dockerfile  requirements.txt
student@CTS-6:~/docker-python-app$ git clone https://github.com/S3nThil-03/Devops.git
```

Step 8: using cd add the repository into it

```
    - .:/app
    restart: always
student@CTS-6:~/docker-python-app$ sudo systemctl enable jenkins
[sudo] password for student:
Synchronizing state of jenkins.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins
student@CTS-6:~/docker-python-app$ sudo systemctl start jenkins
student@CTS-6:~/docker-python-app$ cat dockerfile
FROM python:3.11
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir flask
COPY . .
EXPOSE 5000
CMD ["python", "app.py"]
student@CTS-6:~/docker-python-app$ ls
app.py  docker-compose.yml  dockerfile  requirements.txt
student@CTS-6:~/docker-python-app$ git clone https://github.com/S3nThil-03/Devops.git
Cloning into 'Devops'...
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 14 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (14/14), 4.84 MiB | 4.13 MiB/s, done.
Resolving deltas: 100% (5/5), done.
student@CTS-6:~/docker-python-app$ ls
Devops  app.py  docker-compose.yml  dockerfile  requirements.txt
student@CTS-6:~/docker-python-app$ mv app.py docker-compose.yml  dockerfile requirements.txt Devops/
student@CTS-6:~/docker-python-app$ ls
Devops
student@CTS-6:~/docker-python-app$ cd Devops/
student@CTS-6:~/docker-python-app/Devops$ ls
'Day 1'  README.md  app.py  docker-compose.yml  dockerfile  requirements.txt
student@CTS-6:~/docker-python-app/Devops$
```

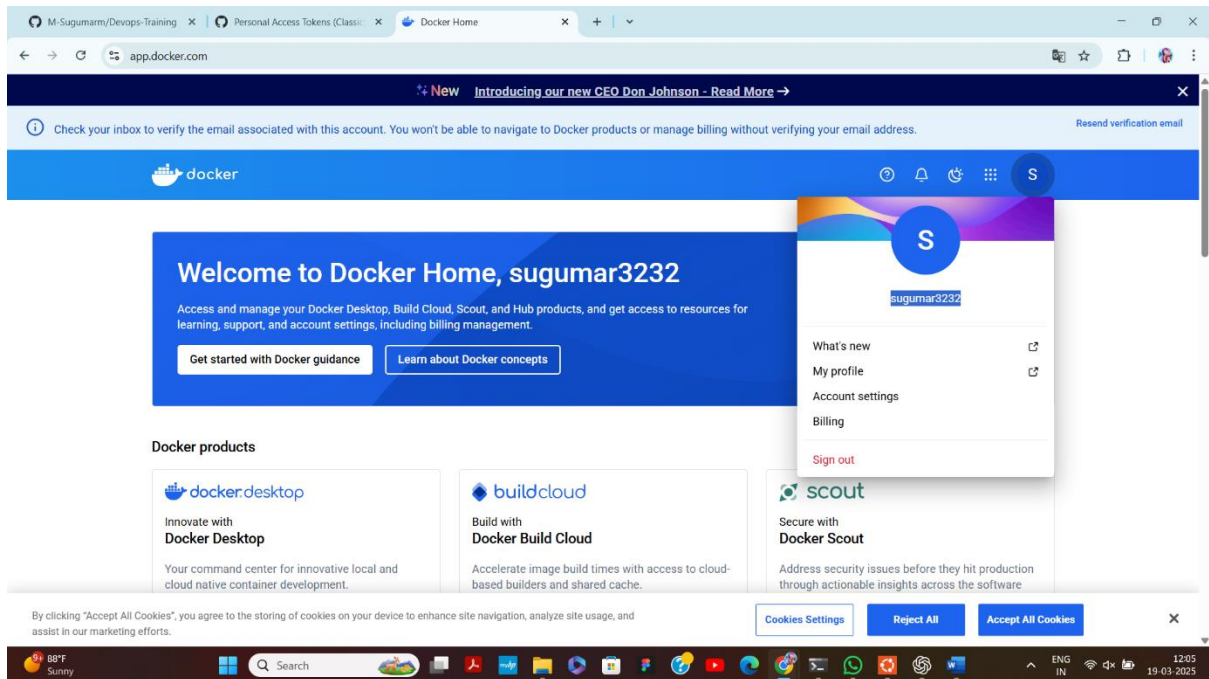## Step 9: using git push command to push all the files into github
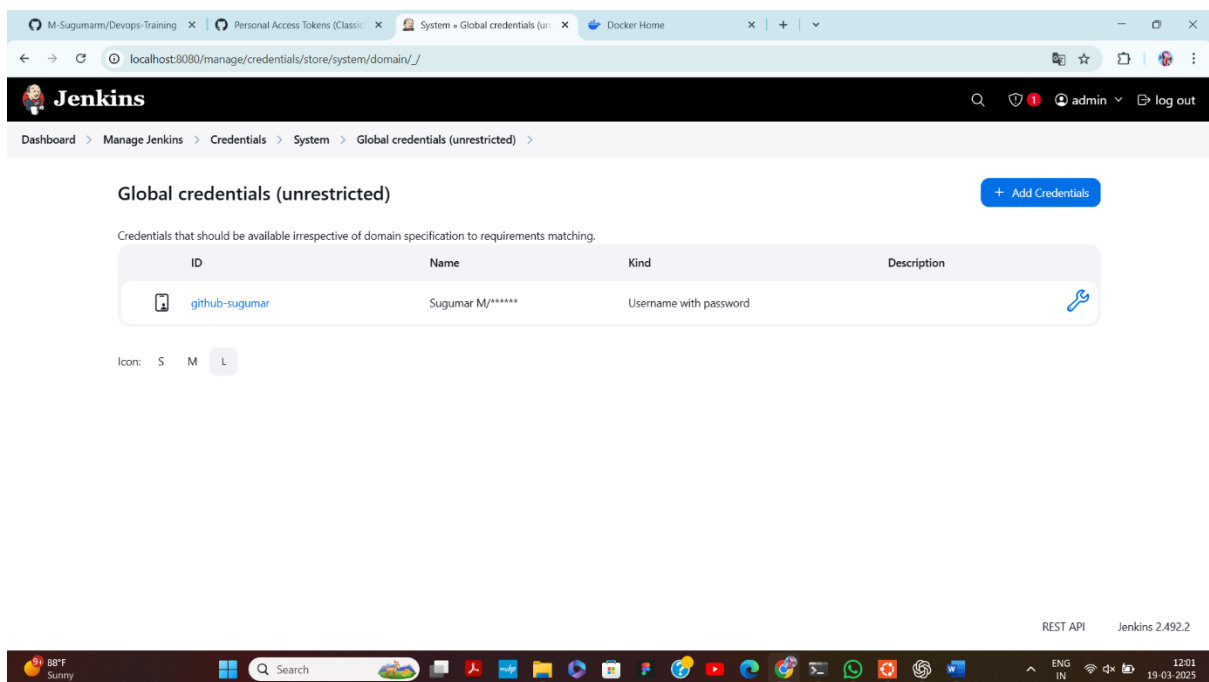

Step 10: check the docker all the files are uploaded in the github repository
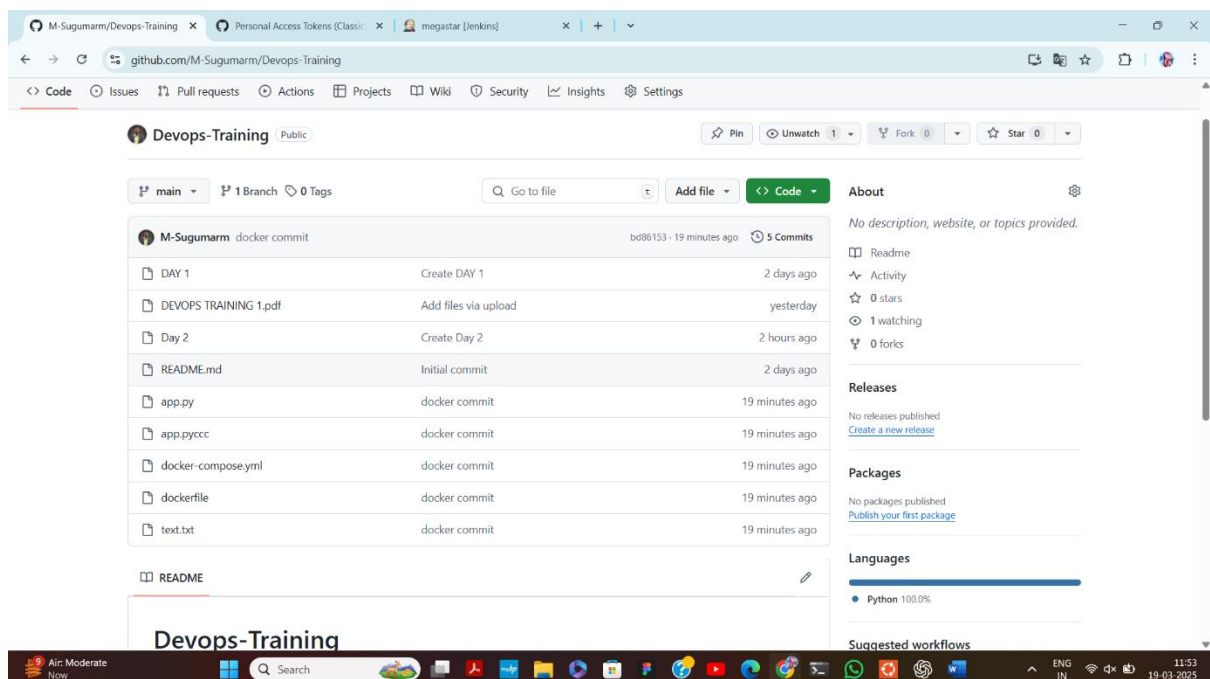
Step 11: go to the docker and login in

Step 12: in jenkins copy the global credentials and change in the jenkins file

Step 13: commit the jenkinsfile into github

Step 14: verify the jenkins file is pushed in the github

Step 15 : using "sudo usermod –aG docker jenkins" and restart the jenkins

Step 16 : build the item and check the output in console output
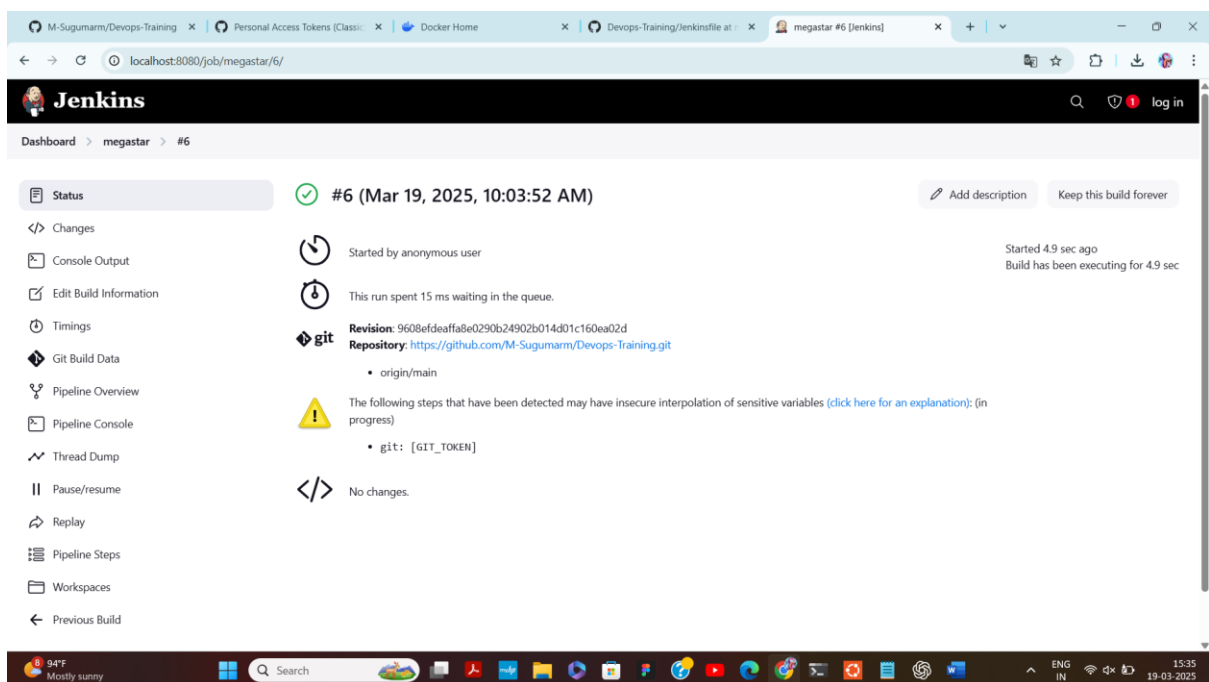
Step 17 : run the localhost:5001

Hello, Flask is running in Docker!
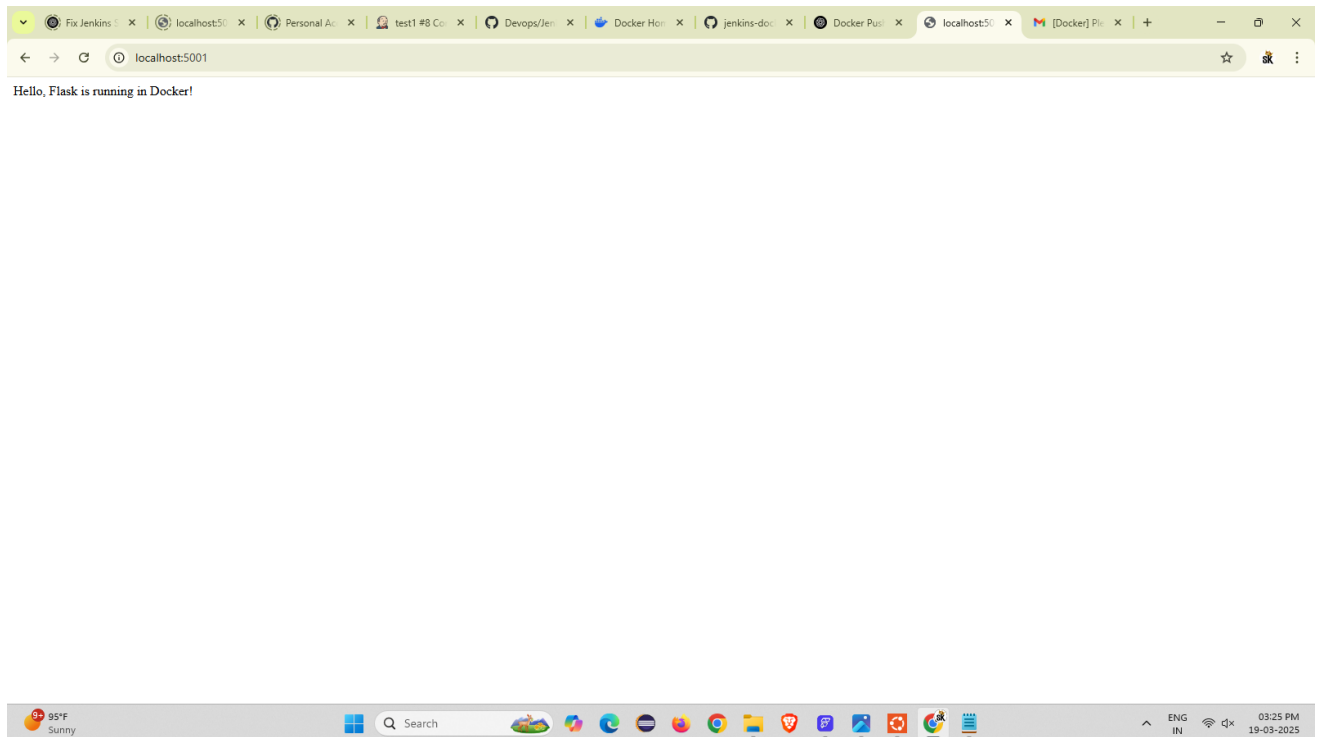
Step 18 : check the image repository in docker