Q. Write code for a simple user registration form for an event.

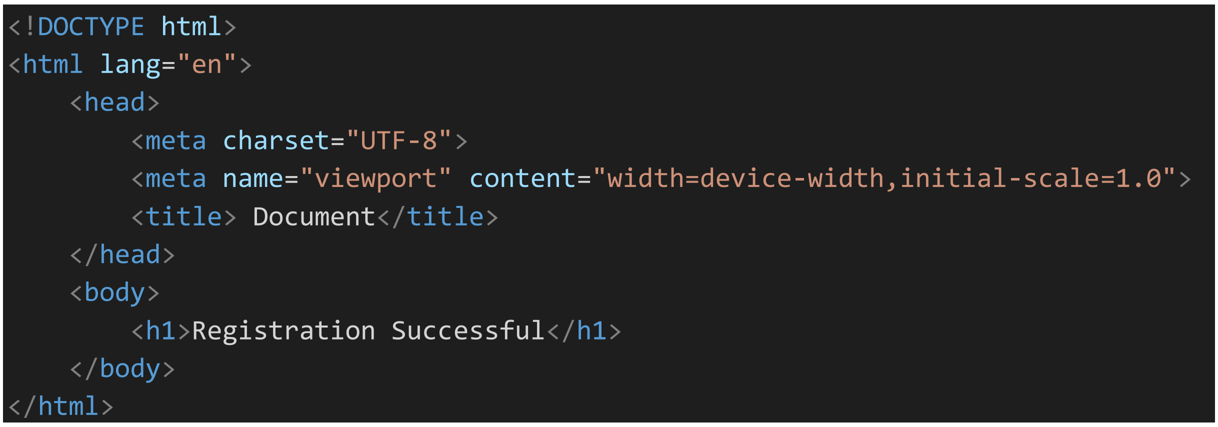
Main.py:

```python
from flask import Flask,render_template,request
app = Flask(__name__)
@app.route('/')
def home():
    return render_template("index.html")
@app.route('/register',methods=['POST'])
def register():
    if request.method=='POST':
        name=request.form['name']
        email=request.form['email']
        password=request.form['password']
        return render_template('success.html')
if __name__=='__main__':
    print('sahya')
    app.run()
```

Index.html:

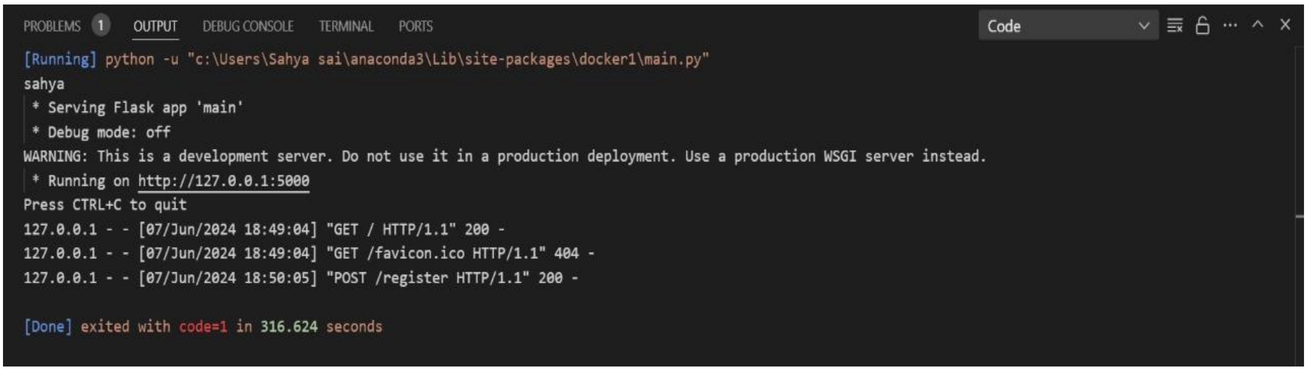
```html
<!index.html>
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport"
        content="width=device-width, initial-scale=1.0">
        <title>Registration Form</title>
    </head>
    <body>
        <h1>Registration Form</h1>
        <form action="/register" method="post">
            <label for="name">Name:</label>
            <input type="text" id="name" name="name" required><br>
            <label for="email">Email:</label>
            <input type="email" id="email"name="email" required><br>
            <label for="password">Password:</label>
            <input type="password" id="password" name="password" required><br>
            <input type="submit" value="Register">
        </form>
    </body>
</html>
```

Success.html:

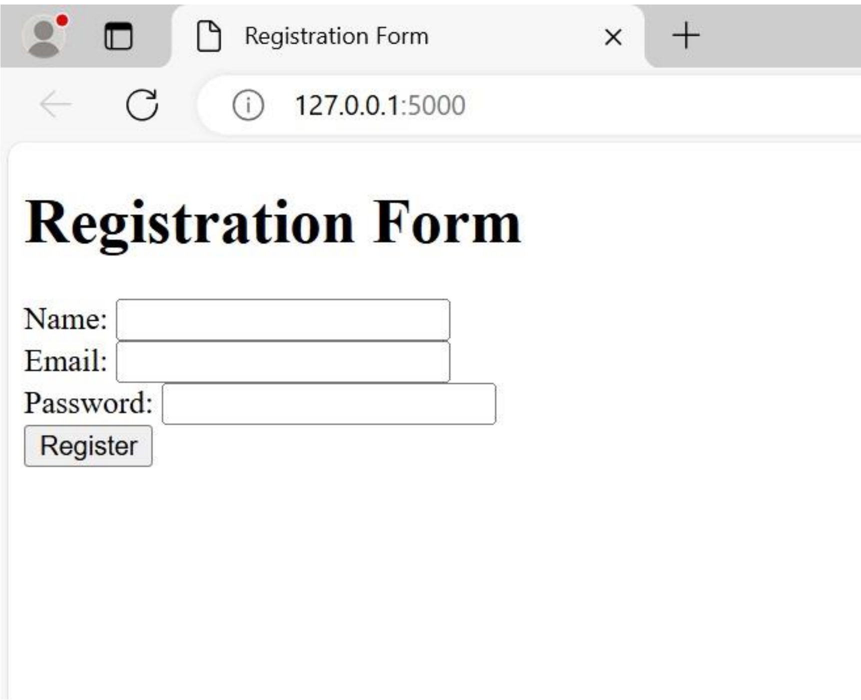


```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,initial-scale=1.0">
        <title> Document</title>
    </head>
    <body>
        <h1>Registration Successful</h1>
    </body>
</html>
```

OUTPUT:

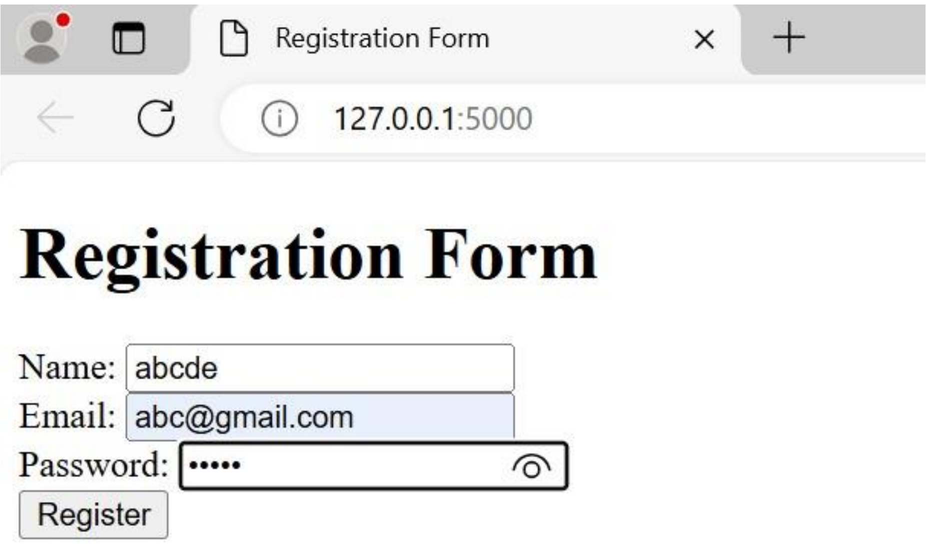


```
[Running] python -u "c:\Users\Sahya sai\anaconda3\Lib\site-packages\docker1\main.py"
sahya
 * Serving Flask app 'main'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [07/Jun/2024 18:49:04] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [07/Jun/2024 18:49:04] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [07/Jun/2024 18:50:05] "POST /register HTTP/1.1" 200 -

[Done] exited with code=1 in 316.624 seconds
```



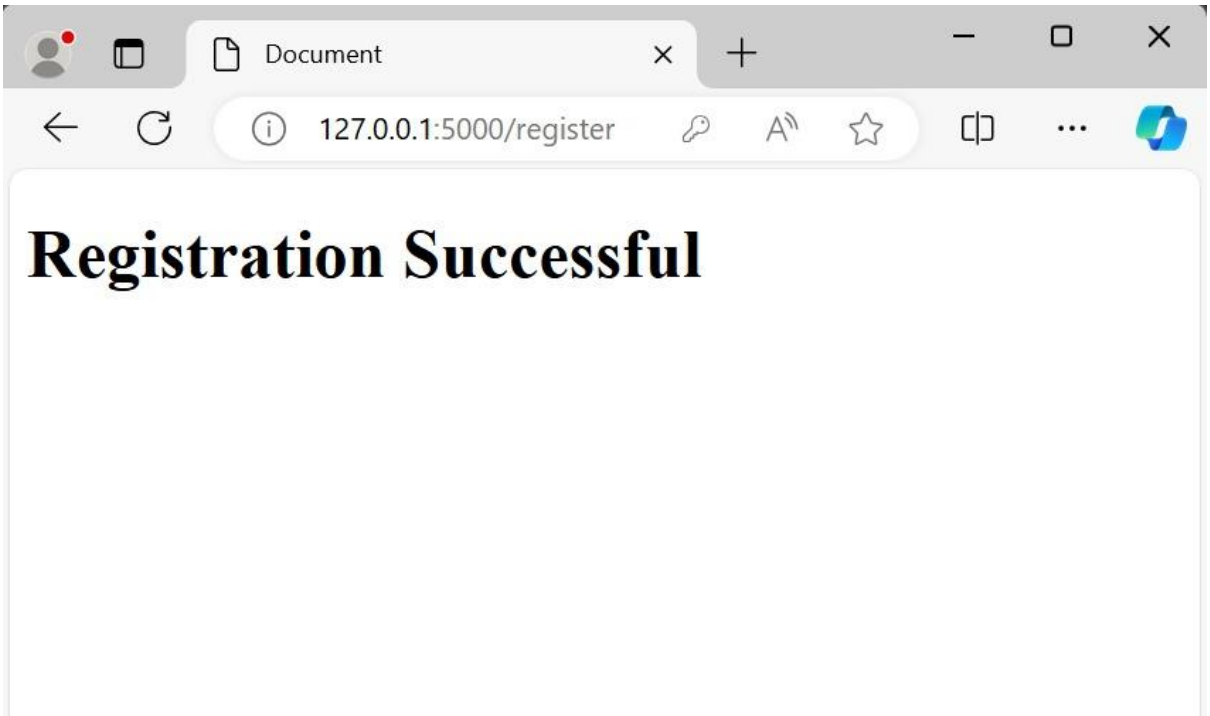
Registration Form

Name:
Email:
Password:
Register



Registration Form

Name: abcde
Email: abc@gmail.com
Password: •••••



Registration Successful

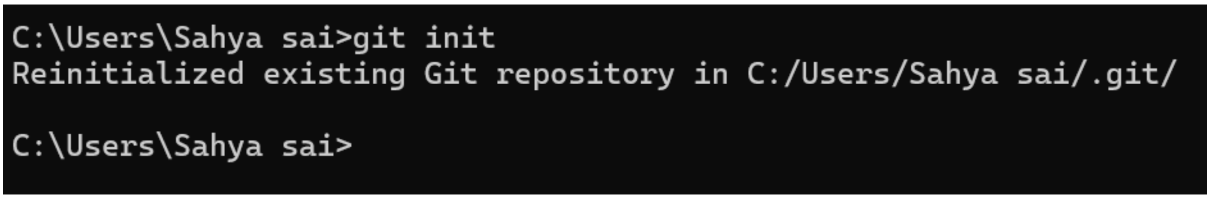
Q. Q. Explore Git and GitHub commands.

Git and GitHub are two of the most popular tools used for version control and collaboration in software development.

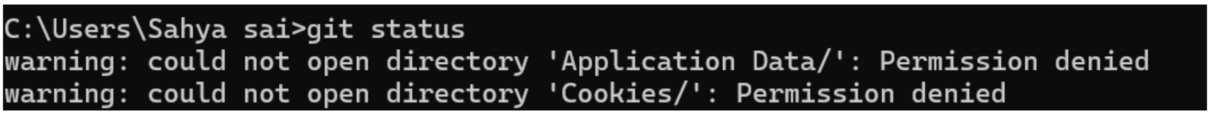
Here are some common Git and GitHub commands.

- Initializing a git repository: $git init
- Checking the status of your repository: $ git status
- Adding files to the stage: $ git add
- Committing changes: $ git commit -m "commit message"
- Checking the commit history: $ git log
- Undoing changes: $ git checkout
- Creating a new branch: $ git branch
- Switching to a different branch: $ git checkout
- Merging two branches: $ git merge
- Pushing changes to a remote repository: $ git push origin
- Cloning a repository from GitHub: $ git clone
- Creating a pull request on GitHub: Go to the repository on GitHub, select the branch you want to merge and click the "New pull request" button.

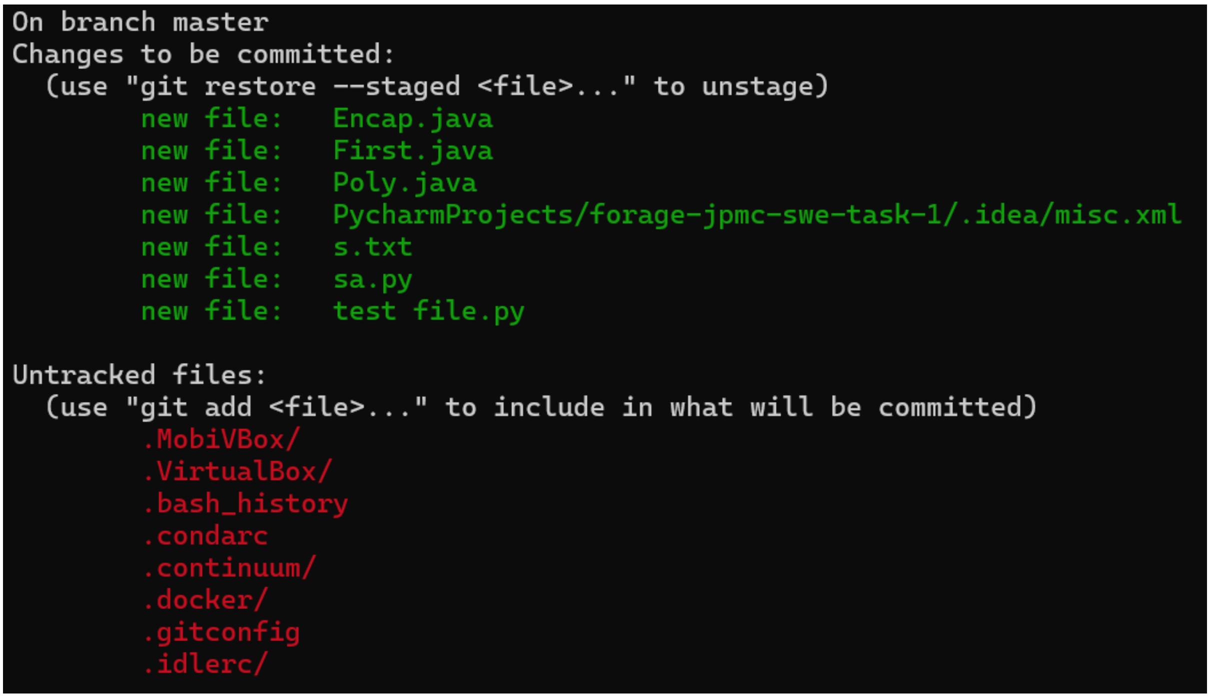
These are just a few of the many Git and GitHub commands available. There are many other Git commands and functionalities that you can explore to suit your needs.



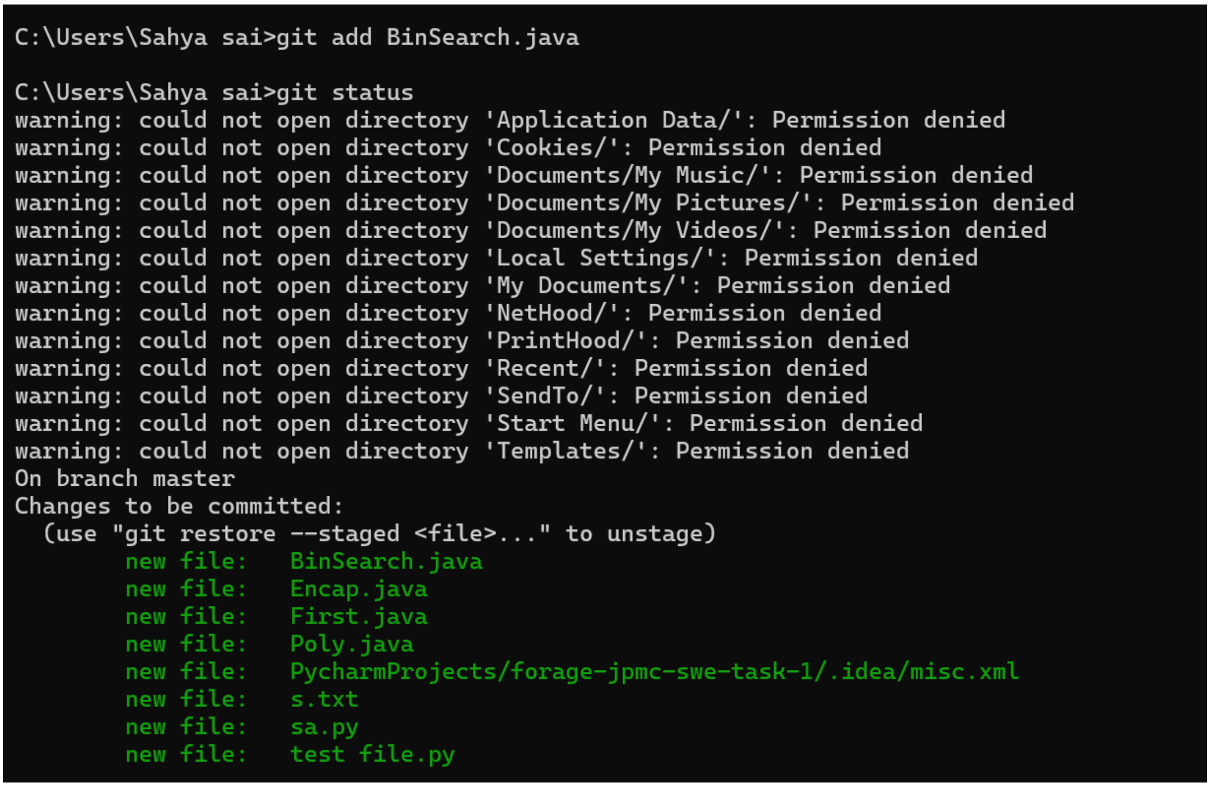
```
C:\Users\Sahya sai>git init
Reinitialized existing Git repository in C:/Users/Sahya sai/.git/

C:\Users\Sahya sai>
```



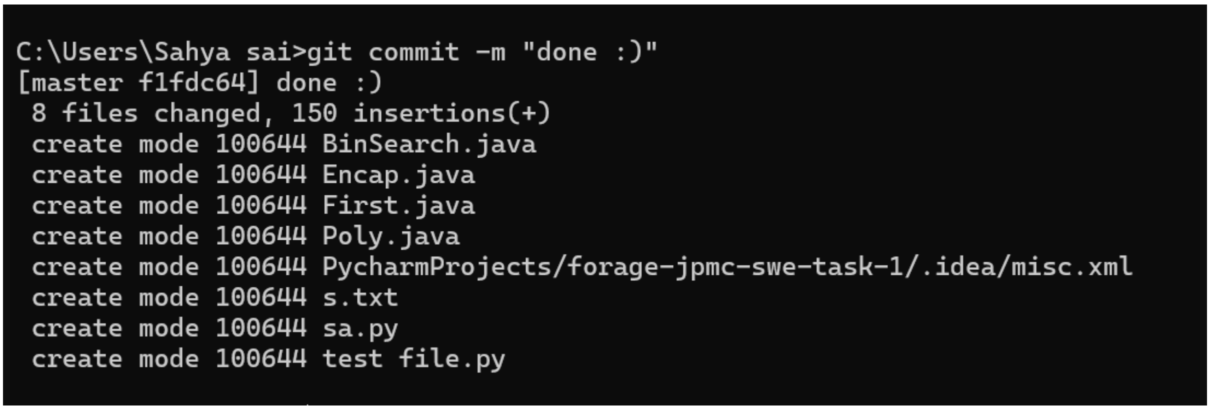
```
C:\Users\Sahya sai>git status
warning: could not open directory 'Application Data/': Permission denied
warning: could not open directory 'Cookies/': Permission denied
```



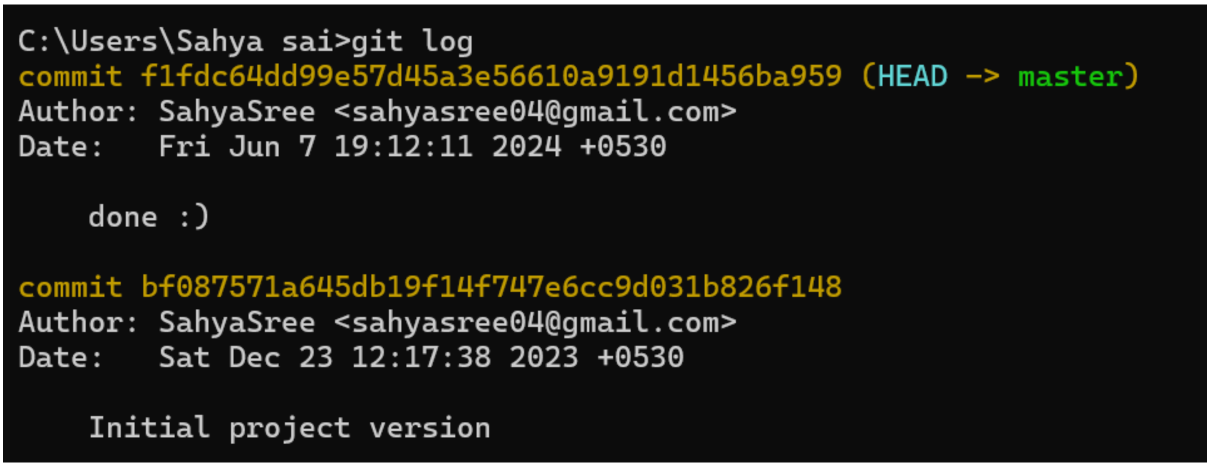
```
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Encap.java
        new file:   First.java
        new file:   Poly.java
        new file:   PycharmProjects/forage-jpmc-swe-task-1/.idea/misc.xml
        new file:   s.txt
        new file:   sa.py
        new file:   test file.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .MobiVBox/
        .VirtualBox/
        .bash_history
        .condarc
        .continuum/
        .docker/
        .gitconfig
        .idlerc/
```



```
C:\Users\Sahya sai>git add BinSearch.java

C:\Users\Sahya sai>git status
warning: could not open directory 'Application Data/': Permission denied
warning: could not open directory 'Cookies/': Permission denied
warning: could not open directory 'Documents/My Music/': Permission denied
warning: could not open directory 'Documents/My Pictures/': Permission denied
warning: could not open directory 'Documents/My Videos/': Permission denied
warning: could not open directory 'Local Settings/': Permission denied
warning: could not open directory 'My Documents/': Permission denied
warning: could not open directory 'NetHood/': Permission denied
warning: could not open directory 'PrintHood/': Permission denied
warning: could not open directory 'Recent/': Permission denied
warning: could not open directory 'SendTo/': Permission denied
warning: could not open directory 'Start Menu/': Permission denied
warning: could not open directory 'Templates/': Permission denied
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   BinSearch.java
        new file:   Encap.java
        new file:   First.java
        new file:   Poly.java
        new file:   PycharmProjects/forage-jpmc-swe-task-1/.idea/misc.xml
        new file:   s.txt
        new file:   sa.py
        new file:   test file.py
```



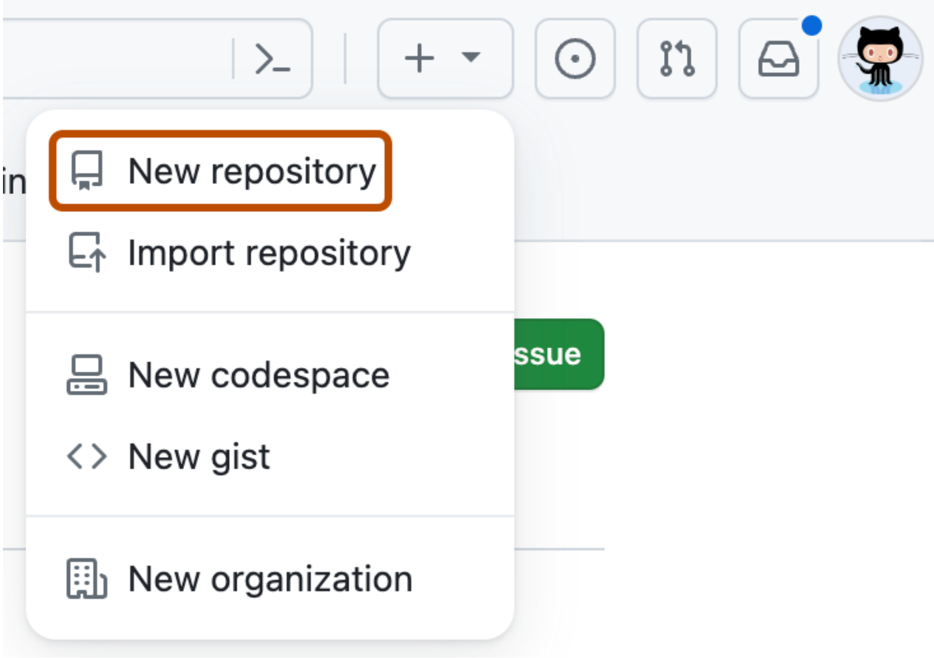
```
C:\Users\Sahya sai>git commit -m "done :)"
[master f1fdc64] done :)
 8 files changed, 150 insertions(+)
 create mode 100644 BinSearch.java
 create mode 100644 Encap.java
 create mode 100644 First.java
 create mode 100644 Poly.java
 create mode 100644 PycharmProjects/forage-jpmc-swe-task-1/.idea/misc.xml
 create mode 100644 s.txt
 create mode 100644 sa.py
 create mode 100644 test file.py
```

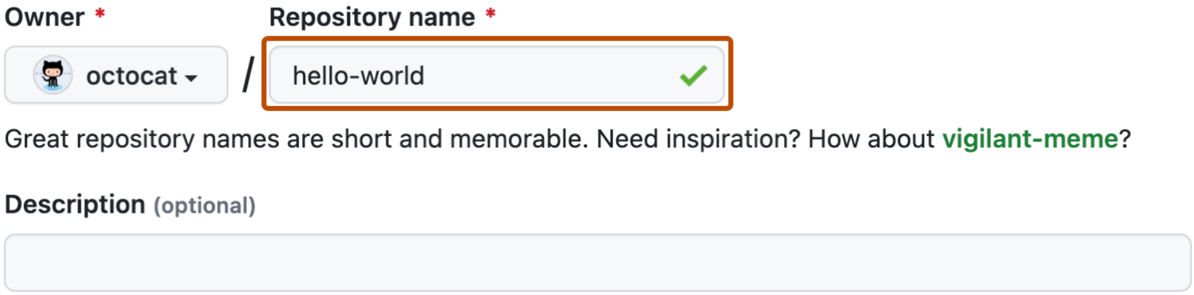


```
C:\Users\Sahya sai>git log
commit f1fdc64dd99e57d45a3e56610a9191d1456ba959 (HEAD -> master)
Author: SahyaSree <sahyasree04@gmail.com>
Date:   Fri Jun 7 19:12:11 2024 +0530

    done :)

commit bf087571a645db19f14f747e6cc9d031b826f148
Author: SahyaSree <sahyasree04@gmail.com>
Date:   Sat Dec 23 12:17:38 2023 +0530

    Initial project version
```

Q. Practice Source code management on GitHub.

- Create new repository on GitHub





Owner *

octocat

/

Repository name *

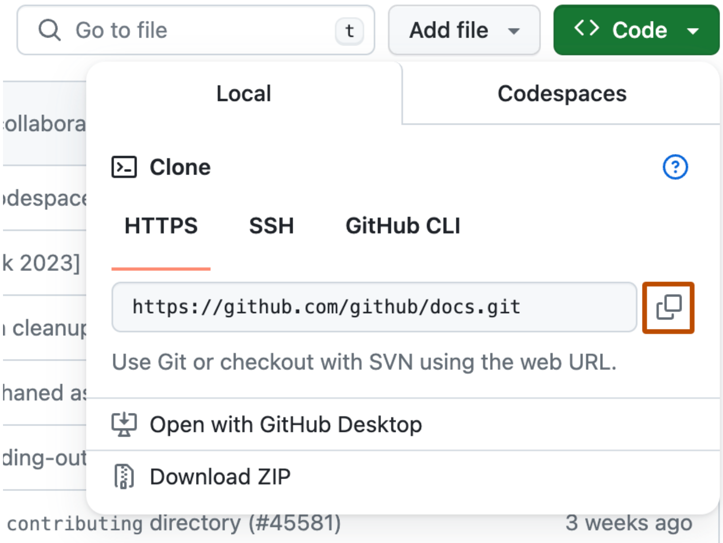
hello-world

Great repository names are short and memorable. Need inspiration? How about vigilant-meme?

Description (optional)

1. Optionally, add a description of your repository. For example, "My first repository on GitHub."
2. Choose a repository visibility. (Public or Private)
3. Select **Initialize this repository with a README**.
4. Click **Create repository**.

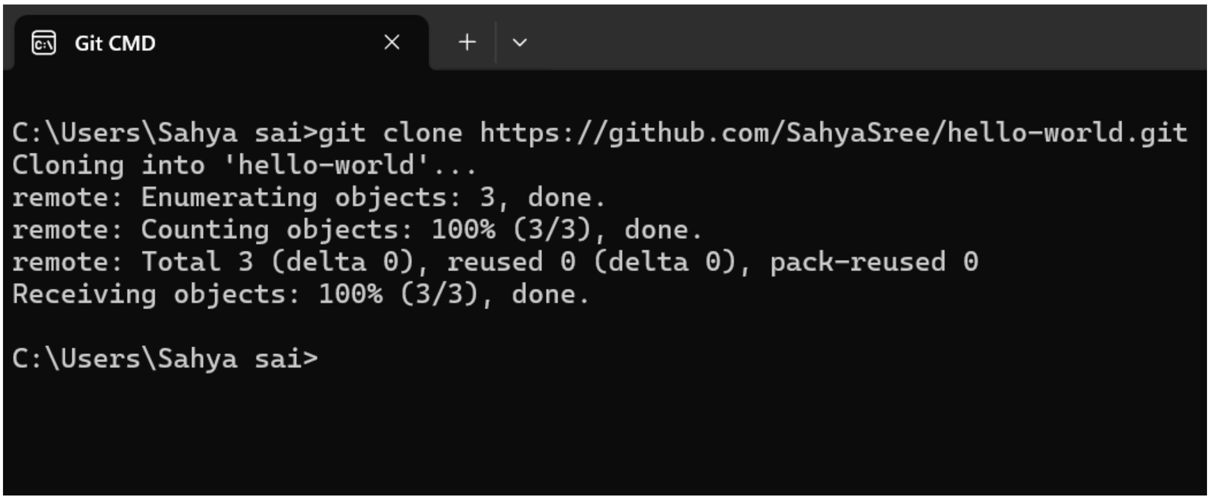
- Clone the repository to your local machine: $ git clone <repository-url>
    1. Copy the URL of the repository to be cloned.



```
https://github.com/github/docs.git
```

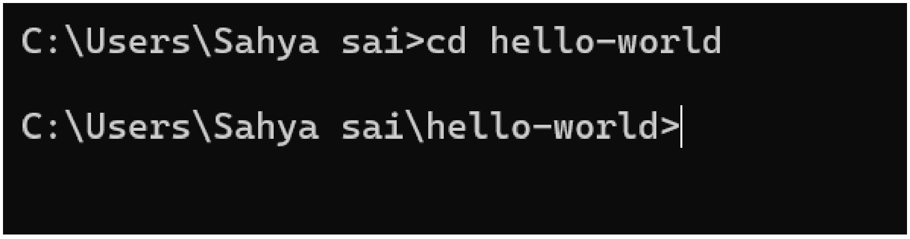
Use Git or checkout with SVN using the web URL.

2. Open Git Bash.
3. Change the current working directory to the location where you want the cloned directory.
4. Type `git clone`, and then paste the URL you copied earlier.



```
C:\Users\Sahya sai>git clone https://github.com/SahyaSree/hello-world.git
Cloning into 'hello-world'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

C:\Users\Sahya sai>
```

- Move to repository directory: $ cd <repository-name>

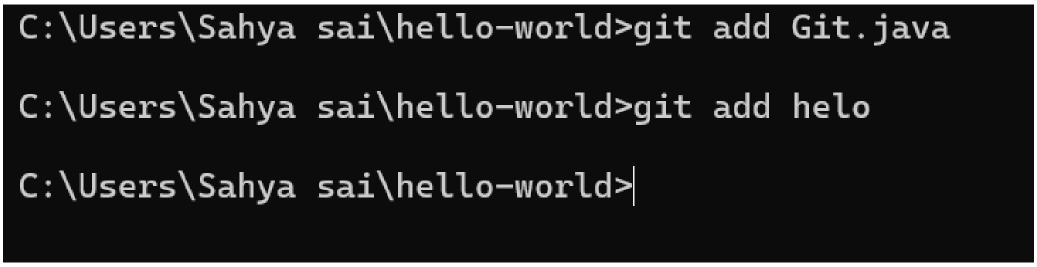


```
C:\Users\Sahya sai>cd hello-world

C:\Users\Sahya sai\hello-world>
```

-

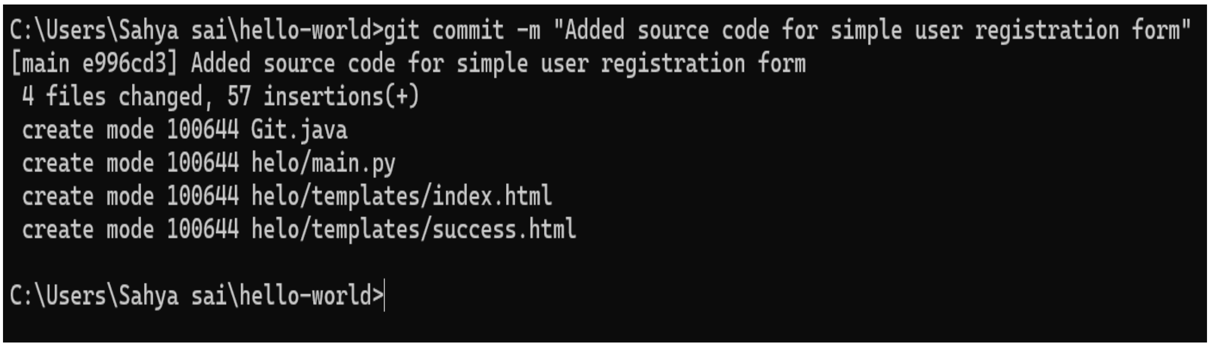
- Create a new file in the repository and add the source code written for user registration form.

- Stage the changes: $ git add <file-name>



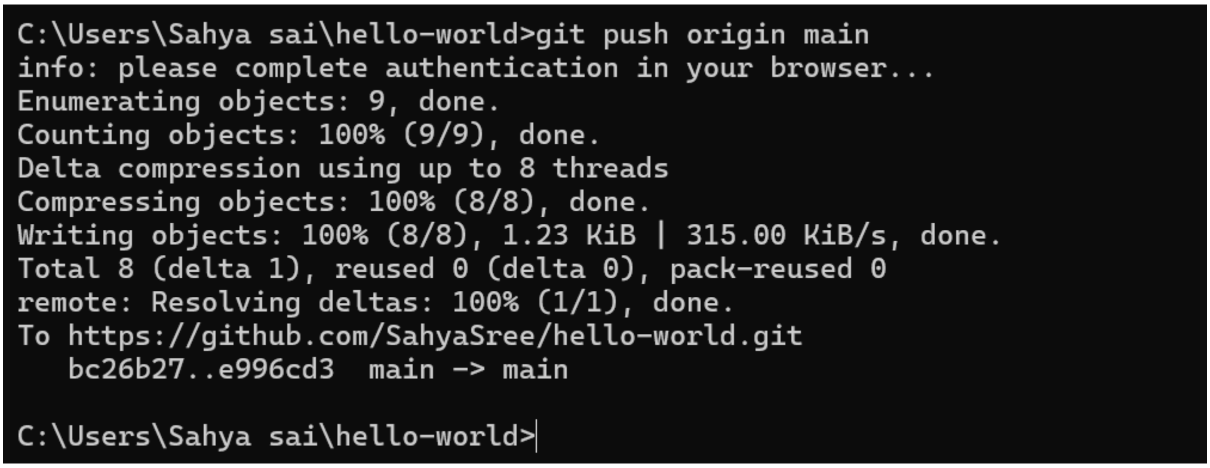
```
C:\Users\Sahya sai\hello-world>git add Git.java

C:\Users\Sahya sai\hello-world>git add helo

C:\Users\Sahya sai\hello-world>
```

- Commit the changes: $ git commit -m “commit message”



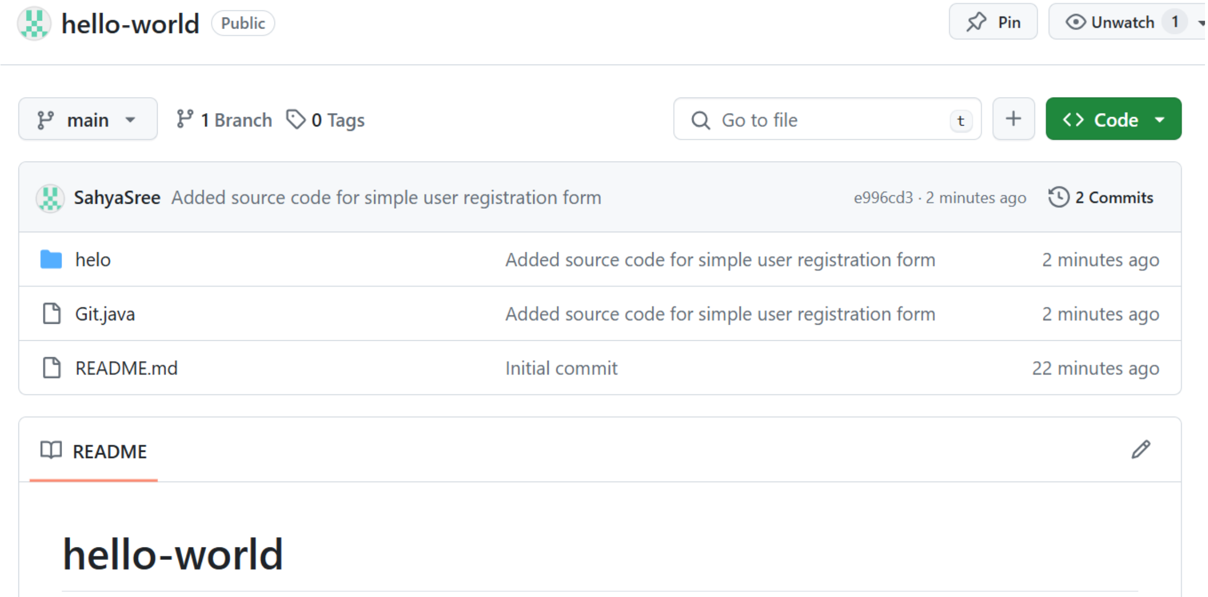
```
C:\Users\Sahya sai\hello-world>git commit -m "Added source code for simple user registration form"
[main e996cd3] Added source code for simple user registration form
 4 files changed, 57 insertions(+)
 create mode 100644 Git.java
 create mode 100644 helo/main.py
 create mode 100644 helo/templates/index.html
 create mode 100644 helo/templates/success.html

C:\Users\Sahya sai\hello-world>
```

- Push the changes to the remote repository: $ git push origin main



```
C:\Users\Sahya sai\hello-world>git push origin main
info: please complete authentication in your browser...
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 1.23 KiB | 315.00 KiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/SahyaSree/hello-world.git
   bc26b27..e996cd3  main -> main

C:\Users\Sahya sai\hello-world>
```

- Verify that the changes are reflected in the repository on GitHub.

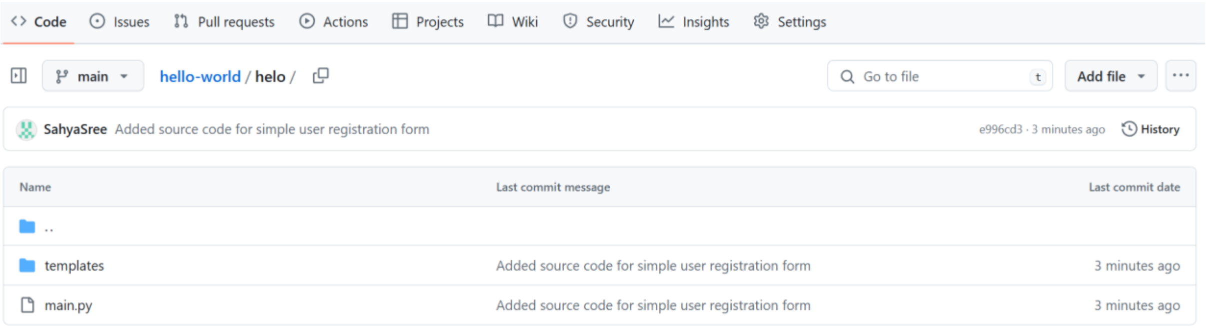


SahyaSree Added source code for simple user registration form e996cd3 · 2 minutes ago 2 Commits

helo	Added source code for simple user registration form	2 minutes ago
Git.java	Added source code for simple user registration form	2 minutes ago
README.md	Initial commit	22 minutes ago

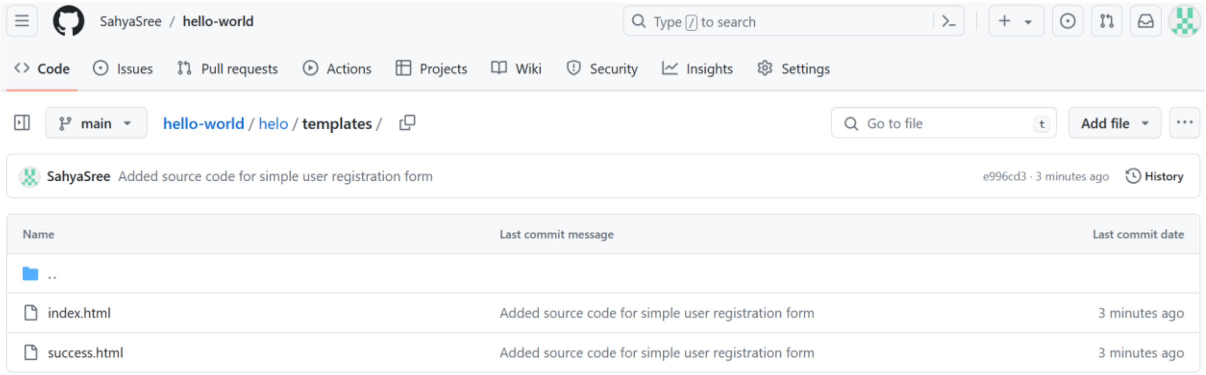
README

hello-world



SahyaSree Added source code for simple user registration form e996cd3 · 3 minutes ago History

Name	Last commit message	Last commit date
..		
templates	Added source code for simple user registration form	3 minutes ago
main.py	Added source code for simple user registration form	3 minutes ago



SahyaSree Added source code for simple user registration form e996cd3 · 3 minutes ago History

Name	Last commit message	Last commit date
..		
index.html	Added source code for simple user registration form	3 minutes ago
success.html	Added source code for simple user registration form	3 minutes ago

These steps demonstrate how to use GitHub for source code management. You can use the same steps to manage any source code projects on GitHub. Additionally, you can also explore GitHub features such as pull requests, code review, and branch management to enhance your source code management workflow.

Q. Jenkins installation and setup, explore the environment.

Install java jdk-21

Set environment varaiable for JDK

Download and install Jenkins

Run Jenkins on local host http://localhost:6969/

Username :admin

Password:5cfe93da2d2a444ebb1485b86c2c95ed

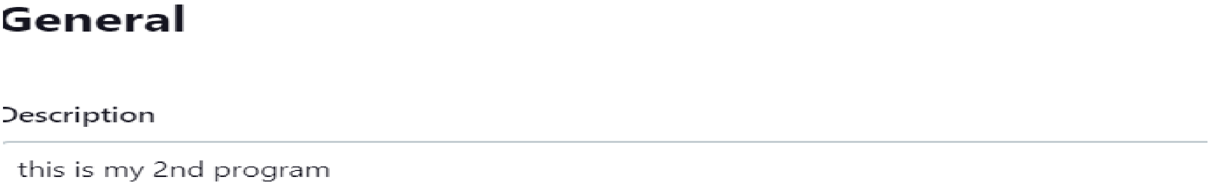
Note:it is different from user to user you have set up this after installation

Steps to run simple python program:

Go to dashboard->new item,then enter an item name choose free style project

Click on ok

Then open general->description(eg:this is my first program)



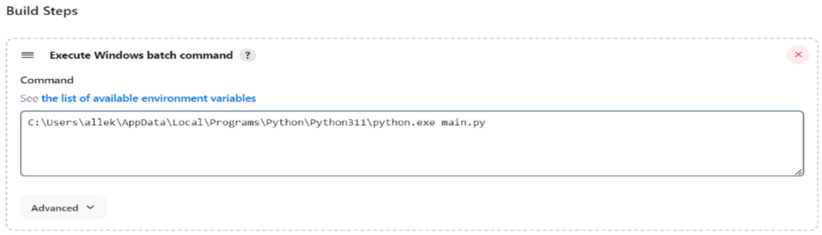
General

Description

this is my 2nd program

Advanced->use custom workspace->enter the path where your python file has saved eg: C:\Users\allek\Downloads\jenkins programs

Go to build steps ->execute windows batch command (you can get get this path from environment variables)



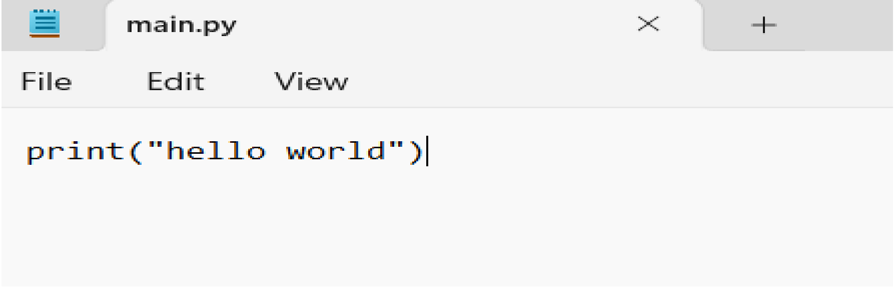
Build Steps

Execute Windows batch command

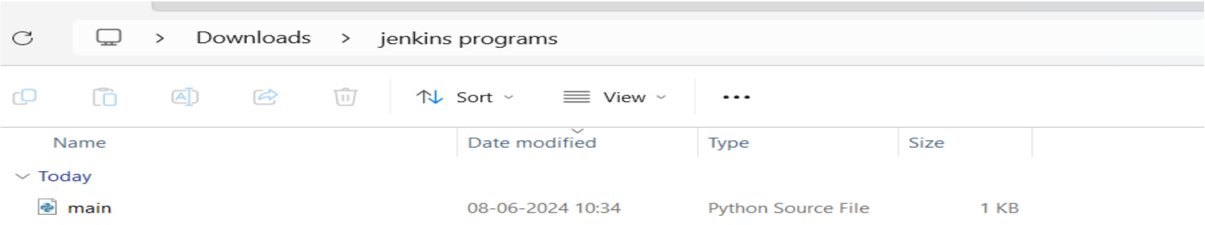
Command

See the list of available environment variables

```
C:\Users\allek\AppData\Local\Programs\Python\Python311\python.exe main.py
```



```
print("hello world")
```

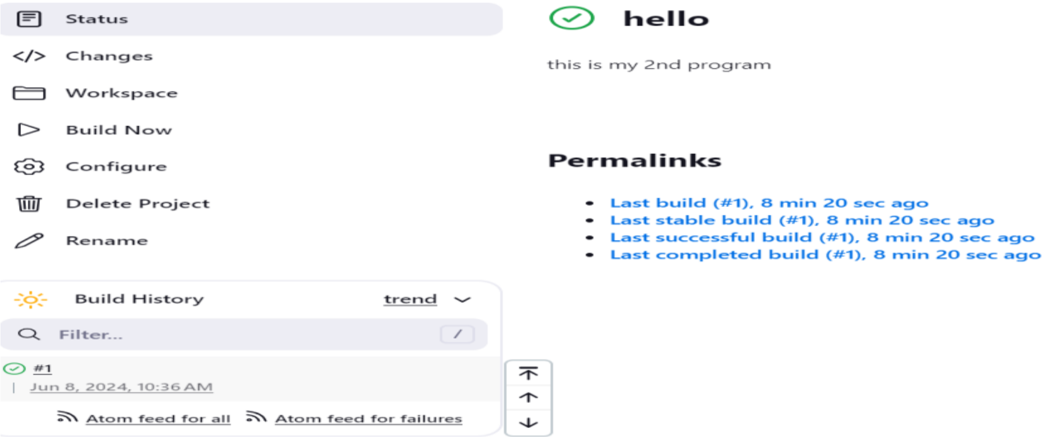


Name	Date modified	Type	Size
Today			
main	08-06-2024 10:34	Python Source File	1 KB

Then click on save

Go to build now,after this you can find build history open link(with date&time) and click on console output

Output will be displayed



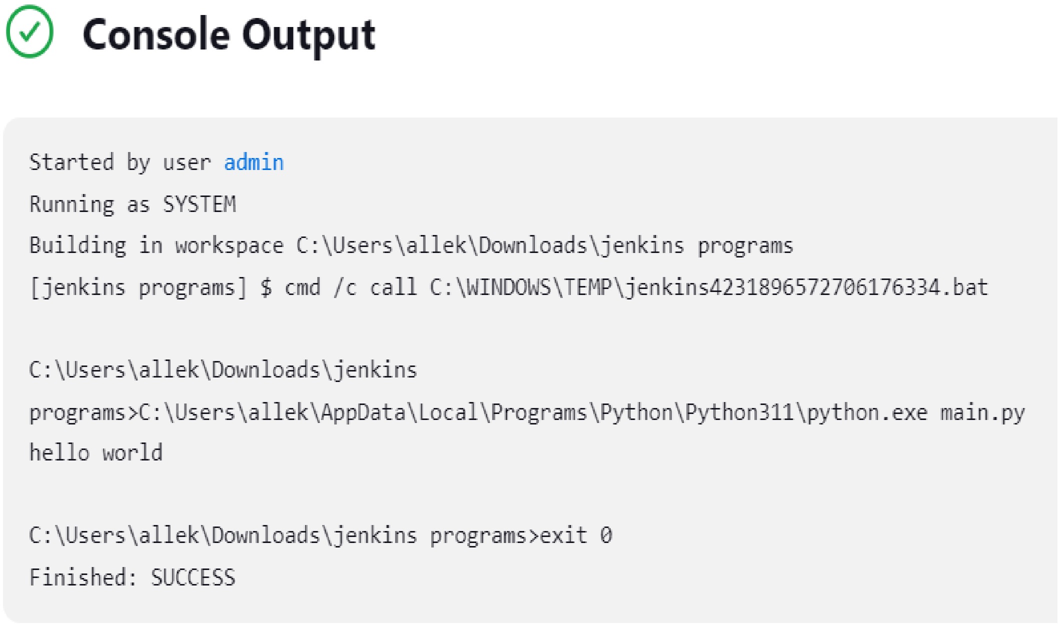
hello

this is my 2nd program

Permalinks

- Last build (#1), 8 min 20 sec ago
- Last stable build (#1), 8 min 20 sec ago
- Last successful build (#1), 8 min 20 sec ago
- Last completed build (#1), 8 min 20 sec ago

#1 Jun 8, 2024, 10:36 AM



Console Output

```
Started by user admin
Running as SYSTEM
Building in workspace C:\Users\allek\Downloads\jenkins programs
[jenkins programs] $ cmd /c call C:\WINDOWS\TEMP\jenkins4231896572706176334.bat

C:\Users\allek\Downloads\jenkins
programs>C:\Users\allek\AppData\Local\Programs\Python\Python311\python.exe main.py
hello world

C:\Users\allek\Downloads\jenkins programs>exit 0
Finished: SUCCESS
```

Q. Demonstrate continuous integration and development using Jenkins.(build pipeline)

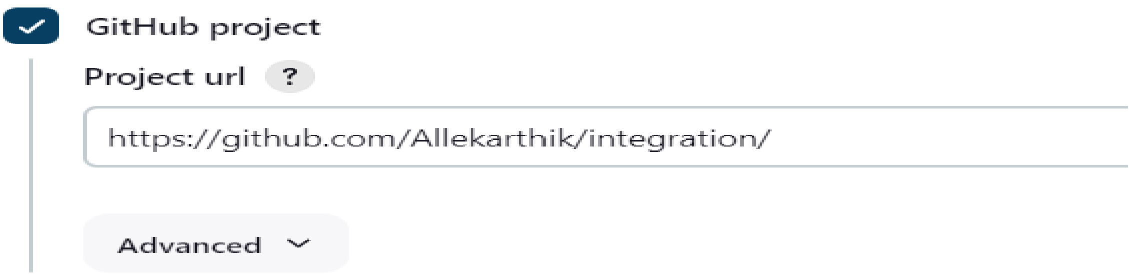
Project url :github.com/Allekarthik/integration

Repository url :https://github.com/Allekarthik/integration.git

For integration of Jenkins with github first of all we need to create a repository   in that place any file lets say eg:index.html

Then create new item ->general ->add any description

In github project add project url

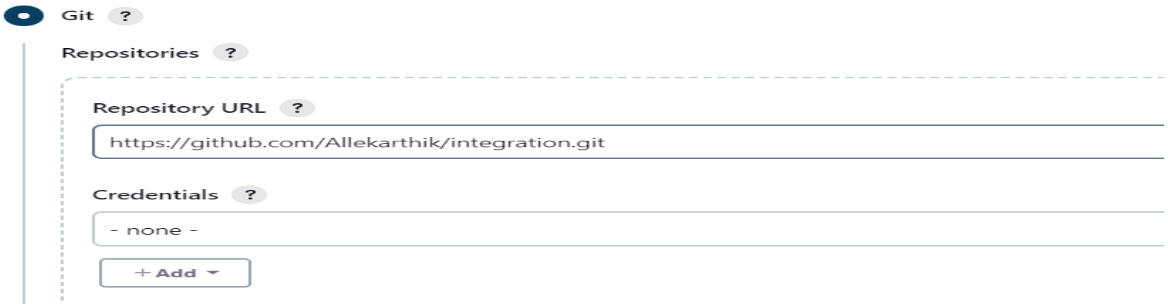


☑ GitHub project

Project url ?

https://github.com/Allekarthik/integration/

Then in git add repository url



● Git ?

Repositories ?

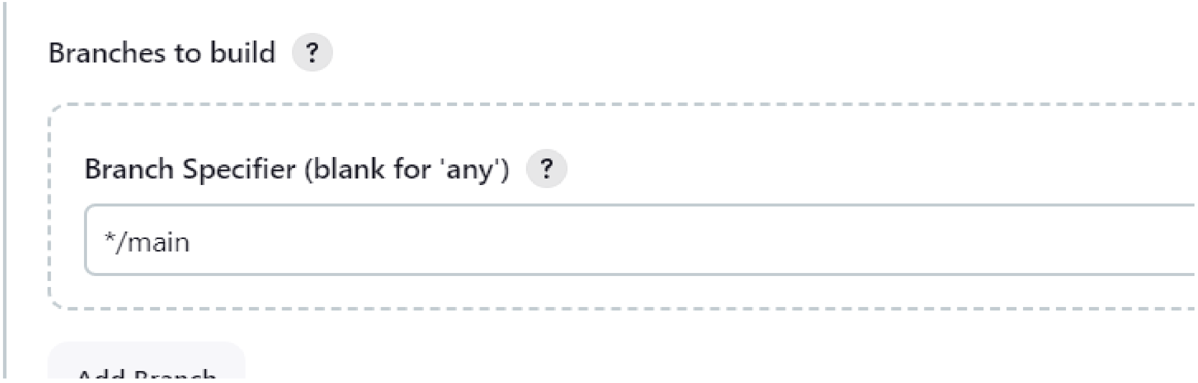
Repository URL ?

https://github.com/Allekarthik/integration.git

Credentials ?

- none -

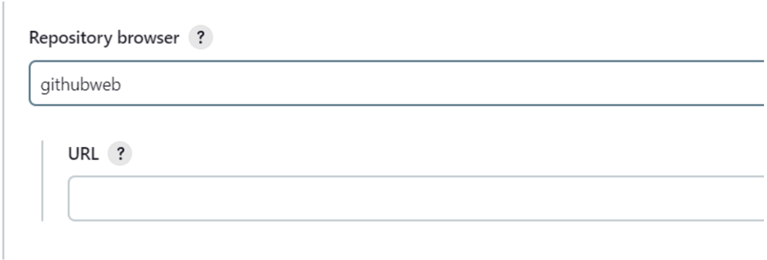
Then select main because my github is stored under main



Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

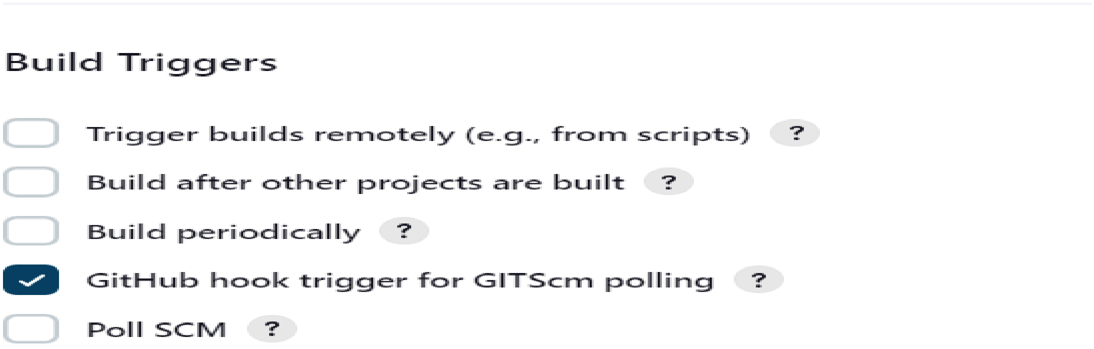


Repository browser ?

githubweb

URL ?

Then select repository browser as githubweb



Build Triggers

□ Trigger builds remotely (e.g., from scripts) ?

□ Build after other projects are built ?

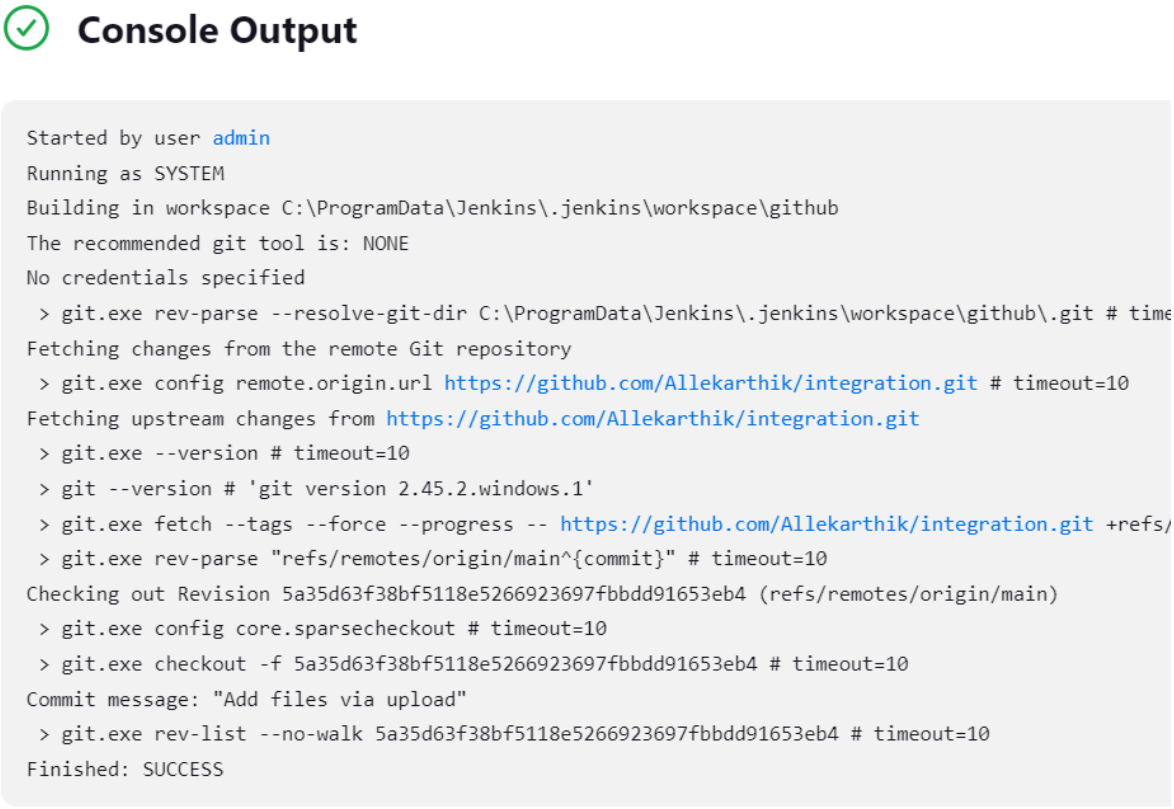
□ Build periodically ?

☑ GitHub hook trigger for GITScm polling ?

□ Poll SCM ?

Then tick the github hook trigger for GITScm polling

Then output will be displayed on the screen .



Console Output

```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\github
The recommended git tool is: NONE
No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\github\.git # time
Fetching changes from the remote Git repository
 > git.exe config remote.origin.url https://github.com/Allekarthik/integration.git # timeout=10
Fetching upstream changes from https://github.com/Allekarthik/integration.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.45.2.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/Allekarthik/integration.git +refs/
 > git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision 5a35d63f38bf5118e5266923697fbbdd91653eb4 (refs/remotes/origin/main)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f 5a35d63f38bf5118e5266923697fbbdd91653eb4 # timeout=10
Commit message: "Add files via upload"
 > git.exe rev-list --no-walk 5a35d63f38bf5118e5266923697fbbdd91653eb4 # timeout=10
Finished: SUCCESS
```

Steps to build pipeline:

Create 3 jobs like job1,job2,job3

We can create by new item->name->apply->save

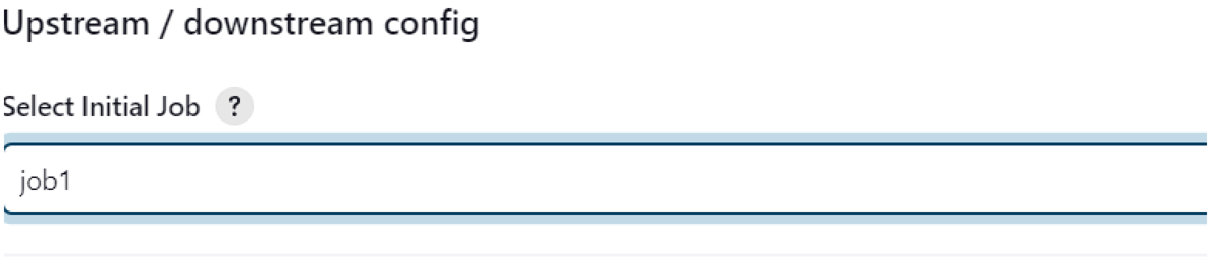
Go to manage jeenkins->pulgins->available pulgins->then install build pipeline

Afer successful installation of pipeline

U can find "+" in main page click on it

Give any name eg:karthik - > select build pipeline view - > create

Then select the initial job eg: job1



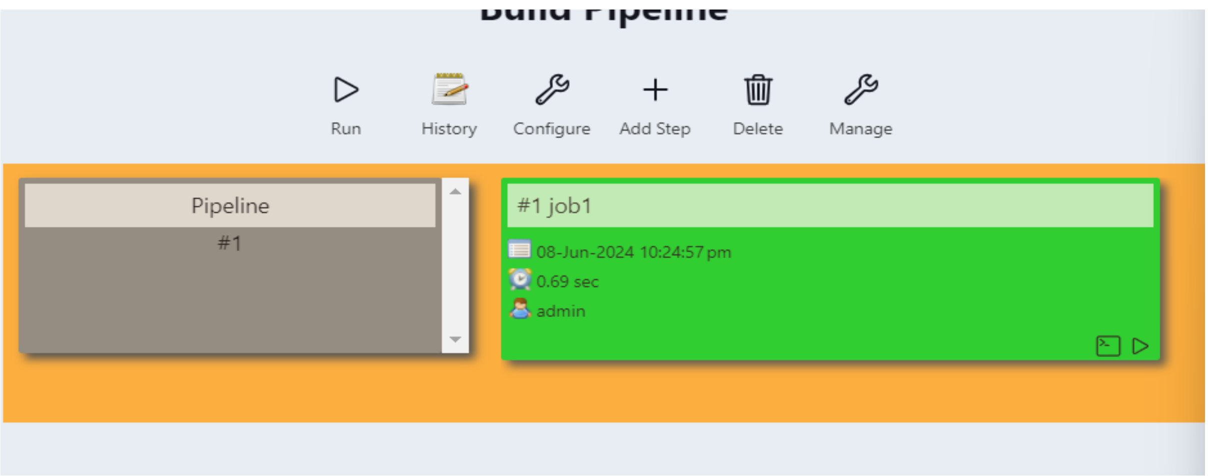
Upstream / downstream config

Select Initial Job ?

job1

Then click on apply -> ok

Then click on run



Pipeline

#1

#1 job1

08-Jun-2024 10:24:57 pm

0.69 sec

admin

Q. **Explore Docker commands for content management**.

Docker is a powerful platform for developing, shipping, and running applications in containers. Content management within Docker involves managing images, containers, volumes, and networks. Here are some essential Docker commands for content management

Docker Commands for Content Management

1. Docker run

- **Description:** Runs a command in a new container. It's one of the most used Docker commands because it creates and starts a new container.
- **Syntax:** docker run [OPTIONS] IMAGE [COMMAND] [ARG...]
- **Example:** $ docker run --name mycontainer -it ubuntu:16.04 /bin/bash
  - **Explanation:**
    - --name mycontainer: Assigns the name "mycontainer" to the container.
    - -it: Combines -i (interactive) and -t (pseudo-TTY) options to keep the container running interactively.

2. Docker start

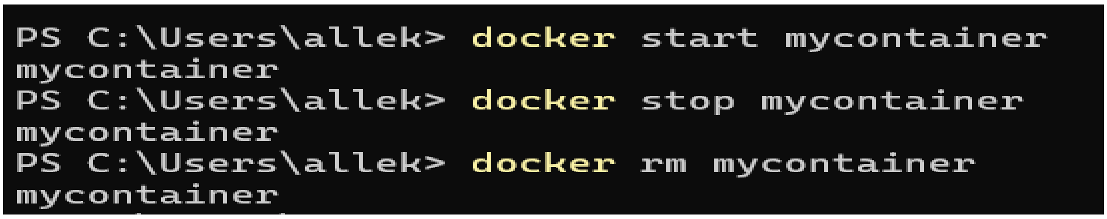
- **Description:** Starts one or more stopped containers. It does not create a new container but starts an existing one.
- **Syntax:** docker start [OPTIONS] CONTAINER [CONTAINER...]
- **Example:** $ docker start mycontainer
  - **Explanation:** Starts the container named "mycontainer".

3. Docker stop

- **Description:** Stops one or more running containers. It sends a SIGTERM signal to the main process inside the container, allowing it to exit gracefully.
- **Syntax:** docker stop [OPTIONS] CONTAINER [CONTAINER...]
- **Example:** $ docker stop mycontainer
  - **Explanation:** Stops the container named "mycontainer".

4. Docker rm

- **Description:** Removes one or more containers. The container must be stopped before it can be removed.
- **Syntax:** docker rm [OPTIONS] CONTAINER [CONTAINER...]
- **Example:** $ docker rm mycontainer
  - **Explanation:** Removes the container named "mycontainer".



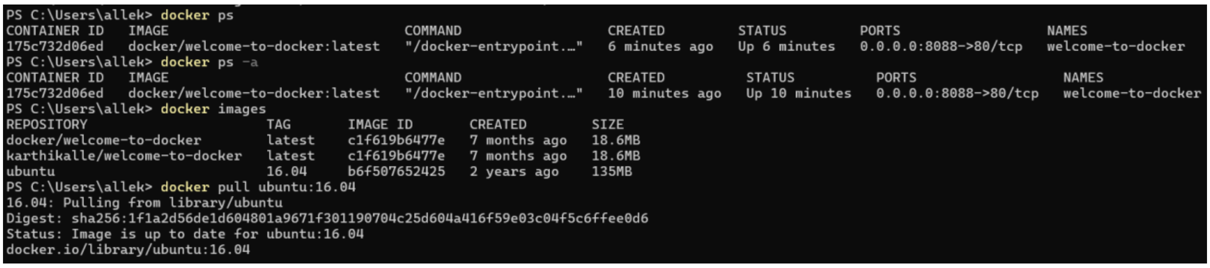
```
PS C:\Users\allek> docker start mycontainer
mycontainer
PS C:\Users\allek> docker stop mycontainer
mycontainer
PS C:\Users\allek> docker rm mycontainer
mycontainer
```

5. Docker ps

- **Description:** Lists containers. By default, it shows only running containers.
- **Syntax:** docker ps [OPTIONS]
- **Example:** $ docker ps
    - **Explanation:** Lists all currently running containers. To list all containers, including stopped ones, use docker ps -a.

6. Docker images

- **Description:** Lists images. It shows all the Docker images available on the local host.
- **Syntax:** docker images [OPTIONS] [REPOSITORY[:TAG]]
- **Example:** $ docker images
    - **Explanation:** Lists all images stored locally on the host.



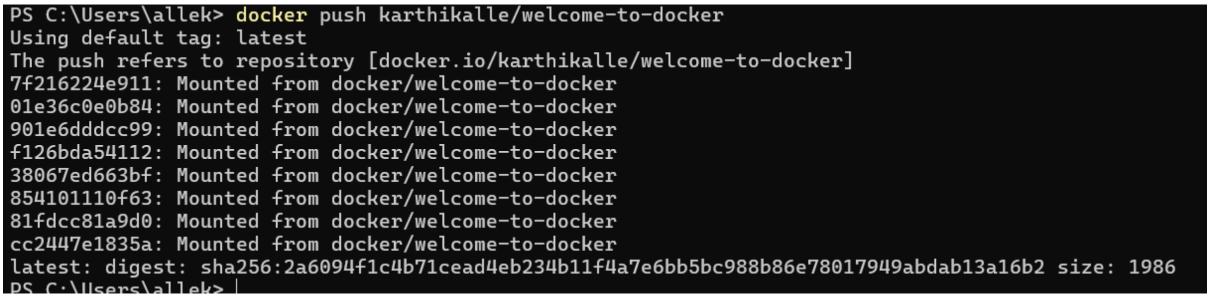
```
PS C:\Users\allek> docker ps
CONTAINER ID   IMAGE                             COMMAND                  CREATED         STATUS         PORTS                  NAMES
175c732d06ed   docker/welcome-to-docker:latest   "/docker-entrypoint.…"   6 minutes ago   Up 6 minutes   0.0.0.0:8088->80/tcp   welcome-to-docker
PS C:\Users\allek> docker ps -a
CONTAINER ID   IMAGE                             COMMAND                  CREATED          STATUS          PORTS                  NAMES
175c732d06ed   docker/welcome-to-docker:latest   "/docker-entrypoint.…"   10 minutes ago   Up 10 minutes   0.0.0.0:8088->80/tcp   welcome-to-docker
PS C:\Users\allek> docker images
REPOSITORY                    TAG       IMAGE ID       CREATED        SIZE
docker/welcome-to-docker      latest    c1f619b6477e   7 months ago   18.6MB
karthikalle/welcome-to-docker latest    c1f619b6477e   7 months ago   18.6MB
ubuntu                        16.04     b6f507652425   2 years ago    135MB
PS C:\Users\allek> docker pull ubuntu:16.04
16.04: Pulling from library/ubuntu
Digest: sha256:1f1a2d56de1d604801a9671f301190704c25d604a416f59e03c04f5c6ffee0d6
Status: Image is up to date for ubuntu:16.04
docker.io/library/ubuntu:16.04
```

7. Docker pull

- **Description:** Pulls an image or a repository from a registry. It downloads the image from a Docker registry like Docker Hub.
- **Syntax:** docker pull [OPTIONS] NAME[:TAG|@DIGEST]
- **Example:** $ docker pull ubuntu:16.04
    - **Explanation:** Pulls the Ubuntu 16.04 image from the Docker Hub registry.

8. Docker push

- **Description:** Pushes an image or a repository to a registry. It uploads the image to a Docker registry.
- **Syntax:** docker push [OPTIONS] NAME[:TAG]
- **Example:** $ docker push myimage
    - **Explanation:** Pushes the image named "myimage" to the Docker Hub registry.



```
PS C:\Users\allek> docker push karthikalle/welcome-to-docker
Using default tag: latest
The push refers to repository [docker.io/karthikalle/welcome-to-docker]
7f216224e911: Mounted from docker/welcome-to-docker
01e36c0e0b84: Mounted from docker/welcome-to-docker
901e6dddcc99: Mounted from docker/welcome-to-docker
f126bda54112: Mounted from docker/welcome-to-docker
38067ed663bf: Mounted from docker/welcome-to-docker
854101110f63: Mounted from docker/welcome-to-docker
81fdcc81a9d0: Mounted from docker/welcome-to-docker
cc2447e1835a: Mounted from docker/welcome-to-docker
latest: digest: sha256:2a6094f1c4b71cead4eb234b11f4a7e6bb5bc988b86e78017949abdab13a16b2 size: 1986
PS C:\Users\allek> |
```

Q. Develop a simple containerized application using Docker:

Here's an example of how you can develop a simple containerized application using Docker: Choose an application:

Before that create a folder in your file manager eg: 22507_Docker ->python_image ->Dockerfile ,app.py

Note:Docker should be opened first

- Choose a simple application that you want to containerize. For example, a Python script that prints "Hello World".

- Create a file named "Dockerfile" in the same directory as the application. In the Dockerfile, specify the base image, copy the application into the container, and specify the command to run the application.

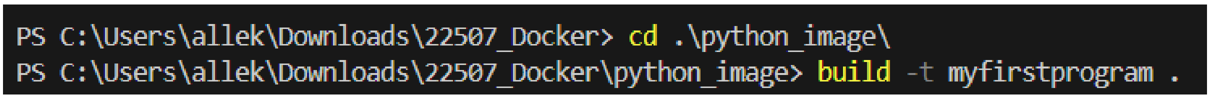
Dockerfile:

```
# Dockerfile Image Container
# Dockerfile is to build images
# Image is a template for running containers
# Container is for running projects
# Use a slim Python 3.8 image as the base
FROM python:3.8-slim

# Set the working directory for the container
WORKDIR /app

# Copy requirements.txt (if you have dependencies)
# COPY requirements.txt .
# RUN pip install -r requirements.txt

# Copy the Python script to the container
COPY main.py .

# Define the command to run when the container starts
CMD ["python", "main.py"]
```

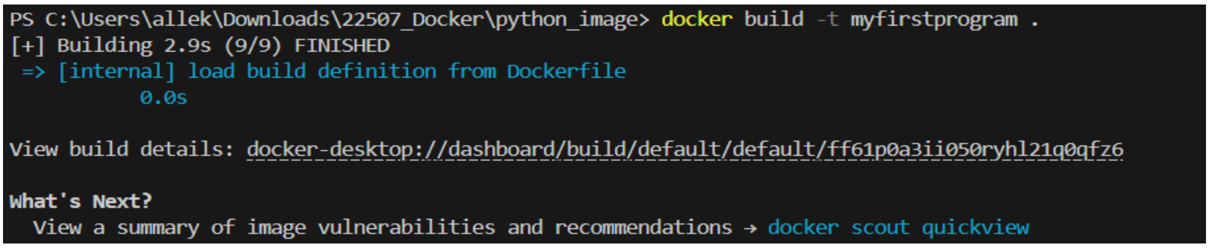
Main.py file:

```
def hello():
    print('hello world')

hello()
```

• Build the Docker image: Run the following command to build the Docker image: $ docker build -t myfirstprogram .



```
PS C:\Users\allek\Downloads\22507_Docker> cd .\python_image\
PS C:\Users\allek\Downloads\22507_Docker\python_image> build -t myfirstprogram .
```

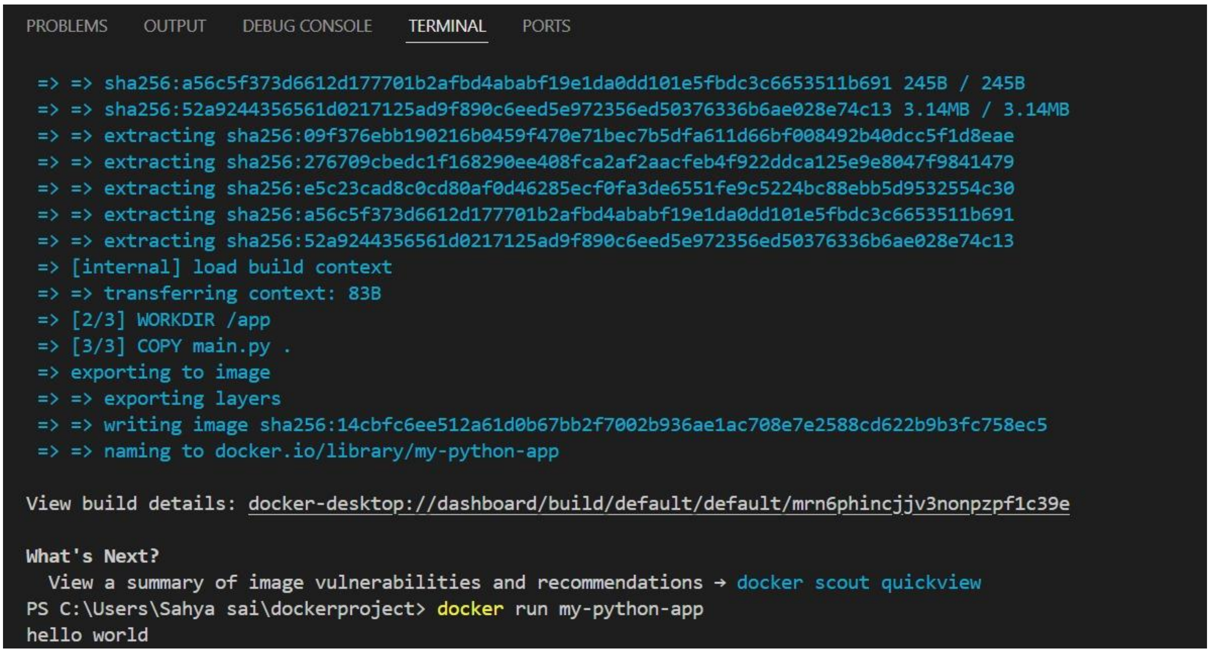


```
PS C:\Users\allek\Downloads\22507_Docker\python_image> docker build -t myfirstprogram .
[+] Building 2.9s (9/9) FINISHED
 => [internal] load build definition from Dockerfile
             0.0s

View build details: docker-desktop://dashboard/build/default/default/ff61p0a3ii050ryhl21q0qfz6

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
```

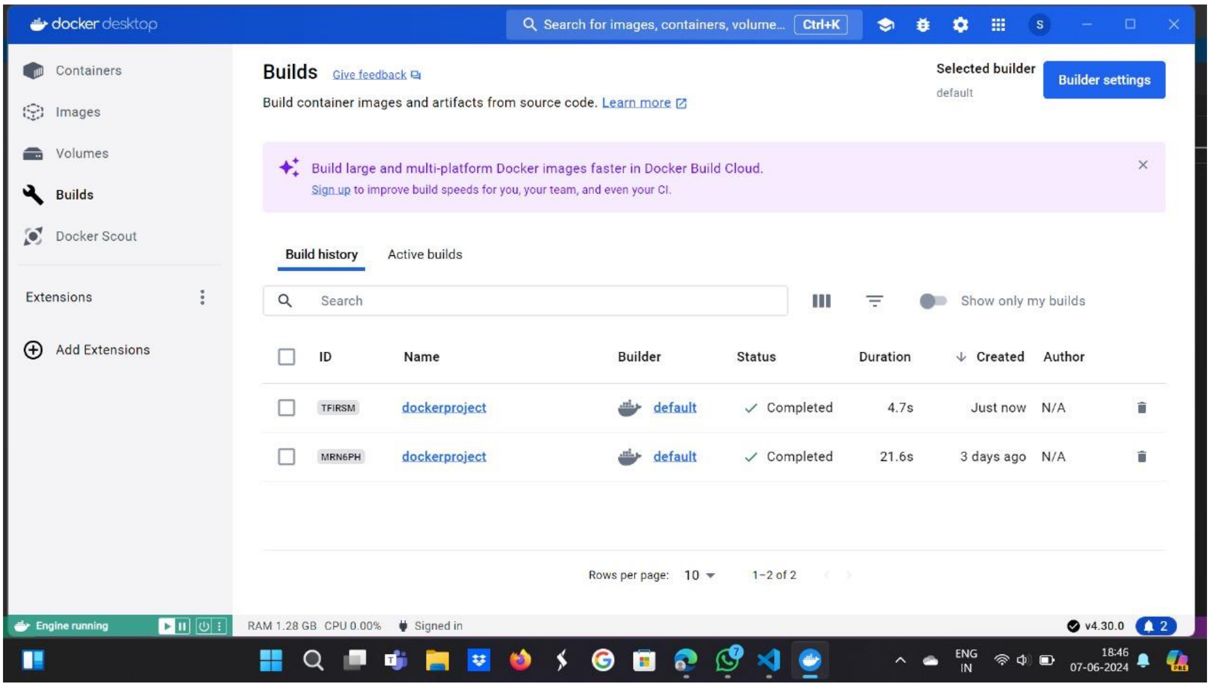
This command builds a new Docker image using the Dockerfile and tags the image with the name "dockerproject".

•Run the Docker container: Run the following command to start a new container based on the image: $ docker run – dockerproject

OUTPUT:



```
 => => sha256:a56c5f373d6612d177701b2afbd4ababf19e1da0dd101e5fbdc3c6653511b691 245B / 245B
 => => sha256:52a9244356561d0217125ad9f890c6eed5e972356ed50376336b6ae028e74c13 3.14MB / 3.14MB
 => => extracting sha256:09f376ebb190216b0459f470e71bec7b5dfa611d66bf008492b40dcc5f1d8eae
 => => extracting sha256:276709cbedc1f168290ee408fca2af2aacfeb4f922ddca125e9e8047f9841479
 => => extracting sha256:e5c23cad8c0cd80af0d46285ecf0fa3de6551fe9c5224bc88ebb5d9532554c30
 => => extracting sha256:a56c5f373d6612d177701b2afbd4ababf19e1da0dd101e5fbdc3c6653511b691
 => => extracting sha256:52a9244356561d0217125ad9f890c6eed5e972356ed50376336b6ae028e74c13
 => [internal] load build context
 => => transferring context: 83B
 => [2/3] WORKDIR /app
 => [3/3] COPY main.py .
 => exporting to image
 => => exporting layers
 => => writing image sha256:14cbfc6ee512a61d0b67bb2f7002b936ae1ac708e7e2588cd622b9b3fc758ec5
 => => naming to docker.io/library/my-python-app

View build details: docker-desktop://dashboard/build/default/default/mrn6phincjjv3nonpzpf1c39e

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
PS C:\Users\Sahya sai\dockerproject> docker run my-python-app
hello world
```



Builds Give feedback

Build container images and artifacts from source code. Learn more

Selected builder
default

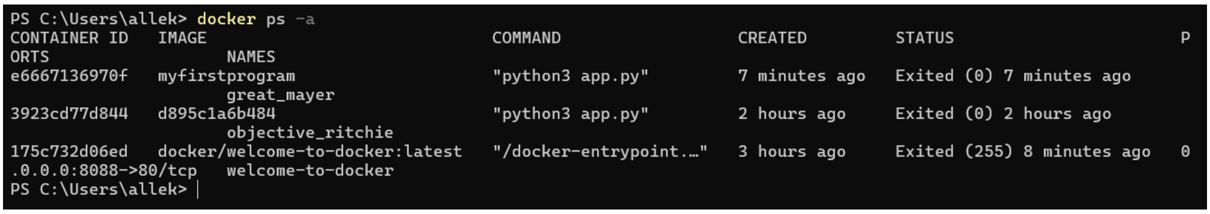
Builder settings

Build large and multi-platform Docker images faster in Docker Build Cloud.
Sign up to improve build speeds for you, your team, and even your CI.

□	ID	Name	Builder	Status	Duration	Created	Author
□	TFIRSM	dockerproject	default	✓ Completed	4.7s	Just now	N/A
□	MRN6PH	dockerproject	default	✓ Completed	21.6s	3 days ago	N/A

Rows per page: 10 1–2 of 2

You can check in Windows powershell also



```
PS C:\Users\allek> docker ps -a
CONTAINER ID   IMAGE                           COMMAND                  CREATED          STATUS                      P
ORTS                  NAMES
e6667136970f   myfirstprogram                  "python3 app.py"         7 minutes ago    Exited (0) 7 minutes ago
                      great_mayer
3923cd77d844   d895c1a6b484                    "python3 app.py"         2 hours ago      Exited (0) 2 hours ago
                      objective_ritchie
175c732d06ed   docker/welcome-to-docker:latest "/docker-entrypoint.…"   3 hours ago      Exited (255) 8 minutes ago  0
.0.0.0:8088->80/tcp   welcome-to-docker
PS C:\Users\allek> |
```

This is a simple example of how you can use Docker to containerize an application. In a real-world scenario, you would likely have more complex requirements, such as running multiple containers, managing network connections, and persisting data. However, this example should give you a good starting point for using Docker to containerize your applications.

Q. Install and Explore Selenium for automated testing or Write a simple program in JavaScript and perform testing using Selenium.

Prerequired:

Download and install Node.js

Download and install vs code

Install selenium-webdriver and install mocha

https://storage.googleapis.com/chrome-for-testing-public/125.0.6422.141/win64/chrome-win64.zip

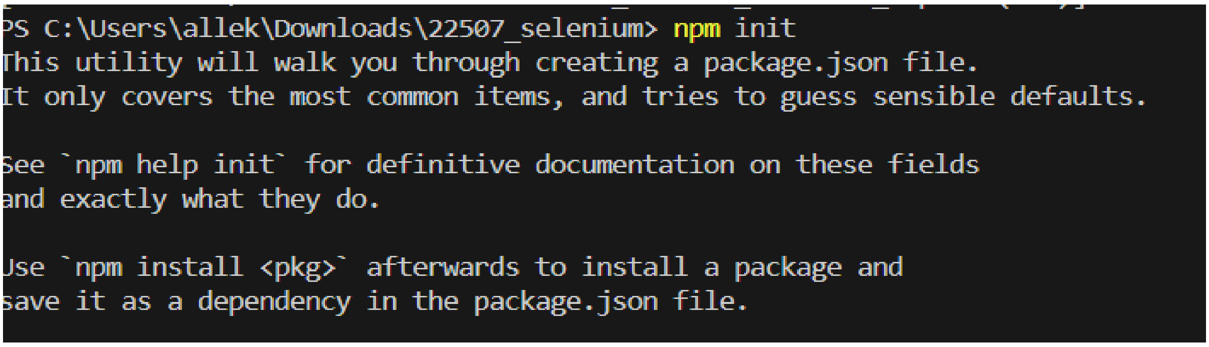
->for installing selenium web driver go to any web browser->selenium web driver install ->Download selenium->javascript stable 4.21->chromedriver.exe->win64

Steps:

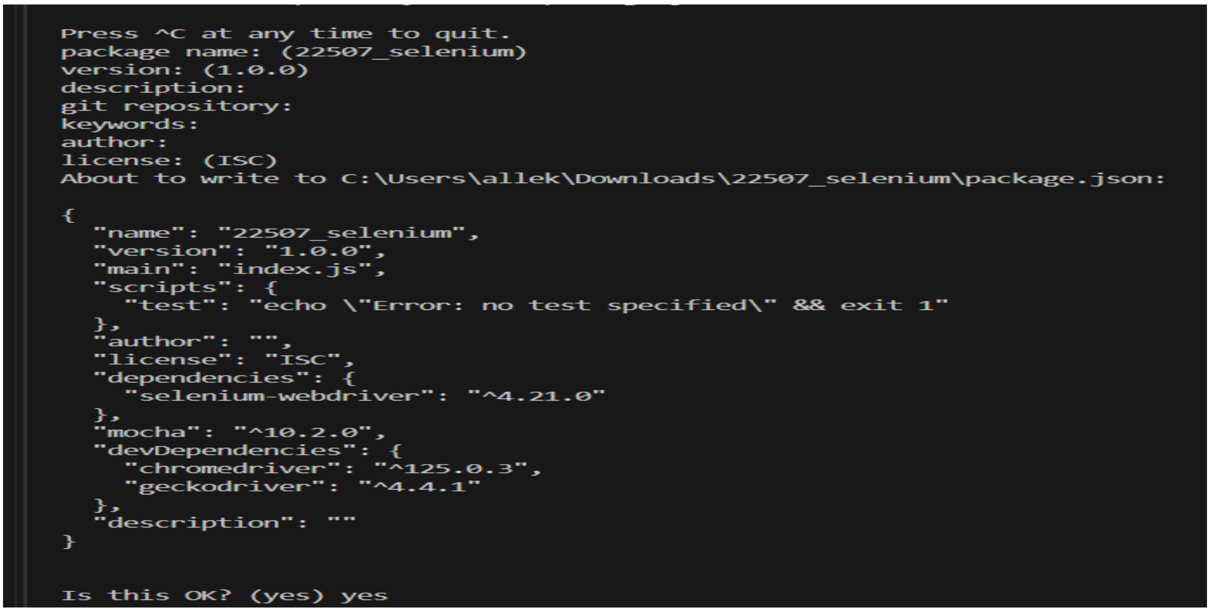
Create a folder called newfolder in your downloads /or any other main folder

Open this folder in vs code and go to termina->new terminal then type the below command i.e npm init where it atomatically creates package.json file

```
{
  "name": "new-folder",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "description": "",
  "dependencies": {
    "selenium-webdriver": "^4.21.0"
  },
  "devDependencies": {
    "chromedriver": "^125.0.3",
    "geckodriver": "^4.4.1"
  },
  "mocha":"^10.2.0" //here u need to add this extra line
}
```



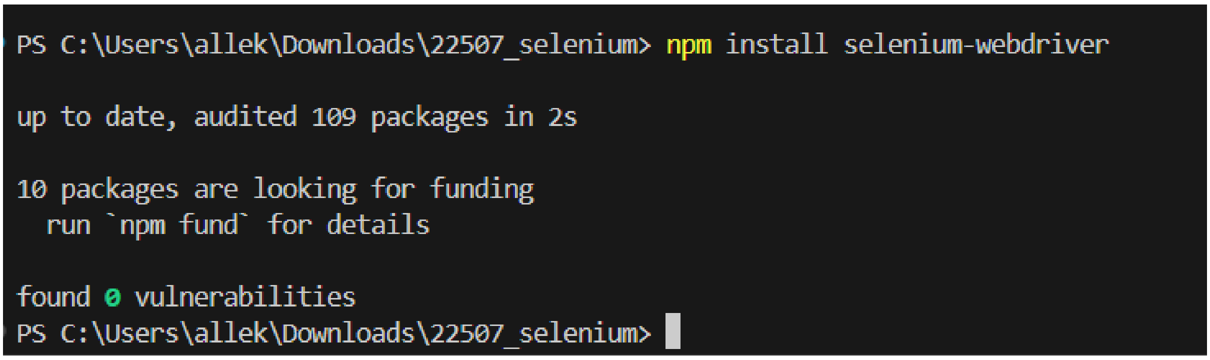
```
PS C:\Users\allek\Downloads\22507_selenium> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.
```



```
Press ^C at any time to quit.
package name: (22507_selenium)
version: (1.0.0)
description:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Users\allek\Downloads\22507_selenium\package.json:

{
  "name": "22507_selenium",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "selenium-webdriver": "^4.21.0"
  },
  "mocha": "^10.2.0",
  "devDependencies": {
    "chromedriver": "^125.0.3",
    "geckodriver": "^4.4.1"
  },
  "description": ""
}


Is this OK? (yes) yes
```

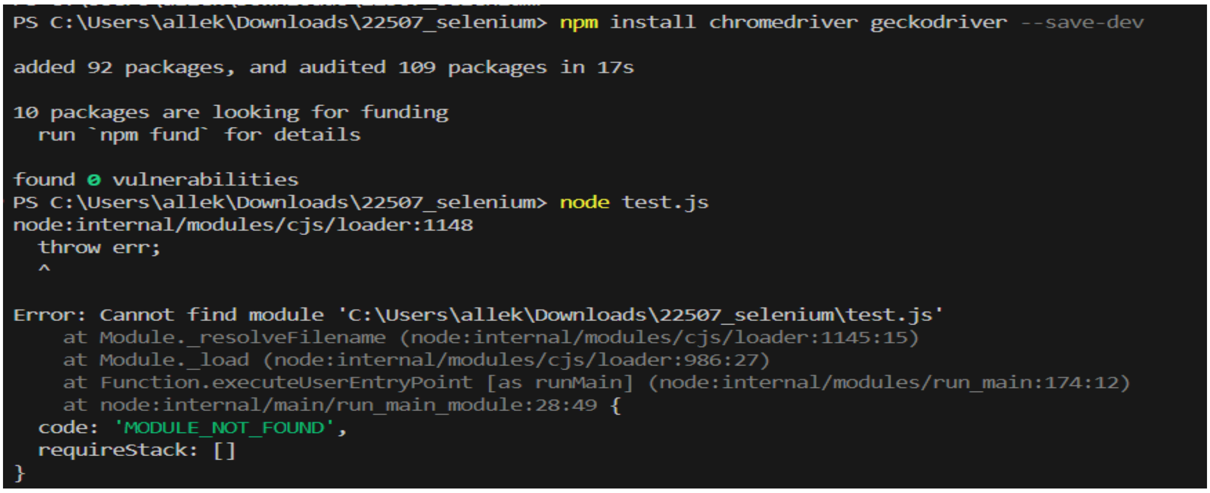
Then next step is to install selenium

After installing it will create an package-lock.json file



```
PS C:\Users\allek\Downloads\22507_selenium> npm install selenium-webdriver

up to date, audited 109 packages in 2s

10 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\allek\Downloads\22507_selenium>
```

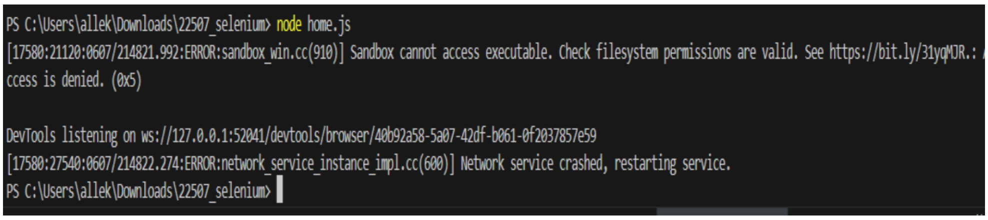
After succesfull installation of selenium we need to install chromedriver



```
PS C:\Users\allek\Downloads\22507_selenium> npm install chromedriver geckodriver --save-dev

added 92 packages, and audited 109 packages in 17s

10 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\allek\Downloads\22507_selenium> node test.js
node:internal/modules/cjs/loader:1148
  throw err;
  ^

Error: Cannot find module 'C:\Users\allek\Downloads\22507_selenium\test.js'
    at Module._resolveFilename (node:internal/modules/cjs/loader:1145:15)
    at Module._load (node:internal/modules/cjs/loader:986:27)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:174:12)
    at node:internal/main/run_main_module:28:49 {
  code: 'MODULE_NOT_FOUND',
  requireStack: []
}
```

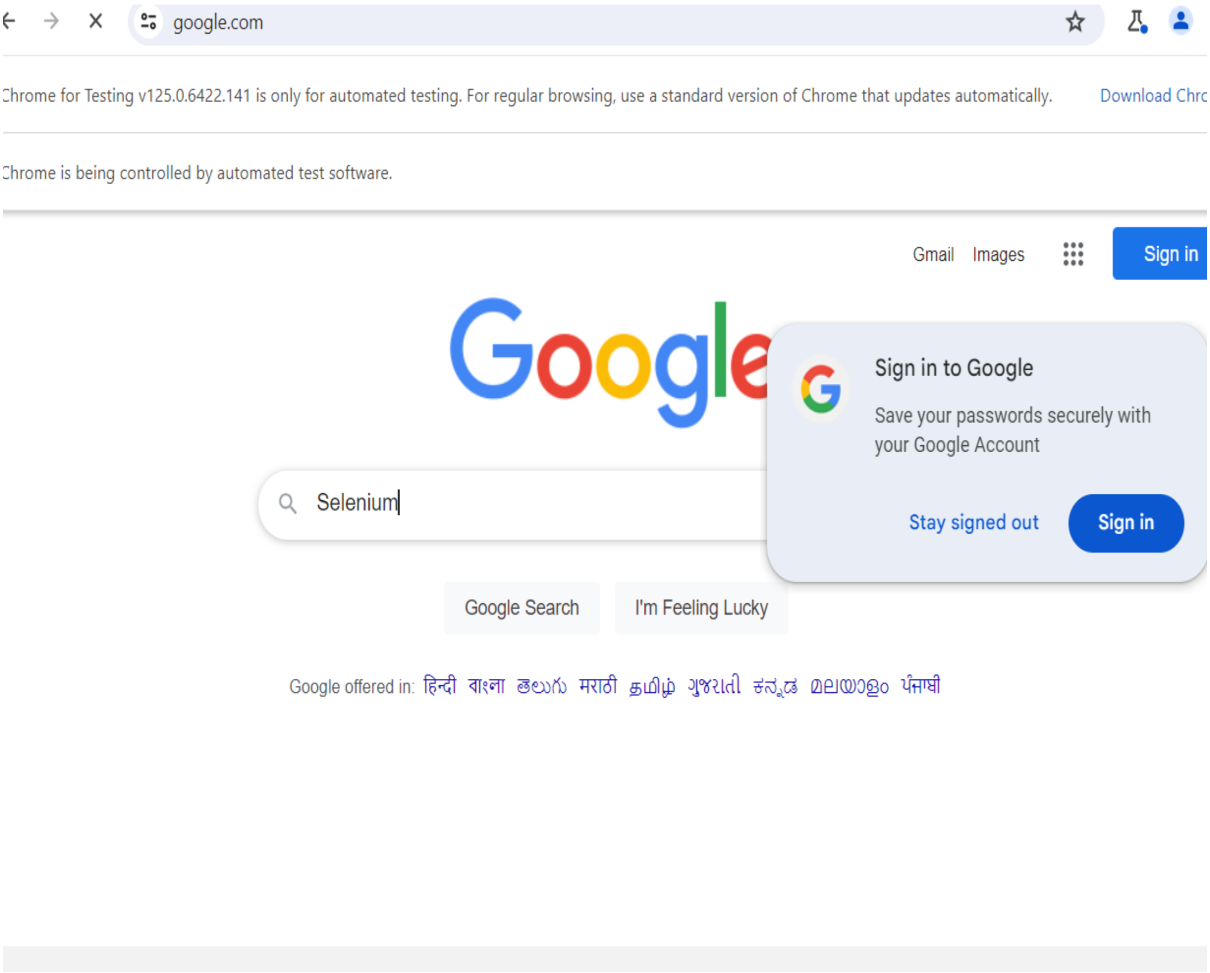
Home.js

```
const { Builder, By, Key, until } = require('selenium-webdriver');
const chrome = require('selenium-webdriver/chrome');


(async () => {
    const driver = await new Builder()
        .forBrowser('chrome')
        .setChromeOptions(new chrome.Options())
        .build();


    try {
        await driver.get('https://www.google.com');
        await driver.findElement(By.name('q')).sendKeys('Selenium', Key.RETURN);
        await driver.wait(until.titleContains('Selenium'), 100000000000);
    } catch (error) {
        console.error('Test failed:', error);
    } finally {
        await driver.quit();
    }
})();
```



```
PS C:\Users\allek\Downloads\22507_selenium> node home.js
[17580:21120:0607/214821.992:ERROR:sandbox_win.cc(910)] Sandbox cannot access executable. Check filesystem permissions are valid. See https://bit.ly/31yqMJR.: 
ccess is denied. (0x5)

DevTools listening on ws://127.0.0.1:52041/devtools/browser/40b92a58-5a07-42df-b061-0f2037857e59
[17580:27540:0607/214822.274:ERROR:network_service_instance_impl.cc(600)] Network service crashed, restarting service.
PS C:\Users\allek\Downloads\22507_selenium> 
```

Output:



google.com

Chrome for Testing v125.0.6422.141 is only for automated testing. For regular browsing, use a standard version of Chrome that updates automatically.

Chrome is being controlled by automated test software.

Sign in to Google

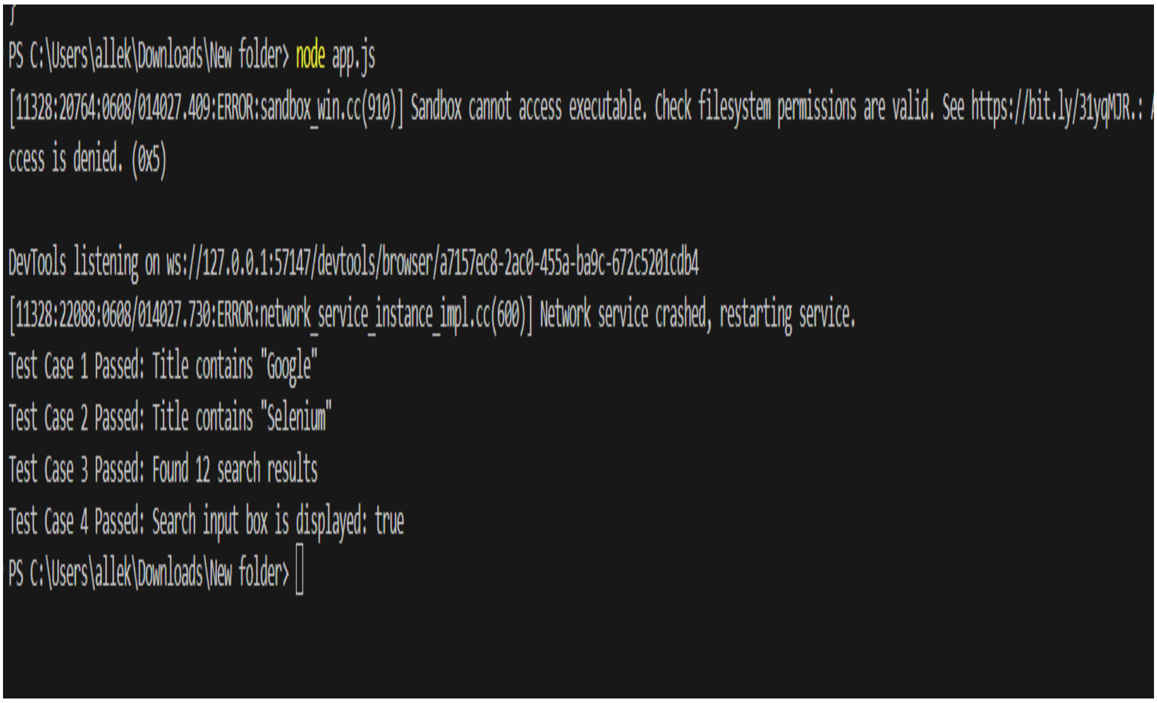
Save your passwords securely with your Google Account

Selenium

Q. Develop test cases for the above containerized application using selenium.

```
const { Builder, By, Key, until } = require('selenium-webdriver');
const chrome = require('selenium-webdriver/chrome');

(async () => {
  const driver = await new Builder()
    .forBrowser('chrome')
    .setChromeOptions(new chrome.Options())
    .build();

  try {
    // Test Case 1: Navigate to Google and verify title
    await driver.get('https://www.google.com');
    await driver.wait(until.titleContains('Google'), 10000);
    console.log('Test Case 1 Passed: Title contains "Google"');

    // Test Case 2: Search for "Selenium" on Google
    await driver.findElement(By.name('q')).sendKeys('Selenium', Key.RETURN);
    await driver.wait(until.titleContains('Selenium'), 10000);
    console.log('Test Case 2 Passed: Title contains "Selenium"');

    // Test Case 3: Verify search results
    const searchResults = await driver.findElements(By.css('div.g'));
    console.log(`Test Case 3 Passed: Found ${searchResults.length} search results`);

    // Test Case 4: Verify the presence of the search input box
    const searchInput = await driver.findElement(By.name('q'));
    const isSearchInputDisplayed = await searchInput.isDisplayed();
    console.log(`Test Case 4 Passed: Search input box is displayed:
${isSearchInputDisplayed}`);
```

```
    } catch (error) {
        console.error('One or more test cases failed:', error);
    } finally {
        await driver.quit();
    }
})();
```



```
}
PS C:\Users\allek\Downloads\New folder> node app.js
[11328:20764:0608/014027.409:ERROR:sandbox_win.cc(910)] Sandbox cannot access executable. Check filesystem permissions are valid. See https://bit.ly/31yqMJR.: A
ccess is denied. (0x5)

DevTools listening on ws://127.0.0.1:57147/devtools/browser/a7157ec8-2ac0-455a-ba9c-672c5201cdb4
[11328:22088:0608/014027.730:ERROR:network_service_instance_impl.cc(600)] Network service crashed, restarting service.
Test Case 1 Passed: Title contains "Google"
Test Case 2 Passed: Title contains "Selenium"
Test Case 3 Passed: Found 12 search results
Test Case 4 Passed: Search input box is displayed: true
PS C:\Users\allek\Downloads\New folder> 
```