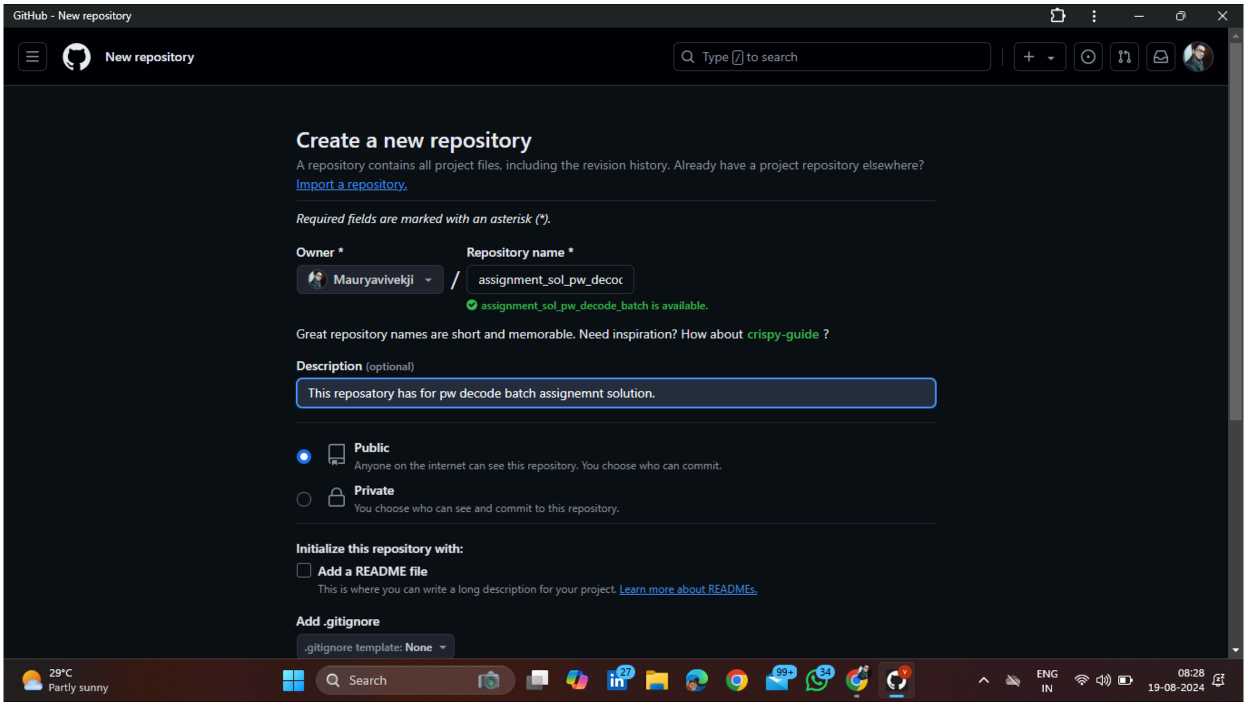
Git & GitHub Assignment



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 * Mauryavivekji / Repository name * assignment_sol_pw_decoc

assignment_sol_pw_decode_batch is available.

Great repository names are short and memorable. Need inspiration? How about crispy-guide ?

Description (optional)

This reposatory has for pw decode batch assignemnt solution.

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

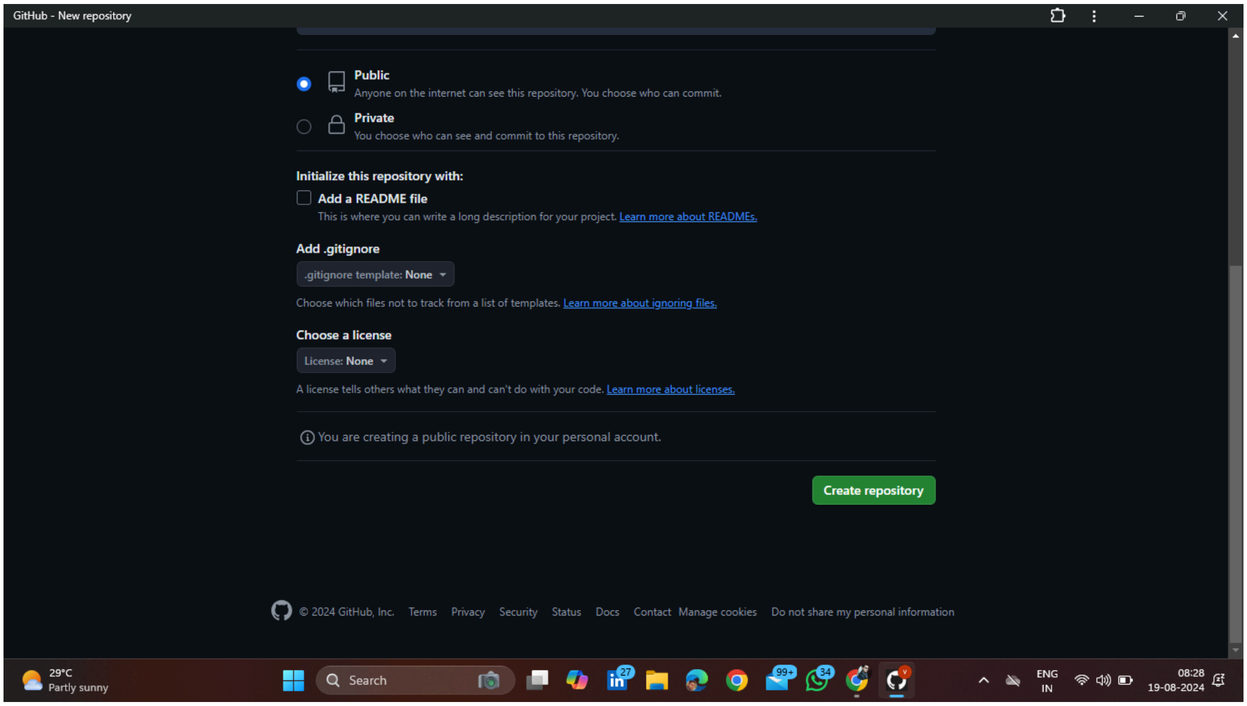
○ Private
You choose who can see and commit to this repository.

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



● 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

Choose which files not to track from a list of templates. Learn more about ignoring files.

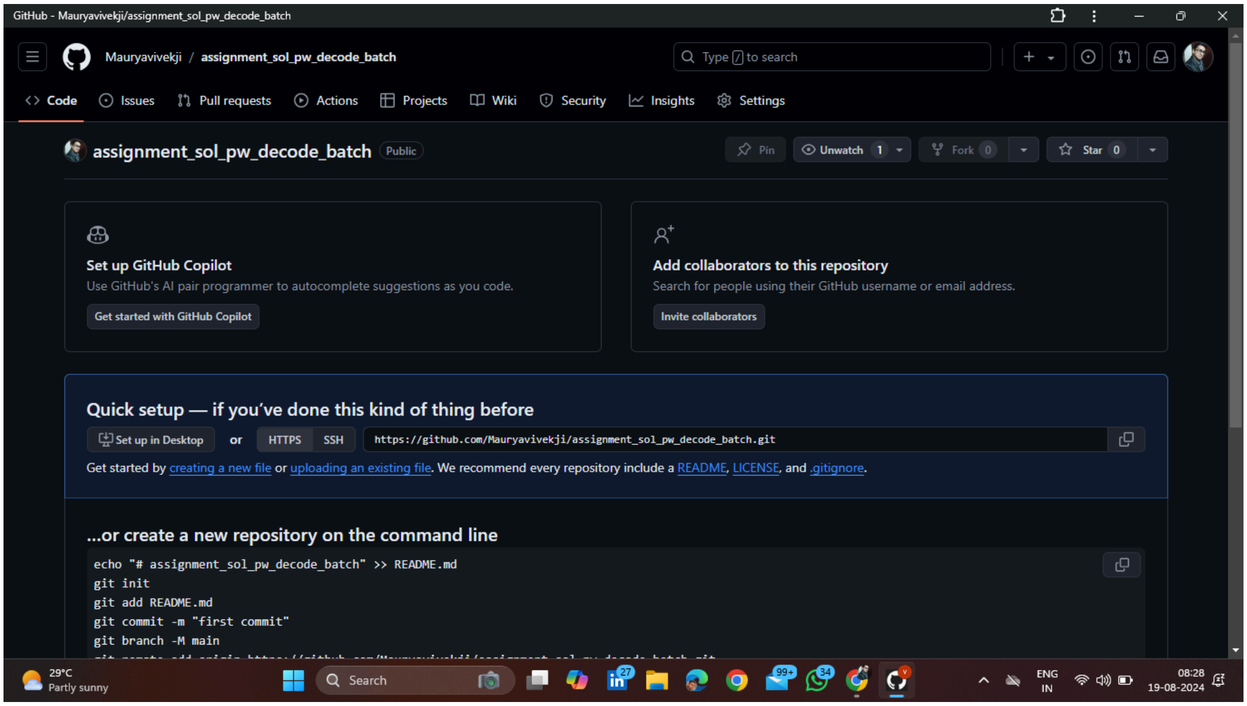
Choose a license

License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

You are creating a public repository in your personal account.

Create repository
© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information



assignment_sol_pw_decode_batch

Set up GitHub Copilot

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Add collaborators to this repository

Search for people using their GitHub username or email address.

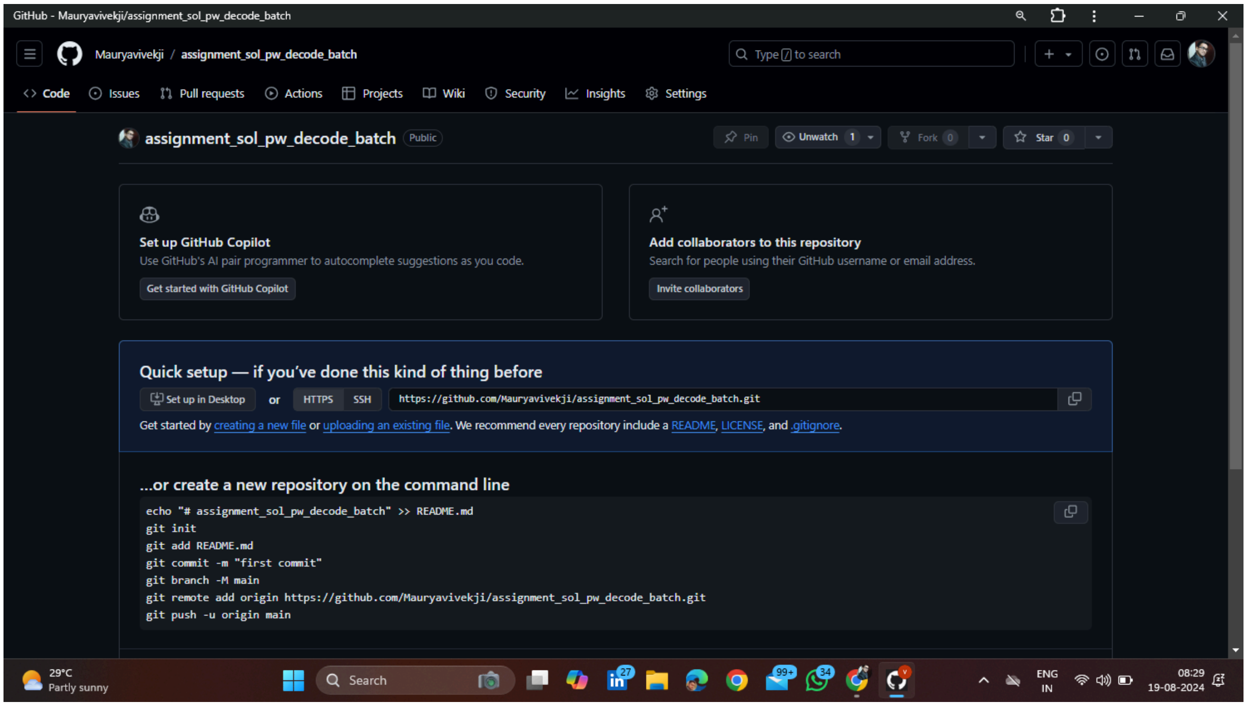
Quick setup — if you've done this kind of thing before

https://github.com/Mauryavivekji/assignment_sol_pw_decode_batch.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

…or create a new repository on the command line

```
echo "# assignment_sol_pw_decode_batch" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Mauryavivekji/assignment_sol_pw_decode_batch.git
```



assignment_sol_pw_decode_batch

Set up GitHub Copilot

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Add collaborators to this repository

Search for people using their GitHub username or email address.

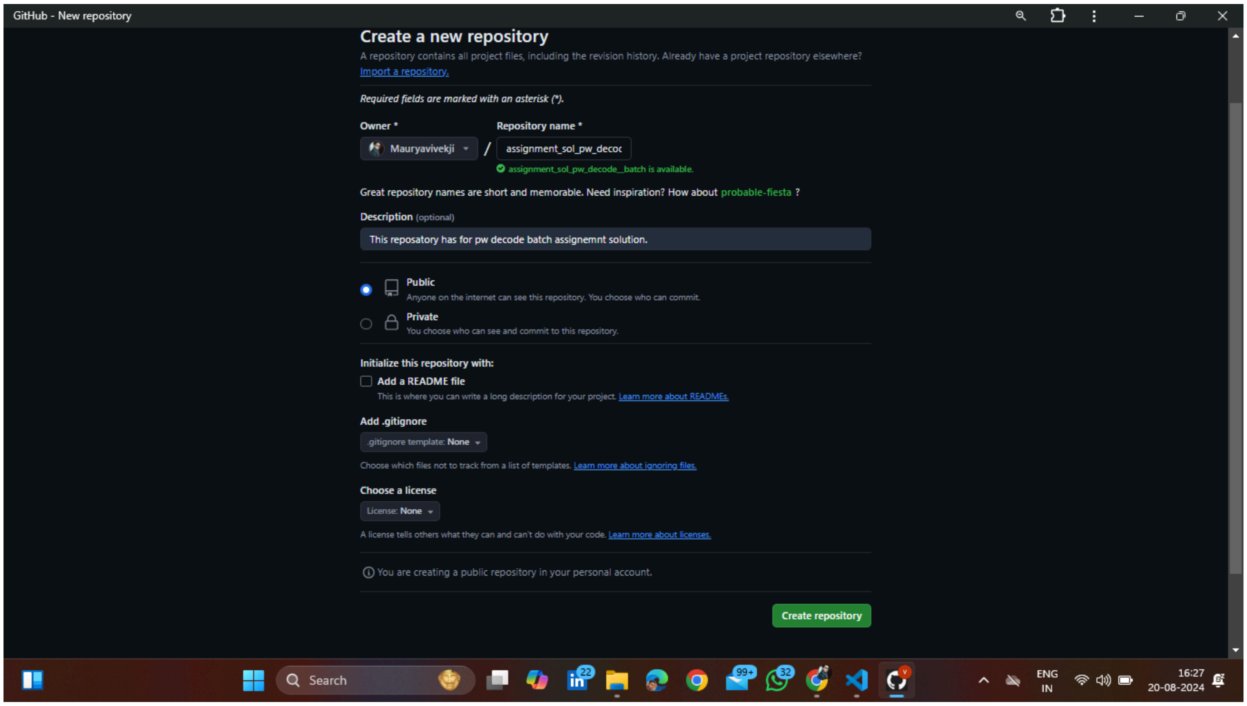
Quick setup — if you've done this kind of thing before

https://github.com/Mauryavivekji/assignment_sol_pw_decode_batch.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

…or create a new repository on the command line

```
echo "# assignment_sol_pw_decode_batch" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Mauryavivekji/assignment_sol_pw_decode_batch.git
git push -u origin main
```



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 * Mauryavivekji / **Repository name *** assignment_sol_pw_decoc

✓ assignment_sol_pw_decode__batch is available.

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

Description (optional)

This reposatory has for pw decode batch assignemnt solution.

● **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

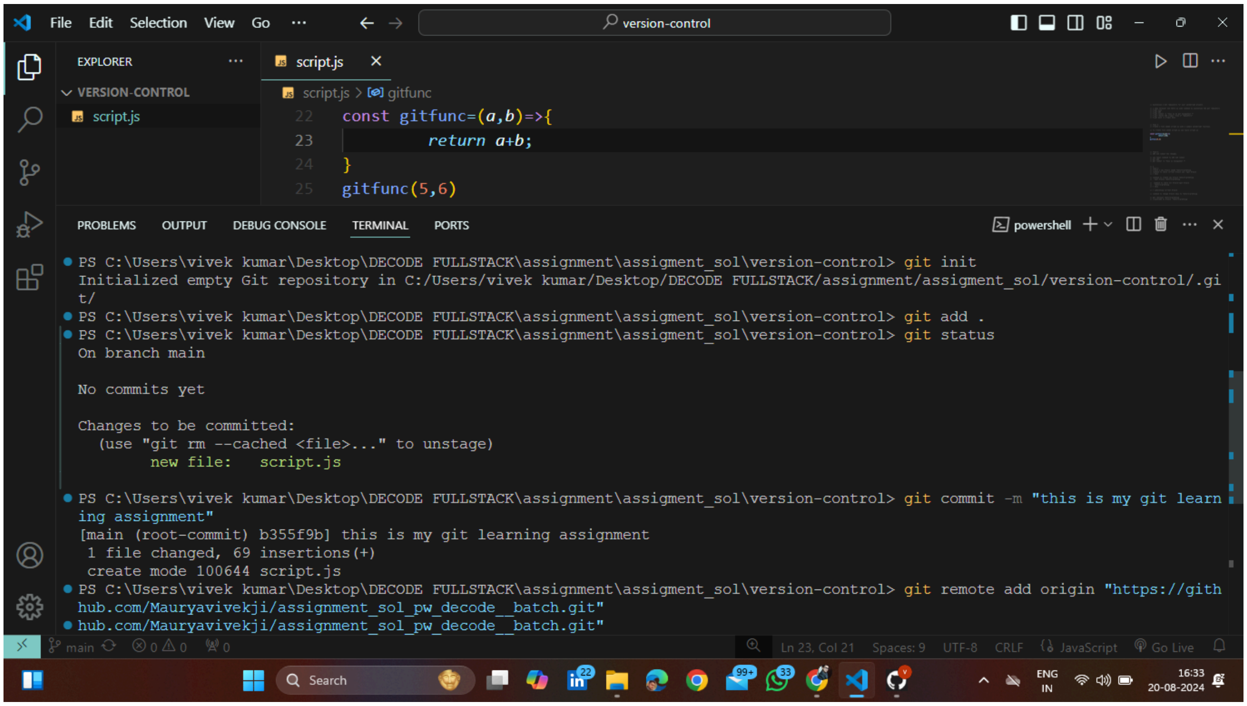
Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

License: None

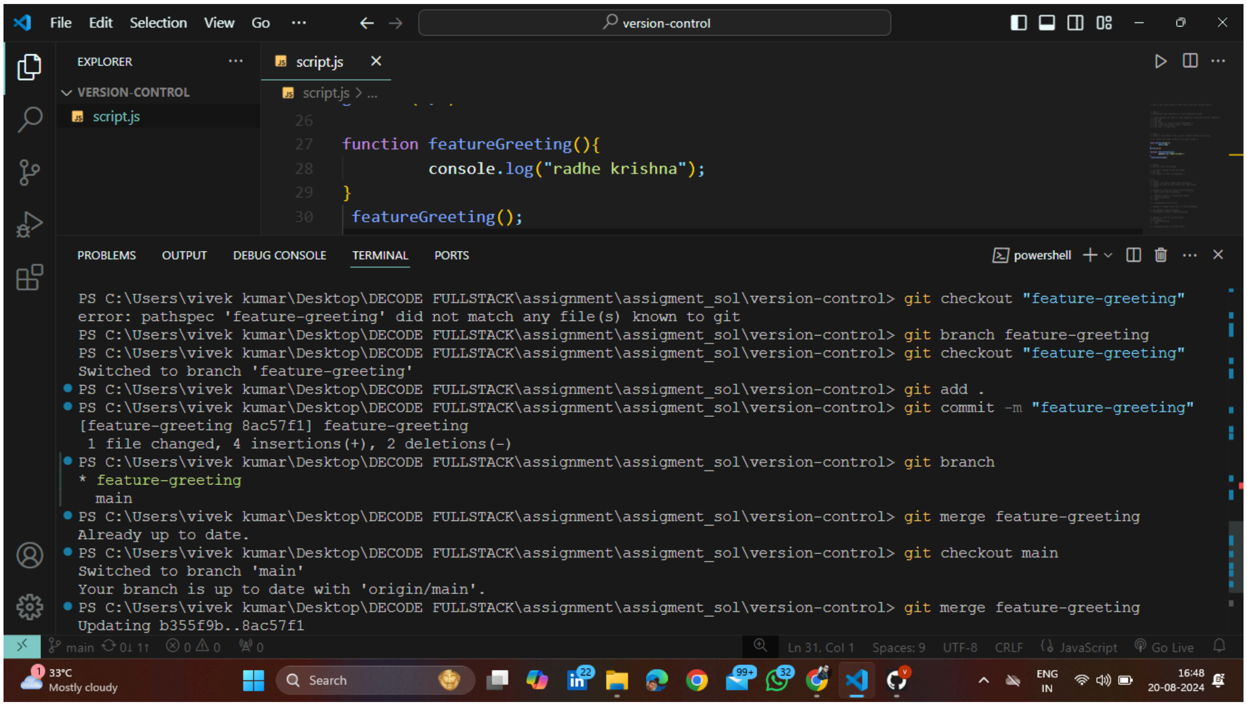
A license tells others what they can and can't do with your code. Learn more about licenses.

ⓘ You are creating a public repository in your personal account.



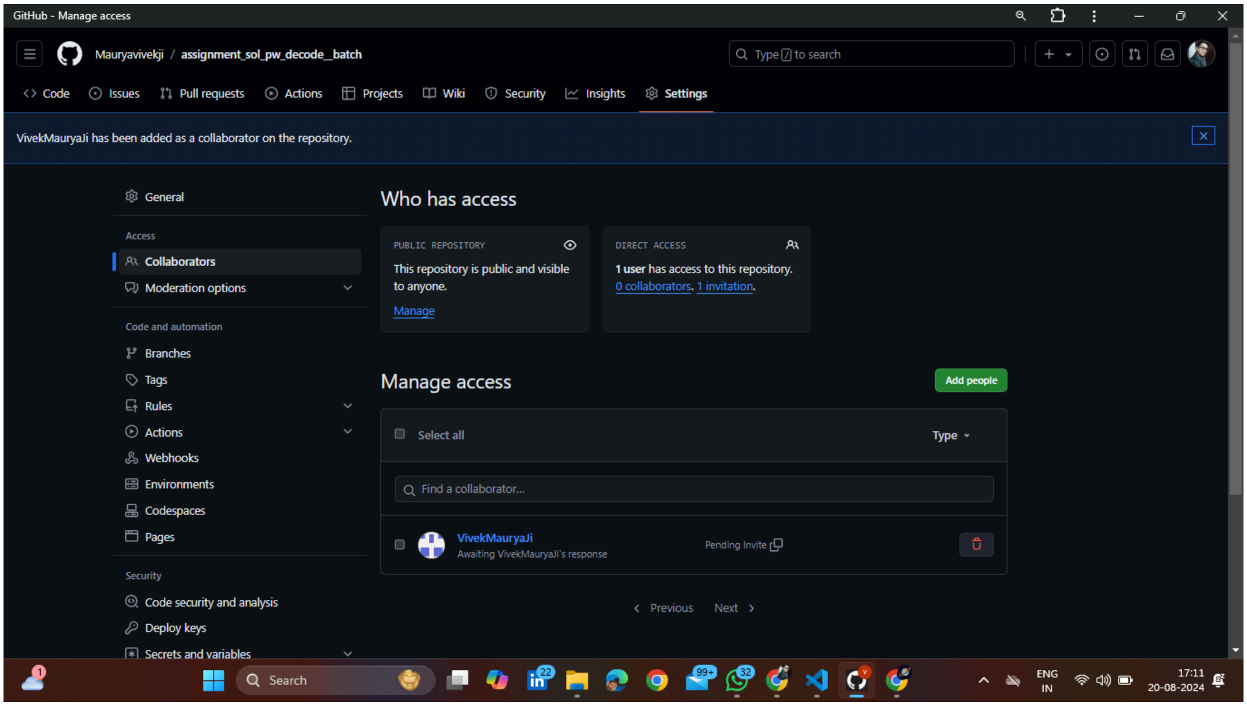
```
const gitfunc=(a,b)=>{
        return a+b;
}
gitfunc(5,6)
```

```
PS C:\Users\vivek kumar\Desktop\DECODE FULLSTACK\assignment\assigment_sol\version-control> git init
Initialized empty Git repository in C:/Users/vivek kumar/Desktop/DECODE FULLSTACK/assignment/assigment_sol/version-control/.gi
t/
PS C:\Users\vivek kumar\Desktop\DECODE FULLSTACK\assignment\assigment_sol\version-control> git add .
PS C:\Users\vivek kumar\Desktop\DECODE FULLSTACK\assignment\assigment_sol\version-control> git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   script.js

PS C:\Users\vivek kumar\Desktop\DECODE FULLSTACK\assignment\assigment_sol\version-control> git commit -m "this is my git learn
ing assignment"
[main (root-commit) b355f9b] this is my git learning assignment
 1 file changed, 69 insertions(+)
 create mode 100644 script.js
PS C:\Users\vivek kumar\Desktop\DECODE FULLSTACK\assignment\assigment_sol\version-control> git remote add origin "https://gith
hub.com/Mauryavivekji/assignment_sol_pw_decode__batch.git"
hub.com/Mauryavivekji/assignment_sol_pw_decode__batch.git"
```


File Edit Selection View Go ···
version-control
EXPLORER
VERSION-CONTROL
script.js
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
main 0↓ 1↑ ⊗ 0 ⚠ 0 0
Ln 31, Col 1 Spaces: 9 UTF-8 CRLF JavaScript Go Live
33°C Mostly cloudy
Search
ENG IN 16:48 20-08-2024

```javascript
function featureGreeting(){
        console.log("radhe krishna");
}
 featureGreeting();
```

```
PS C:\Users\vivek kumar\Desktop\DECODE FULLSTACK\assignment\assigment_sol\version-control> git checkout "feature-greeting"
error: pathspec 'feature-greeting' did not match any file(s) known to git
PS C:\Users\vivek kumar\Desktop\DECODE FULLSTACK\assignment\assigment_sol\version-control> git branch feature-greeting
PS C:\Users\vivek kumar\Desktop\DECODE FULLSTACK\assignment\assigment_sol\version-control> git checkout "feature-greeting"
Switched to branch 'feature-greeting'
PS C:\Users\vivek kumar\Desktop\DECODE FULLSTACK\assignment\assigment_sol\version-control> git add .
PS C:\Users\vivek kumar\Desktop\DECODE FULLSTACK\assignment\assigment_sol\version-control> git commit -m "feature-greeting"
[feature-greeting 8ac57f1] feature-greeting
 1 file changed, 4 insertions(+), 2 deletions(-)
PS C:\Users\vivek kumar\Desktop\DECODE FULLSTACK\assignment\assigment_sol\version-control> git branch
* feature-greeting
  main
PS C:\Users\vivek kumar\Desktop\DECODE FULLSTACK\assignment\assigment_sol\version-control> git merge feature-greeting
Already up to date.
PS C:\Users\vivek kumar\Desktop\DECODE FULLSTACK\assignment\assigment_sol\version-control> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\vivek kumar\Desktop\DECODE FULLSTACK\assignment\assigment_sol\version-control> git merge feature-greeting
Updating b355f9b..8ac57f1
```



VivekMauryaJi has been added as a collaborator on the repository.

Who has access

PUBLIC REPOSITORY

This repository is public and visible to anyone.

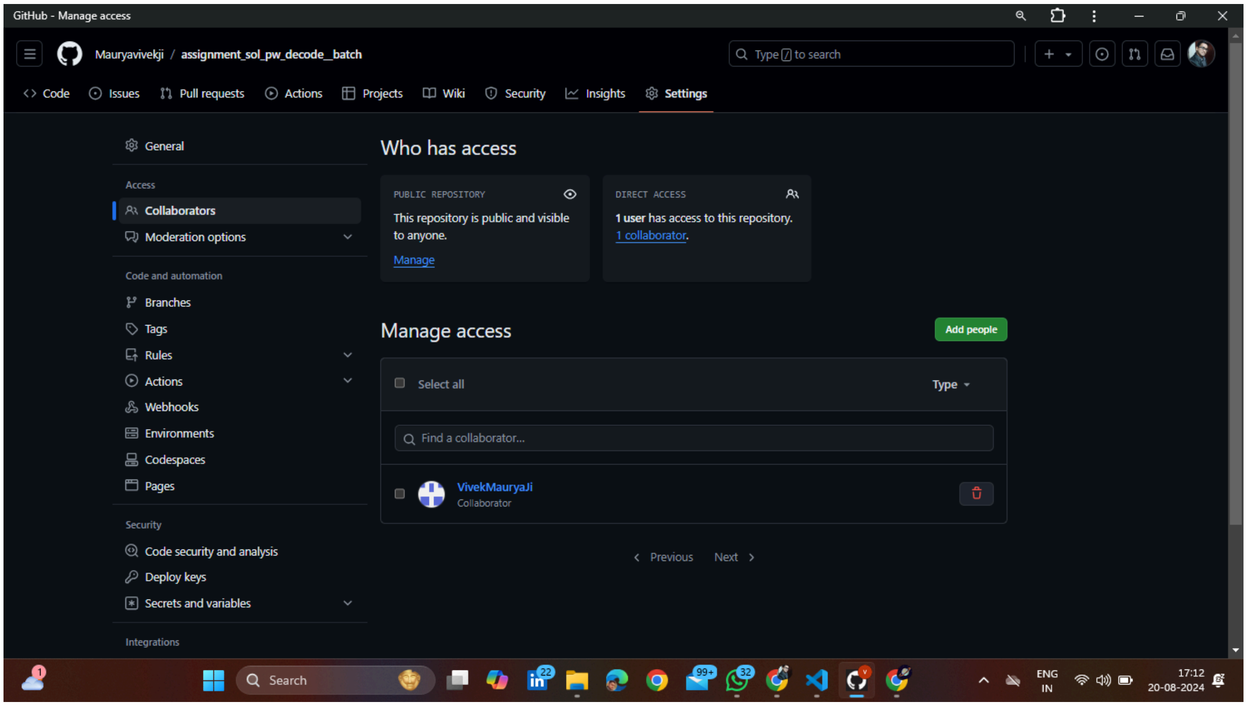
Manage

DIRECT ACCESS

1 user has access to this repository. 0 collaborators. 1 invitation.

Manage access

VivekMauryaJi
Awaiting VivekMauryaJi's response
Pending Invite



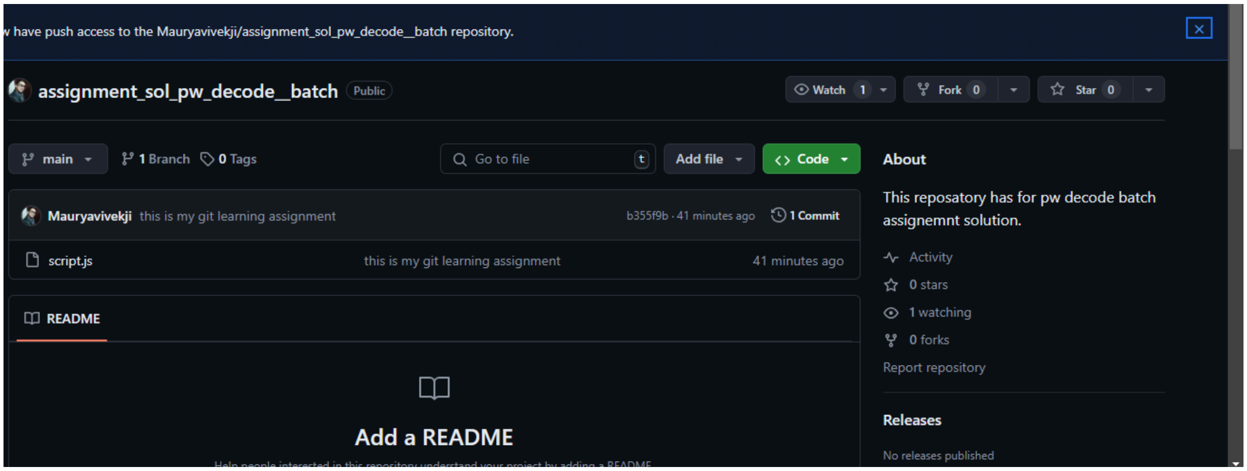
Who has access

PUBLIC REPOSITORY
This repository is public and visible to anyone.
Manage

DIRECT ACCESS
1 user has access to this repository. 1 collaborator.

Manage access

VivekMauryaJi
Collaborator



w have push access to the Mauryavivekji/assignment_sol_pw_decode_batch repository.

Mauryavivekji this is my git learning assignment b355f9b · 41 minutes ago 1 Commit

script.js this is my git learning assignment 41 minutes ago

This reposatory has for pw decode batch assignemnt solution.

Add a README

Help people interested in this repository understand your project by adding a README.

No releases published

Task 1: Initialize a Git repository

1. Open your terminal or command prompt.
2. Navigate to your project directory.

Run the following command to initialize a Git repository:
bash

```
git init
```

3.

Task 2: Create a file named `script.js`

1. In your project directory, create a new file named `script.js`.

Add a simple JavaScript function to the file. For example:

javascript

```
function greet() {
    console.log("Hello, World!");
}
```

2.

Task 3: Add and commit the changes

Stage the new file:

bash

```
git add script.js
```

1.

Commit the changes with a message:

bash

```
git commit -m "Add script.js with a greeting function"
```

2.

Creating a New Feature Branch

Task 1: Create a new branch named `feature-greeting`

In your terminal, run:

bash

```
git branch feature-greeting
```

1.

Task 2: Switch to the new branch

Switch to the `feature-greeting` branch:
bash

```
git checkout feature-greeting
```

1.

Task 3: Add a new function in `script.js` for a personalized greeting

Edit `script.js` to add the following function:
javascript

```
function personalizedGreet(name) {
    console.log(`Hello, ${name}!`);
}
```

1.

Task 4: Commit the changes in the feature branch

Stage and commit your changes:
bash

```
git add script.js
git commit -m "Add personalizedGreet function"
```

1.

Task 5: Merge `feature-greeting` into the main branch

Switch back to the main branch:
bash
e

```
git checkout main
```

1.

Merge the `feature-greeting` branch:
bash

```
git merge feature-greeting
```

2.

Task 6: Switch back to the main branch

Ensure you are on the main branch:
bash

```
git checkout main
```

1.

Working with GitHub

Task 1: Create a GitHub repository

1. On GitHub, create a new repository.
2. Follow the instructions provided by GitHub to link your local repository.

Task 2: Push your local repository to the GitHub repository

Add the GitHub remote repository:
bash

```
git remote add origin <your-github-repo-url>
```

1.

Push your changes to GitHub:
bash

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

2.

Task 3: Invite a team member to collaborate

1. On GitHub, go to the repository settings.
2. Invite your team member by adding them as a collaborator.

Task 4: Team member clones the repository and makes a change

Your team member should clone the repository:
bash

```
git clone <your-github-repo-url>
```

1.
2. They can then make a change, e.g., editing `script.js`.

Task 5: Fetch and merge the changes made by your team member

Fetch the changes from the remote repository:
bash

```
git fetch origin
```

1.

Merge the changes:
bash

```
git merge origin/main
```

2.

Handling Branches and Merge Conflicts

Task 1: Create a new branch named `feature-update`

Create the branch:
bash

```
git branch feature-update
```

1.

Task 2: Switch to the new branch and make changes to `script.js`

Switch to the branch:
bash

```
git checkout feature-update
```

1.
2. Make some changes in `script.js`.

Task 3: Commit the changes in the feature branch

Stage and commit your changes:
bash

```
git add script.js
git commit -m "Update script.js in feature-update branch"
```

1.

Task 4: Switch back to the main branch and make different changes to `script.js`

Switch to the main branch:
bash

```
git checkout main
```

1.
2. Make different changes in `script.js`.

Task 5: Commit the changes in the main branch

Stage and commit your changes:
bash

```
git add script.js
git commit -m "Update script.js in main branch"
```

1.

Task 6: Attempt to merge `feature-update` into the main branch

Attempt the merge:
bash

```
git merge feature-update
```

1.
2. If a conflict occurs, Git will indicate which files have conflicts.

Task 7: Resolve any merge conflicts

1. Open the files with conflicts.
2. Edit the files to resolve the conflicts, then save them.

Stage and commit the resolved changes:
bash

```
git add script.js
git commit -m "Resolve merge conflict between main and feature-update"
```

3.

Managing Obsolete Files

Task 1: Create a new file named `obsolete.js`

1. Create `obsolete.js` in your project directory.

Task 2: Add and commit the file to the repository

Stage and commit the file:
bash

```
git add obsolete.js
git commit -m "Add obsolete.js"
```

1.

Task 3: Realize that `obsolete.js` is unnecessary and should be removed

Remove the file:
bash

```
git rm obsolete.js
```

1.

Task 4: Undo the last commit where `obsolete.js` was added

1. Reset the last commit:
   bash

   ```
   git reset --hard HEAD~1
   ```