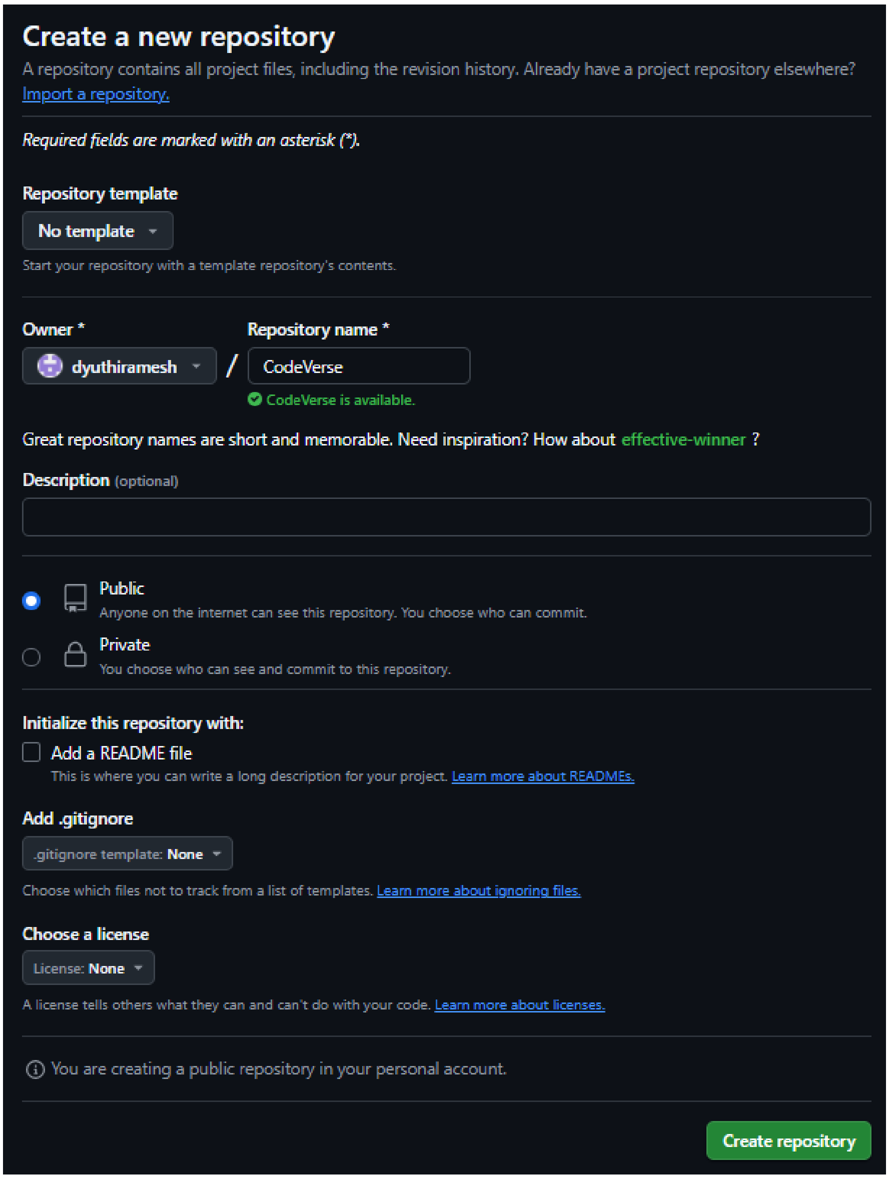
CodeVerse – GitHub guide (using Git)

Step 1: Create a New Repository on GitHub

1. Go to GitHub
2. Click **New** repository.
3. Repository name: CodeVerse
4. Description: (optional)
5. Keep it **Public**.
6. Do **not** initialize with a README (since you have local files).
7. Click **Create repository**.

You will now see some commands GitHub suggests. Keep this page open.



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 (*).

Repository template

No template

Start your repository with a template repository's contents.

Owner * : dyuthiramesh / Repository name * : CodeVerse

CodeVerse is available.

Great repository names are short and memorable. Need inspiration? How about effective-winner ?

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

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.

Step 2: Set up Git locally

Assuming you already have Git installed (if not, install from here).

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

```
>>cd path/to/your/CodeVerse
```

For example:

```
>>cd ~/Desktop/CodeVerse
```

3. Initialize Git:

```
>>>git init
```

4. Add your files to the staging area:

```
>>>git add .
```

5. Commit the files:

```
>>>git commit -m "Initial commit of CodeVerse with Week1 and Week2 structure"
```

Step 3: Connect to GitHub repository

From the GitHub page, copy the repo URL.

If you chose **HTTPS**, it will look like:

```
>>>git remote add origin https://github.com/your-username/CodeVerse.git
```

If you prefer **SSH** (requires SSH setup):

```
>>>git remote add origin git@github.com:your-username/CodeVerse.git
```

Step 4: Push the code

Now push your code to GitHub:

```
>>>git branch -M main
```

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

✔ Done! Your CodeVerse repo is now on GitHub 🎉

Optional: Add a README and .gitignore (Recommended)

Before pushing, you can also create:

- A README.md (to explain the project)
- A .gitignore file (to exclude files like __pycache__/, .DS_Store, etc.)

✅ Steps to Update the Repo Later

Whenever you add new files or folders to your CodeVerse project locally (e.g., Week3/SampleProblem.py), follow these steps:

1. **Open your terminal and navigate to the project:**

```
>>>cd path/to/your/CodeVerse
```

2. **Check the status to see new files or changes:**

```
>>>git status
```

You'll see untracked files (new files/folders you've added).

3. **Stage the new files:** To add **all** new/modified files:

```
>>>git add .
```

Or, to add specific files:

```
>>>git add Week3/SampleProblem.py
```

4. **Commit the changes:** Write a meaningful commit message:

```
>>>git commit -m "Added Week3 with SampleProblem.py"
```

5. **Push to GitHub:**

```
>>>git push
```

✔️ Done! Changes are live on your GitHub repository 🎉

🧩 Pro Tip: If you just want to see what changed before committing:

```
>>>git diff
```

This shows the actual changes inside the files.

🚀 Quick Shortcut for Regular Workflow

If you're frequently updating, here's a quick 3-command flow:

```
>>>git add .
```

```
>>>git commit -m "Your message here"
```

```
>>>git push
```