Learning Objectives
By the end, learners will be able to:

1. Authenticate to GitHub from the command line (SSH or HTTPS).
2. Create a remote repository on GitHub for **the mini_finance project**.
3. Clone it locally.
4. Make a code change, pull new updates, and push your change.
5. Open a Pull Request to propose your modification.

Steps to Complete

Step 0 — Access Existing Mini-Finance Code

The upstream repository exists at:

`https://github.com/pravinmishraaws/mini_finance`

We'll fork this repository to your GitHub account, clone it, and work with it.

Step 1 — Fork & Authenticate

1. Log in to GitHub and **Fork** the `mini_finance` repository into your account.
2. In your terminal, configure authentication if not already set:

   SSH (recommended):

   ```
   ssh-keygen -t ed25519 -C "your.email@example.com"
   eval "$(ssh-agent -s)"
   ssh-add ~/.ssh/id_ed25519
   # Copy contents of ~/.ssh/id_ed25519.pub into your
   GitHub account (Settings → SSH and GPG keys)
   git config --global url."git@github.com:".insteadOf
   "https://github.com/"
   ```

   OR HTTPS:

   ```
   git config --global credential.helper cache
   ```

3. Test with:

   ```
   git ls-remote git@github.com:yourusername/mini_finance.git
   ```

4. **Expected outcome:** You have forked the repo and your terminal is authenticated for Git ops.

Step 2 — Clone Your Fork Locally

```
git clone git@github.com:yourusername/mini_finance.git
cd mini_finance
git remote -v
```

- `origin` should point to your fork.
- Add an `upstream` remote to the original repo:

  ```
  git remote add upstream
  https://github.com/pravinmishraaws/mini_finance.git
  ```

Expected outcome: Local clone with `origin` (your fork) and `upstream` properly set.

Step 3 — Create a Feature Branch & Make a Change

1. Create a new branch:

   ```
   git checkout -b feature-readme-update
   ```

2. Open `README.md` and add a new section:
   You may write: “This project demonstrates Git operations like clone, pull, push, PR—a hands-on Mini-Finance tool.”
3. Save, then stage and commit:

```
git add README.md
git commit -m "docs: update README with assignment note"
```

Step 4 — Pull From Upstream & Push to Origin

1. Sync changes from upstream’s `main`:

```
git fetch upstream
git checkout main
git merge upstream/main
```

2. Switch back to your feature branch:

```
git checkout feature-readme-update
git rebase main     # optional but recommended
```

3. Push your branch to your fork:

```
git push -u origin feature-readme-update
```

Expected outcome: Your feature branch is available on GitHub under your fork.

Step 5 — Create a Pull Request

1. Go to your fork on GitHub.
2. Click **Compare & Pull Request**.
3. Make sure it’s targeting `pravinmishraaws/mini_finance:main` from your `feature-readme-update` branch.
4. Title: “docs: update README with assignment note”
5. In the body, add a short description:
   “This PR adds a new section to the README explaining the project's purpose in the context of this GitHub assignment.”
6. Submit the **Pull Request**.

How to Submit Your Assignment?

Live Cohort Students:

If you are part of the live cohort, please follow the assignment submission guidelines as explained here.

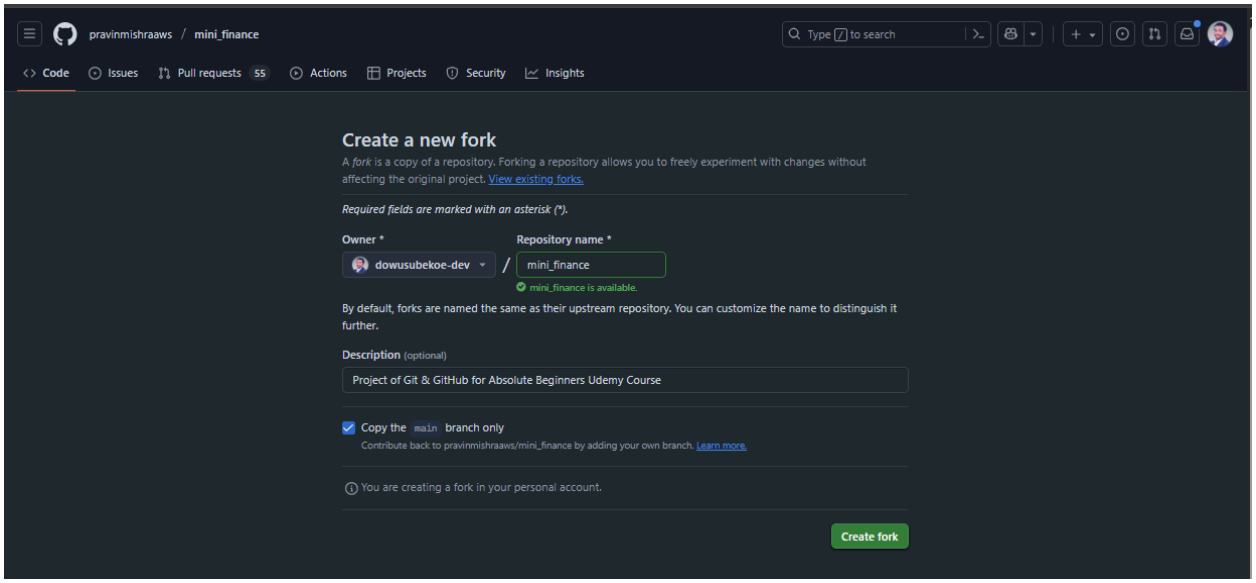
Self-Paced on Udemy:

If you are taking the course self-paced on Udemy, submit the assignment by clicking **“Next”** and answering the provided questions.

Questions for this assignment

Fork & Clone (Screenshot + Explanation)

- Screenshot A: The GitHub page showing your **forked mini_finance repo** under your account (URL visible).



- Screenshot B: Terminal output of git remote -v after cloning, showing both origin and upstream.

```
automate-iac@windows-main:/mnt/e/main-pod/1 - cloud-computing_devops/courses/aws/devops-by-pravin-mishra/mini_finance$ git remote -v
origin  git@github.com:dowusubekoe-dev/mini_finance.git (fetch)
origin  git@github.com:dowusubekoe-dev/mini_finance.git (push)
upstream        https://github.com/pravinmishraaws/mini_finance.git (fetch)
upstream        https://github.com/pravinmishraaws/mini_finance.git (push)
automate-iac@windows-main:/mnt/e/main-pod/1 - cloud-computing_devops/courses/aws/devops-by-pravin-mishra/mini_finance$
```

- Text (1–2 lines): What is the purpose of the upstream remote in your workflow?

The upstream remote is the link to the original repository from which you forked or cloned. Its purpose is to let you pull in new changes and updates from that source, so your copy doesn't fall behind. It's like keeping your notebook synced with the teacher's master copy.

Branching, Rebase, Push (Screenshot + Explanation)

- Screenshot C: Terminal showing your commit on feature-readme-update. (e.g., git log --oneline -n 3)

```
automate-iac@windows-main mini_finance (feature-readme-update =)$ git log --oneline -n 3
47a792f (HEAD -> feature-readme-update, origin/feature-readme-update) docs: update README with assignment note
c77e2b1 (upstream/main, origin/main, origin/HEAD, main) Merge pull request #50 from rut-a/feature-readme-update
6faa8a1 docs: update README with assignment note
```

- Screenshot D: Confirmation of git push -u origin feature-readme-update.

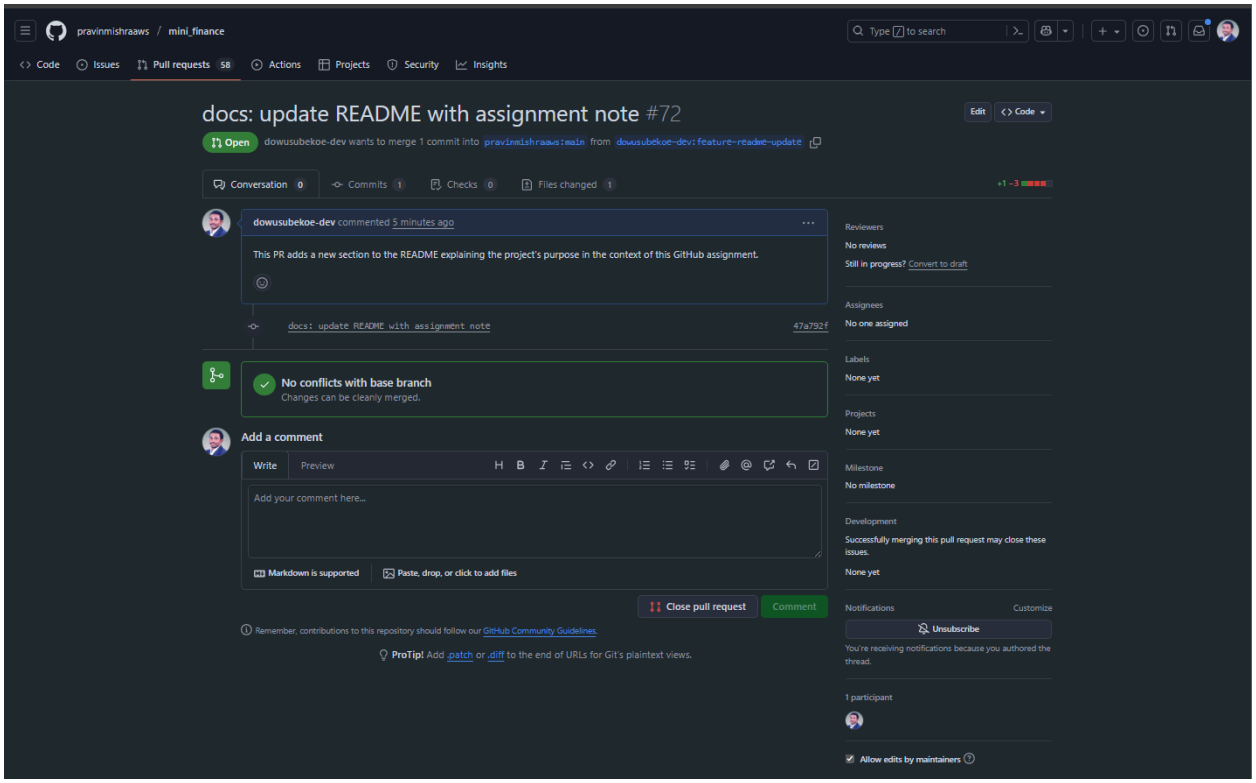
```
automate-iac@windows-main mini_finance (feature-readme-update)$ git push -u origin feature-readme-update
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 347 bytes | 28.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'feature-readme-update' on GitHub by visiting:
remote:      https://github.com/dowusubekoe-dev/mini_finance/pull/new/feature-readme-update
remote:
To github.com:dowusubekoe-dev/mini_finance.git
 * [new branch]      feature-readme-update -> feature-readme-update
branch 'feature-readme-update' set up to track 'origin/feature-readme-update'.
```

- Text (1–2 lines): Why did you rebase or merge from upstream/main before pushing?

To ensure your branch has the latest updates from the main project before pushing. Rebasing or merging from `upstream/main` prevents conflicts, keeps your work aligned with everyone else's changes, and avoids breaking the codebase when your updates get added.

Pull Request (Screenshot + Text)

- Screenshot E: The GitHub UI showing your **open Pull Request** (title and description visible).



- Text (1–2 lines): Why is creating a Pull Request an important step in team collaboration?

Creating a **Pull Request** is vital because it lets teammates review, discuss, and approve changes before merging them, ensuring code quality and preventing errors. It also keeps the project history transparent and collaborative.