

# Git Collaboration Project Report

## Part 2: Git Collaboration Project:

### 1. Project Setup:

For this part of the project, I followed the steps outlined to initialise a new Git repository and set up the project on GitHub.

Here's a detailed breakdown of the process:

**1.1.** I navigated to my local project directory using the terminal and executed the command `git init` to initialise a new Git repository.

**1.2.** In the project directory "ASSIGNMENT3", I created a Python script named "AMAZON WEB SCRAPER.ipynb" that included tasks to scrape contents from a website and store it in csv on a daily basis.

**1.3.** Using the commands `<git add .>` and `<git commit -m "Initial commit">`, I staged and committed the initial files to the repository with a meaningful commit message.

**1.4.** Next, I visited GitHub and created a new repository named "Assignment3".

**1.5.** To link my local repository to the GitHub repository, I executed the command `<git remote add origin <GitHub_repo_url>>`.

**1.6.** Finally, I pushed the contents of my local repository to GitHub's main branch using the command `<git push -u origin main>`.

## 2. Branching and Development:

In this phase, I focused on branching off for feature development, making commits with appropriate messages, and ensuring a seamless workflow:

**2.1.** To work on a new email sending script, I created a new branch called "feature\_branch" using the command **<git checkout -b feature\_branch>**.

**2.2.** While on the "feature\_branch," I made incremental changes to the "AMAZON WEB SCRAPER.ipynb" script, implementing the email sending feature.

**2.3.** At each point of development, I used the command **<git add .>** to stage the changes and follow it up with a **<git commit -m "Adding Email sending feature">** to commit the changes along with a descriptive message.

### Adding E-mail sending feature #1

**Merged** Aishwarya-j10 merged 1 commit into `main` from `feature_branch` 12 hours ago

Conversation 0 Commits 1 Checks 0 Files changed 1

Commits on Aug 26, 2023

Adding E-mail sending feature

Aishwarya-j10 committed 8 minutes ago

### **3. Collaboration Simulation:**

This stage involved simulating collaboration by involving a peer, performing pull requests, and understanding the pull request review process:

#### **3.1. Cloning the repository:**

I cloned the "Assignment3" repository to a directory on my peer's local machine using the command **<git clone <GitHub\_repo\_url>>**.

#### **3.2. Creating a New Branch:**

Changed to the repository's directory and created a new branch **<peer\_feature\_branch>**.

- **cd <repository\_directory>**
- **git checkout -b peer\_feature\_branch**

#### **3.3. Making Changes and Commit:**

Made the necessary changes to the "AMAZON WEB SCRAPER.ipynb" script. After making changes, committed them with informative commit messages:

- **git add .**
- **git commit -m "Updated Amazon Webscraper.ipynb"**

#### **3.4. Pushing to GitHub:**

Pushed the changes to the remote repository on GitHub:

- **git push origin peer\_feature\_branch**

#### **3.5. Creating a Pull Request:**

On the GitHub website, I navigated to my repository and generated a pull request by clicking on "New Pull Request" under the Pull Requests tab. Setting the base branch to "main", and comparing the branch to "peer\_feature\_branch", also provided the title and description of the pull request.

## Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: main ← compare: peer\_feature\_branch ✓ Able to merge. These branches can be automatically merged.

Updated amazon webscrapper.ipynb

Write

Preview

H B I ≡ <> 🔗 ≡ ≡ ≡ @ ↻ ↶

Merging peer\_feature\_branch to main.

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

Open Updated amazon webscrapper.ipynb #2

Aishwarya-j10 wants to merge 1 commit into main from peer\_feature\_branch

Pull request accepted.

Add more commits by pushing to the peer\_feature\_branch branch on Aishwarya-j10/Assignment3.

Merge pull request #2 from Aishwarya-j10/peer\_feature\_branch

Updated amazon webscrapper.ipynb

This commit will be authored by 123799540+ Aishwarya-j10@users.noreply.github.com

Confirm merge Cancel

## 3.6. Reviewing and Merging:

On my side as the repository owner, I reviewed the pull request, provided feedback and comments and approved the pull request and merged it.

Merged Updated amazon webscrapper.ipynb #2

Aishwarya-j10 merged 1 commit into main from peer\_feature\_branch 12 hours ago

Merging peer\_feature\_branch to main.

Updated amazon webscrapper.ipynb c69dd56

Aishwarya-j10 commented 12 hours ago

Owner Author

Pull request accepted.

Aishwarya-j10 merged commit 2d574ec into main 12 hours ago

Revert

Pull request successfully merged and closed

You're all set—the peer\_feature\_branch branch can be safely deleted.

Delete branch

## 4. Handling Merge Conflicts:

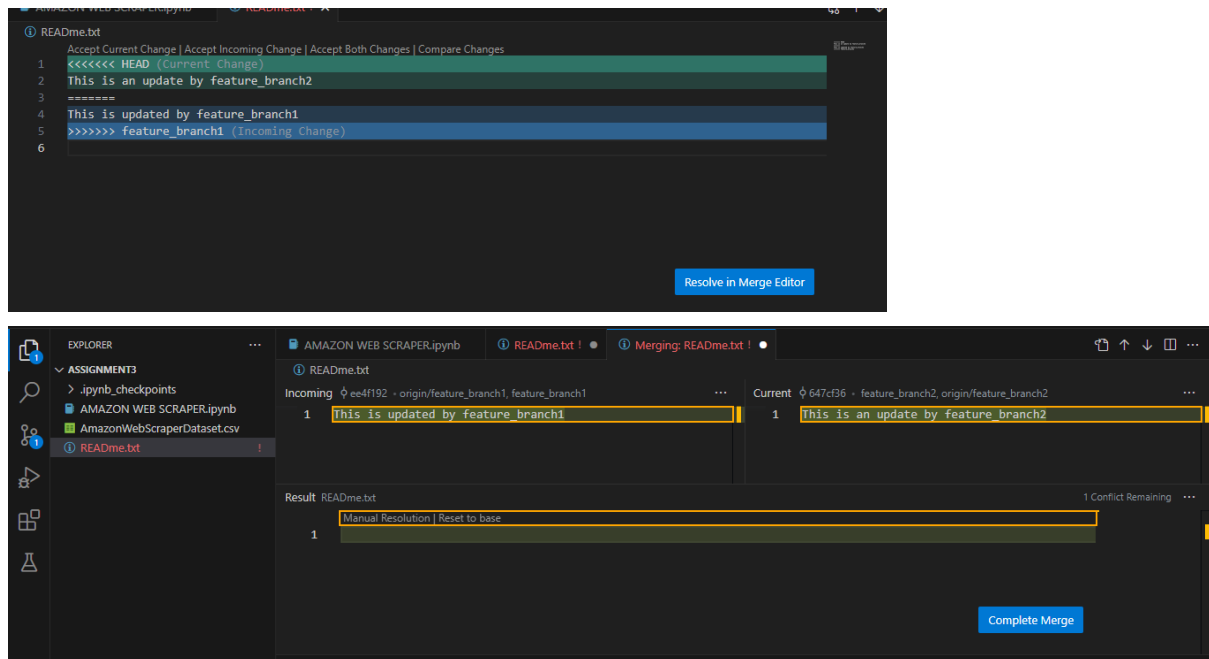
To practise conflict resolution, I purposefully created and resolved a merge conflict:

**4.1.** To simulate a merge conflict, I intentionally edited the same part of a file within the "feature\_branch1" and "feature\_branch2."

**4.2.** I attempted to merge the changes from "feature\_branch1" into "feature\_branch2" using the command **<git merge feature\_branch1>**.

**4.3.** This action led to a merge conflict, which I detected by examining the conflicting file.

**4.4.** I navigated to the conflicted file, resolved the conflicting sections manually, and saved the changes.



**4.5.** Subsequently, I used the **<git add . >** command to stage the resolved file and executed **<git commit>** to finalize the merge conflict resolution.

## 5). Rebasing and History Cleanup:

5.1. On a new branch "rebase\_branch," I made several incremental commits.

- **git checkout -b rebase\_branch**

5.2. Making several incremental commits on this branch:

- **git commit -m "commit1"**

- **git commit -m "commit2"**

- **git commit -m "commit3"**

```
PS C:\Users\admin\Downloads\Aishwarya\Week3\ASSIGNMENT3> git checkout -b rebase_branch
Switched to a new branch 'rebase_branch'
PS C:\Users\admin\Downloads\Aishwarya\Week3\ASSIGNMENT3> git add .
PS C:\Users\admin\Downloads\Aishwarya\Week3\ASSIGNMENT3> git commit -m "commit1"
[rebase_branch f69ba02] commit1
1 file changed, 3 insertions(+), 1 deletion(-)
PS C:\Users\admin\Downloads\Aishwarya\Week3\ASSIGNMENT3> git add .
PS C:\Users\admin\Downloads\Aishwarya\Week3\ASSIGNMENT3> git commit -m "commit2"
[rebase_branch b75a4b9] commit2
1 file changed, 2 insertions(+), 1 deletion(-)
PS C:\Users\admin\Downloads\Aishwarya\Week3\ASSIGNMENT3> git add .
PS C:\Users\admin\Downloads\Aishwarya\Week3\ASSIGNMENT3> git commit -m "commit3"
[rebase_branch 2b5c2c8] commit3
1 file changed, 2 insertions(+), 1 deletion(-)
PS C:\Users\admin\Downloads\Aishwarya\Week3\ASSIGNMENT3>
```

5.3. Initiating Interactive Rebase (To reorganise, squash, or modify commits)

- **git rebase -i HEAD~3**

5.4. Squashed commit2 and commit3 as one single commit1 in order to achieve a clean commit history while pushing it to main.

5.5. Switched to main branch :<**git checkout main**>

5.6. Finally, I performed a fast-forward merge by executing <**git merge rebase-branch**>.

