Intro:

Git is a distributed version control system that allows you to track changes in your project over time. It enables multiple developers to collaborate on projects efficiently by managing different versions of the project's history. We can use sourcetree for easily doing the same job as well.
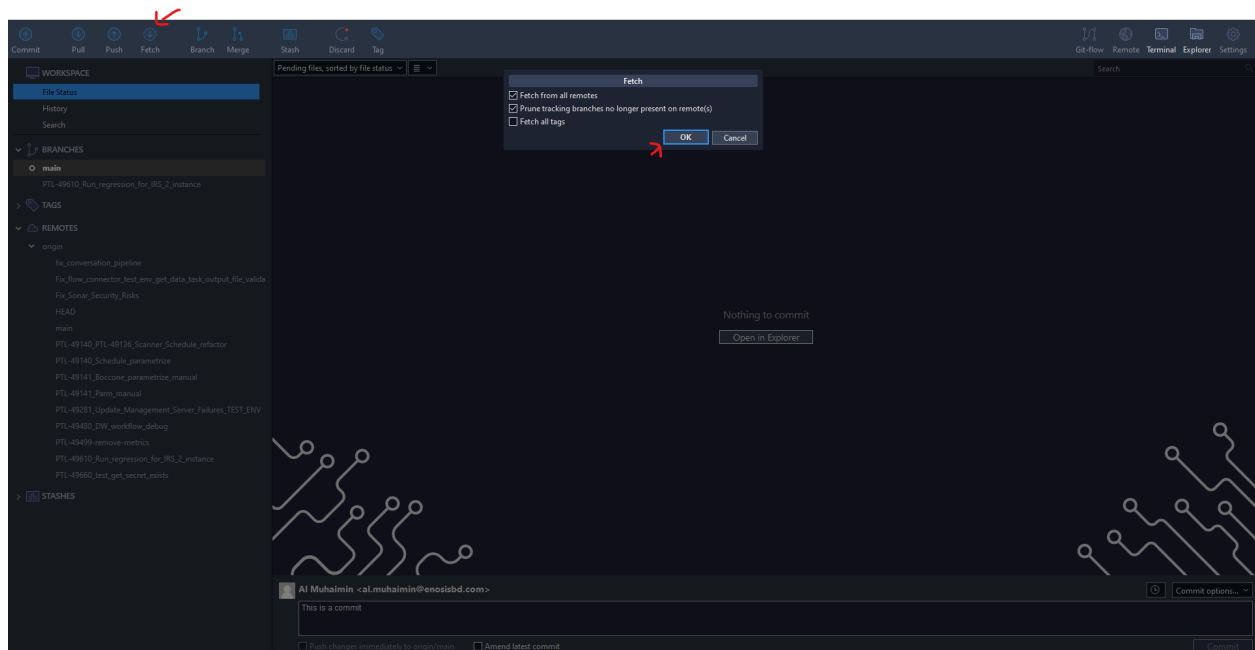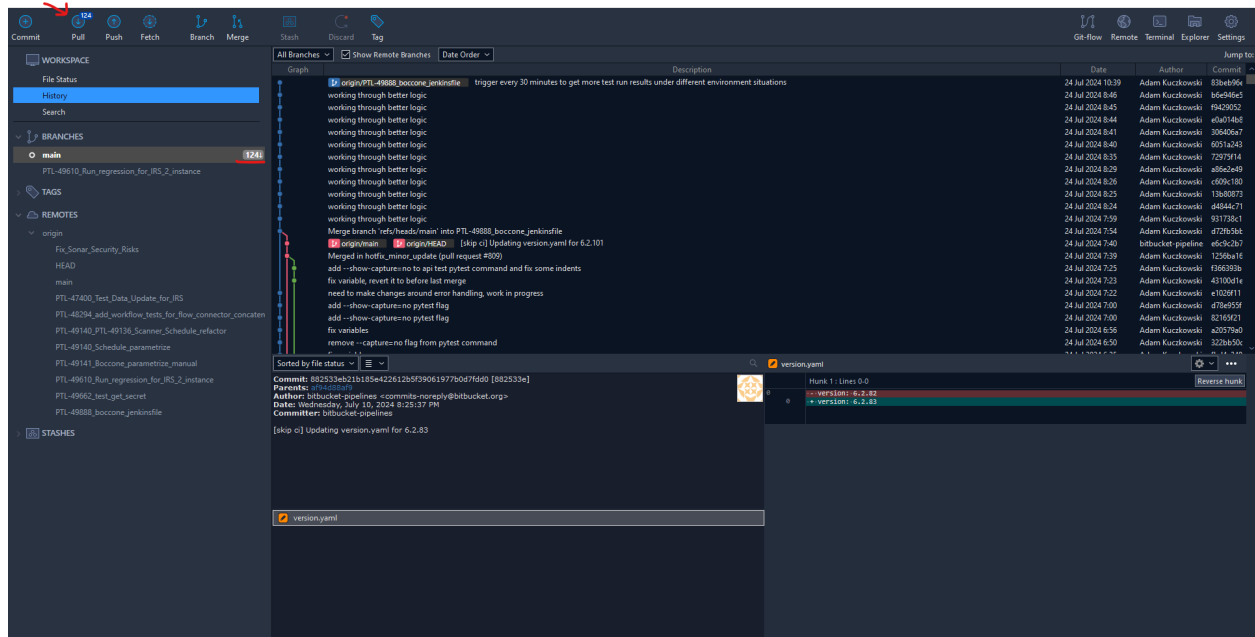
- Fetch

Command:

```
git fetch origin
```

Explanation: Downloads all the branches and commits from the remote repository named origin to your local .git directory. This command does not affect your local working directory.

Attachment from sourcetree:

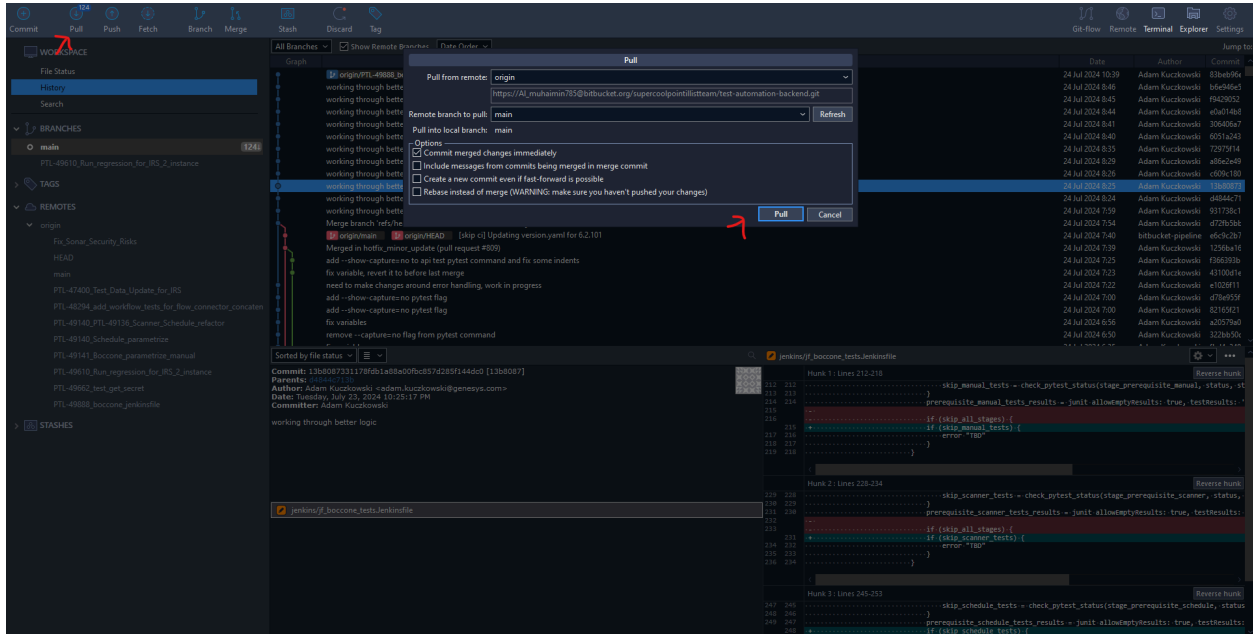When the fetch is completed, changes would appear to the branch such as:

- **Pull**

Command:

git pull

Explanation: Fetches changes from a remote repository and merges them into the current branch.

Attachment from sourcetree: After completing the fetch, we need to pull the changes from the main branch
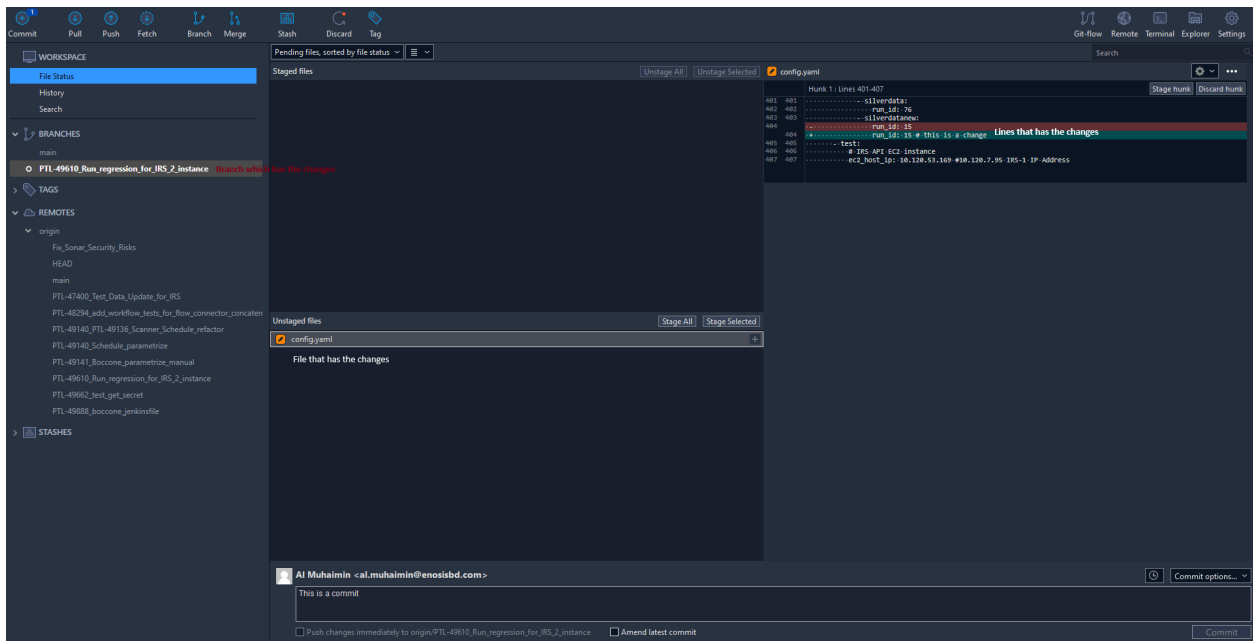
- **Commit**

Command:

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

Explanation: Records changes to the repository with a descriptive message.
Attachment from sourcetree: when a change is made to the code the following appears:
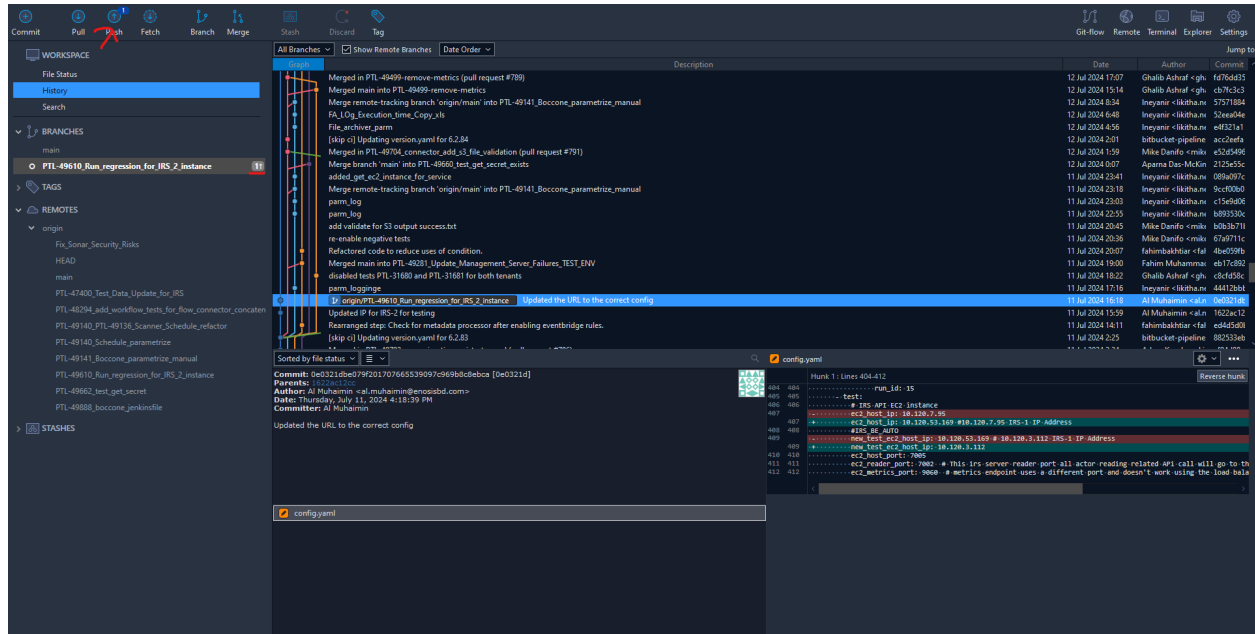
- **Push**

Command:

git push

Explanation: Sends local commits to a remote repository, updating the remote branch.
Attachment from sourcetree: After changes are committed, we need to push the changes to the remote repository

- **Merge**

Command:

git merge main

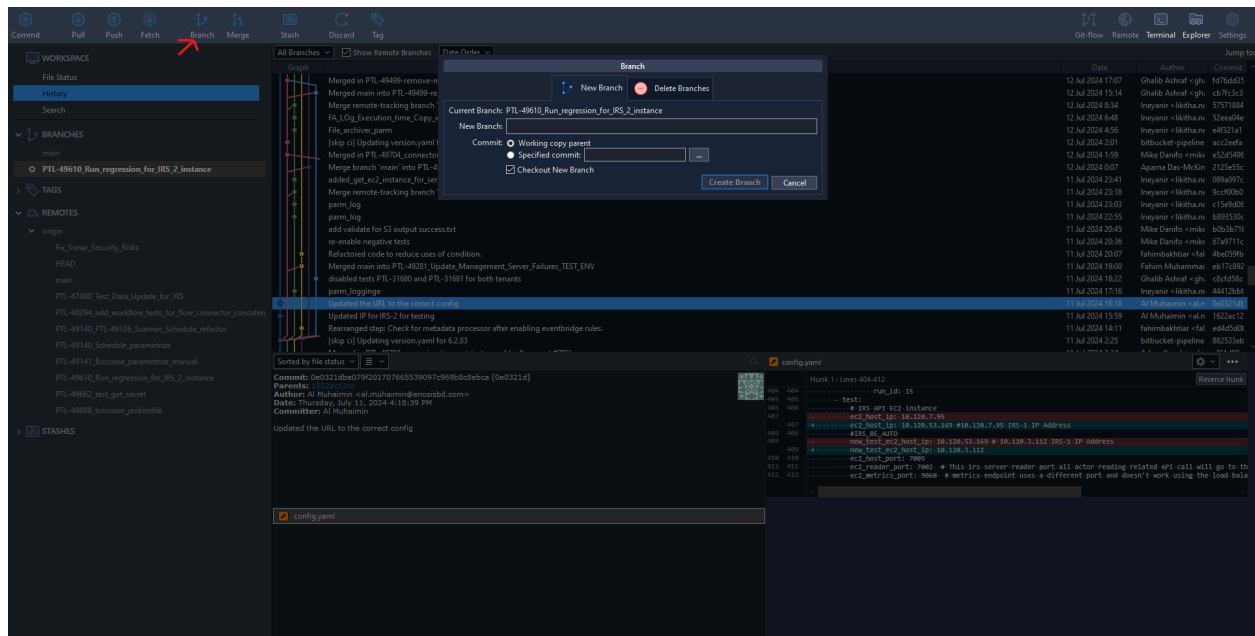**Explanation**: Combines the specified branch into the current branch, integrating changes made in both branches.

- **Create Branch**

Command:

git branch <branch-name>

Explanation: Creates a new branch starting from the current branch.

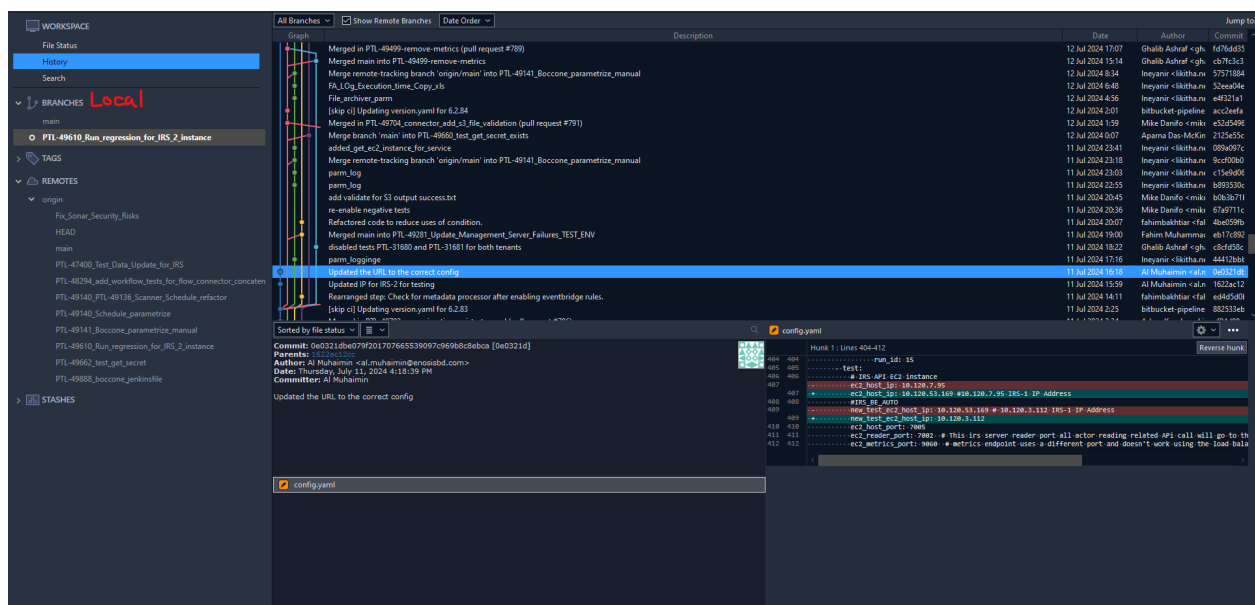**Attachment from sourcetree**: entering an unique branch name will create the branch

- **View All Branches (Local and Remote)**

Command:

```
git branch -a
```

Explanation: Lists all local and remote branches. This command provides a comprehensive view of all branches available in your repository, regardless of their location.

Attachment from sourcetree:

- View Remote Branch

Command:

```
git branch -r
```

Explanation: Lists all remote branches, allowing you to see branches in the remote repository.

- Detailed Information About Branches

Command:

```
git branch -vv or git branch -vva
```

Explanation: Displays detailed information about each branch, including the last commit on the branch and the upstream branch (for remote branches). This can be particularly useful for understanding the status of branches and their relationship to each other.

- Switch to a Specific Branch

Command:

```
git checkout <branch-name>
```

Example: git checkout develop
Explanation: Changes the active branch to <branch-name>, updating the working directory to match the version stored in that branch. This command is essential for navigating between different features or stages of development.

- Create and Switch to a New Branch

Command:

```
git checkout -b <new-branch-name>
```

Example: git checkout -b feature/new-feature
Explanation: Simultaneously creates a new branch named <new-branch> and switches to it. This is a convenient way to start working on a new feature or bug fix without affecting the current branch.

- **Checkout a Remote Branch**

Command:

> **git checkout -b <local-branch> origin/<remote-branch>**

Example: git checkout -b feature/new-feature origin/feature/new-feature
Explanation: Creates a new local branch that tracks a remote branch, allowing you to work on features developed by others or integrate changes from a remote repository.
Attachment from sourcetree: Right click on a remote branch and click check out
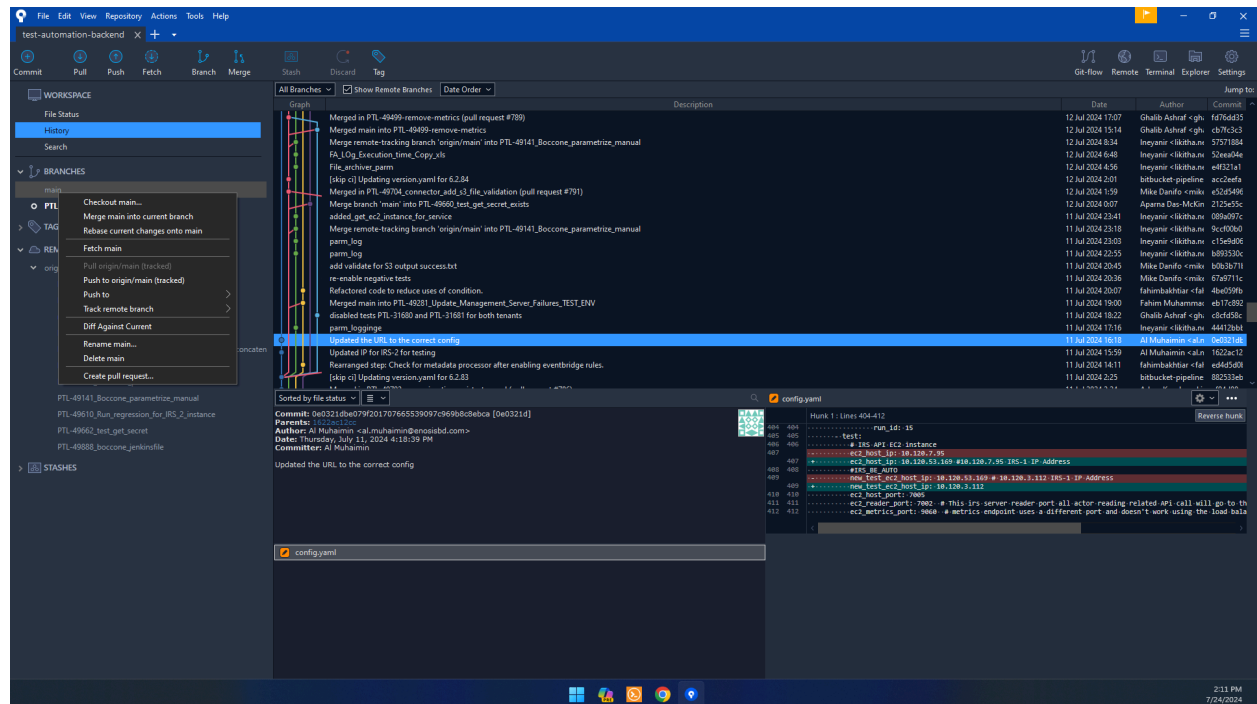
Exercise

Prerequisites

1. Git installed on your computer.
2. Sourcetree installed and set up.
3. Access to a Git repository (you can use an existing one or create a new repository).

Exercise 1: Clone a Repository

1. Open Sourcetree.
2. Click on "Clone/New" at the top left corner.
3. Clone the following repository: https://github.com/Muhaimin785/Git-Exercise.git

4. Choose a local path for the cloned repository.
5. Click "Clone".

Exercise 2: Fetch Changes from the Remote Repository

1. In Sourcetree, select the cloned repository.
2. Right-click on the repository and choose "Fetch".
3. Observe the changes in the repository after the fetch operation completes.

Exercise 3: Pull Changes from the Main Branch

1. With the repository selected, right-click on the main branch (usually named main or master).
2. Choose "Pull".
3. Review the changes pulled from the remote repository.

Exercise 4: Make Some Changes

1. Open the repository folder in your preferred text editor or IDE.
2. Modify some files and save your changes.
3. Step 5: Commit Your Changes
4. Return to Sourcetree and select the modified files.
5. Right-click on the selected files and choose "Commit".
6. Enter a meaningful commit message describing your changes.
7. Click "Commit".

Exercise 6: Push Your Changes to the Remote Repository

1. With the repository selected in Sourcetree, right-click on the main branch.
2. Choose "Push".
3. Review the changes being pushed to the remote repository.

Exercise 7: Create a New Branch

1. In Sourcetree, right-click on the main branch.
2. Hover over "New Branch", then enter a name for your new branch.
3. Press Enter to create the branch.
4. Step 8: Switch to Your New Branch
5. Right-click on the newly created branch.
6. Choose "Checkout".
7. Your working directory now reflects the state of the new branch.

Exercise 9: Merge Changes from the Main Branch

1. Right-click on the main branch again.
2. Choose "Merge Into Current".

3. Confirm the merge operation.

Exercise 10: View All Branches

1. In Sourcetree, click on the "Branches" button at the bottom.
2. Notice the list of all local and remote branches.

Exercise 11: Checkout a Remote Branch

1. Find a remote branch you'd like to work on.
2. Right-click on the remote branch and choose "Checkout".
3. A new local branch tracking the remote branch will be created and checked out.