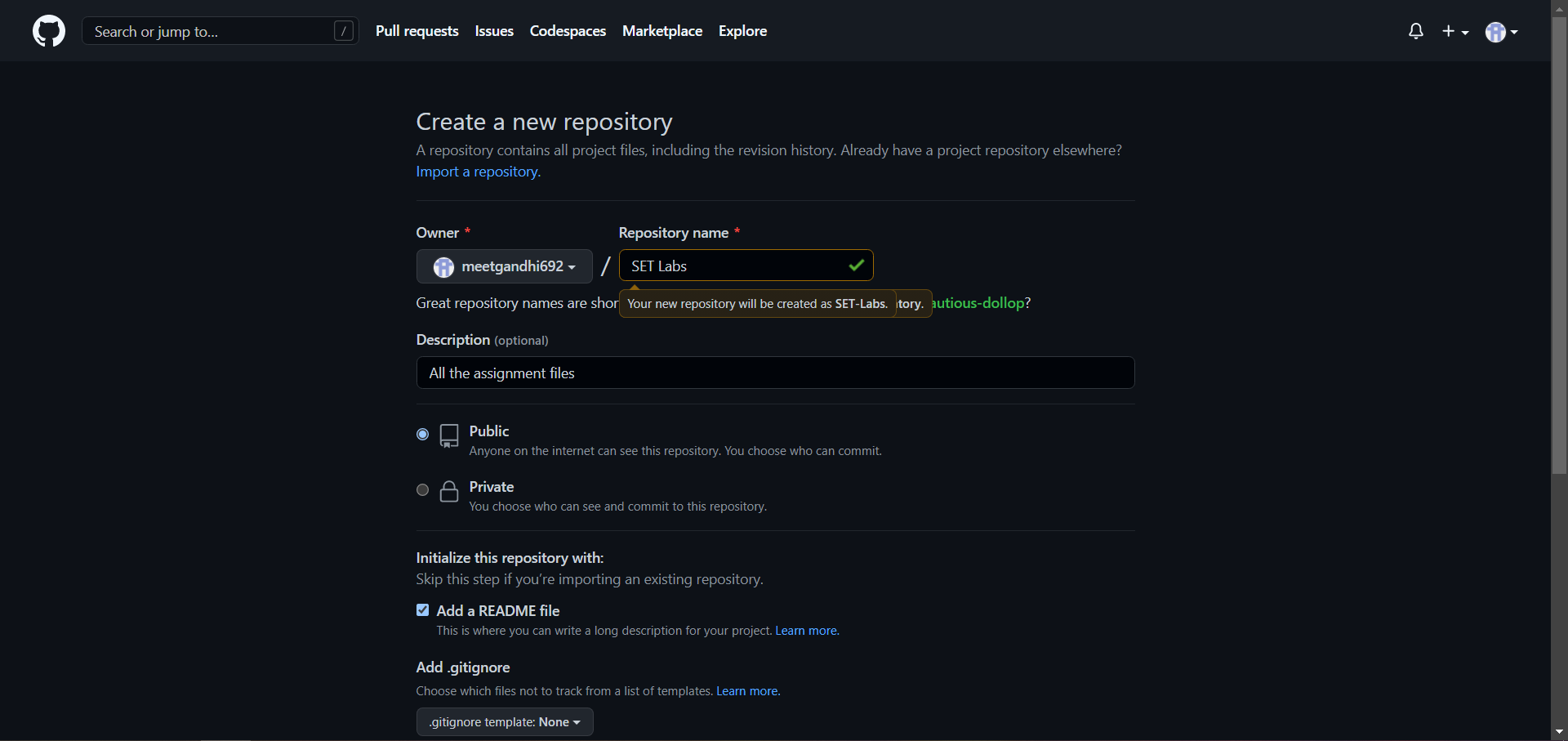
**Name:** Meet Gandhi

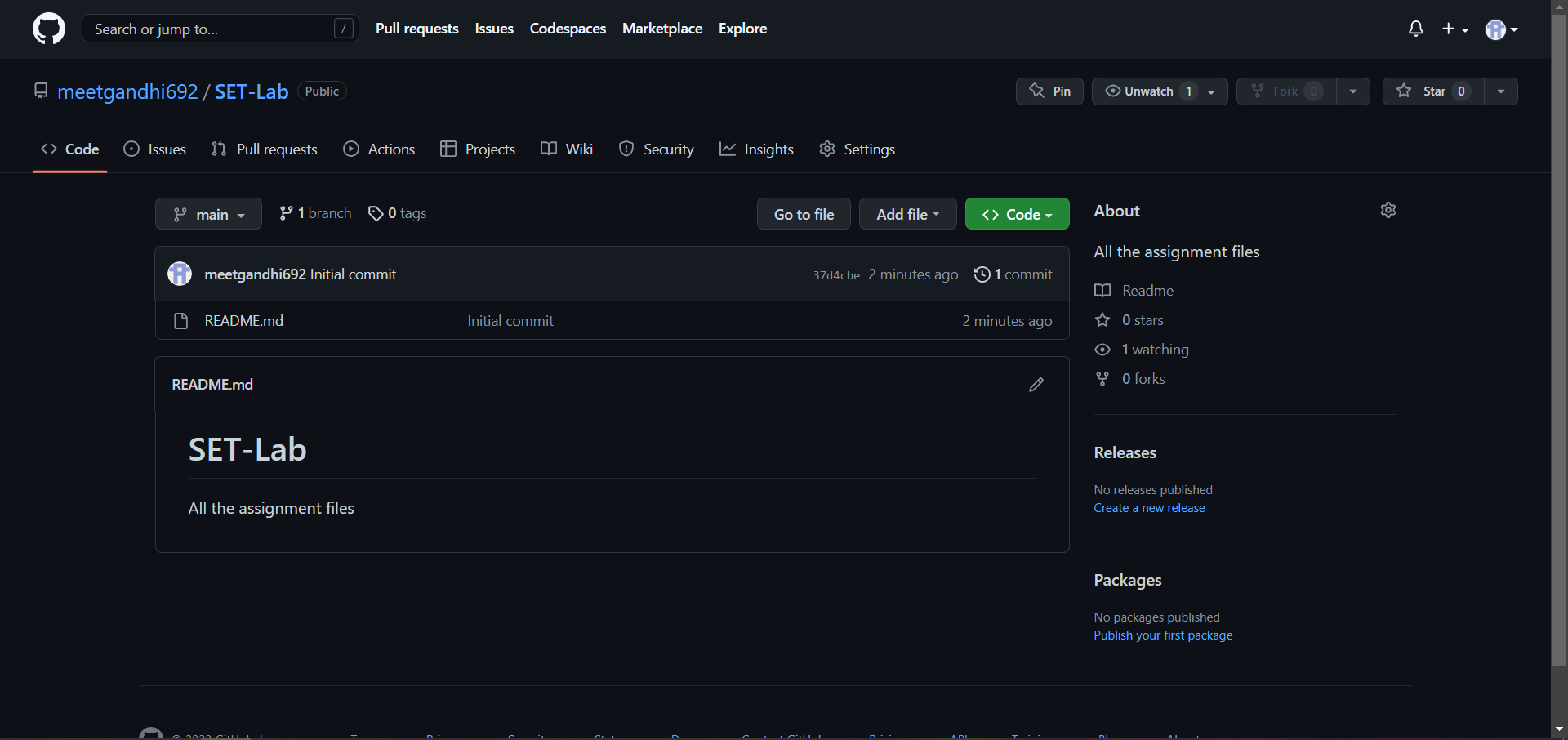
**Batch:** T8

**PRN:** 2020BTECS00112

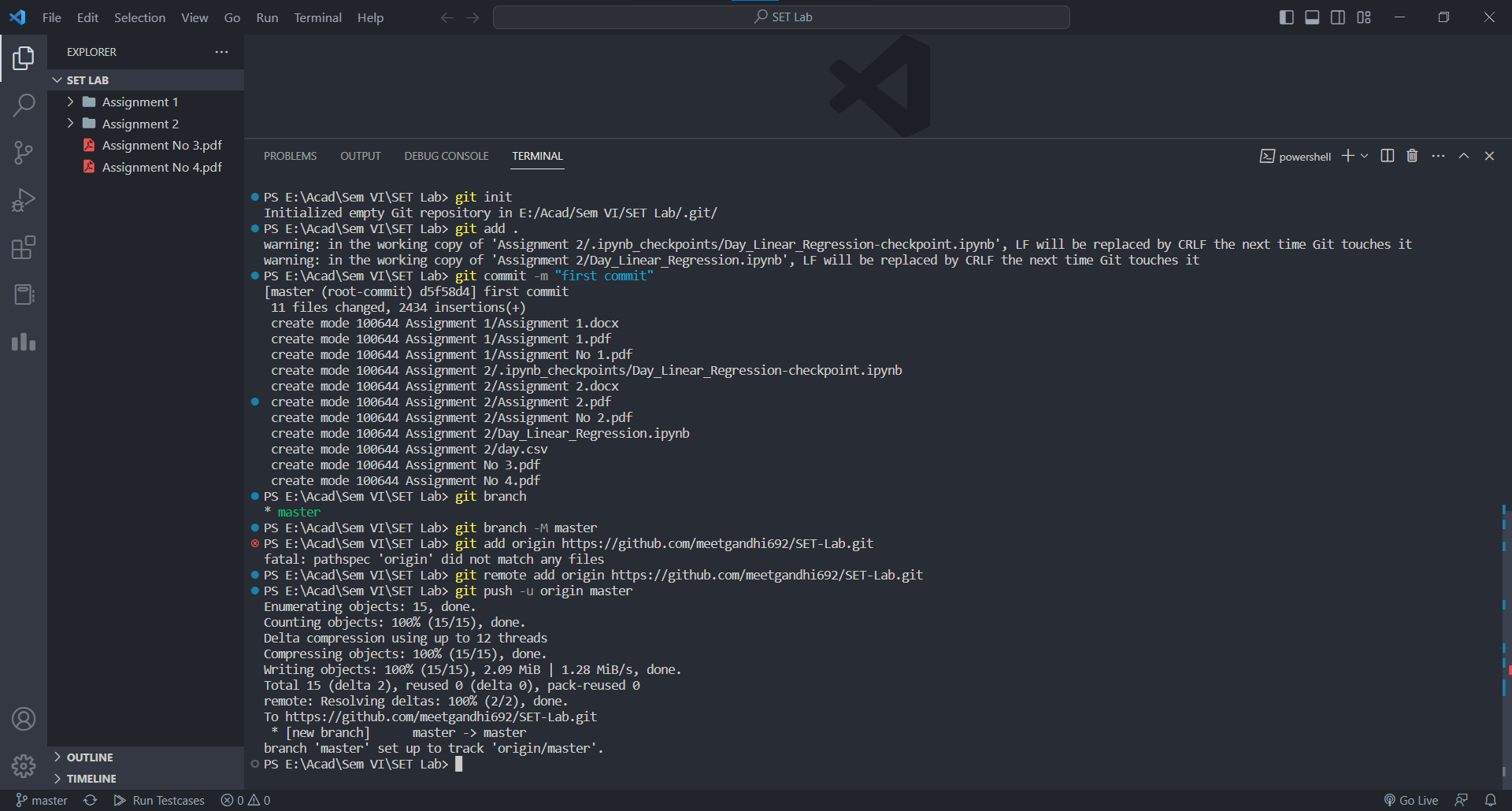
Assignment 4 – GitHub

1. Create a repository on GitHub named SET Lab and add files into it (you can add implementation files of previous assignment) perform below operations on it. (Add screenshot as an answer to every question)

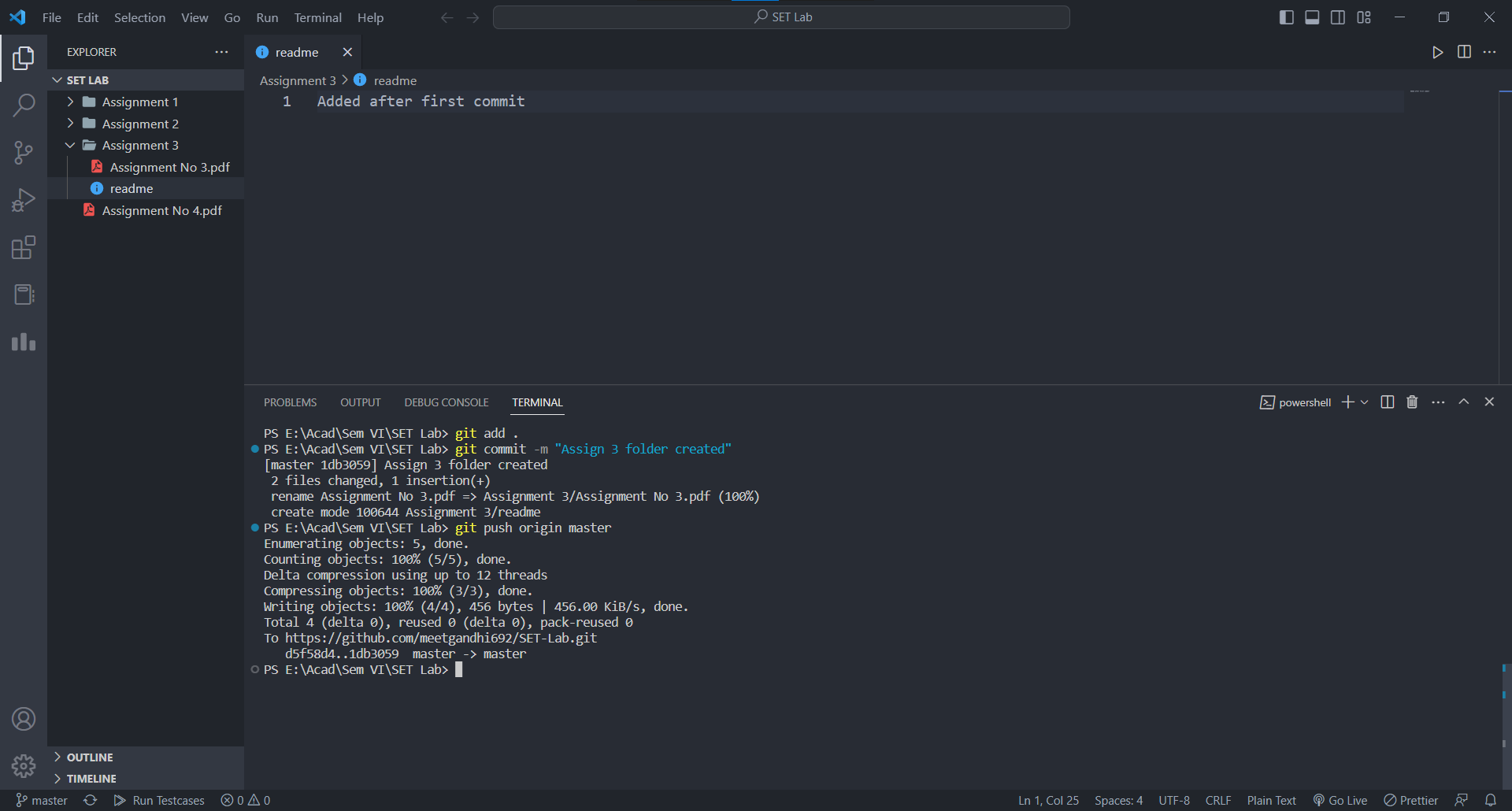




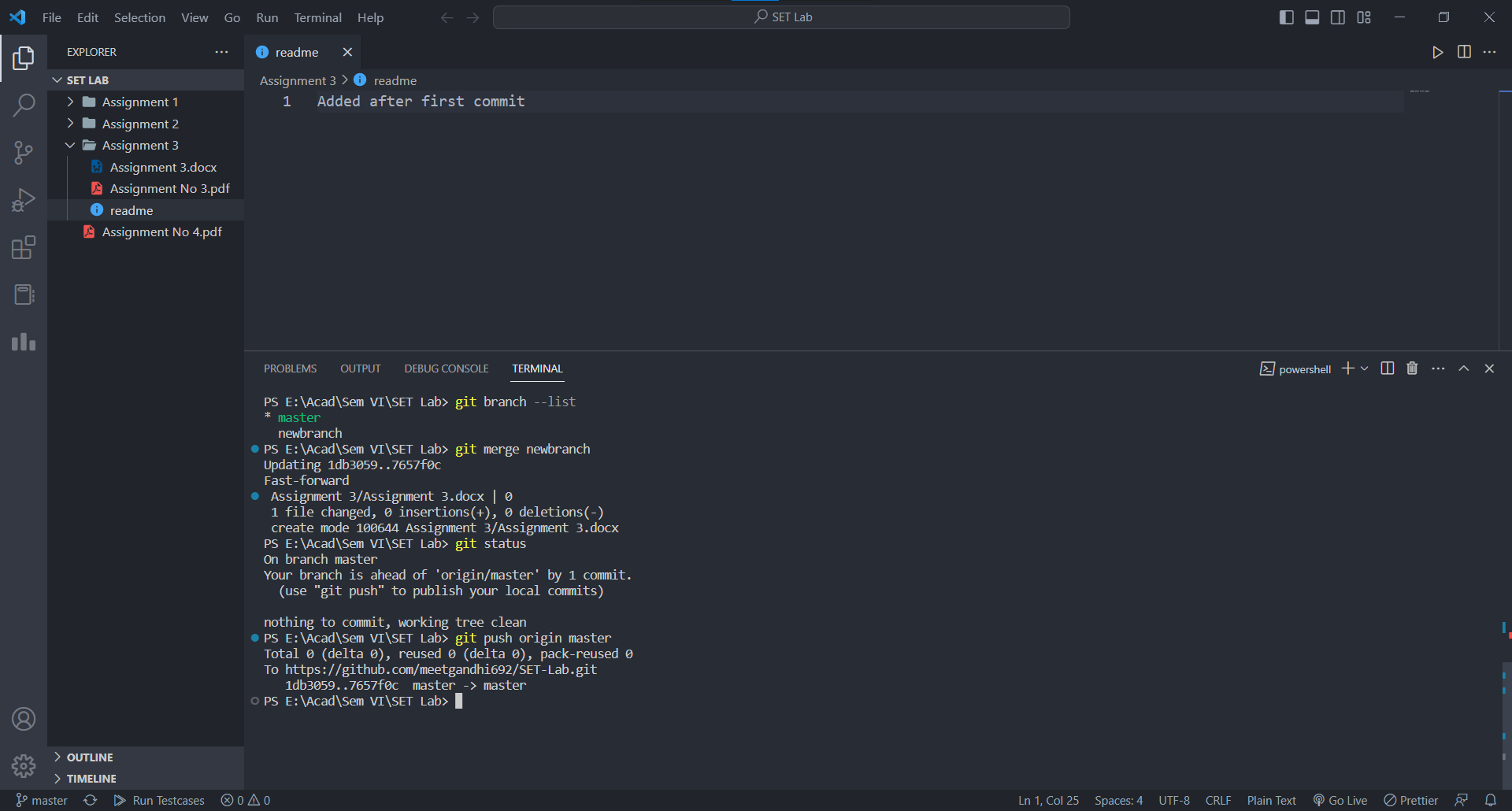
1. Perform commit on added files



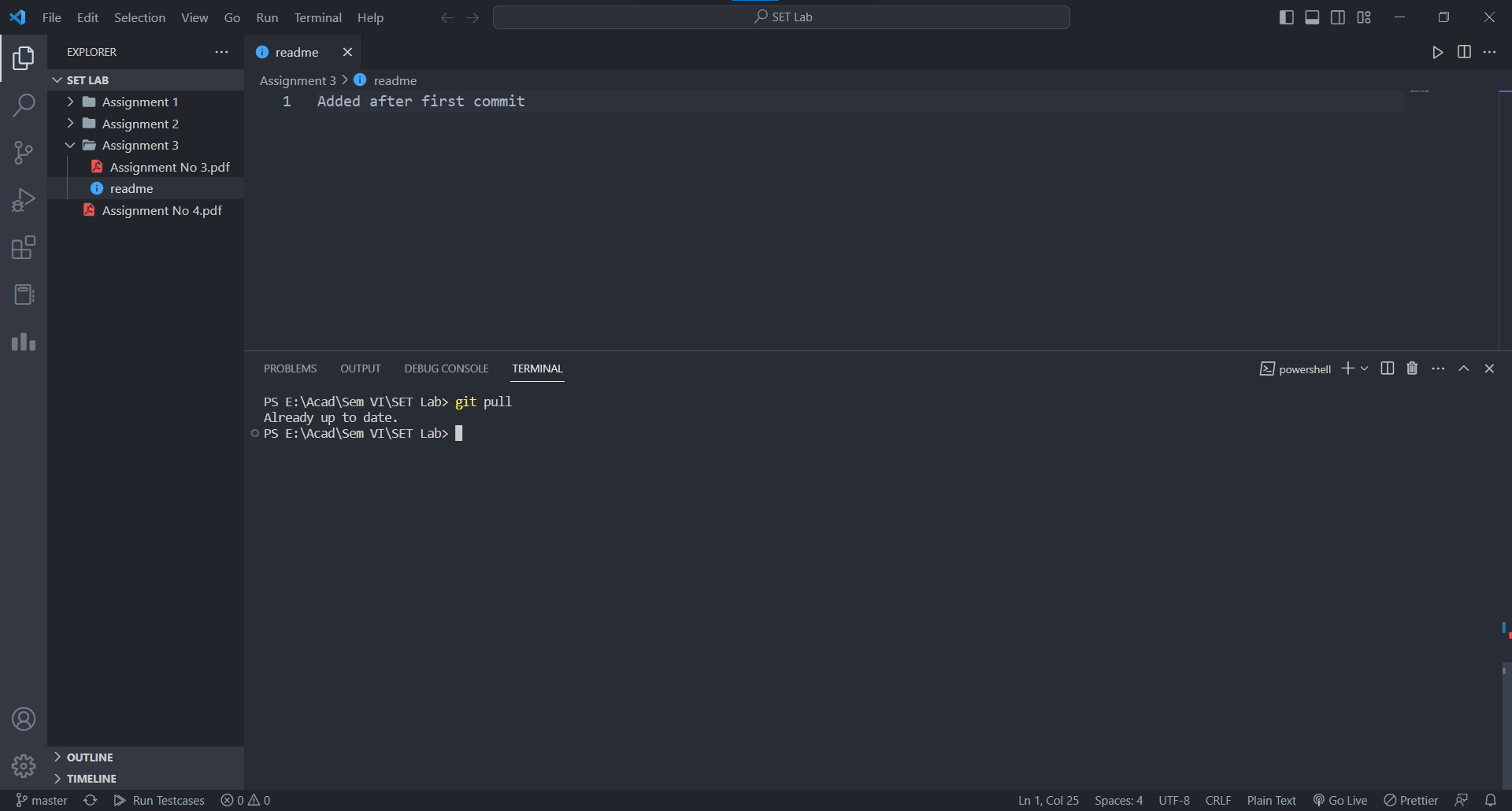
1. Perform update to the existing files (show history)



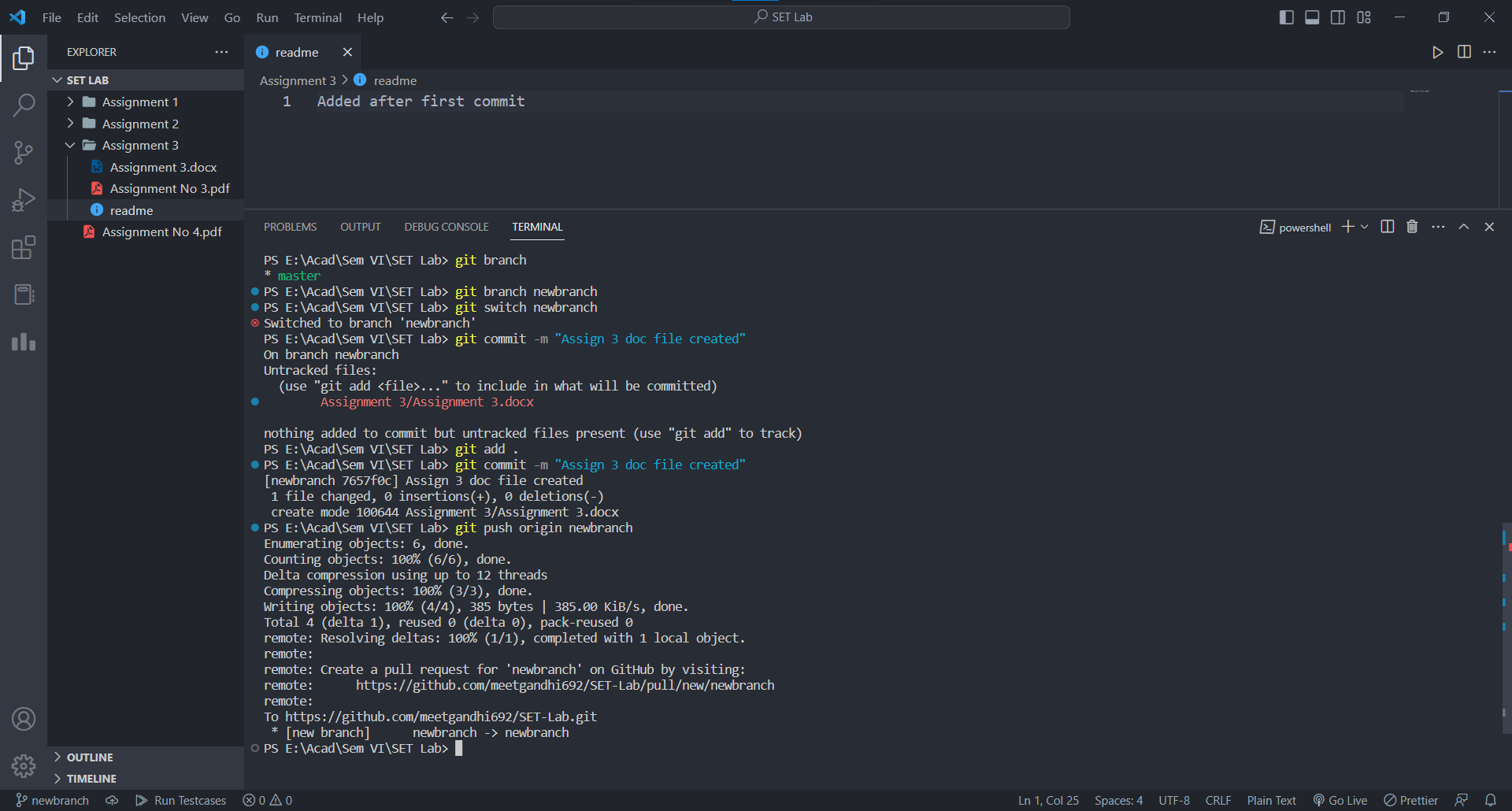
1. Create another branch



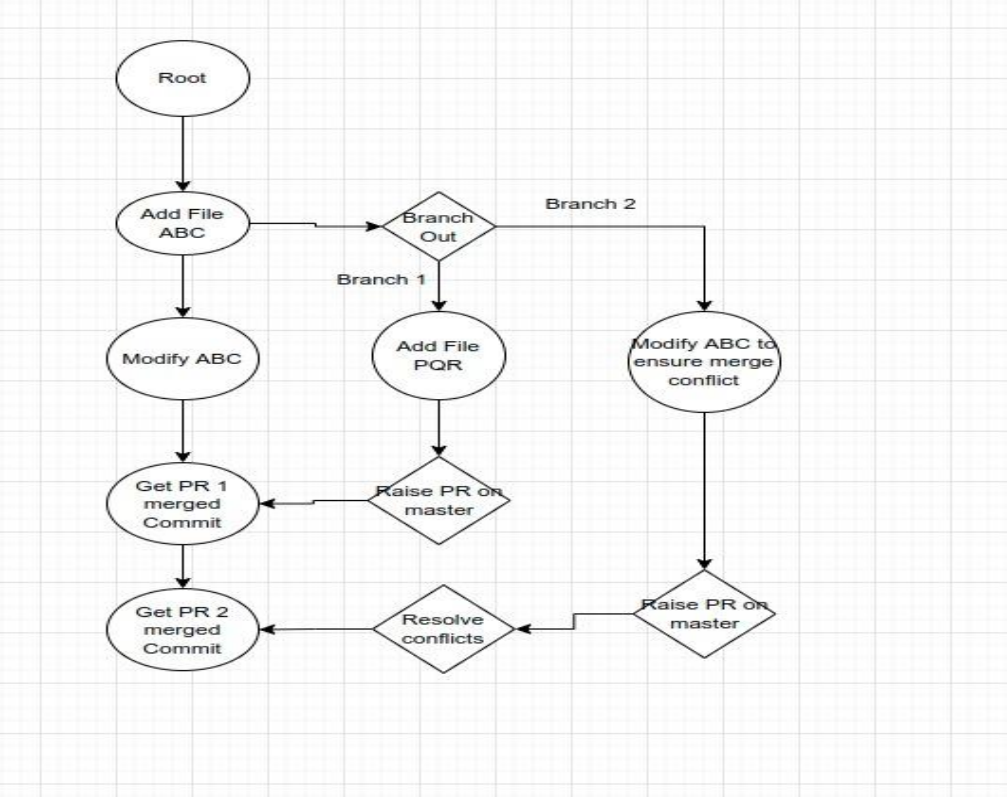
1. Create pull request



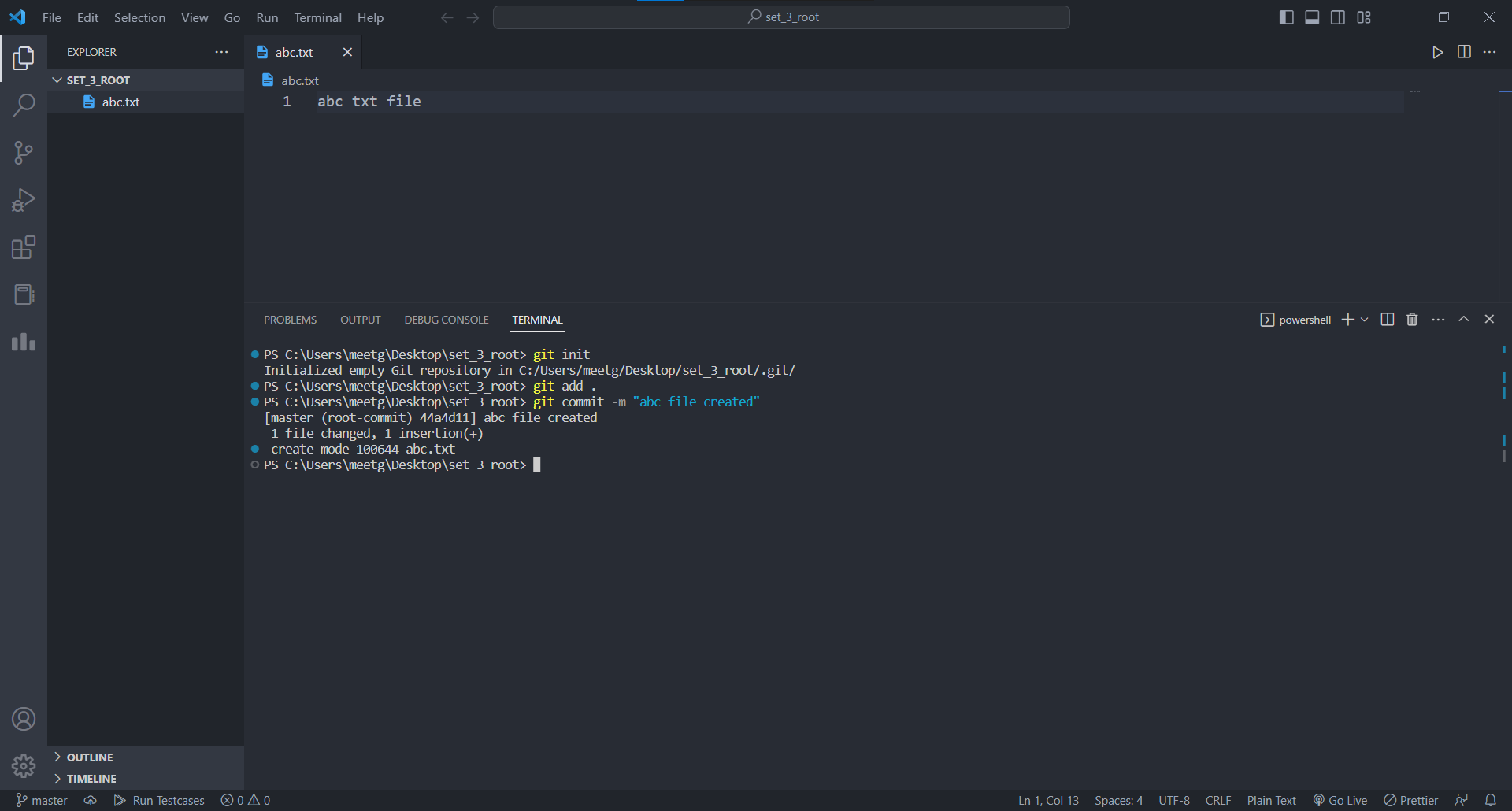
1. Perform merging of both branches



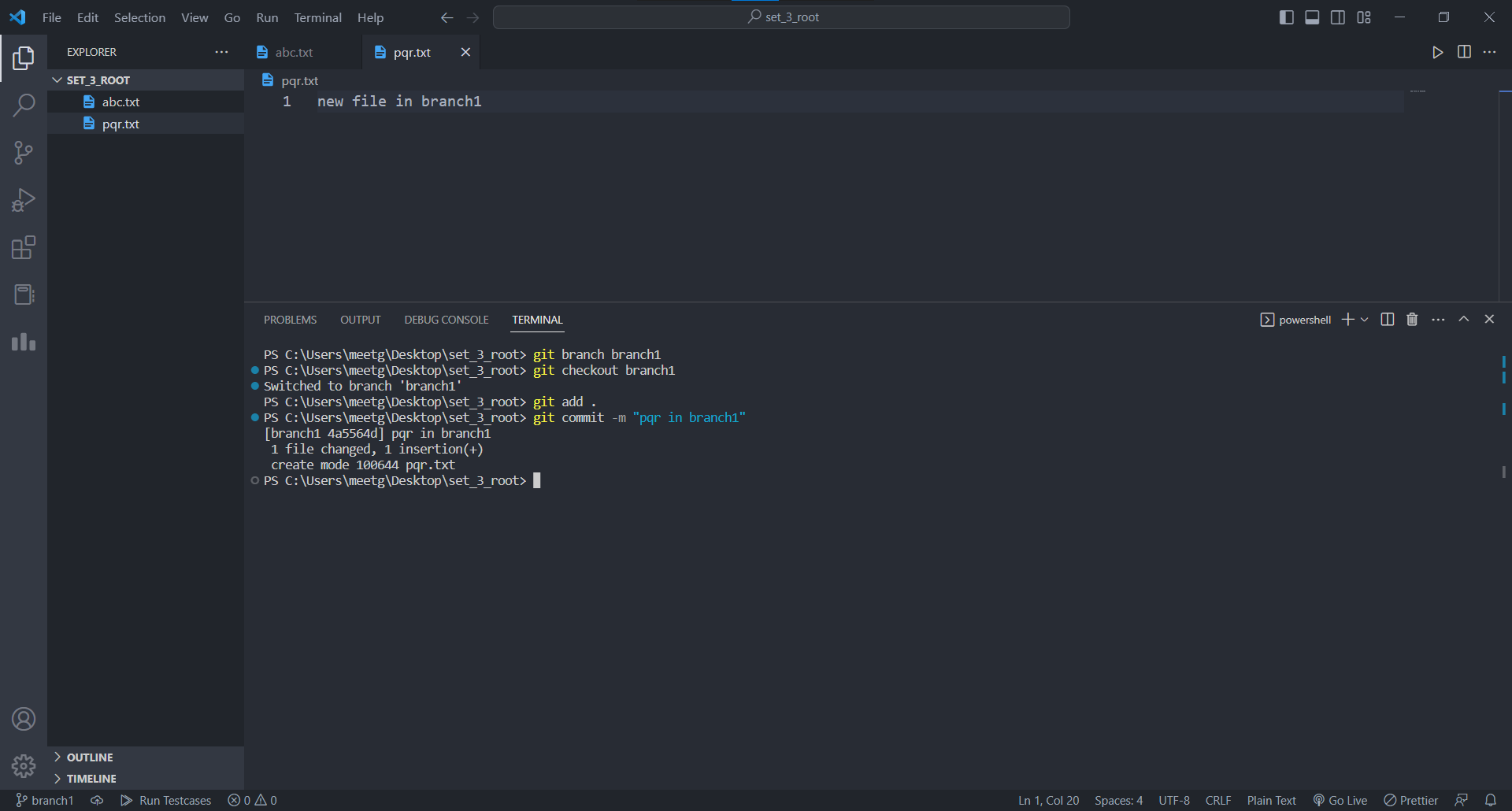
1. Perform Fork operation
2. For the diagram given below create a GitHub repository and perform operations given in the diagram. (Perform commit operations as given) (Add screenshots as an answer to this question)



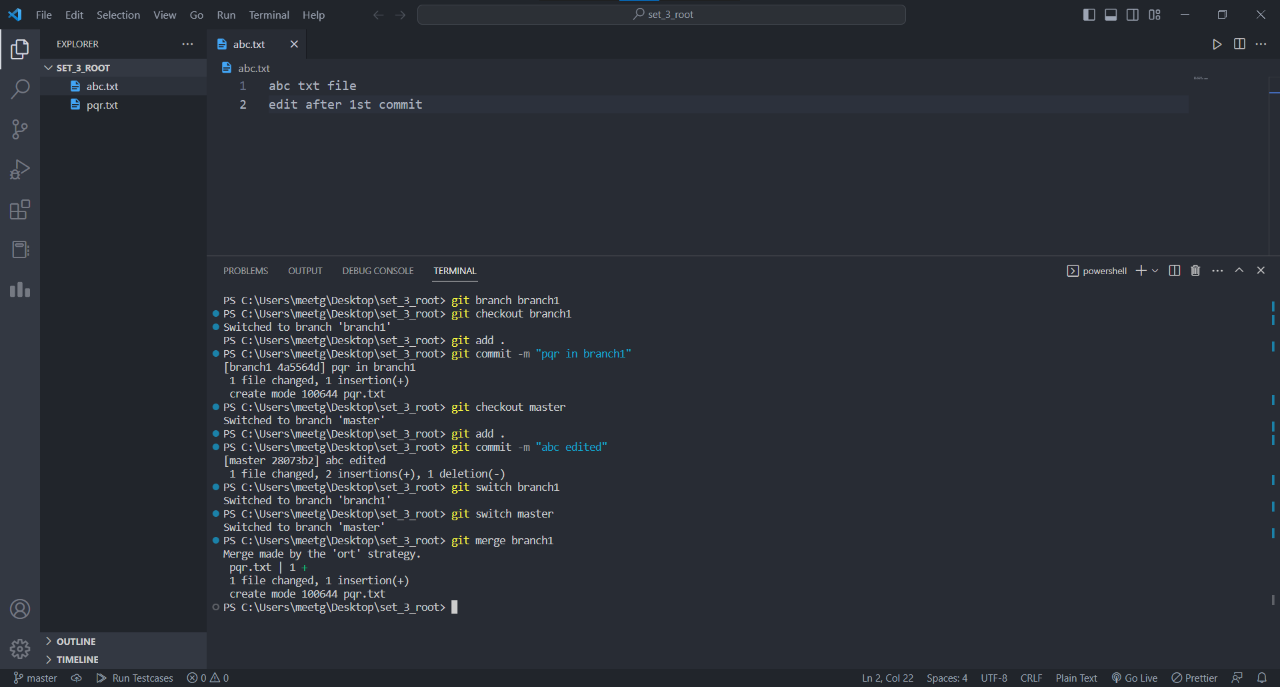
1. Created file abc.txt in master



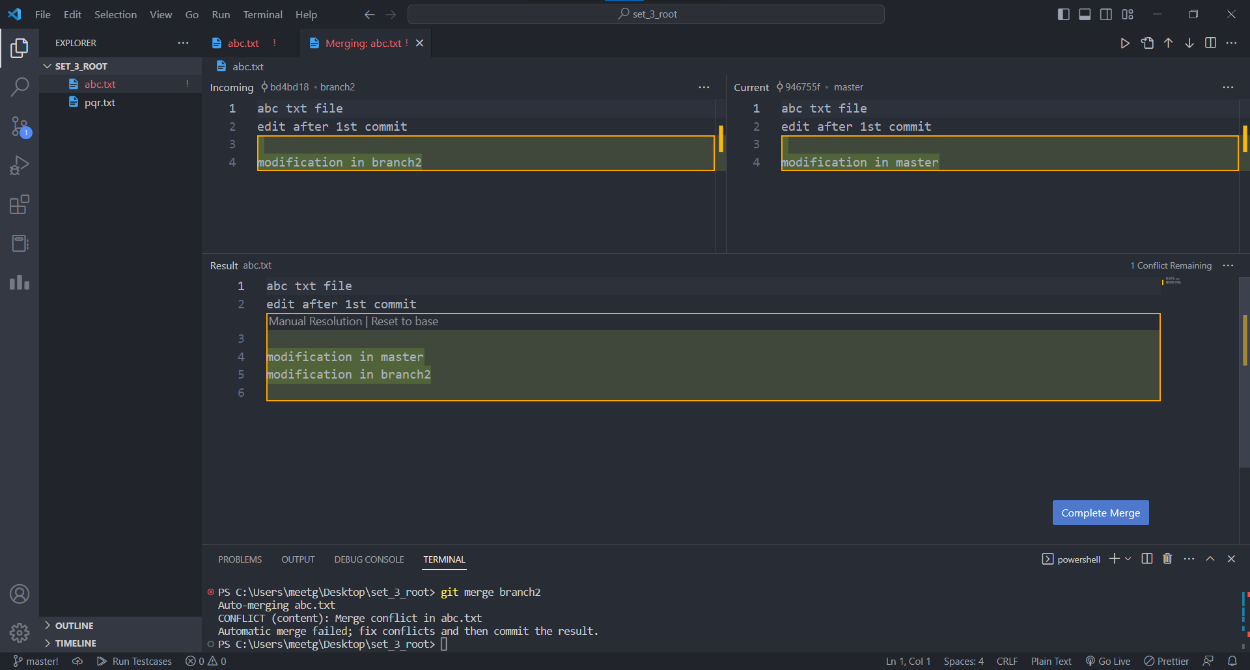
1. Created branch branch1 and added file pqr.txt



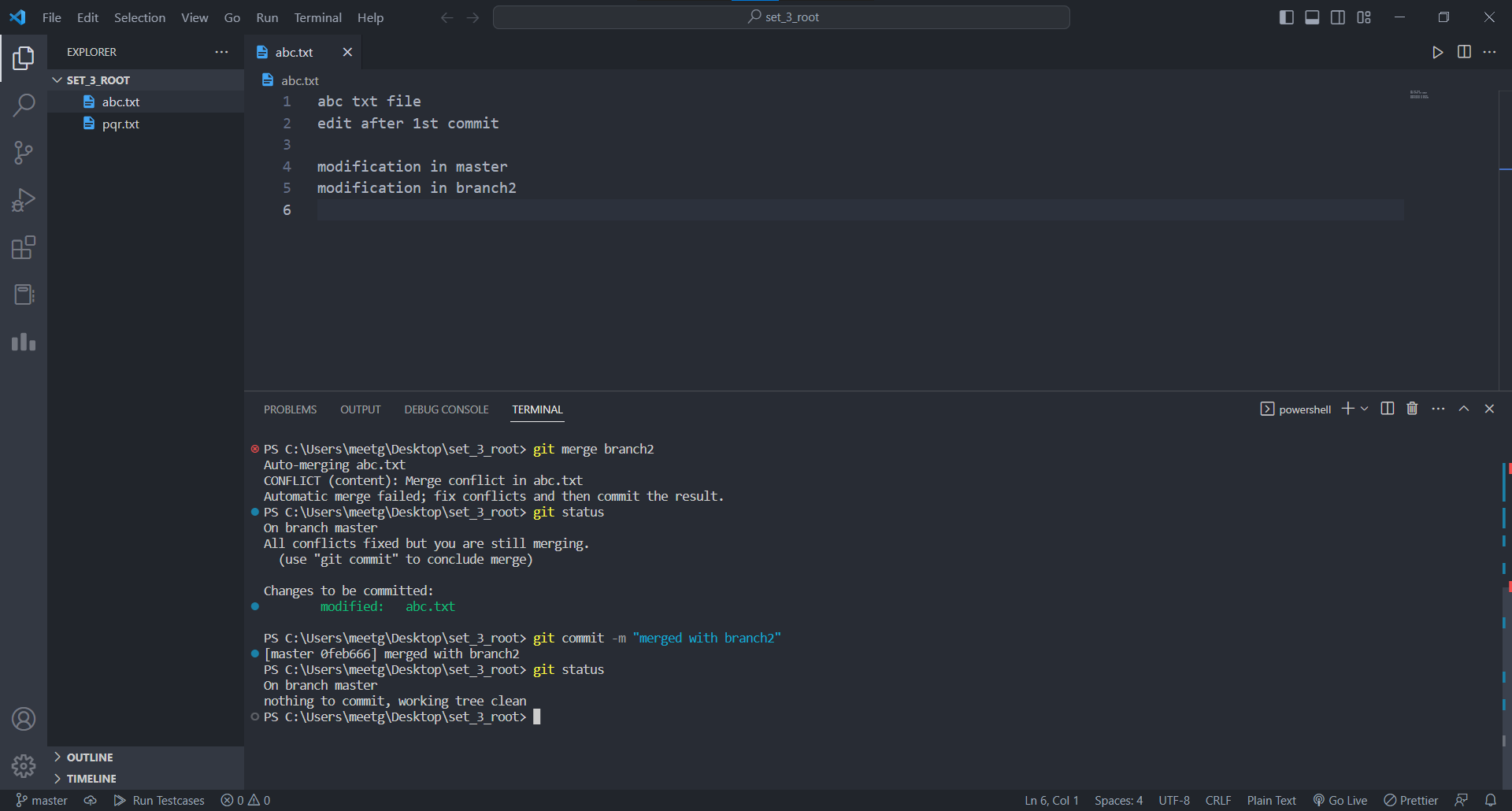
1. Merged branches master and branch1



1. Created branch branch2 and modified file abc both in master and branch2 to ensure merge conflict.
2. Encountering merge conflict



1. Solving merge conflict and commit the code in master branch

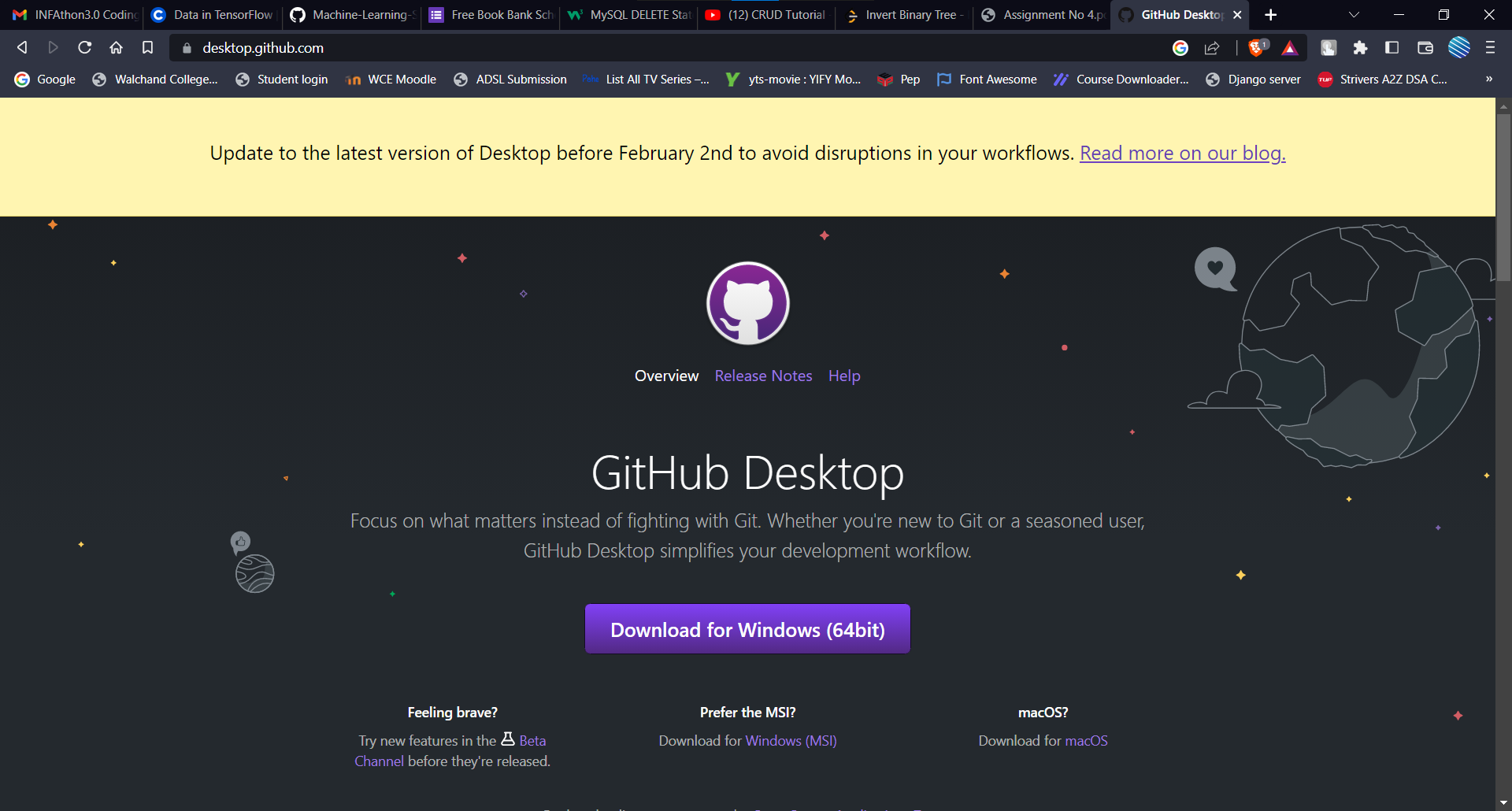


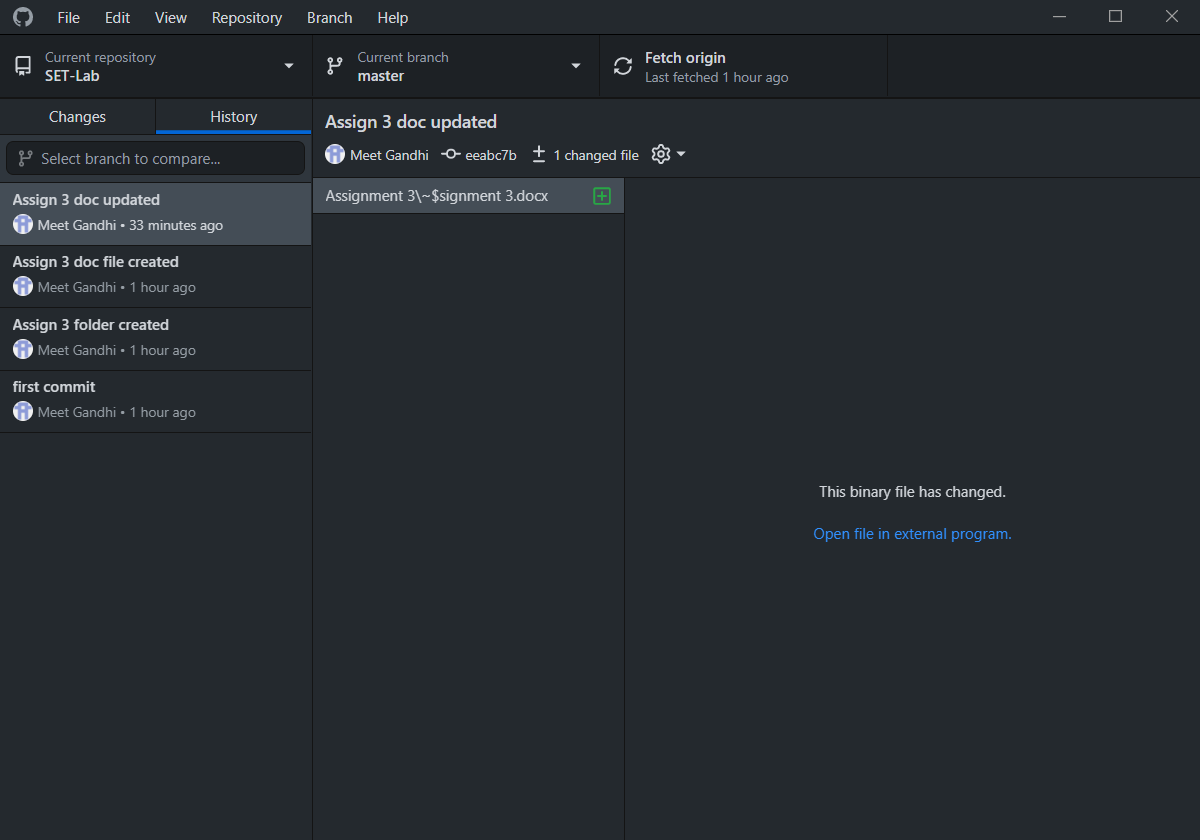
1. What is GitHub desktop? How to install GitHub on local machine? Install GitHub on your local machine and access repository created in question no 1 (add screenshots).

GitHub Desktop is a desktop application that provides a graphical user interface (GUI) for working with Git repositories hosted on GitHub. It allows you to manage and collaborate on your code and project files, track changes, and work with your team in an easy-to-use interface. To install GitHub Desktop on your local machine, you can follow these steps:

1. Go to the GitHub Desktop website at https://desktop.github.com/ and click the "Download for [your operating system]" button.
2. Once the installer has finished downloading, run it and follow the prompts to complete the installation process.
3. After the installation is complete, launch GitHub Desktop and sign in with your GitHub account. If you don't have an account, you can create one for free on the GitHub website.
4. Once you're signed in, you can use GitHub Desktop to clone repositories from GitHub, create new repositories, make changes to your files, and collaborate with other developers on your team.

That's it! With GitHub Desktop installed on your local machine, you'll have a powerful tool for managing your code and projects, and you can easily keep your work in sync with your team.





1. Differentiate in between GitHub, Git and GitLab.

GitHub, Git, and GitLab are all related to version control and code management, but they serve different purposes and have different features.

1. Git: Git is a command-line tool that is used for version control, which allows developers to keep track of changes made to their code over time. It is a distributed version control system, which means that each developer has a complete copy of the code repository on their local machine, allowing them to work offline and merge their changes with the main codebase later. Git is free and open-source software, and it is widely used in the software development industry.
2. GitHub: GitHub is a web-based hosting platform for Git repositories. It provides a web-based graphical interface for managing Git repositories, including tools for code review, bug tracking, project management, and collaboration. Developers can use GitHub to host their Git repositories, share their code with others, and collaborate with other developers on the same project. GitHub also provides a number of additional features and integrations, such as continuous integration and deployment, code scanning, and project management tools.
3. GitLab: GitLab is a web-based Git repository manager that provides a complete DevOps platform. It is similar to GitHub in that it provides a web-based interface for managing Git repositories, but it also includes features for continuous integration and deployment, issue tracking, code review, and project management. GitLab is an open-source software, which means that you can host your own instance of GitLab on your own servers or use the GitLab.com cloud service. GitLab is used by developers and organizations of all sizes for their code management needs.

In summary, Git is a command-line tool for version control, GitHub is a web-based hosting platform for Git repositories that provides additional collaboration and project management features, and GitLab is a web-based Git repository manager that provides a complete DevOps platform.

1. What is version control? Explain with example.

Version control is a system that allows developers to manage changes to their code over time. It provides a way to track changes, compare different versions of the code, and collaborate with other developers on the same project. Version control is an important tool for software development, as it enables teams to work on the same codebase without interfering with each other's work.

One of the most popular version control systems is Git. With Git, developers can create a repository for their code, which tracks all changes made to the code over time. When a developer makes changes to the code, they commit those changes to the repository, creating a new version of the code. Each version of the code is stored in the repository, allowing developers to go back in time and see previous versions of the code.

For example, let's say that a team of developers is working on a web application. One developer creates a new feature, while another fixes a bug. Each developer commits their changes to the Git repository, creating a new version of the code. If a bug is introduced in a later version, the team can use Git to go back to the previous version of the code to find and fix the bug. Git also allows developers to work on different branches of the code, so that they can experiment with new features or bug fixes without affecting the main codebase.

Version control systems like Git are an essential tool for software development, as they allow developers to work collaboratively on the same codebase while keeping track of changes and ensuring the integrity of the code.