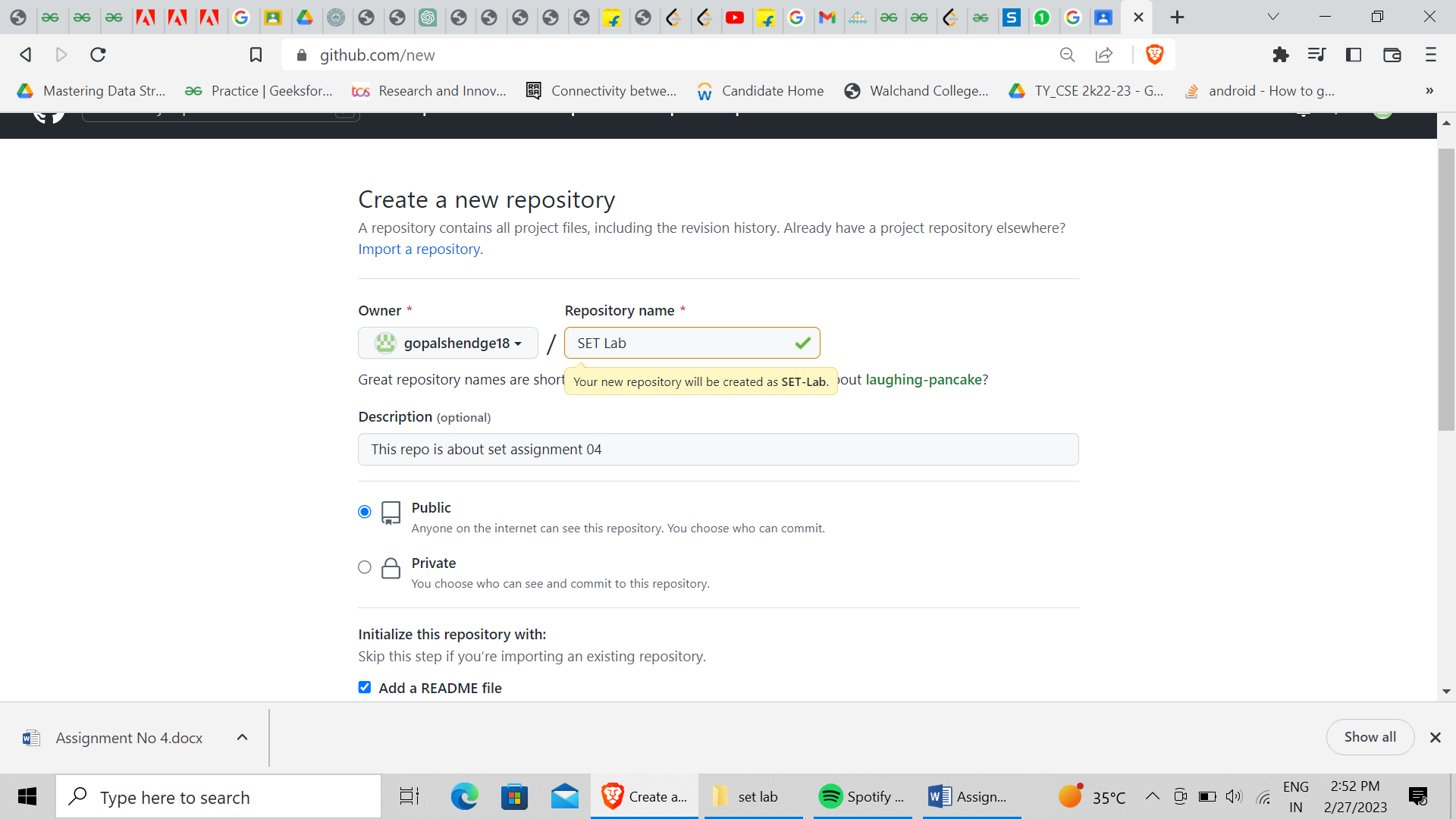
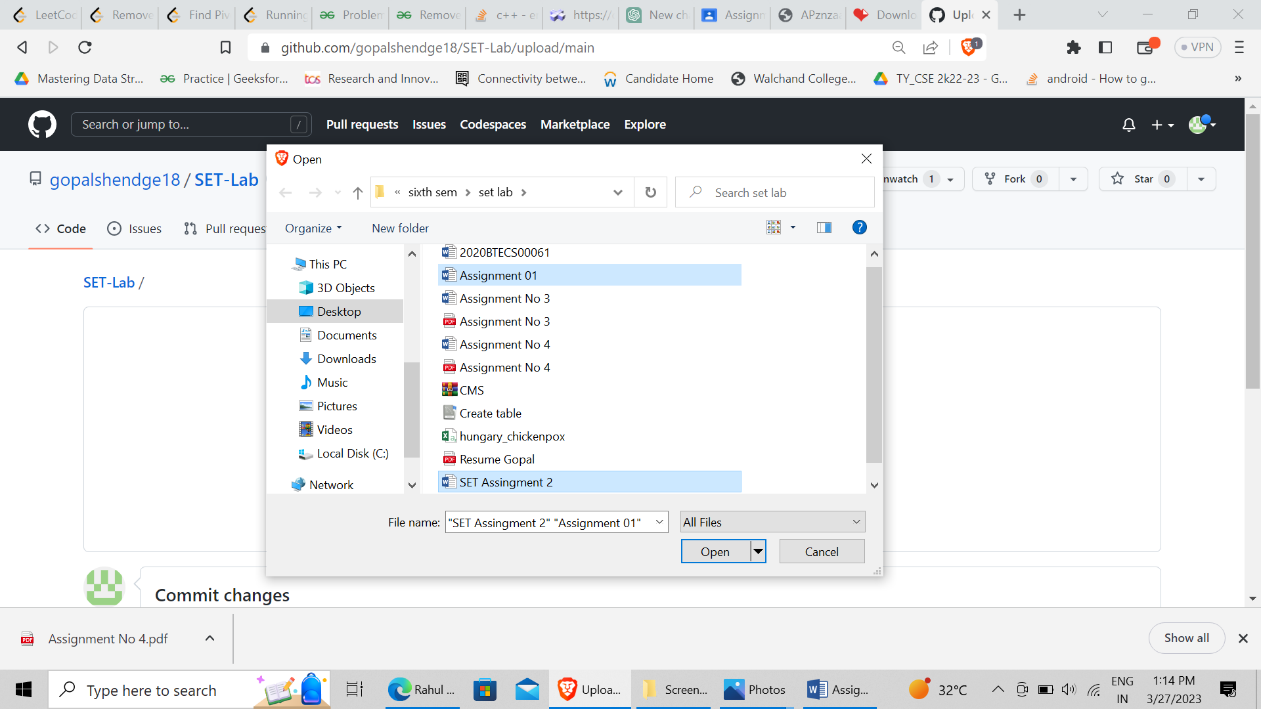
**Software Engineering Tools Lab Assignment No-4**

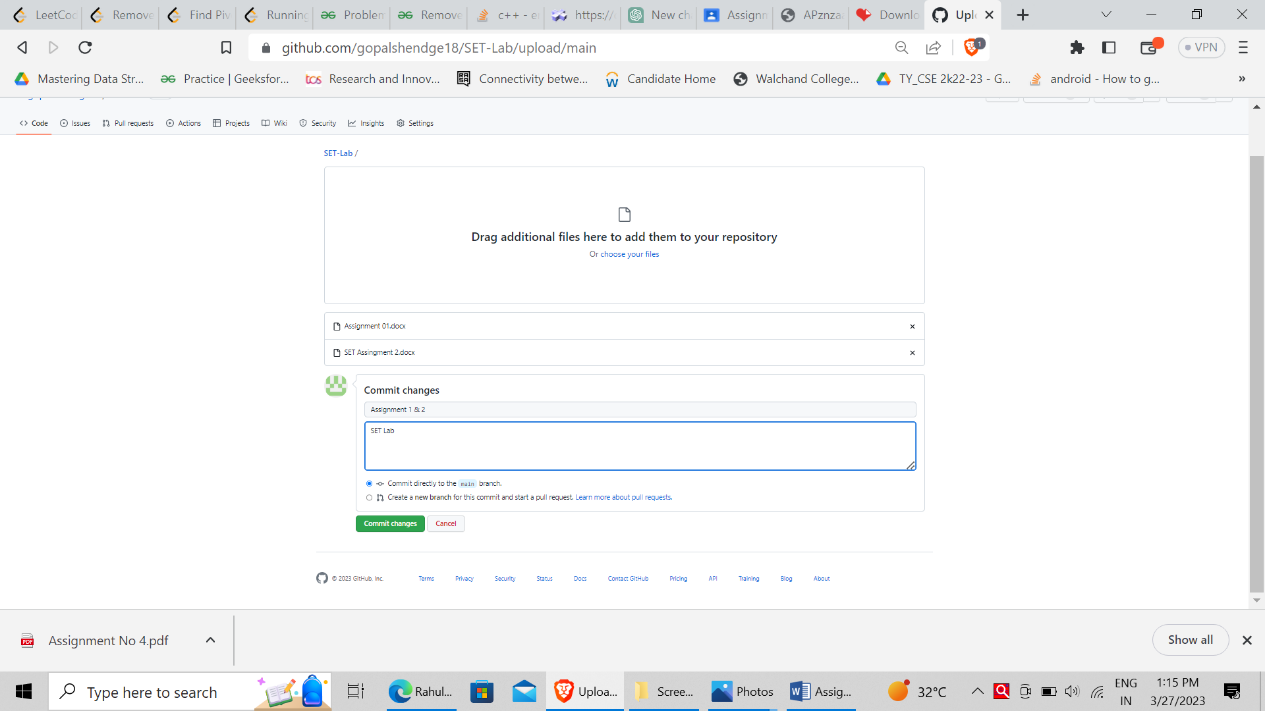
(Module 3- GitHub) **Due date-24/02/2023**

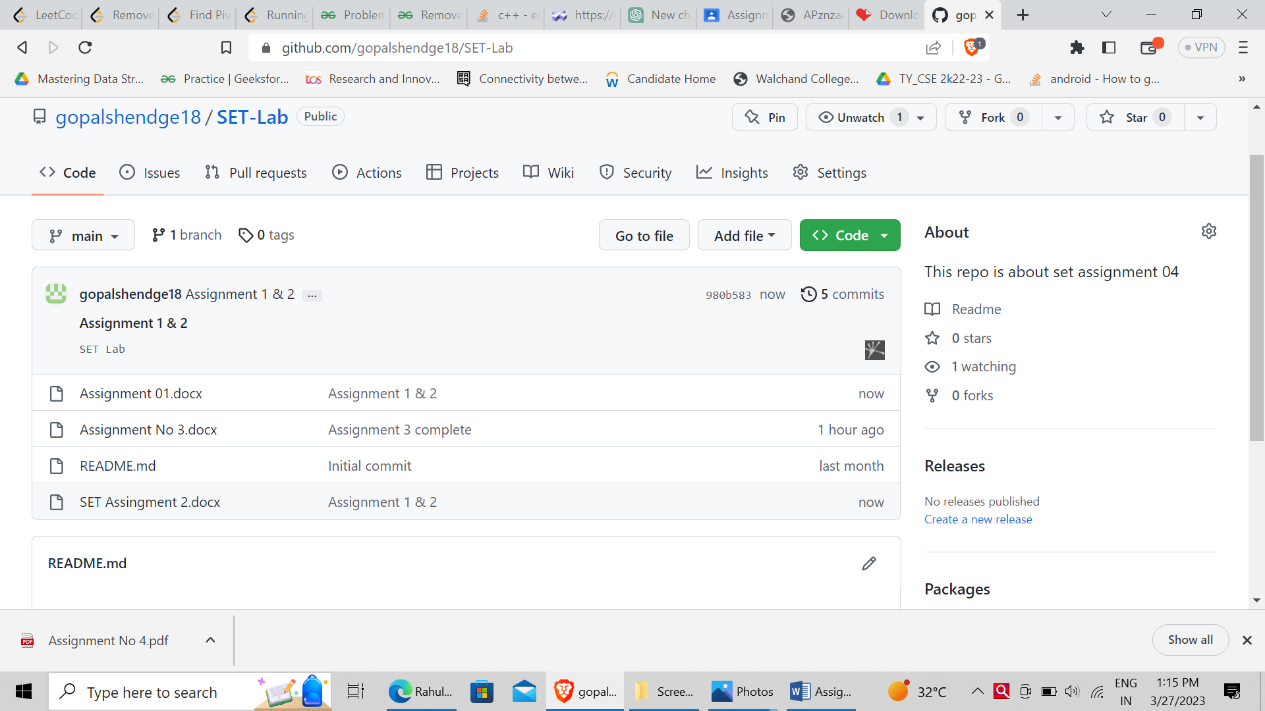
Q 1. Create a repository on GitHub named SET Lab and add files into it (you can add implementation files of previous assignment) perform below operations on it. (Add screenshot as an answer to every question)



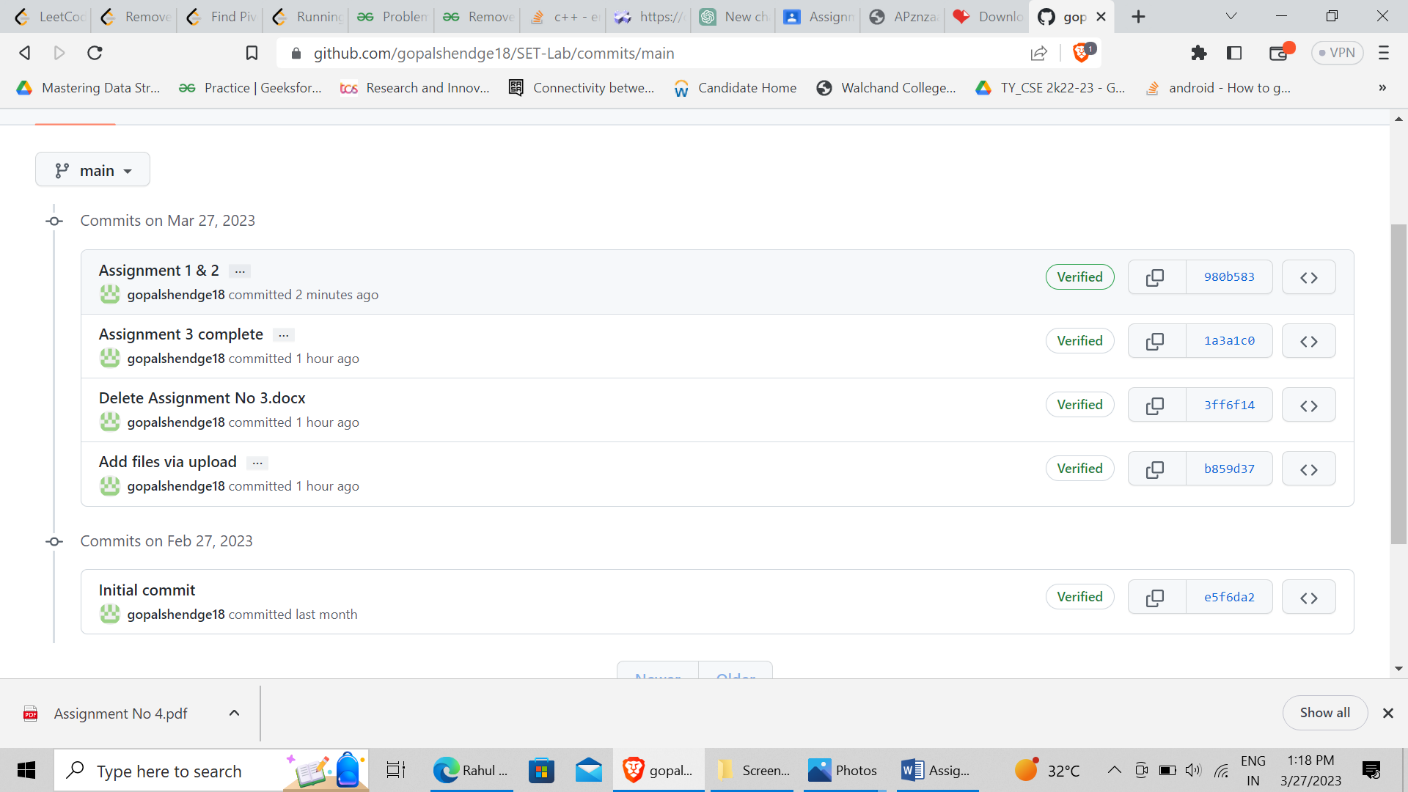
1. Perform commit on added files



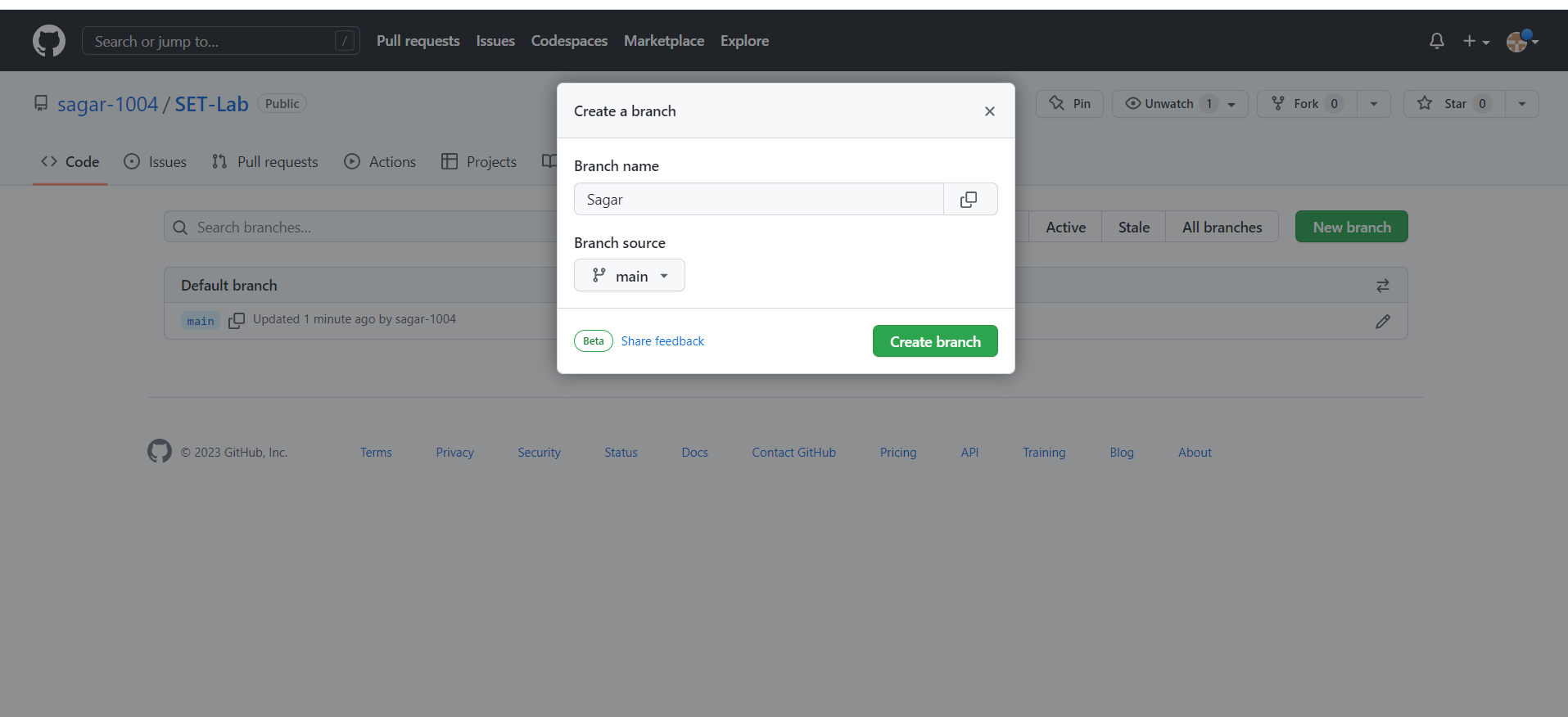


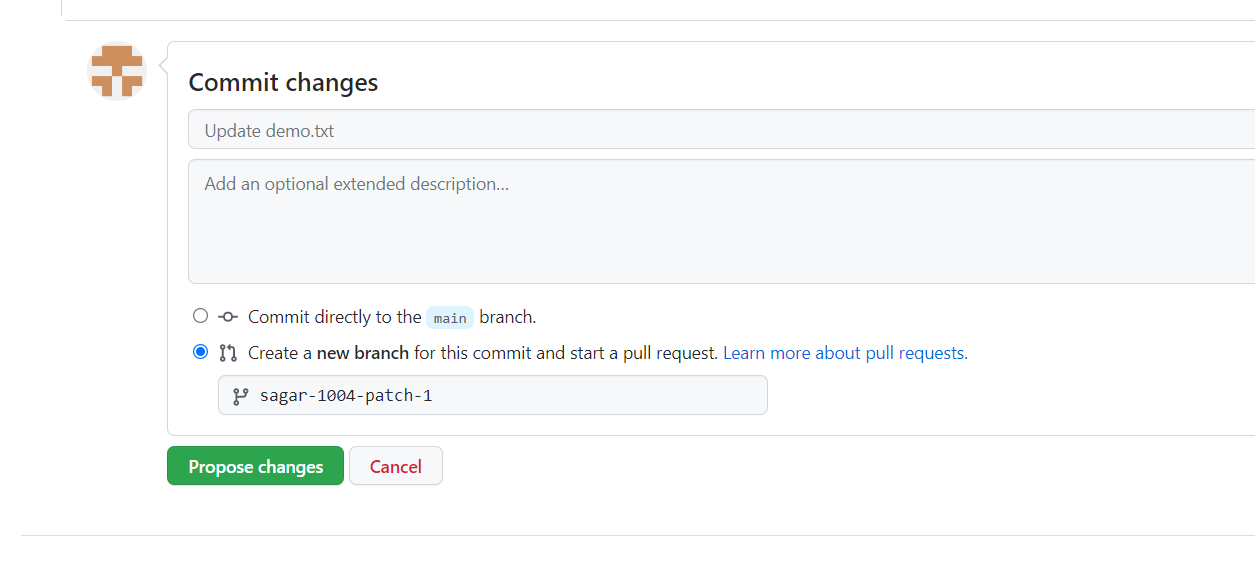


1. Perform update to the existing files (show history)

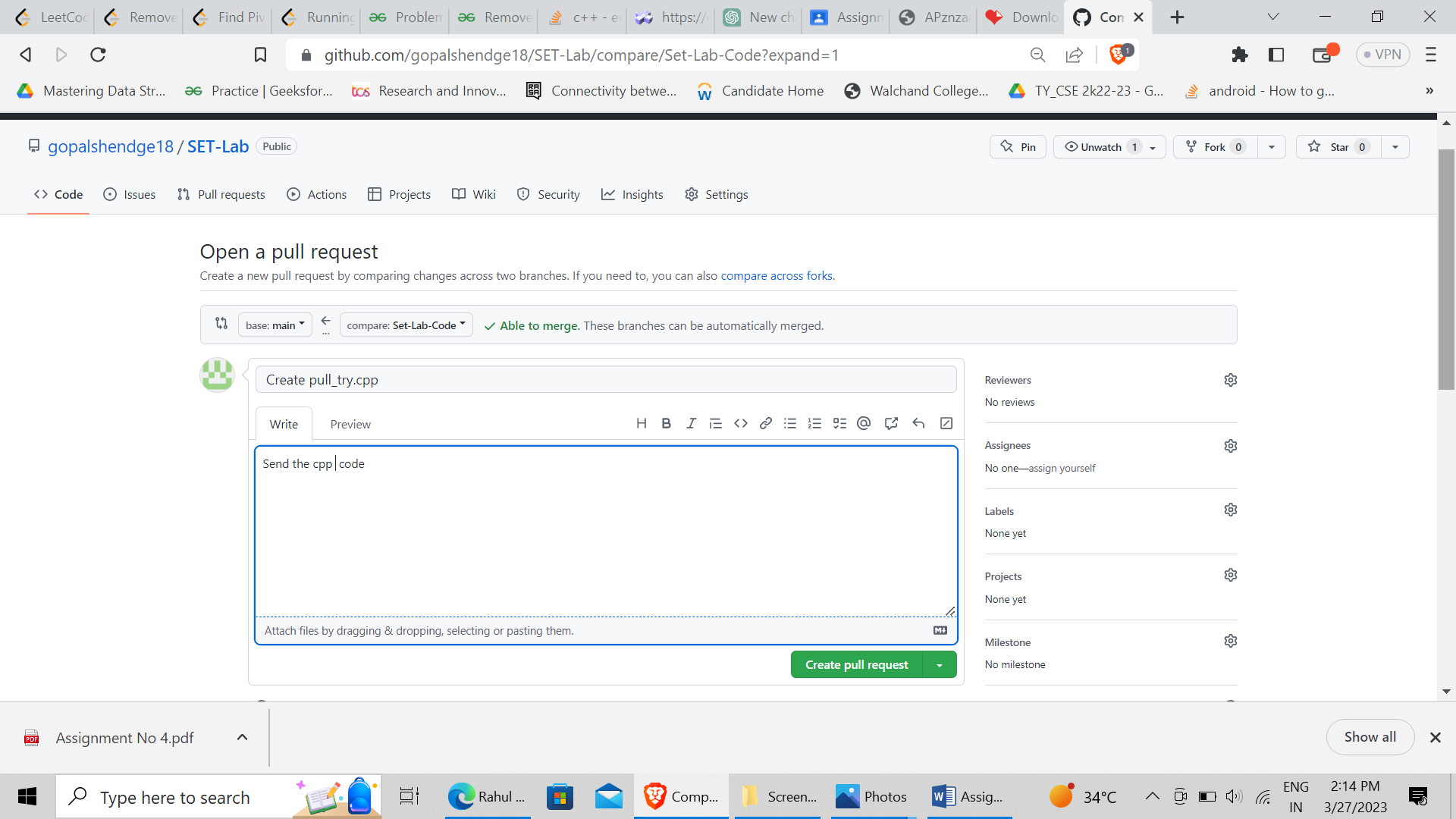


1. Create another branch

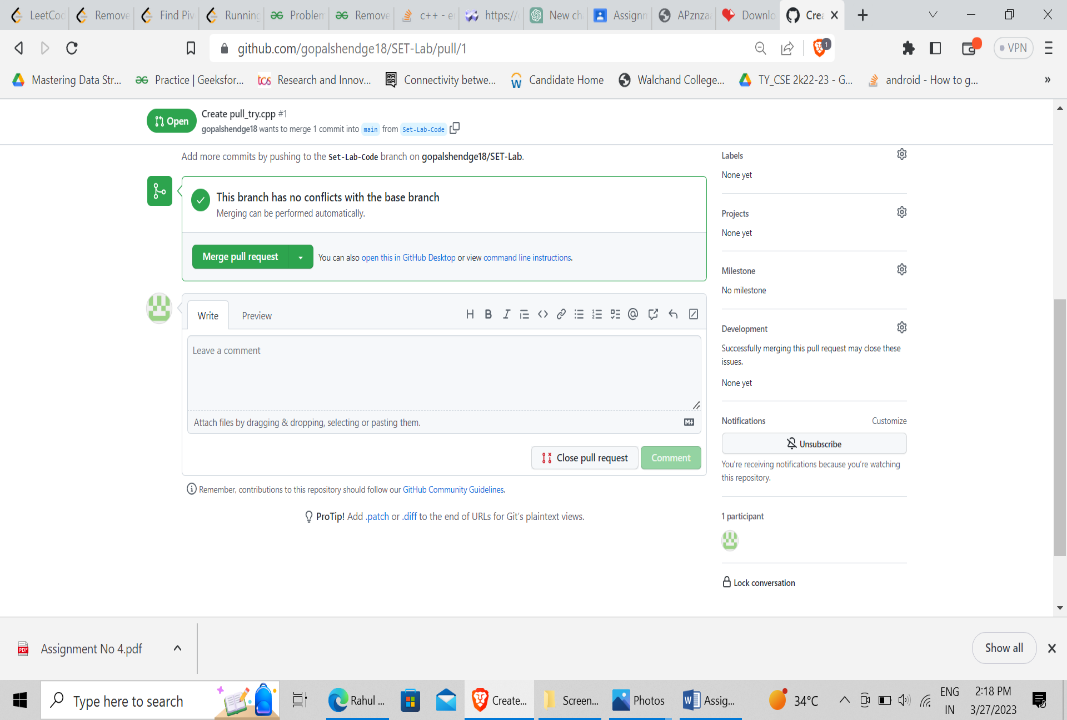


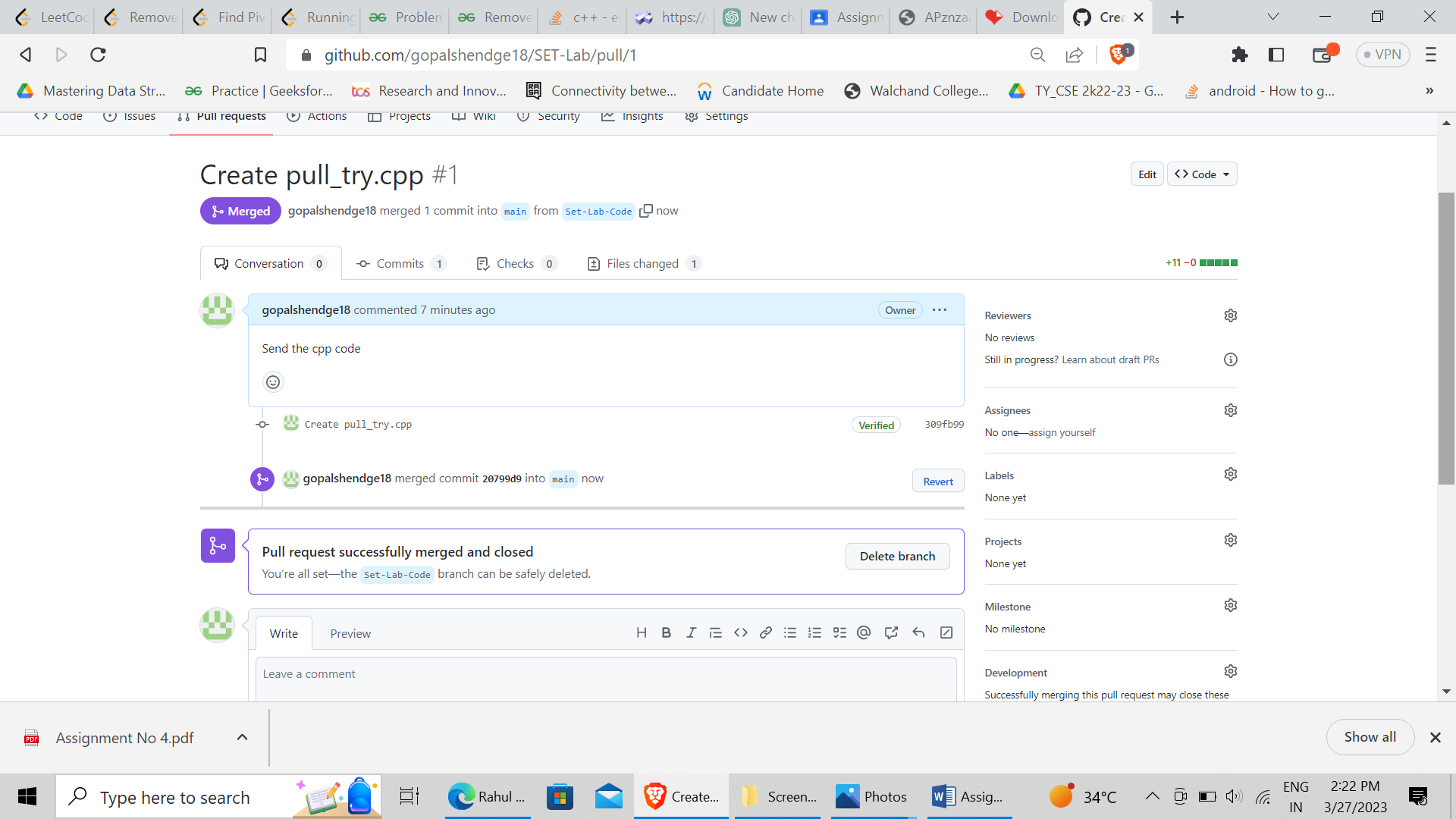


1. Create pull request

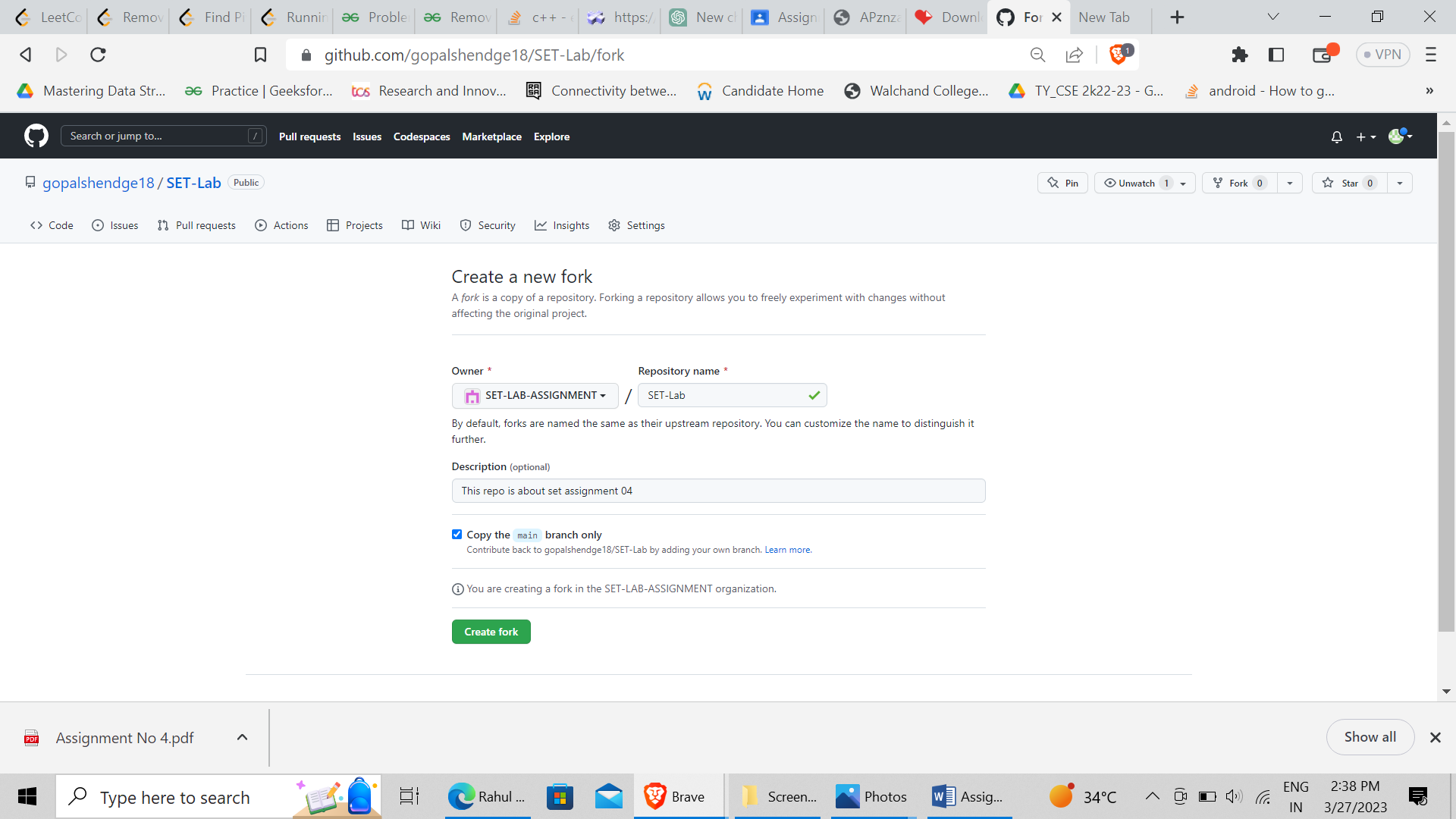


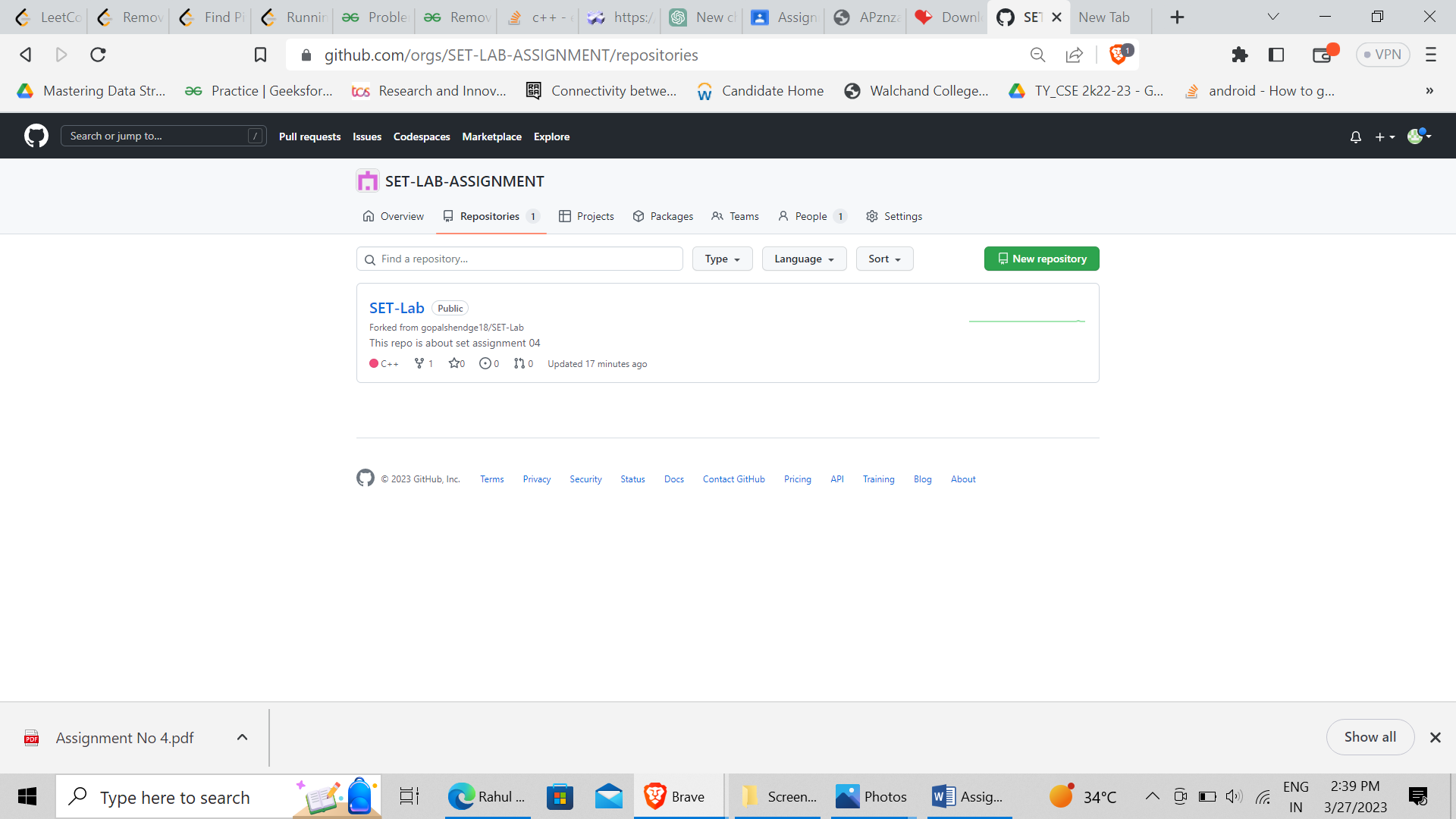
1. Perform merging of both branches



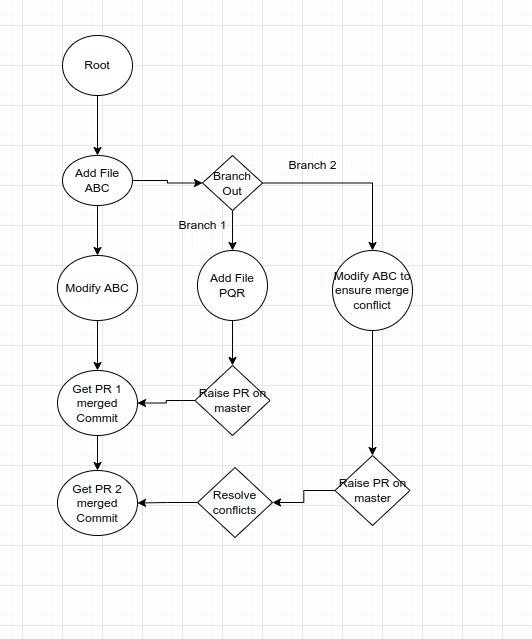


1. Perform Fork operation

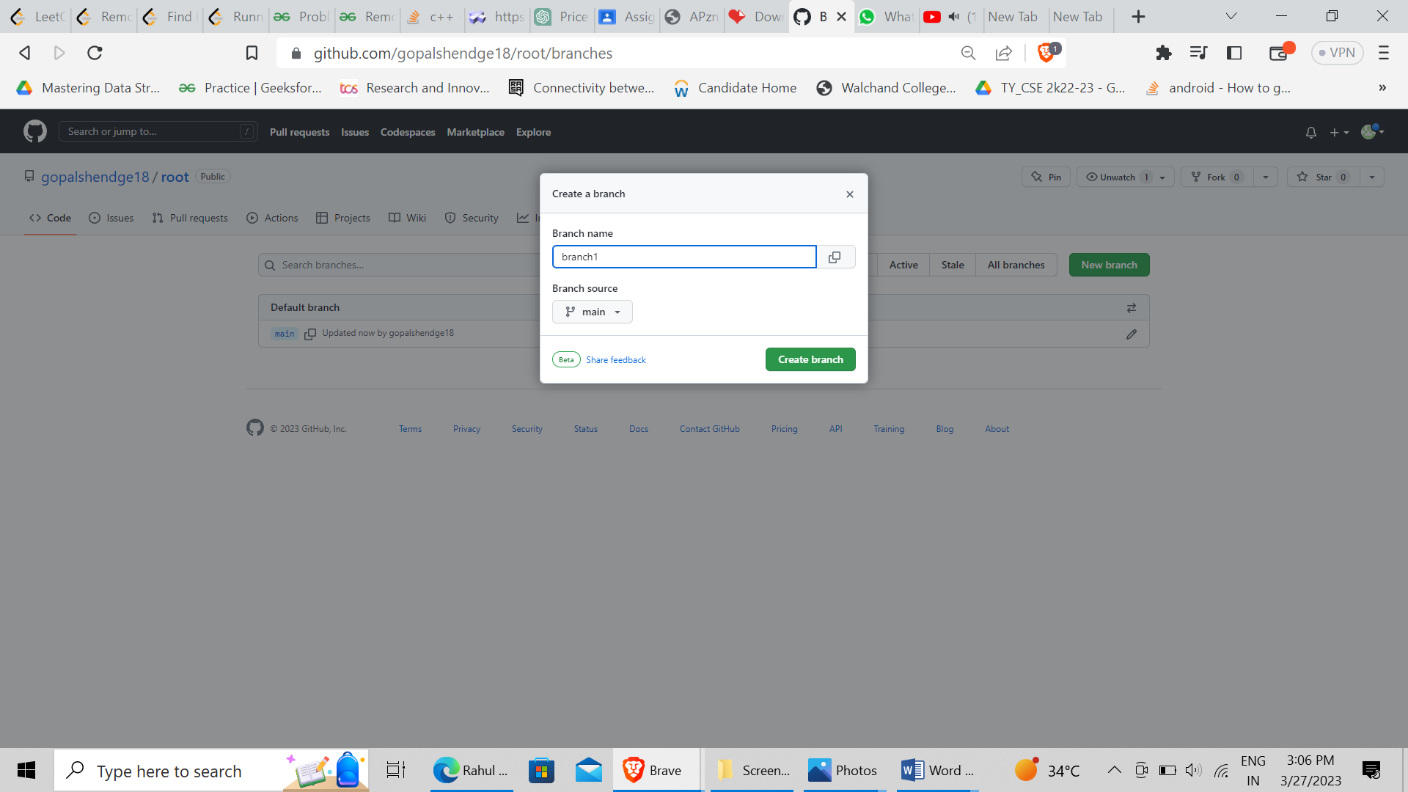


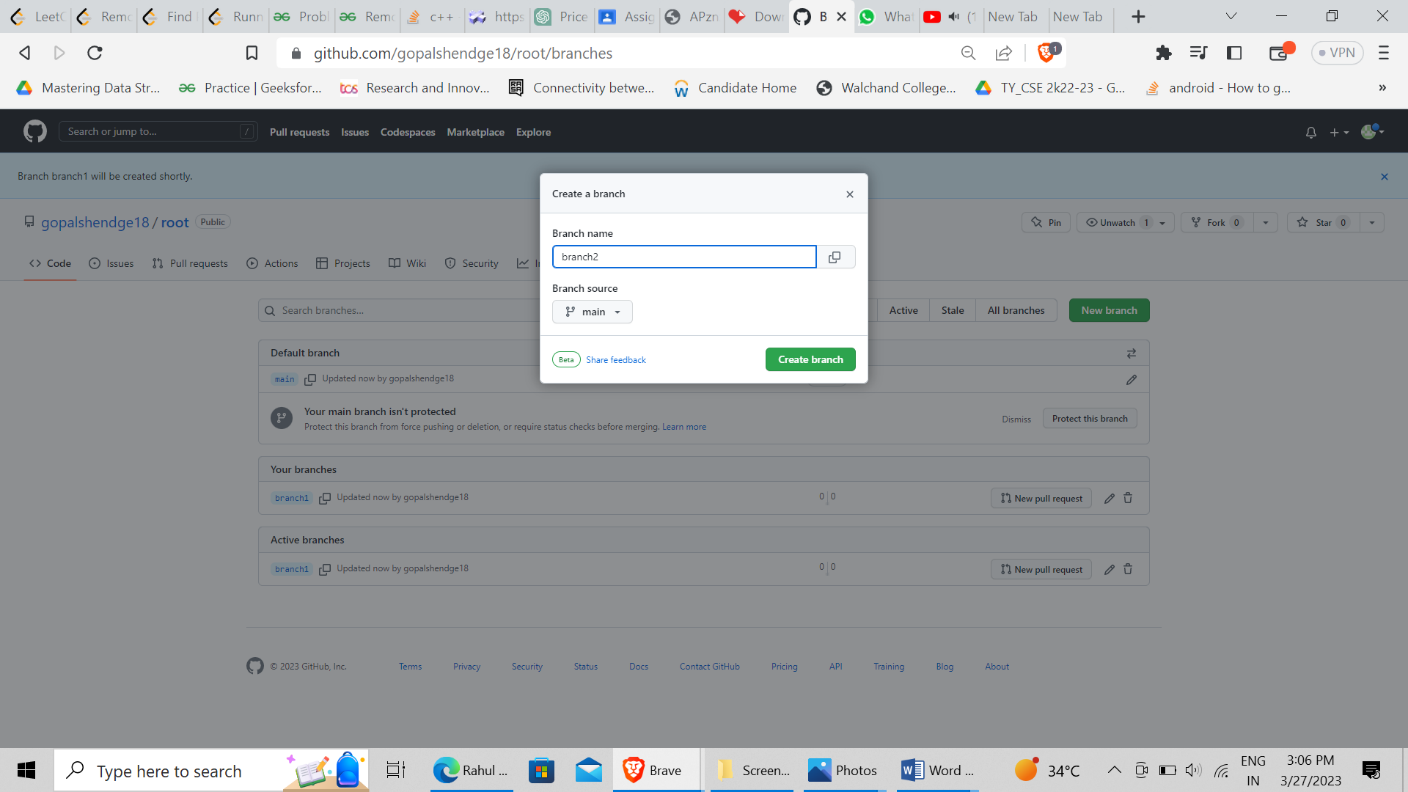


Q 2. For the diagram given below create a GitHub repository and perform operations given in the diagram. (Perform commit operations as given)(Add screenshots as an answer to this question)

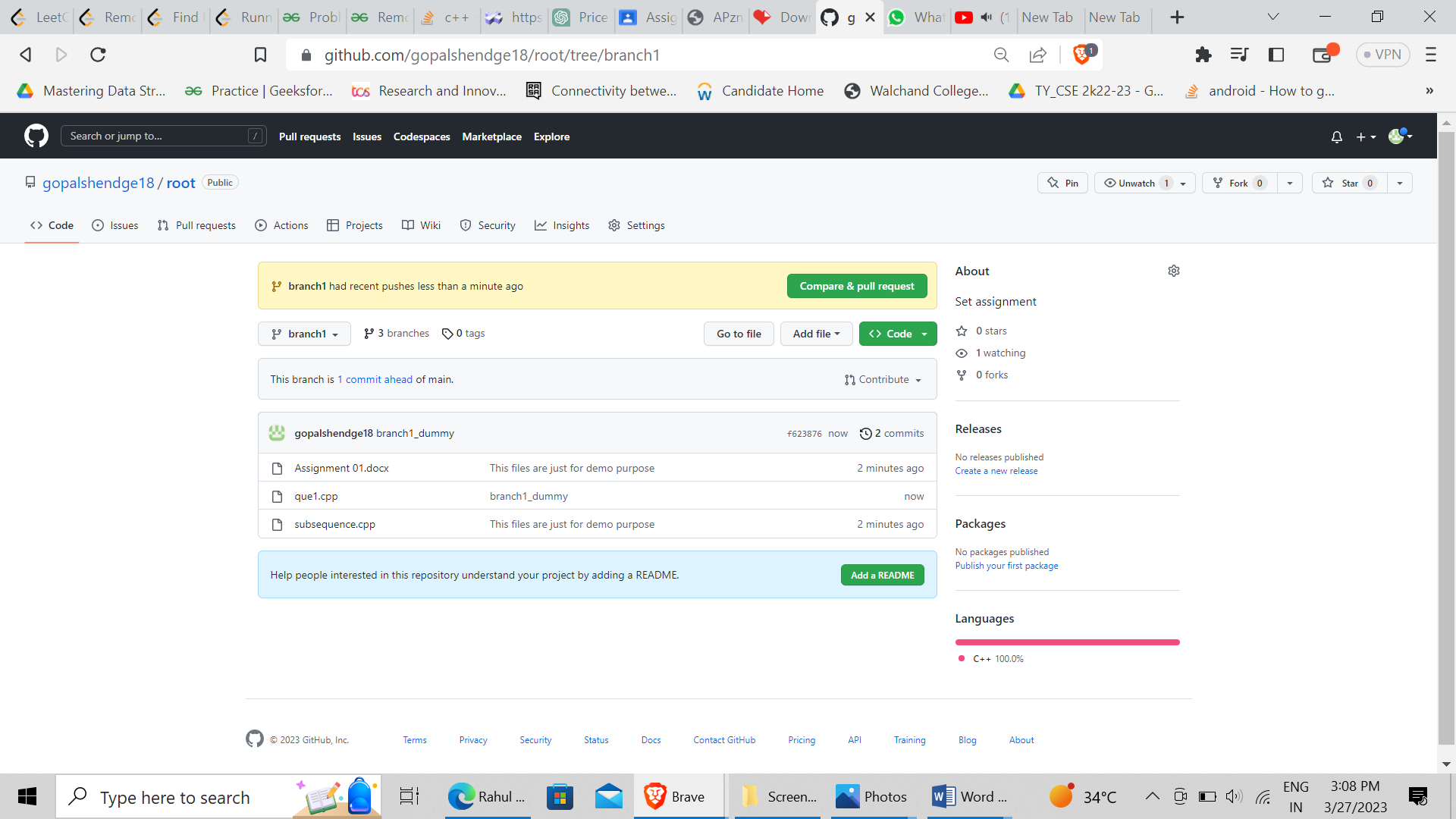


Creating branch1 and branch2 in main

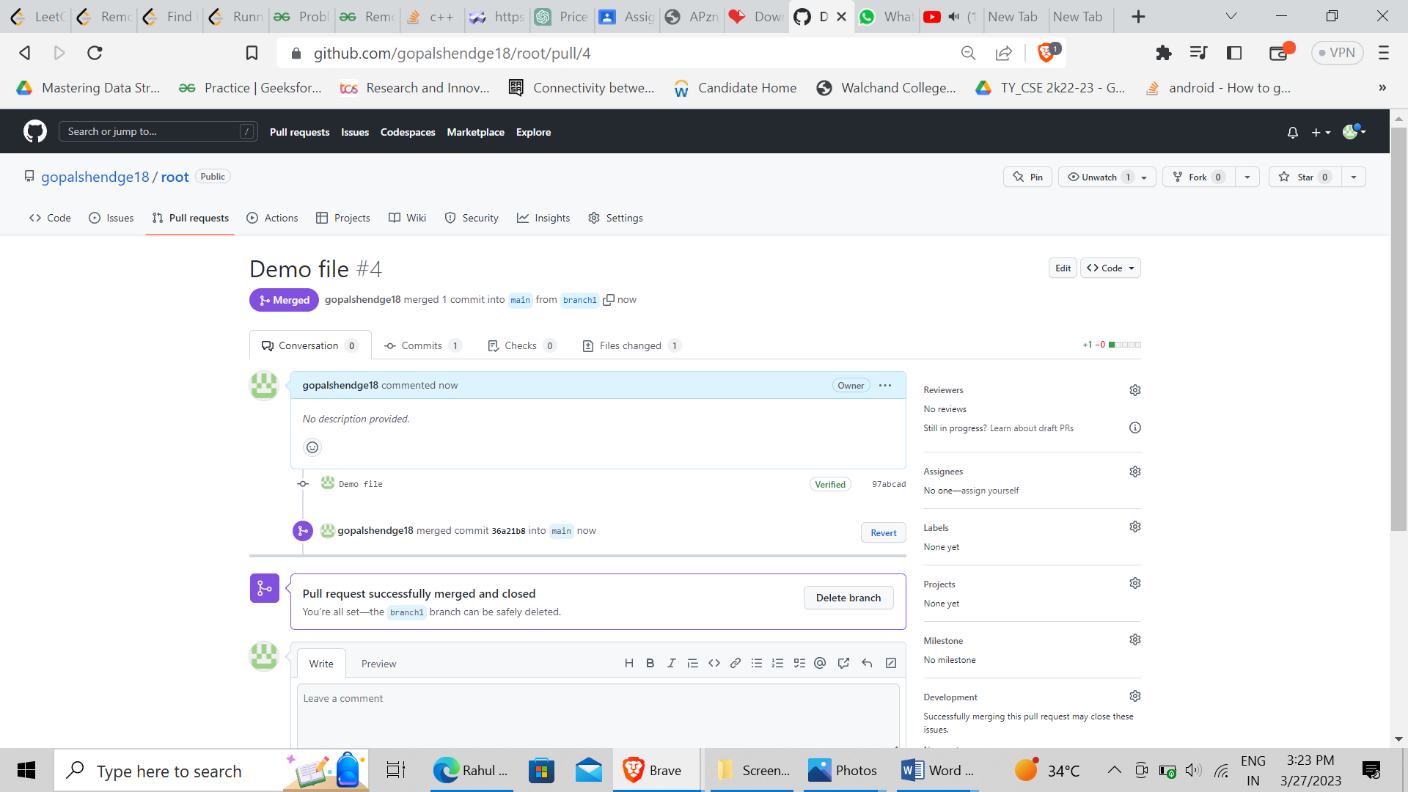
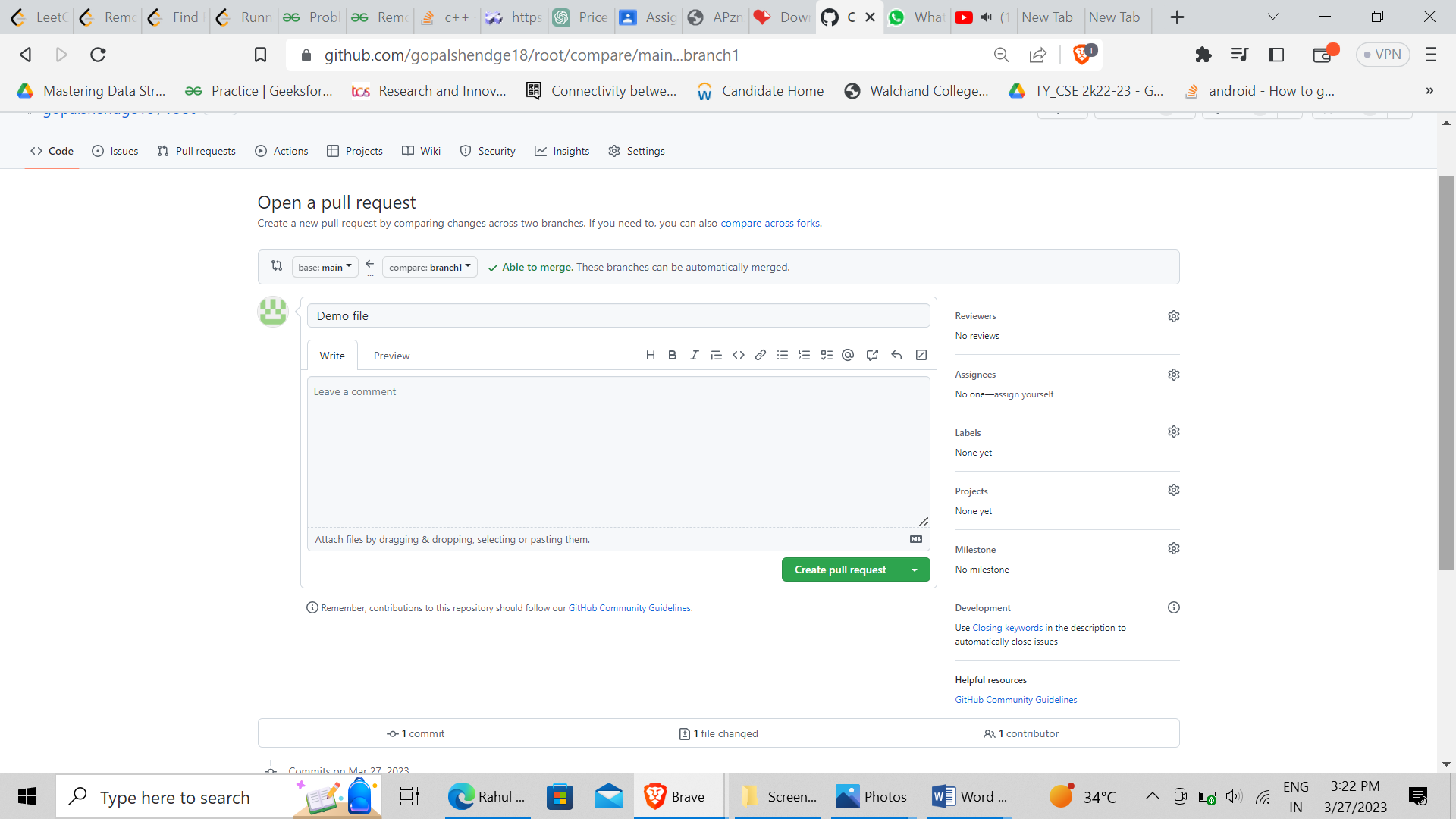
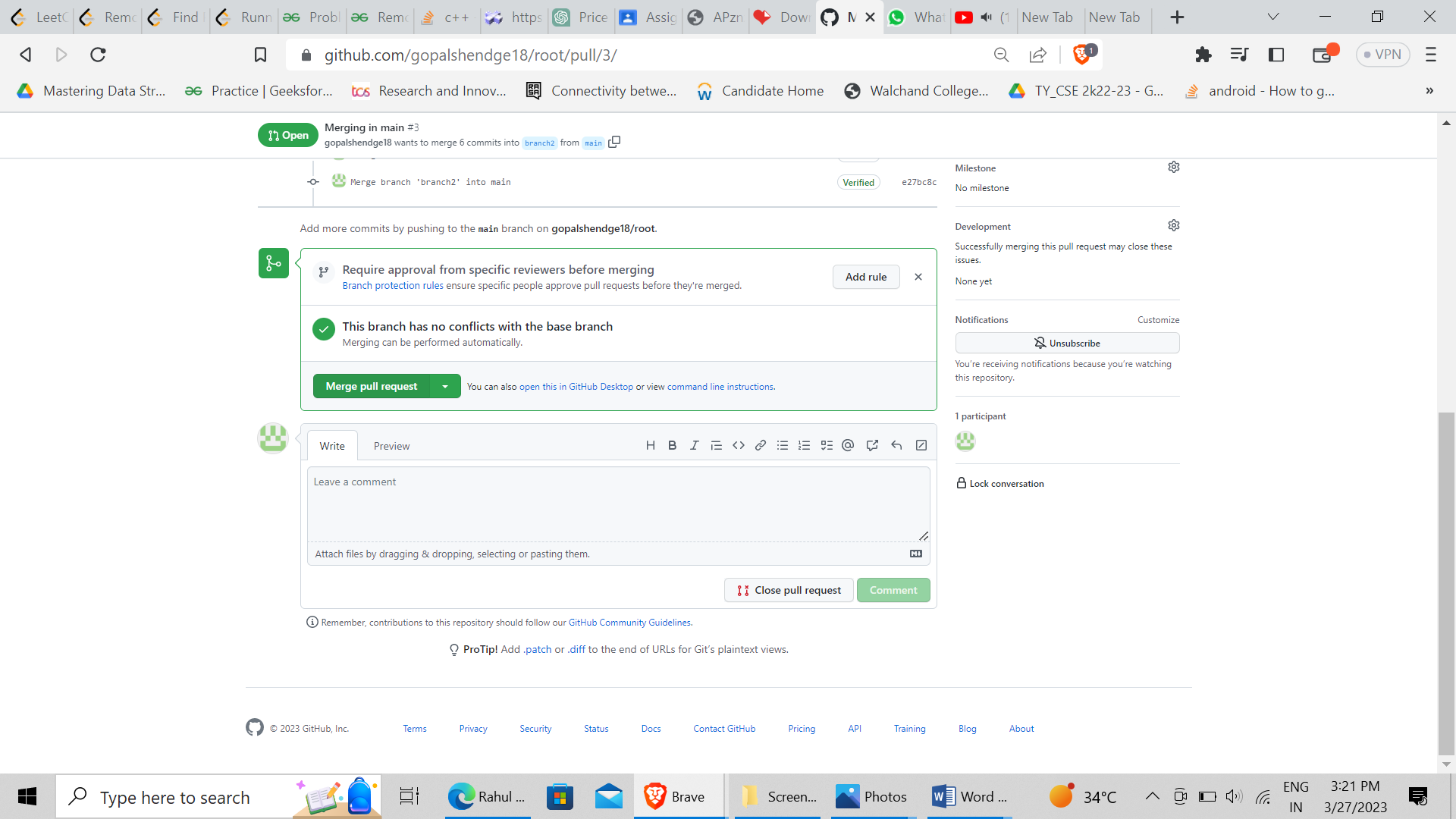
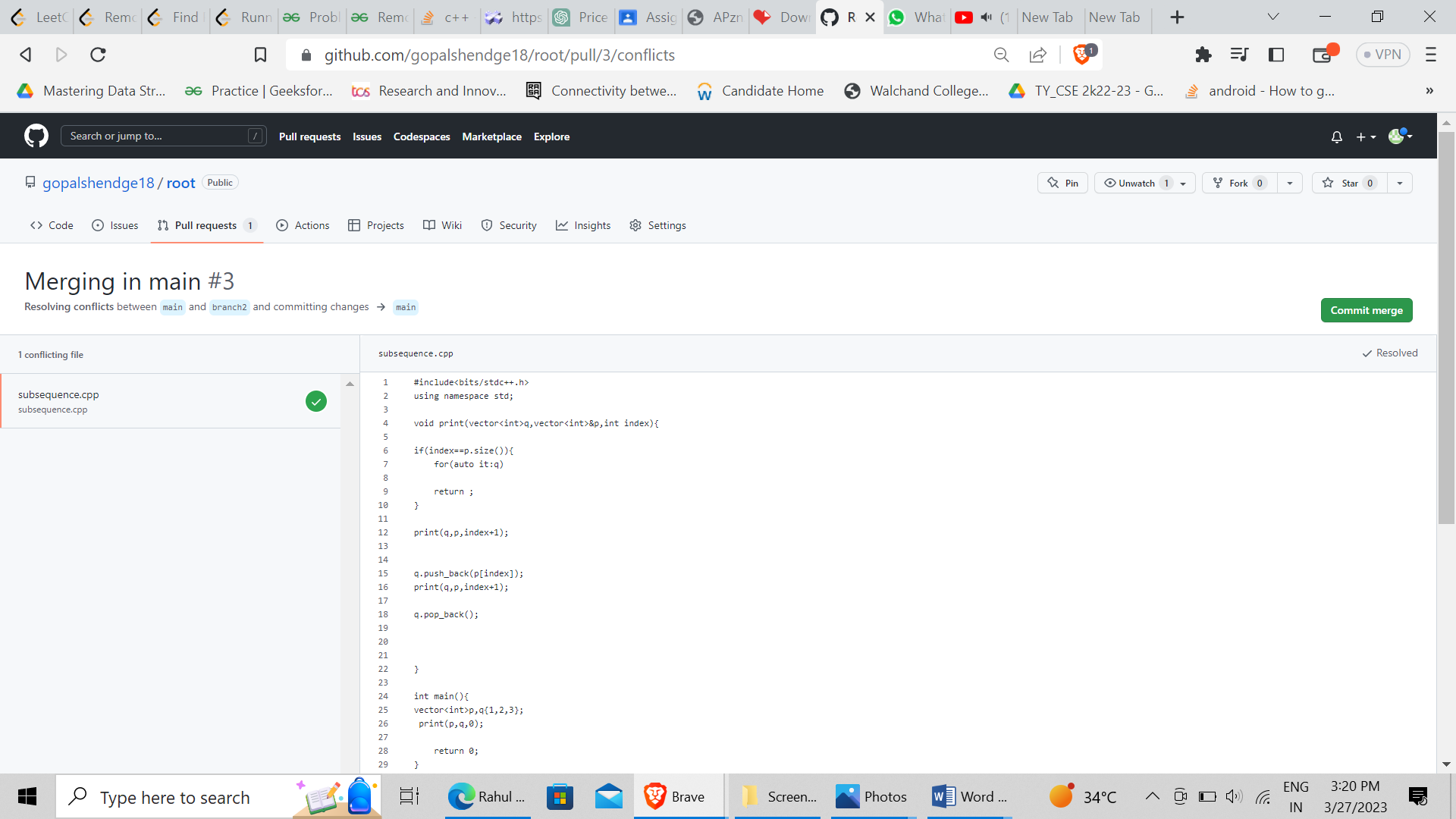
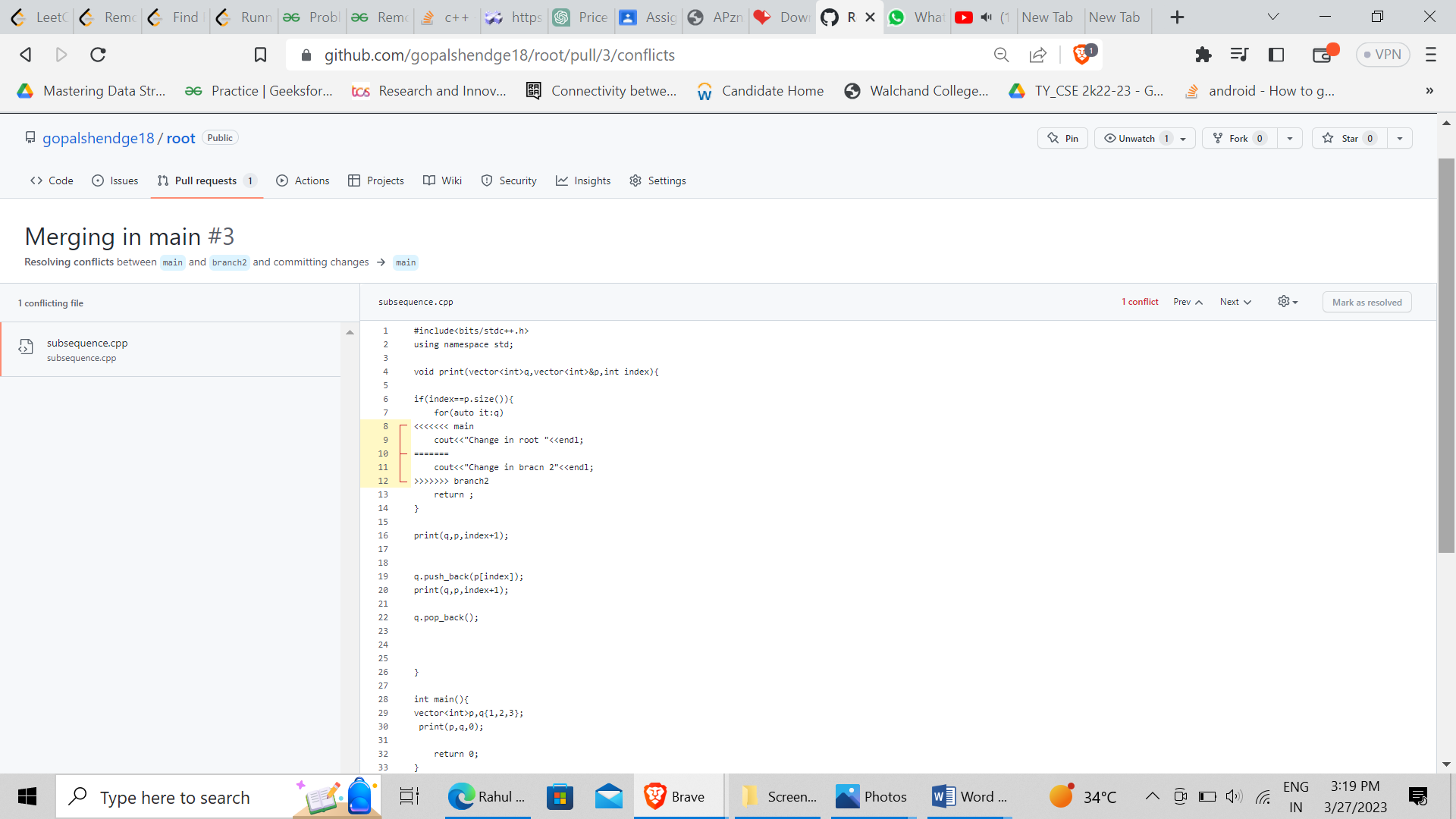
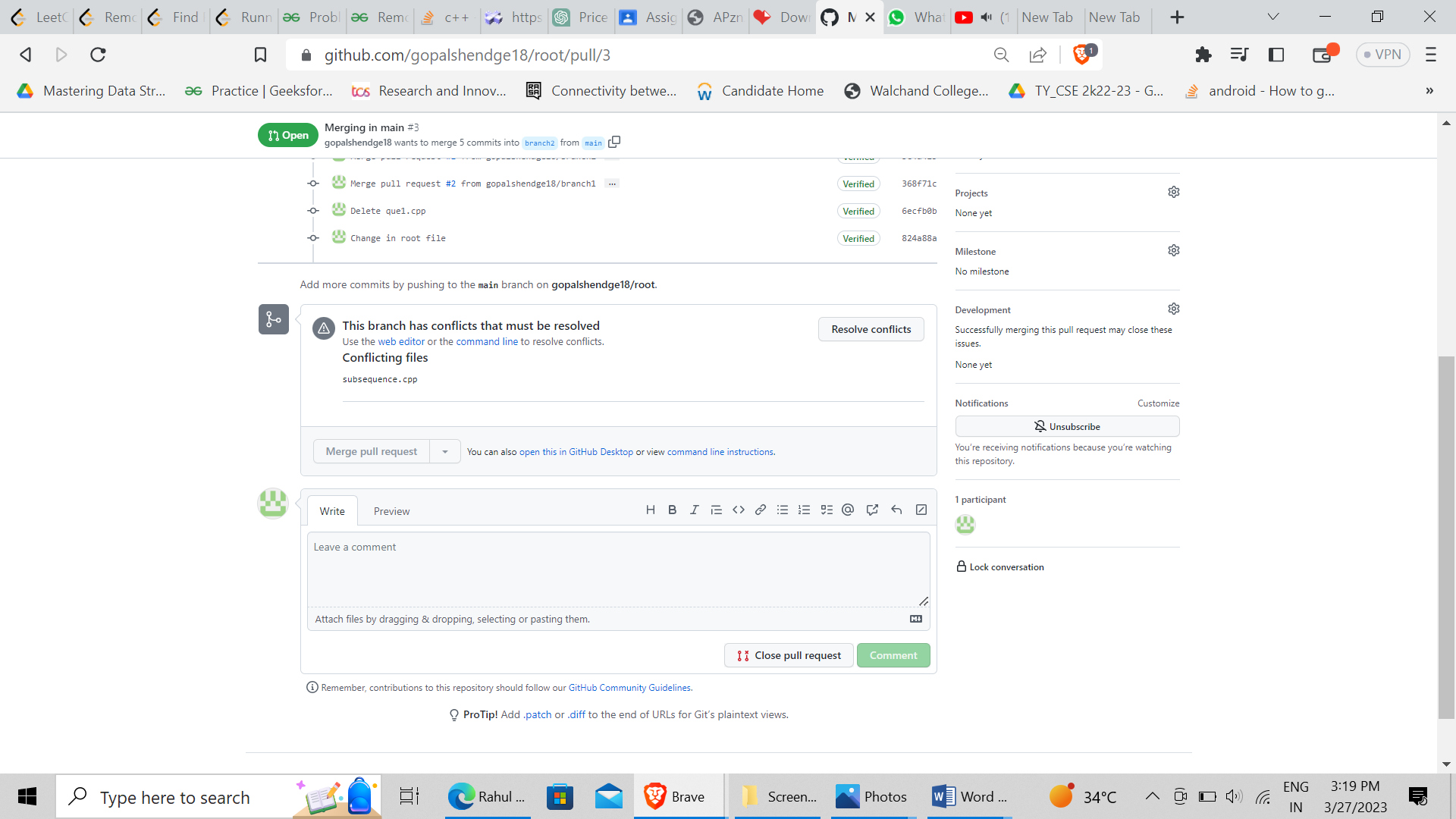
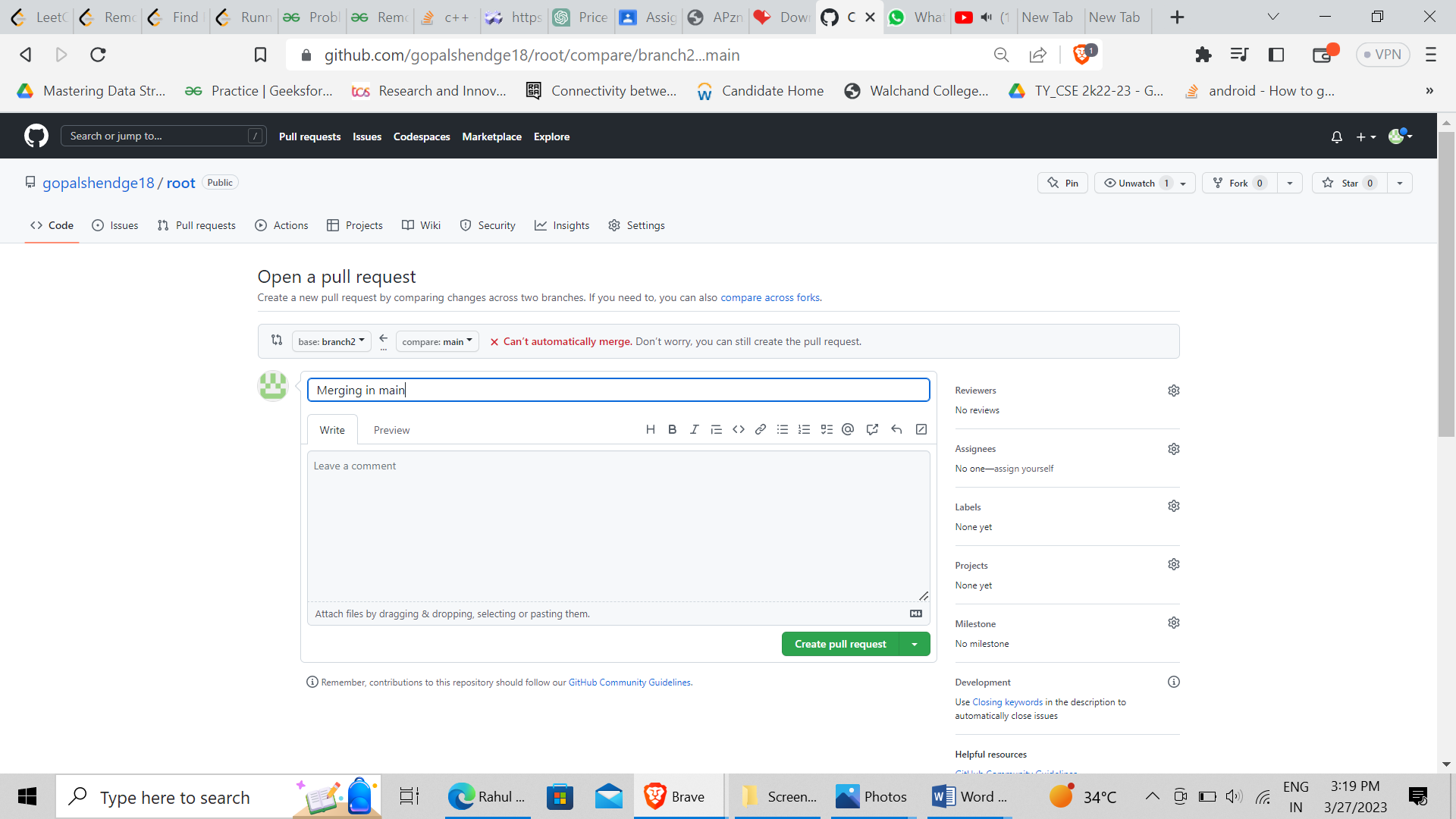
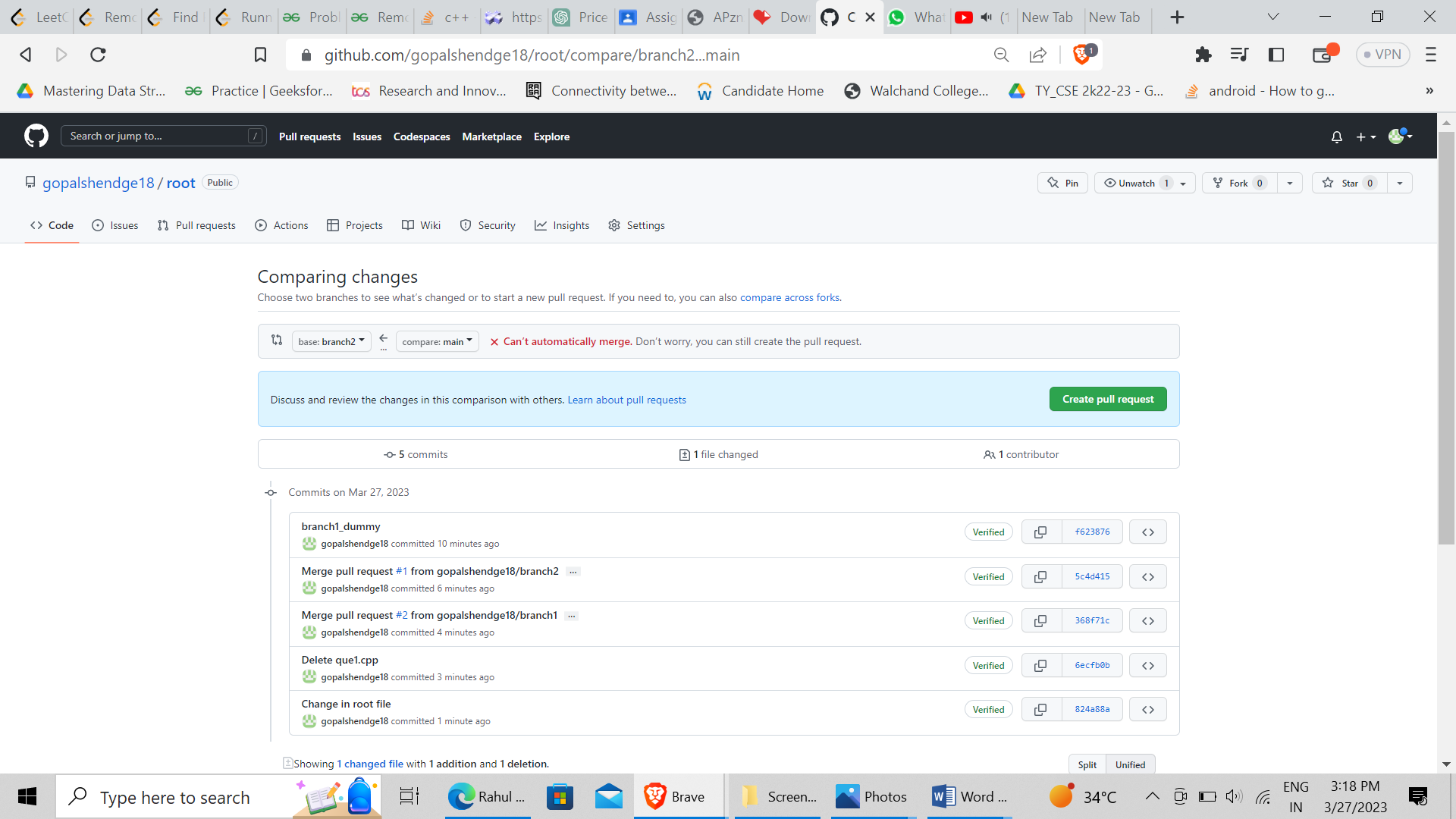
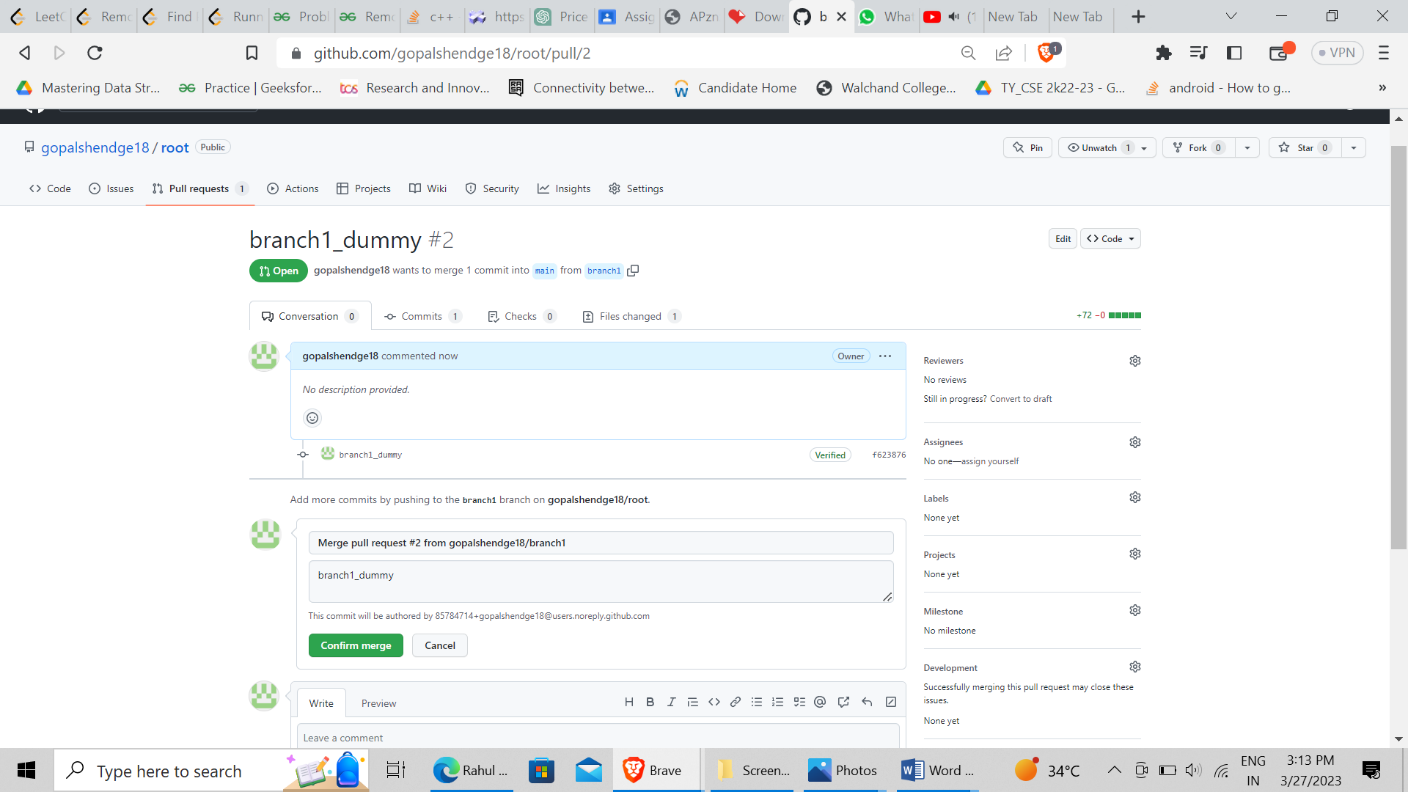
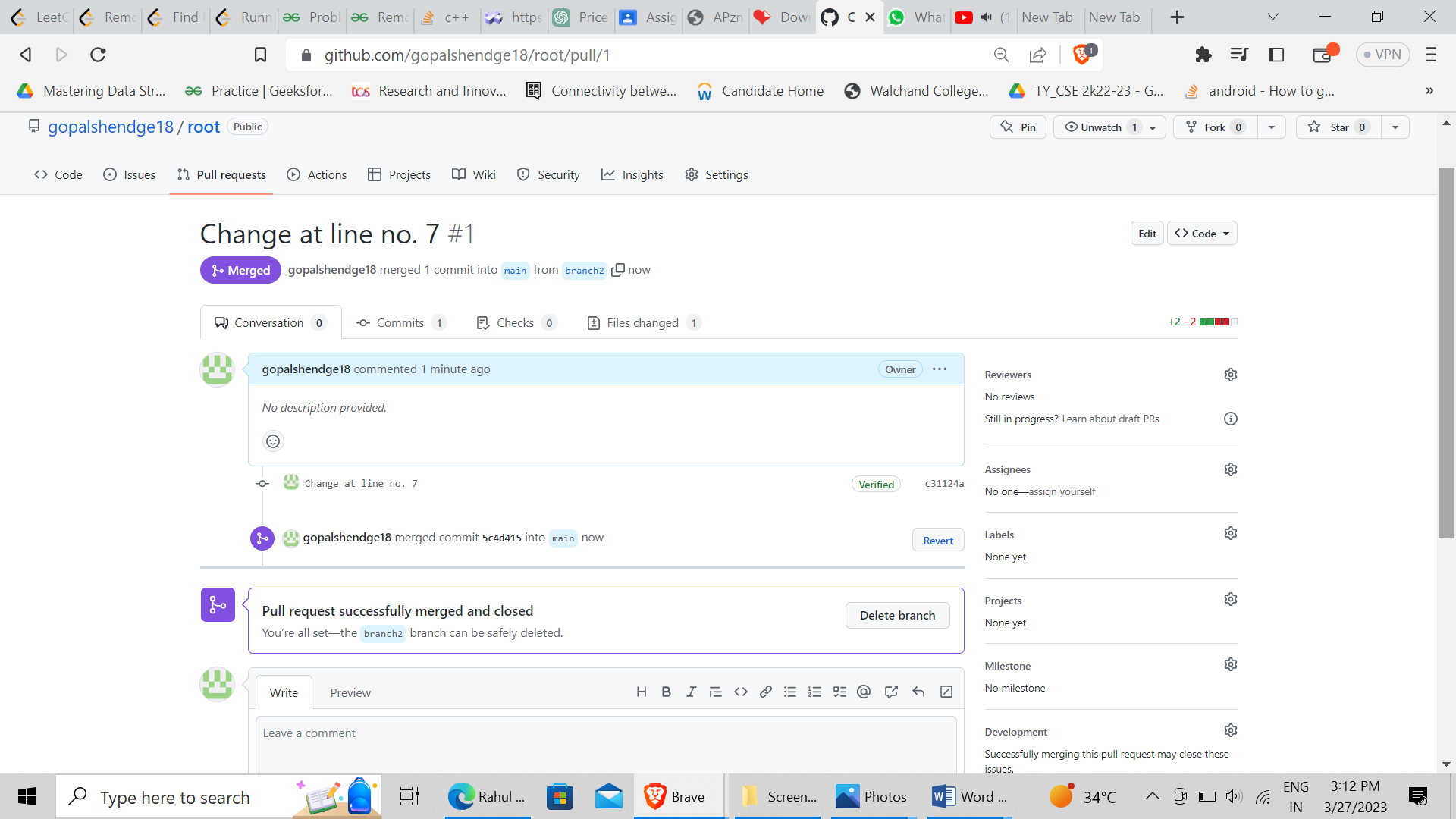
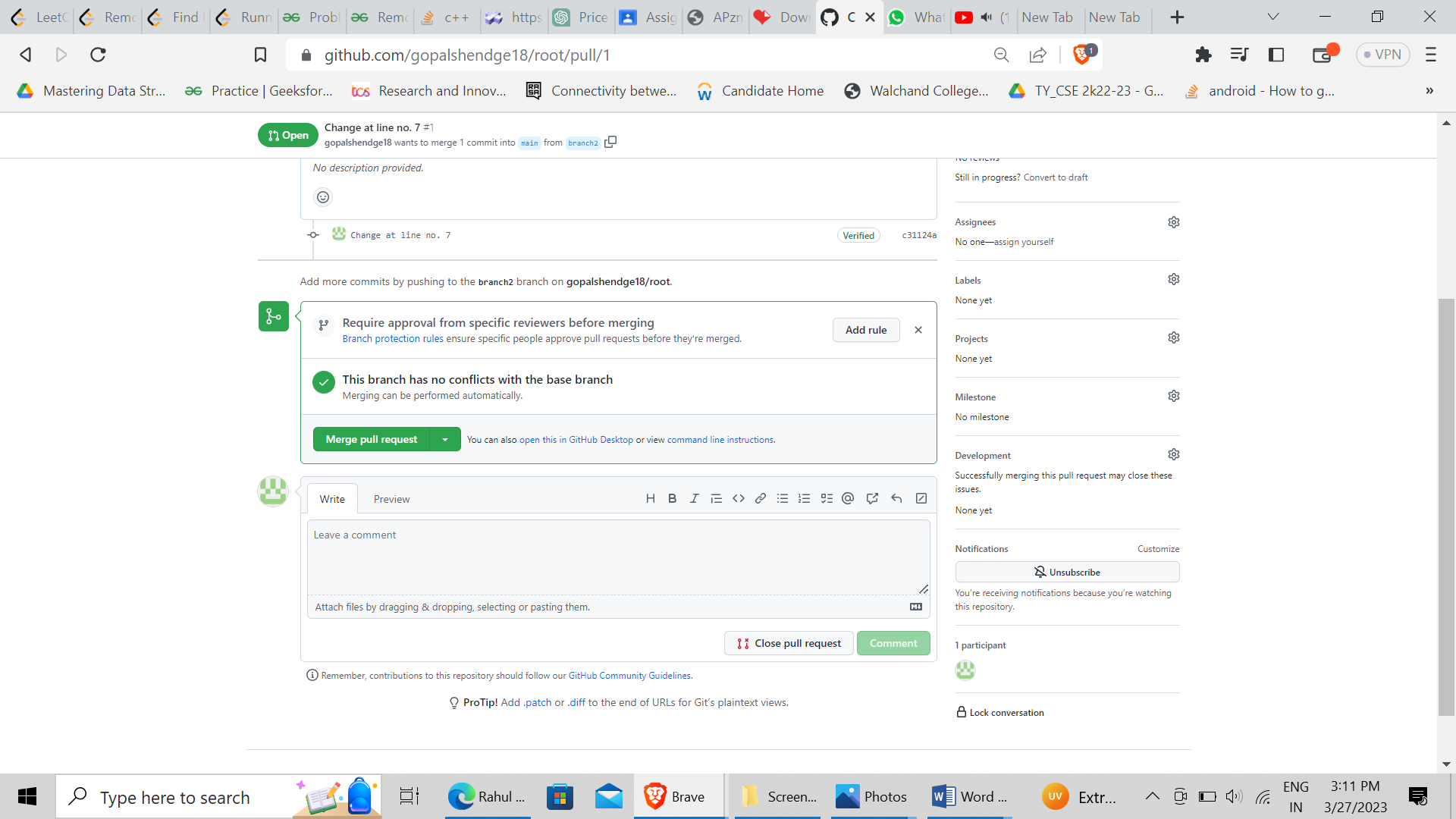
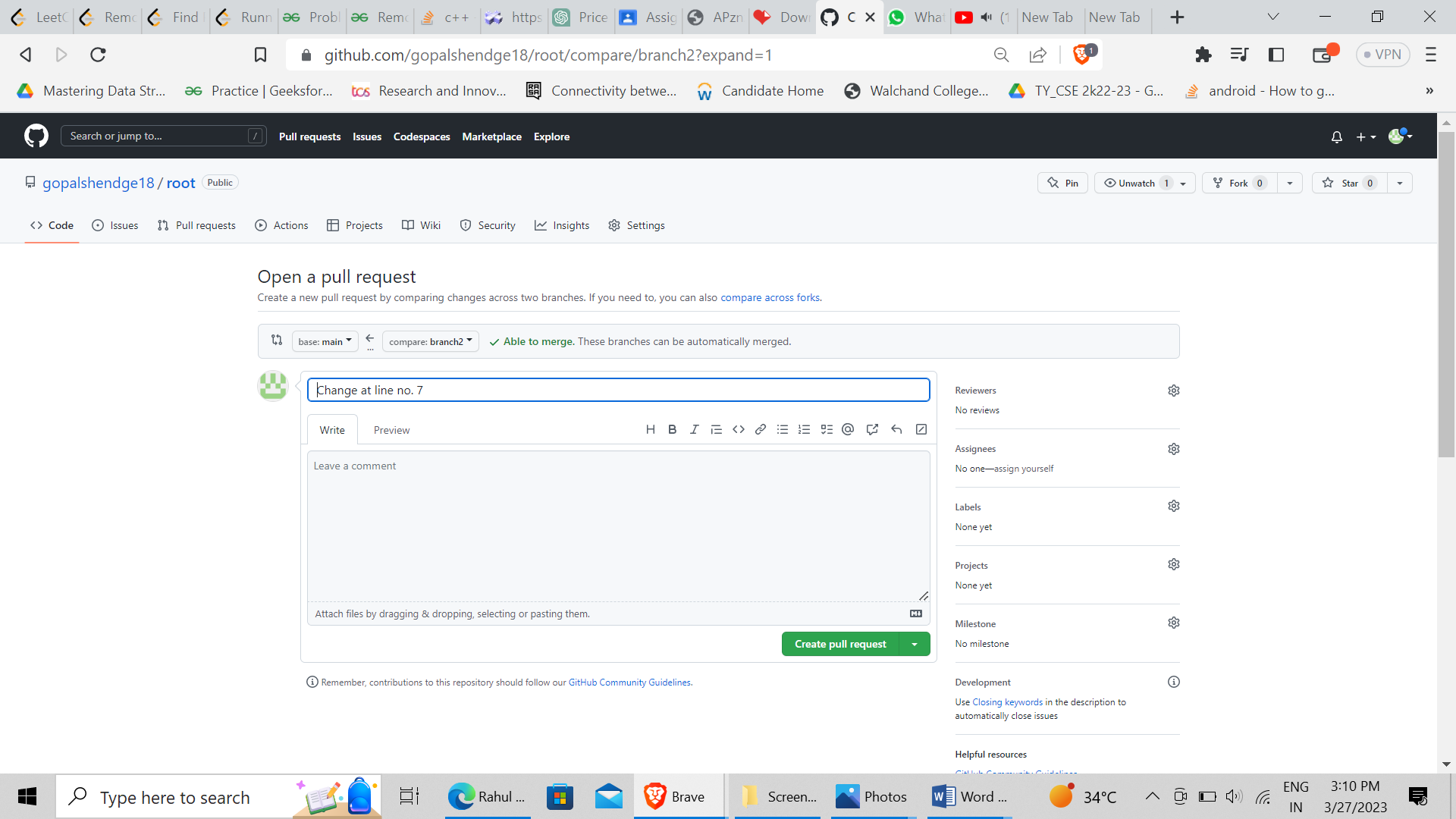




File added in main branch



Modifying file from main branch



Q 3. 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 graphical user interface (GUI) for managing Git repositories. It provides an easy-to-use interface for creating and managing repositories, committing changes, branching and merging, and collaborating with others.

To install GitHub Desktop on your local machine, follow these steps:

Go to the GitHub Desktop download page at https://desktop.github.com/.

Click the "Download for [your operating system]" button.

Once the download is complete, open the installer and follow the on-screen instructions to install GitHub Desktop on your local machine.

Here are the steps to access a repository on GitHub Desktop:

Open GitHub Desktop and sign in to your GitHub account.

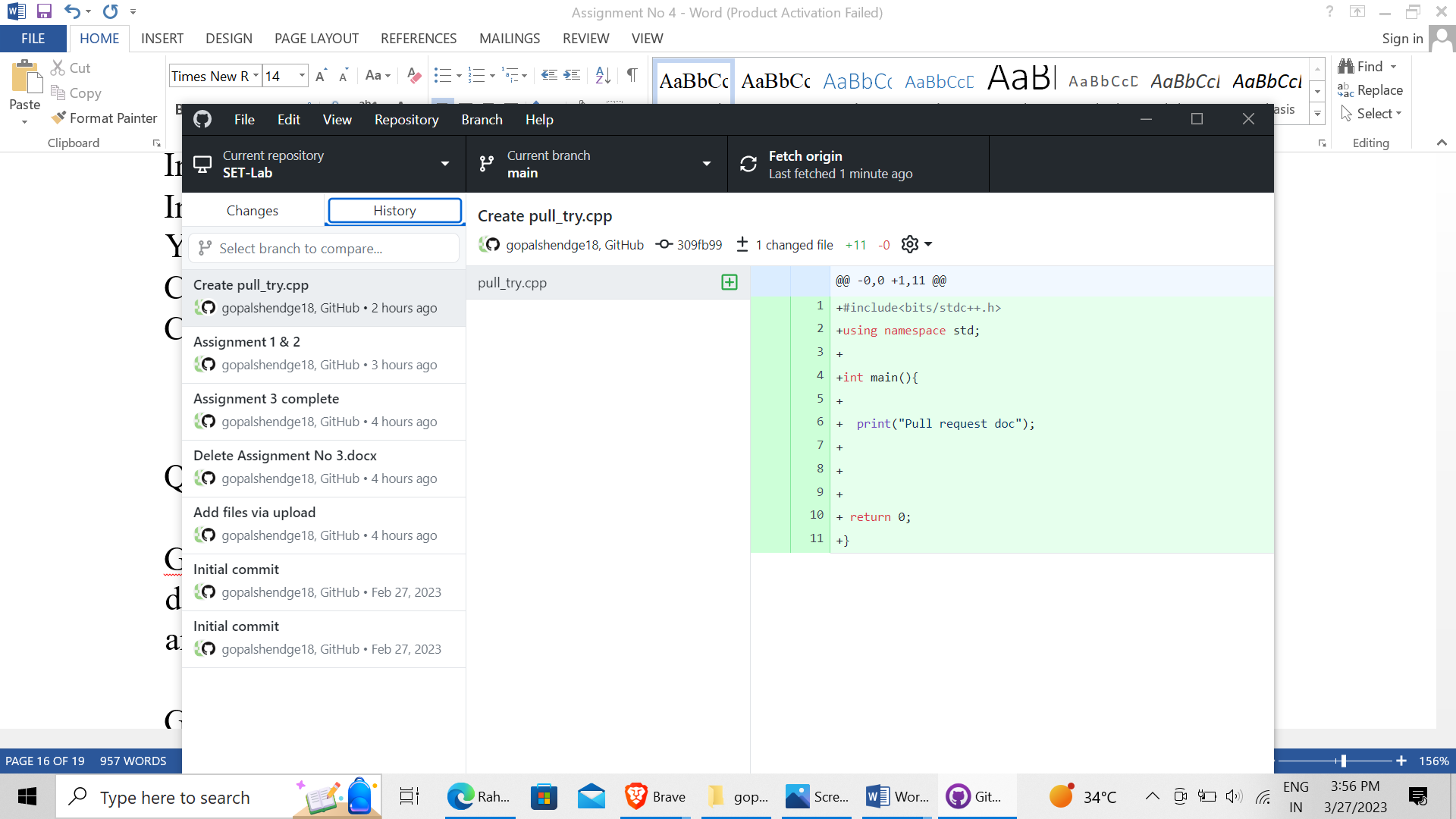
Click the "Add" button in the top left corner and select "Clone Repository".

In the "Clone a Repository" window, select the "URL" tab.

In the "Repository URL" field, enter the URL of the repository you want to clone. You can copy the URL of the repository from the browser address bar.

Choose the local path where you want to save the repository.

Click the "Clone" button to clone the repository.



Q 4. Differentiate in between GitHub, Git and GitLab.

GitHub, Git, and GitLab are all related to version control and software development, but they serve different purposes and have different features. Here are some key differences between them:

Git: Git is a distributed version control system that allows developers to manage changes to their source code over time. It was created by Linus Torvalds in 2005 and is used by developers around the world. Git is a command-line tool that allows developers to commit changes, create branches, merge code, and collaborate with others.

GitHub: GitHub is a web-based platform that provides hosting for Git repositories. It was founded in 2008 and is now owned by Microsoft. GitHub provides a user-friendly web interface for managing Git repositories, as well as features such as issue tracking, pull requests, and code reviews. It is commonly used by open source projects, individual developers, and organizations to collaborate on software development.

GitLab: GitLab is another web-based platform that provides hosting for Git repositories. It was founded in 2011 and is now owned by GitLab Inc. GitLab provides similar features to GitHub, such as issue tracking and code reviews, but also includes additional features such as continuous integration and continuous deployment (CI/CD). GitLab is often used by enterprises and organizations that require more advanced features for their software development process.

In summary, Git is the command-line tool that developers use to manage their code changes, while GitHub and GitLab are web-based platforms that provide hosting for Git repositories and additional features to support software development. GitHub is widely used for open source projects and individual developers, while GitLab is often used by enterprises and organizations that require more advanced features.

Q 5. What is version control? Explain with example.

Version control is the practice of managing changes to a codebase or document over time. It allows developers and collaborators to track modifications, compare versions, and revert to earlier versions if necessary. The main purpose of version control is to keep a record of changes made to a project, and enable collaboration among multiple contributors.

Here is an example of how version control works in software development:

Suppose a team of developers is working on a web application. They use version control to manage the codebase of the application. The team uses Git, a popular version control system.

At the beginning of the project, they create a Git repository to store the code. Each developer on the team has a copy of the repository on their local machine. They work on different features of the application and make changes to the code.

When a developer makes changes to the code, they commit their changes to the local repository. They include a message that describes the changes they made. For example, "Added login form to the homepage" or "Fixed bug in the search feature."

Once the developer is ready to share their changes with the rest of the team, they push their commits to the remote repository on a server, such as GitHub or GitLab. The remote repository stores a complete history of all changes made to the code.

If a developer needs to see the changes made to a specific file, they can use Git to compare different versions of the file. They can see what lines of code were added, deleted, or modified.

If a developer makes a mistake or introduces a bug, they can use Git to revert to an earlier version of the code. This helps to avoid lost work and makes it easier to fix issues.

In summary, version control enables developers to manage code changes, collaborate with others, and maintain a record of the project history. Git is a popular version control system used by developers around the world to manage their codebases.