PAVITHRA PS

21ADL064

GIT PROJECT

KEC VECHICLE SERVICE

# Create a New Repository on GitHub:

* Go to GitHub.com and log in to your account.
* Click on the "+" icon in the top right corner and select "New repository."
* Fill in the repository name, description, and choose other settings as needed.
* Click "Create repository."

# Initialize a Git Repository Locally:

* Open Git Bash on your computer.
* Navigate to the directory where you want to create your local repository using the

`cd` command.

**cd E:\Faculty Leave Management System git init -** Initialize a new Git repository

# Link Local Repository to the GitHub Repository:

Copy the URL of your GitHub repository.

- Link your local repository to the GitHub repository

# git remote add origin https://github.com/Ruthranayaki213/FLMS\_Web.git

1. **Add Files to the Local Repository:**

* Place the files you want to add to the repository in the directory.
* Use the following command to stage the files for commit

**git add** - This command stages all the files in the directory.

If you only want to add specific files, replace `**.`** with the file names.

# Commit the Changes:

- Commit the staged files with a descriptive message

# git commit -m "Initial commit”

Replace `"Initial commit"` with your own descriptive message.

# Push Changes to GitHub:

- Push the committed changes to the GitHub repository

# git push -u origin master

This command pushes the changes to the `master` branch of the remote repository.

# Check status

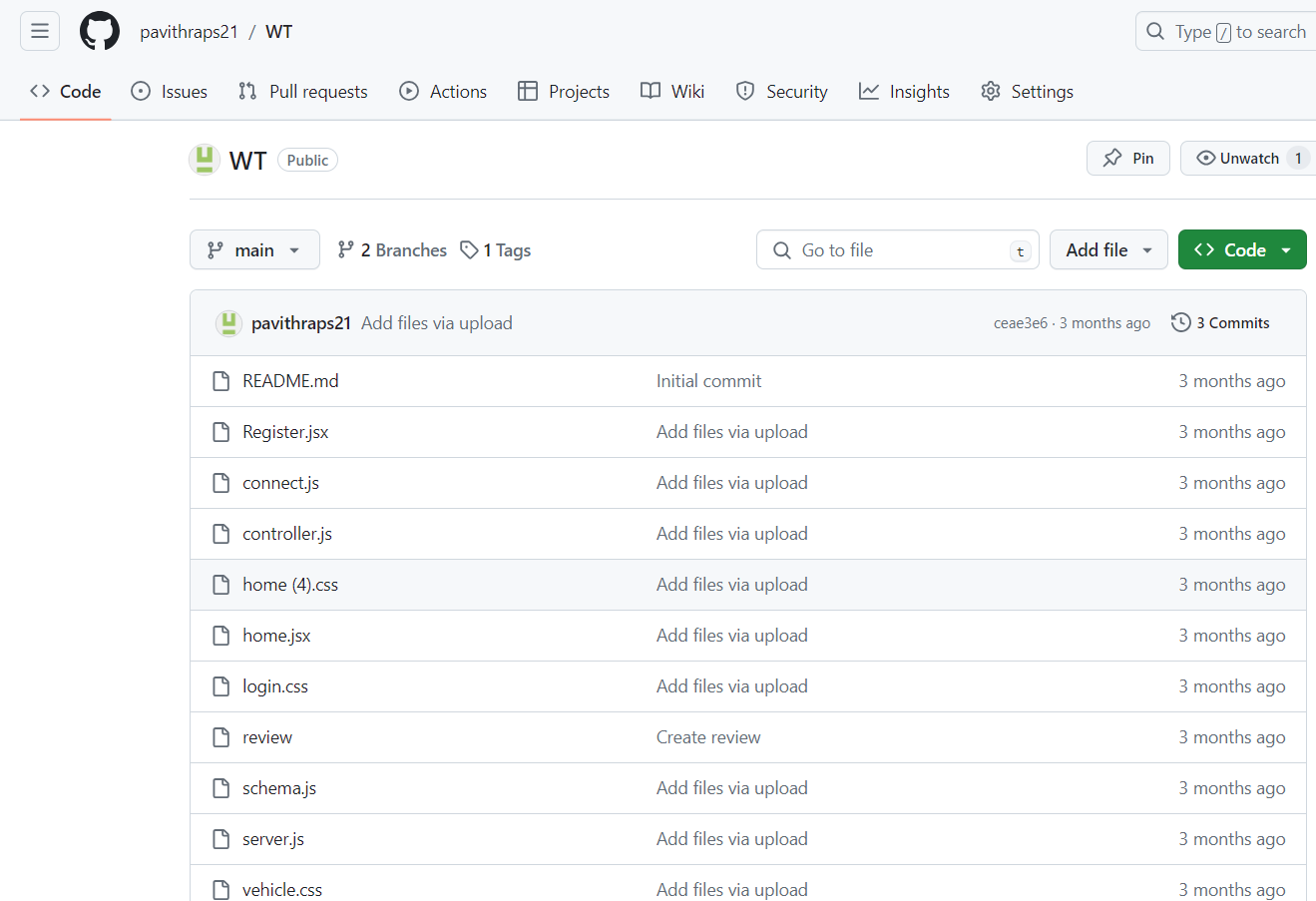
- The current status of your repository, including any changes not yet staged or committed.

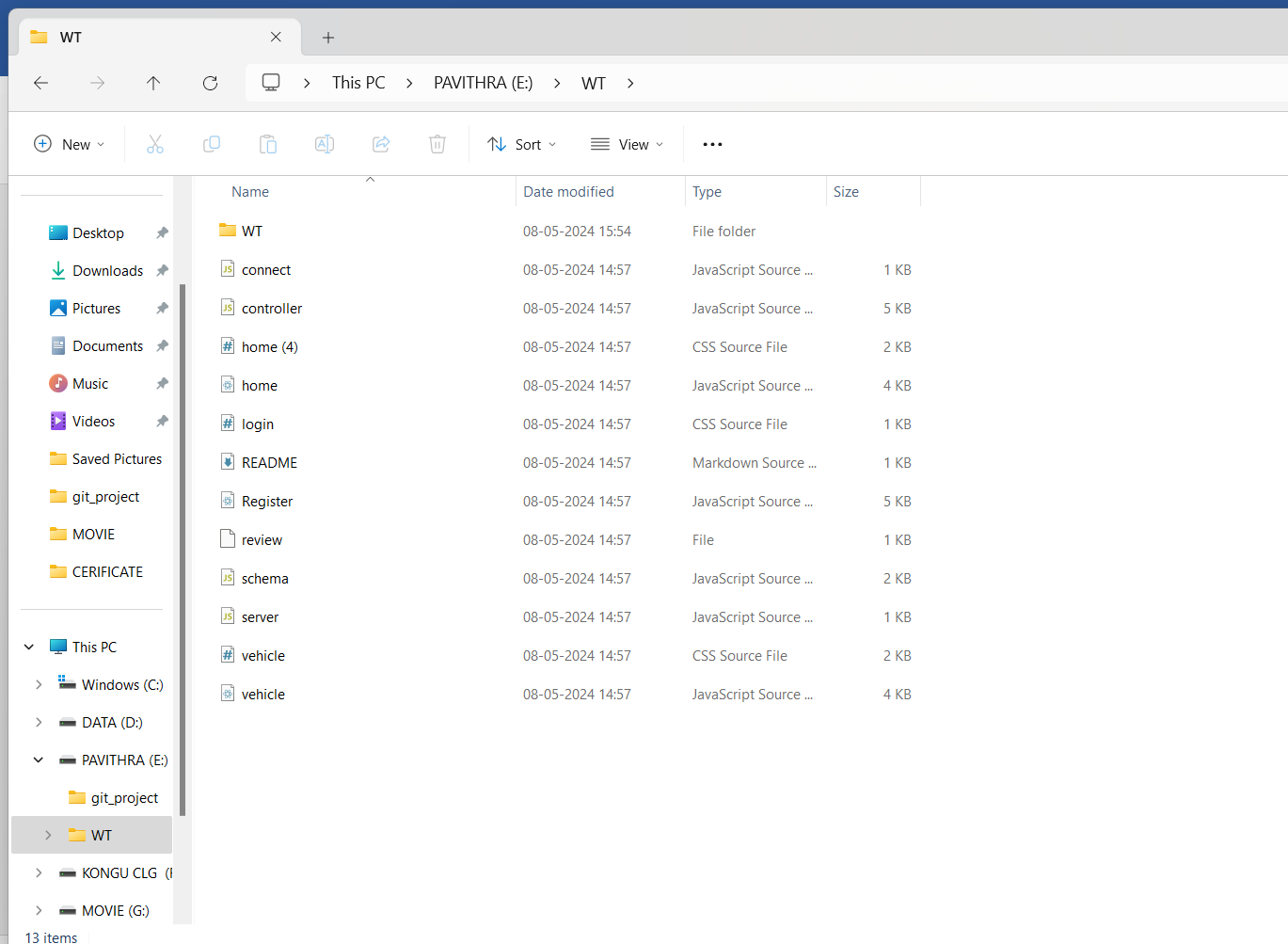
# git status

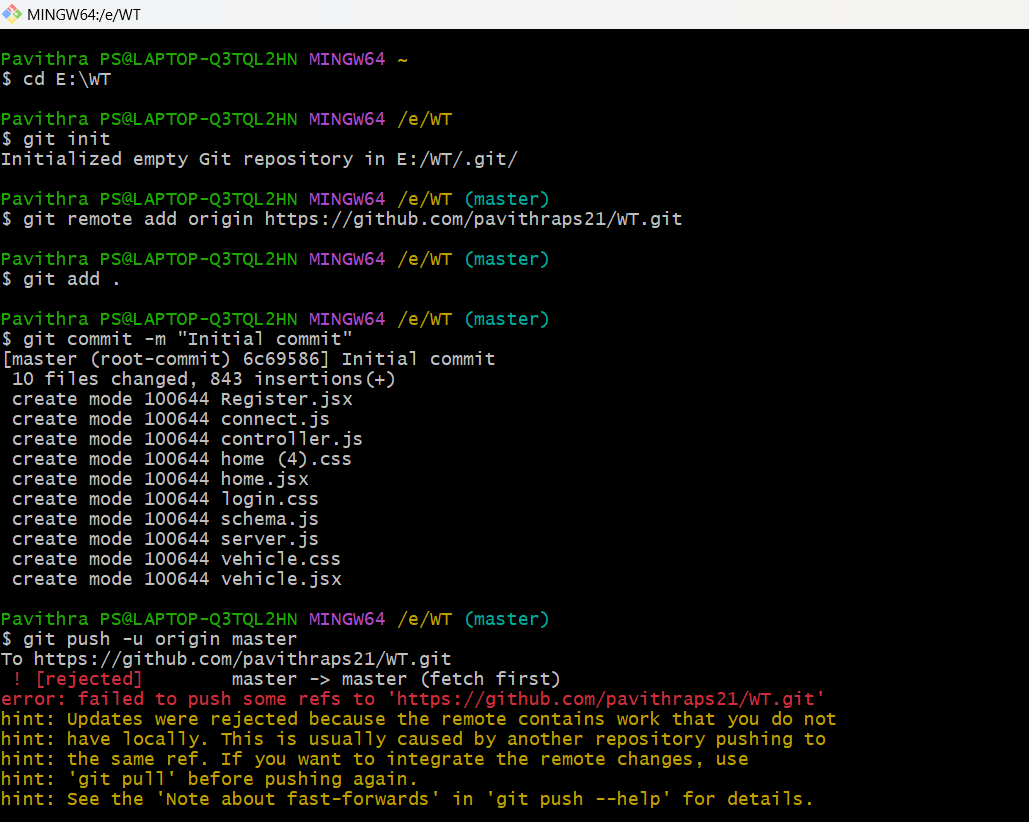
1. **Pull Changes from Remote**

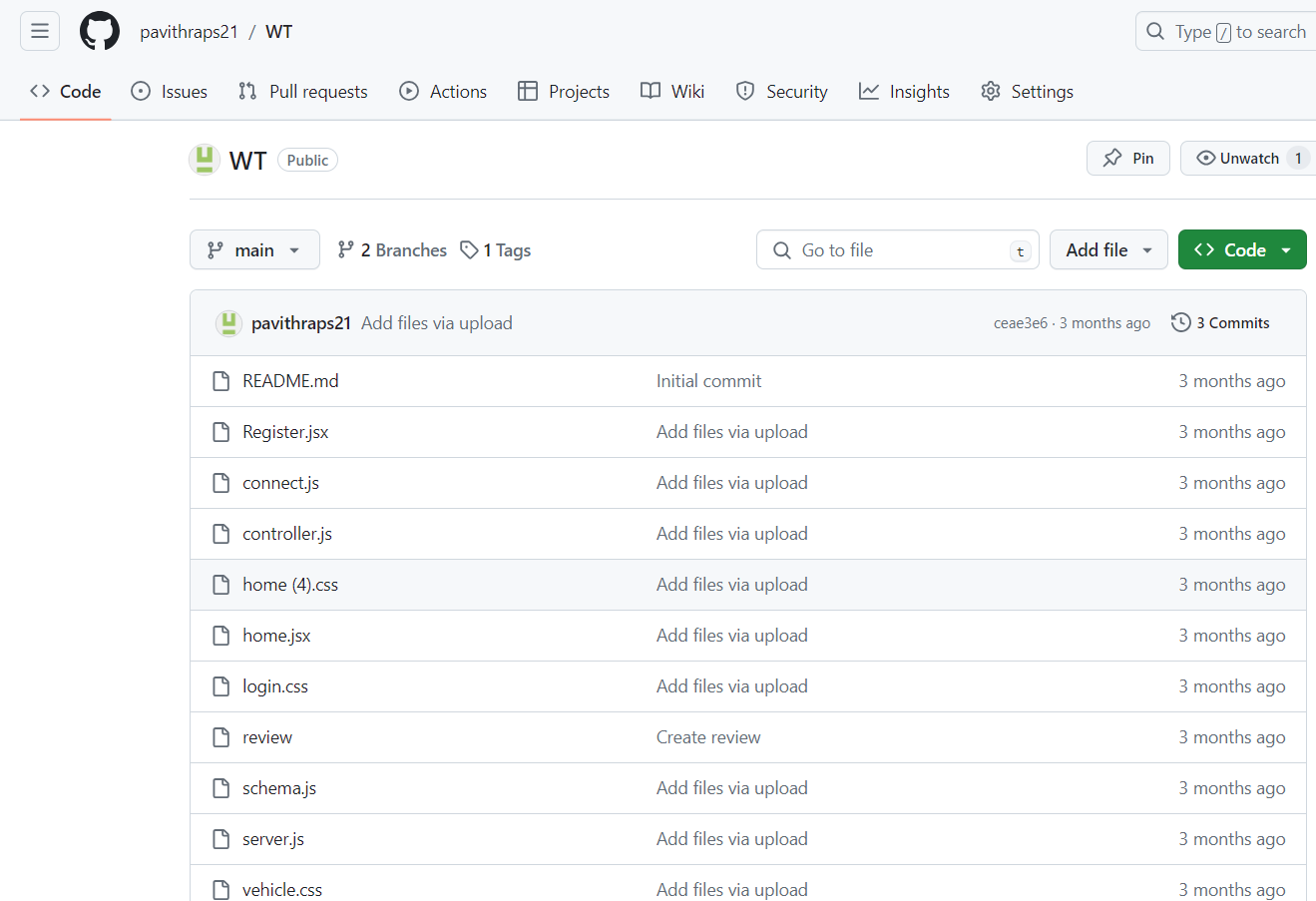
- fetch and merge changes from the remote repository into your local branch

# git pull origin master



****





1. **Create and Switch Branches:**

# git checkout –b facultydetail

Use this command to create a new branch and switch to it.

# Add and Commit Changes:

**git add \*.txt**

stage specific files for the next commit.

# git commit -m "new file added"

This command commits the staged changes with a descriptive message.

# Push Changes to a Different Branch:

**git push origin master**

# View Commit History:

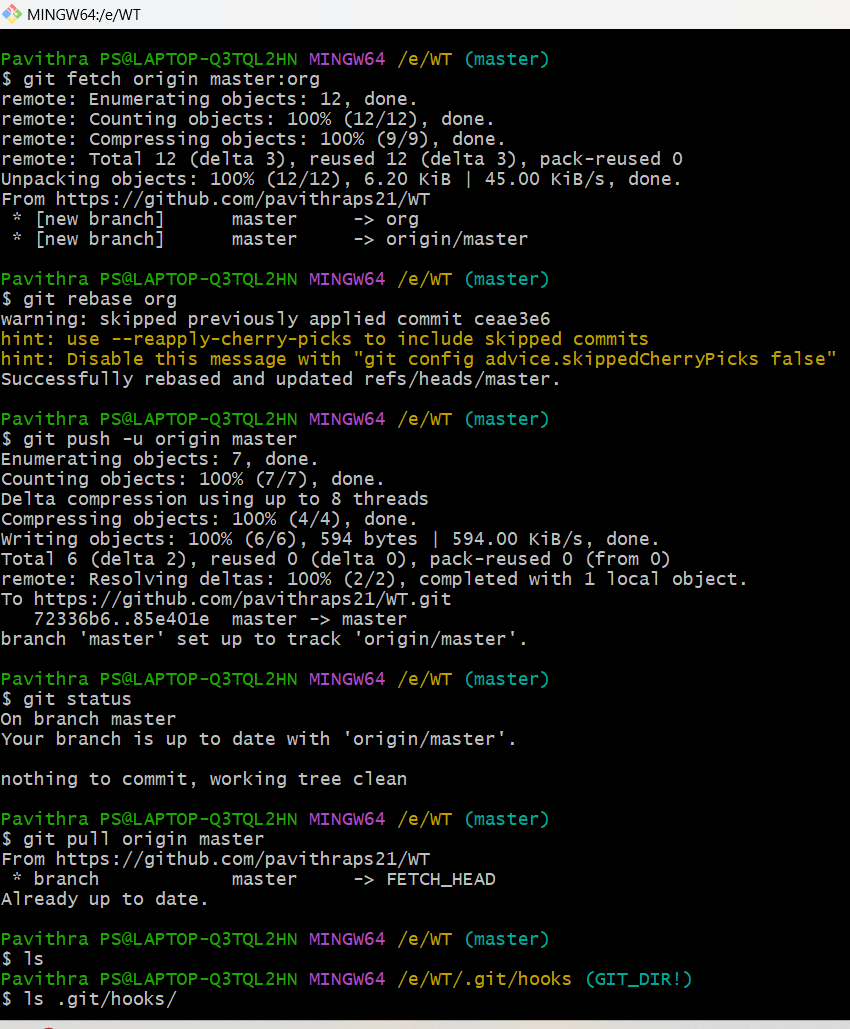
**git log**

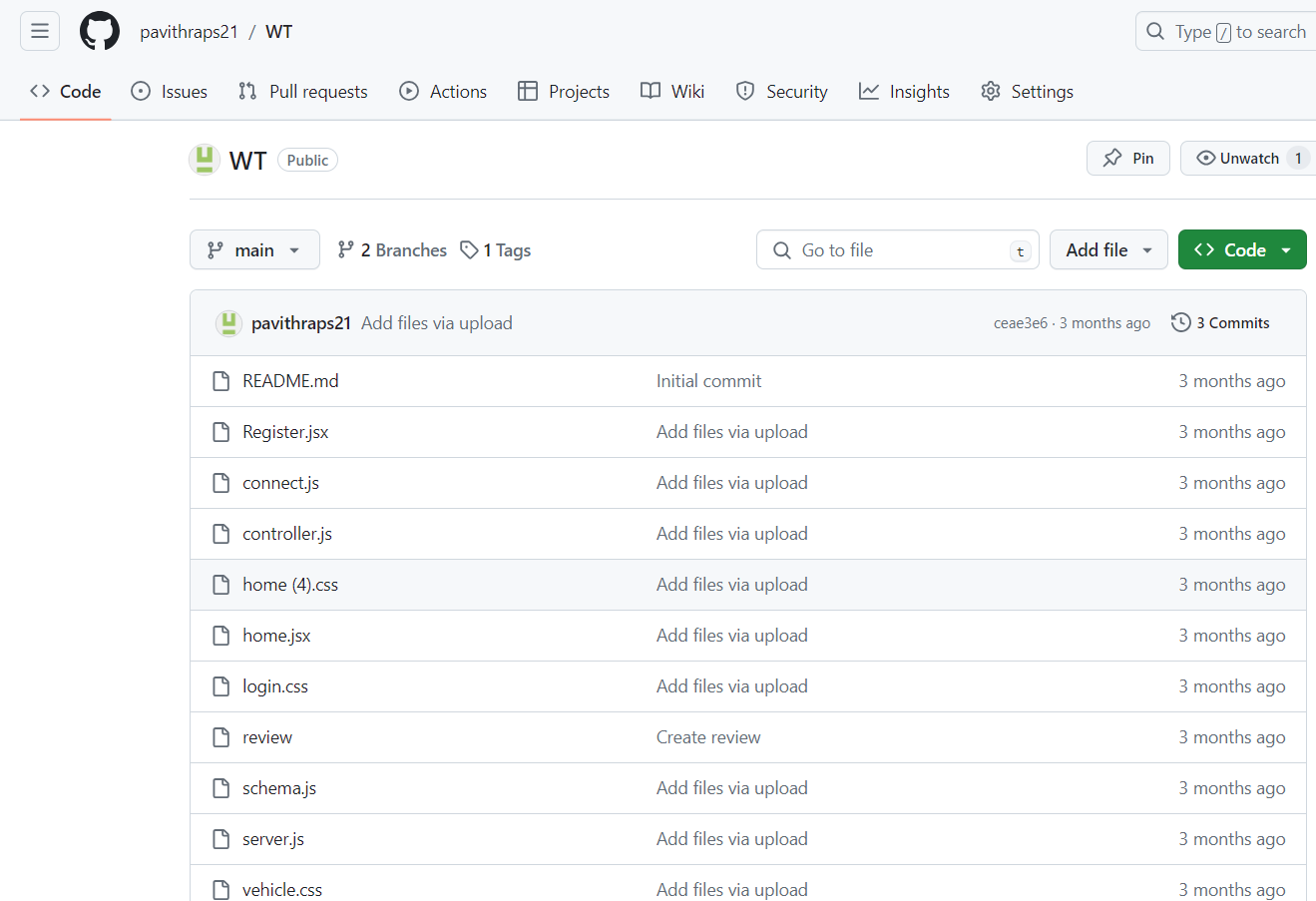
- displays a list of commits in reverse chronological order, showing the commit hash, author, date, and commit message.

# Check the Current Directory: ls

-use the ls command to list the files and directories in the current directory

# Review and Merge: git checkout master git merge facultydetail git push origin master





1. **Branching:**

Create a new branch

**git checkout –b leavedetail** Switch to an existing branch **git checkout master**

List all branches

# git branch

**s**ummarized version of the commit history

# git log --oneline

**pull changes from the remote repository into the current branch git clone https://github.com/pavithraps21/WT**

# Working with Tags:

List all tags:

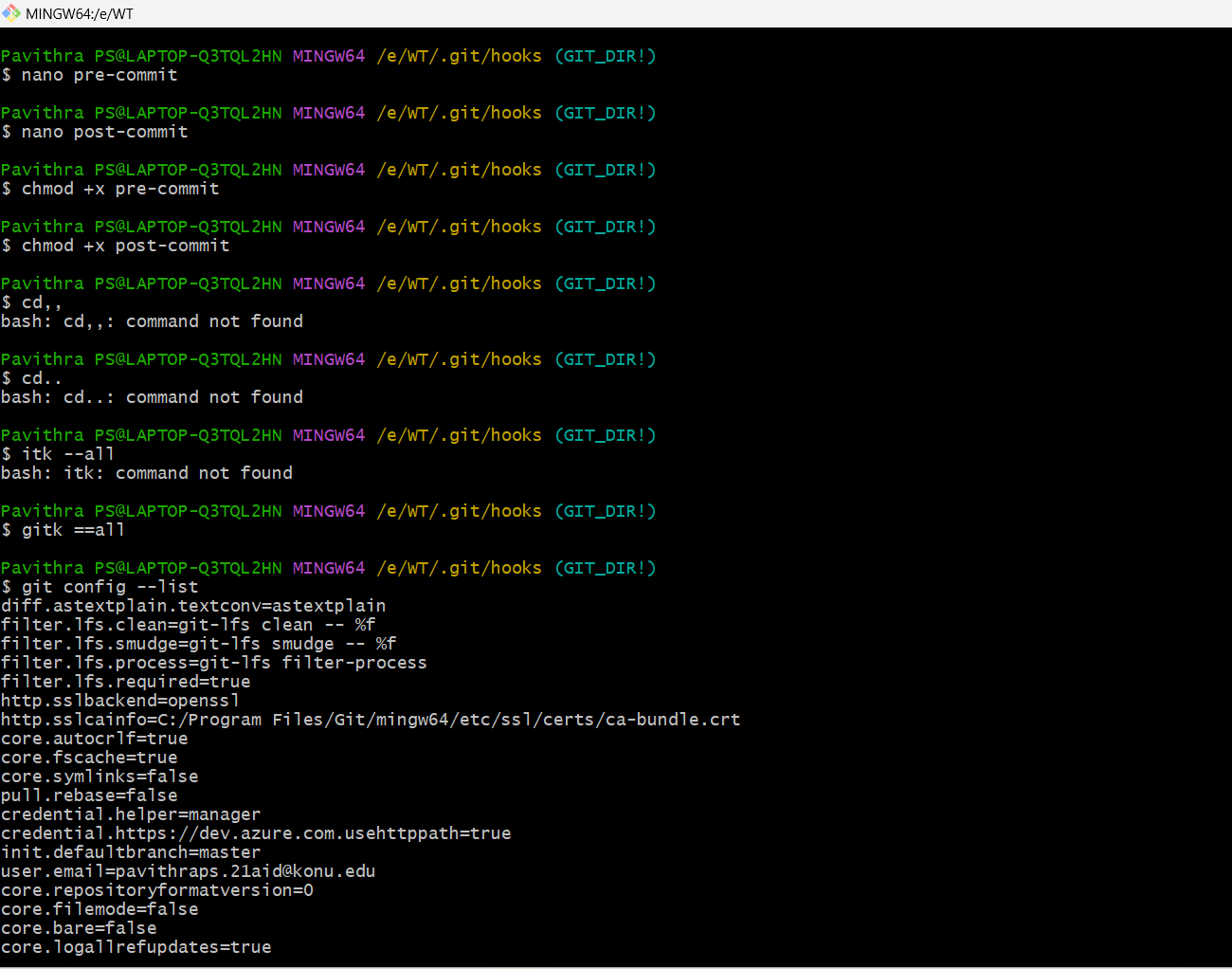
# git tag

Create a new tag:

# git tag flms

Push tags to the remote repository:

# git push –tags



1. **Stage the Changes:**

After making the necessary changes, stage the modified file using the `git add` command

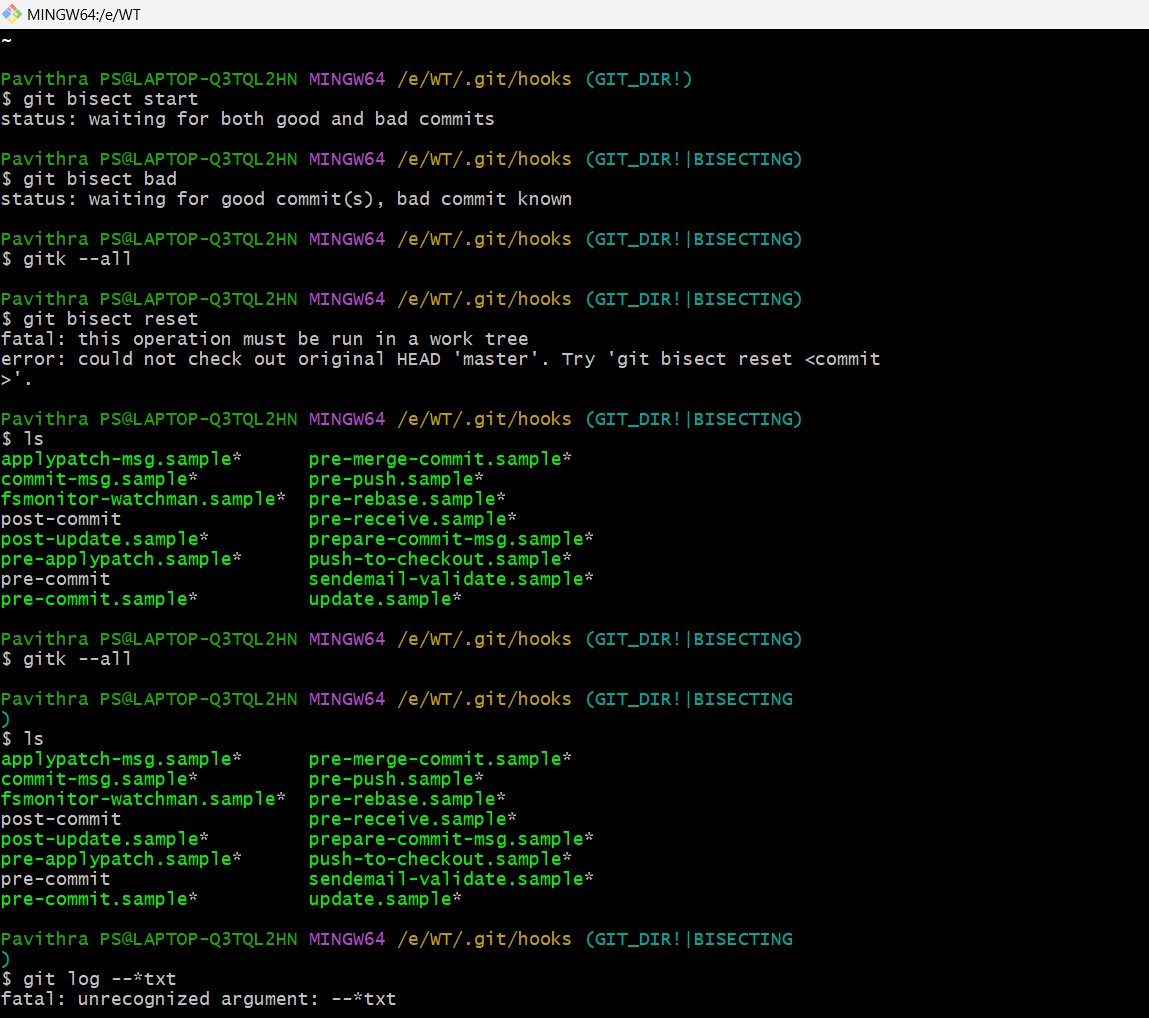
# git add .\*txt

**Commit the Change**

# 

# git commit --amend

**git commit --amend --no-edit**



# Push the Changes

pushed the original commit to a remote repository, you may need to force-push the amended commit

# git push --force

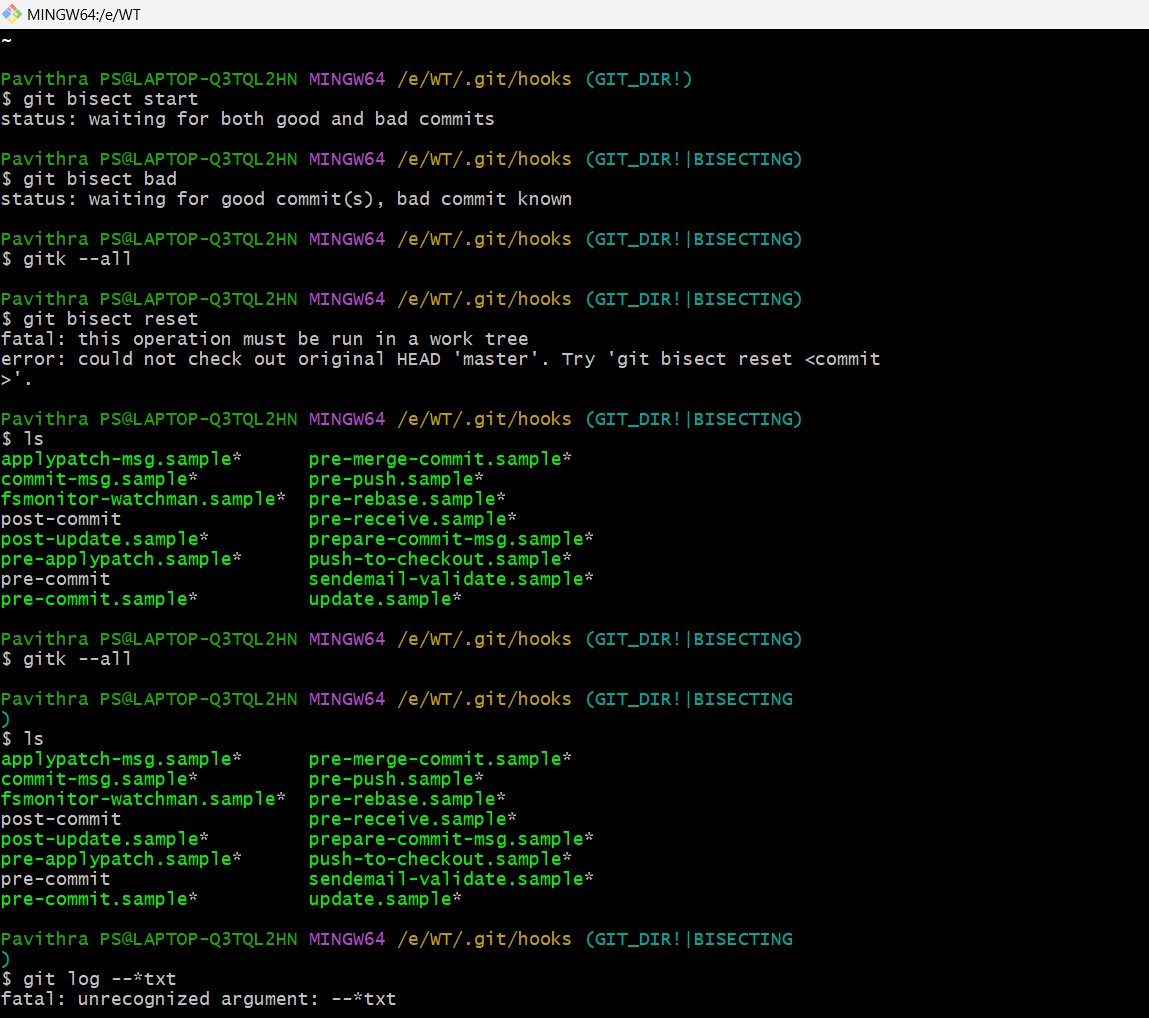
1. **Browsing History**

View the commit history as a graph

# git log --graph --oneline –all

1. **Searching**

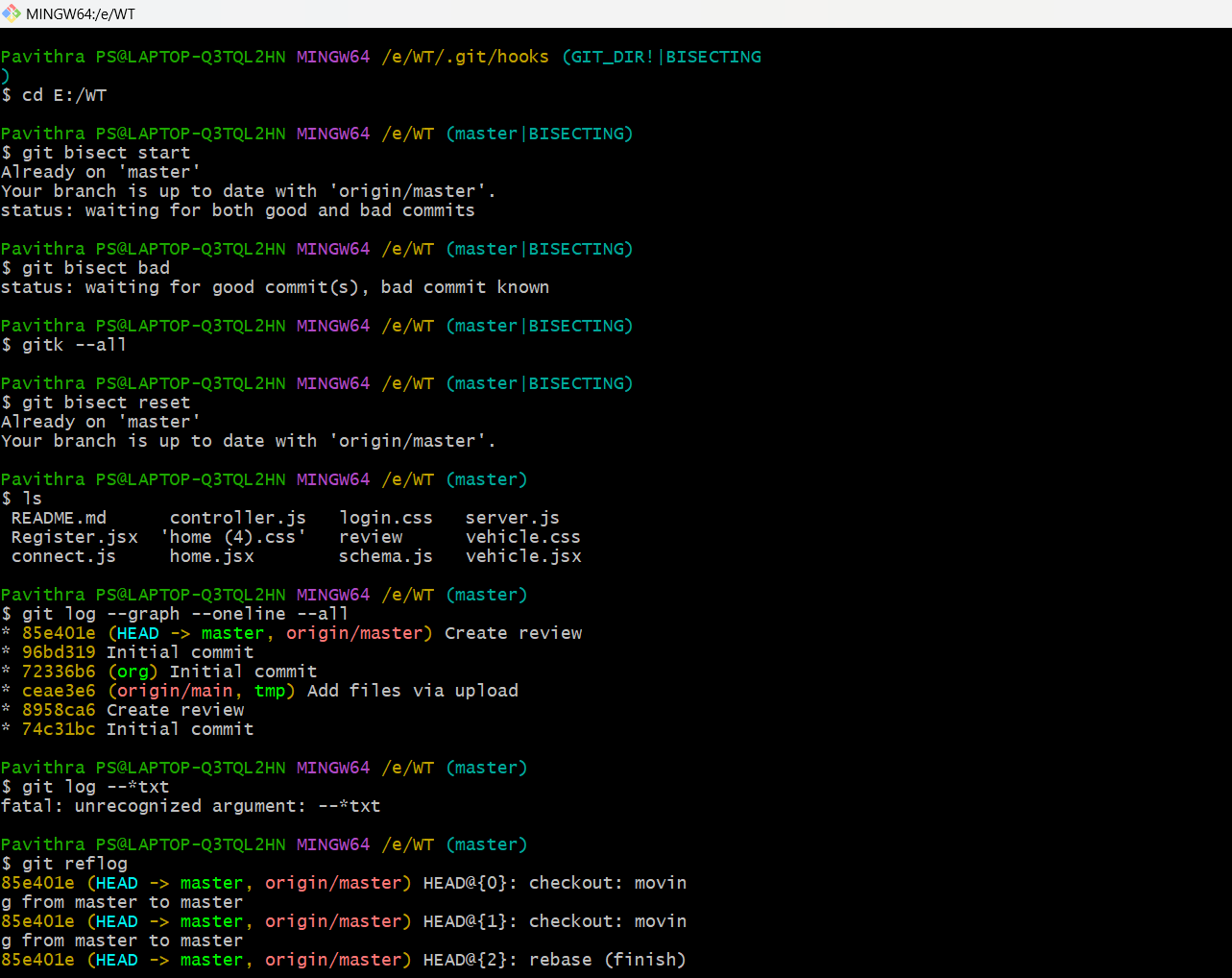
# git log --grep="<keyword>"Searching



1. **Reflogs:**

View the reflog, which records changes to branch tips

# git reflog



1. **Annotating Files:**

View file annotations (blame) to see who last modified each line

# git blame

1. **Working with Tags:**

# List all tags:

**git tag**

Show information about a specific tag

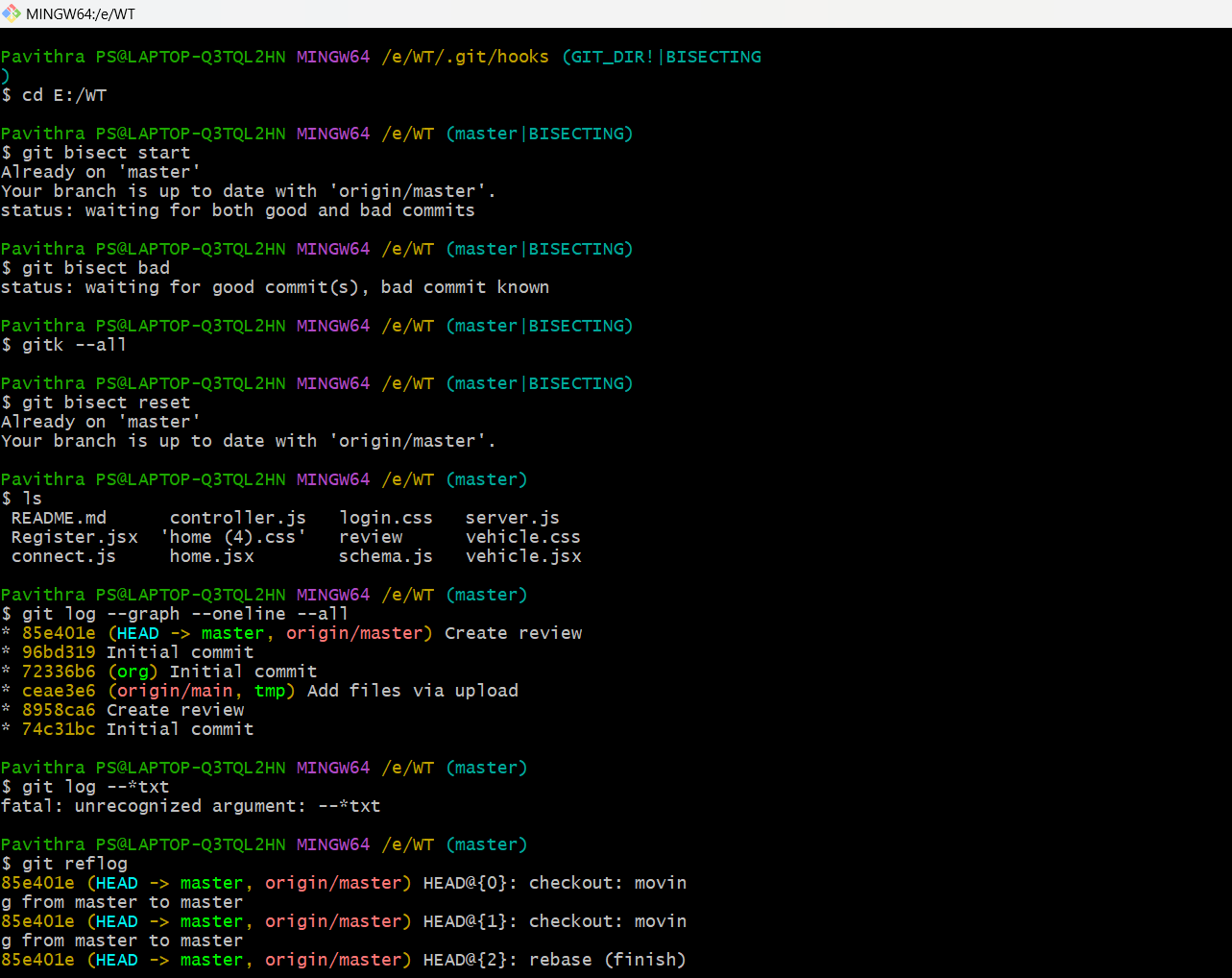
# git show flms

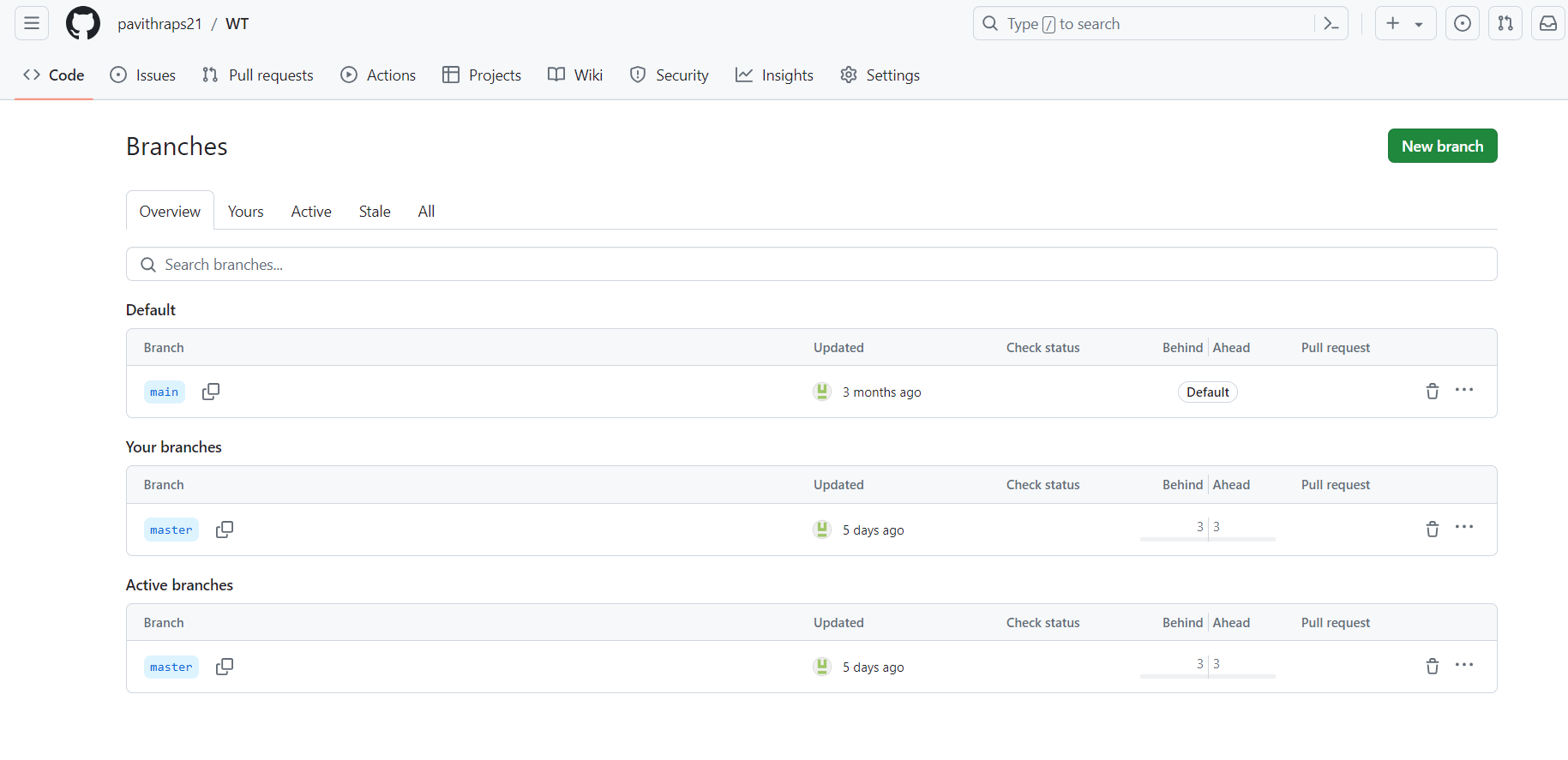
1. **Bisecting:**

Use binary search to find the commit that introduced a bug

# git bisect start git bisect bad

**git bisect good <commit\_hash>**



****

# Navigate to the Hooks Director

**cd /git/hooks**

List Available Hook

# ls

View Hook Sample Scripts

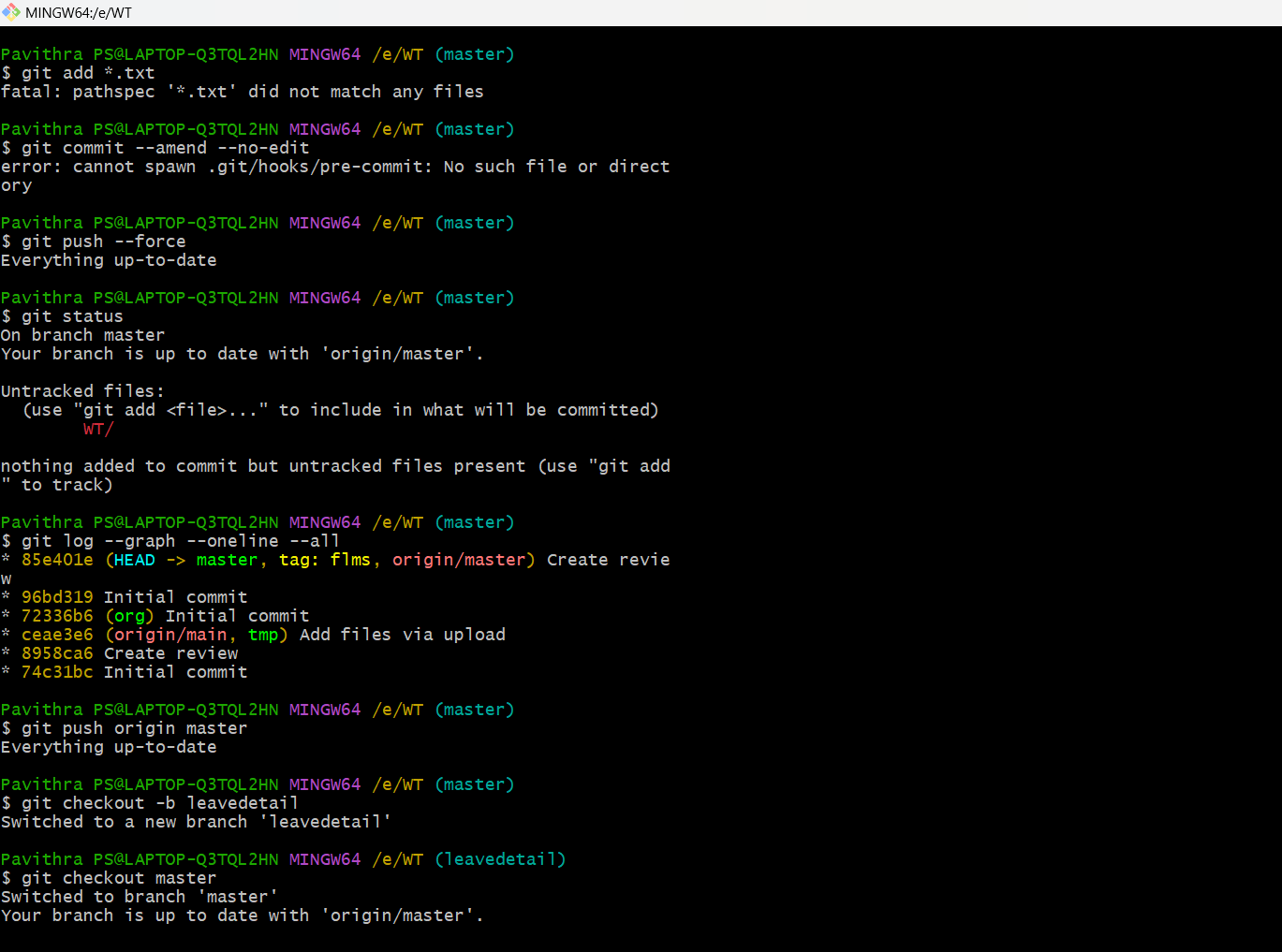
# ls \*.sample

Create a New Hook Script

**touch pre-commit** Edit the Hook Script **Nano pre-commit**

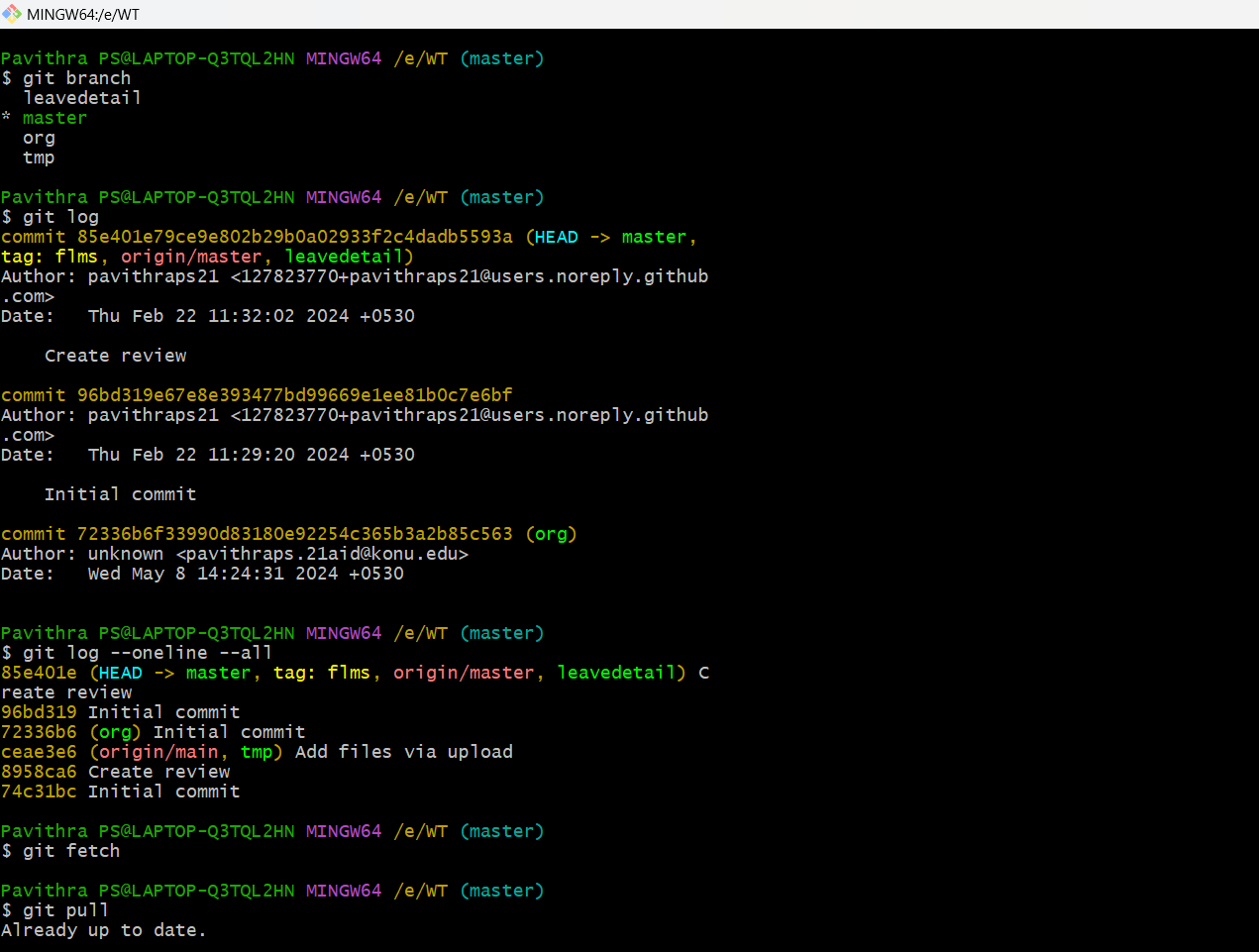
Make the Script Executable

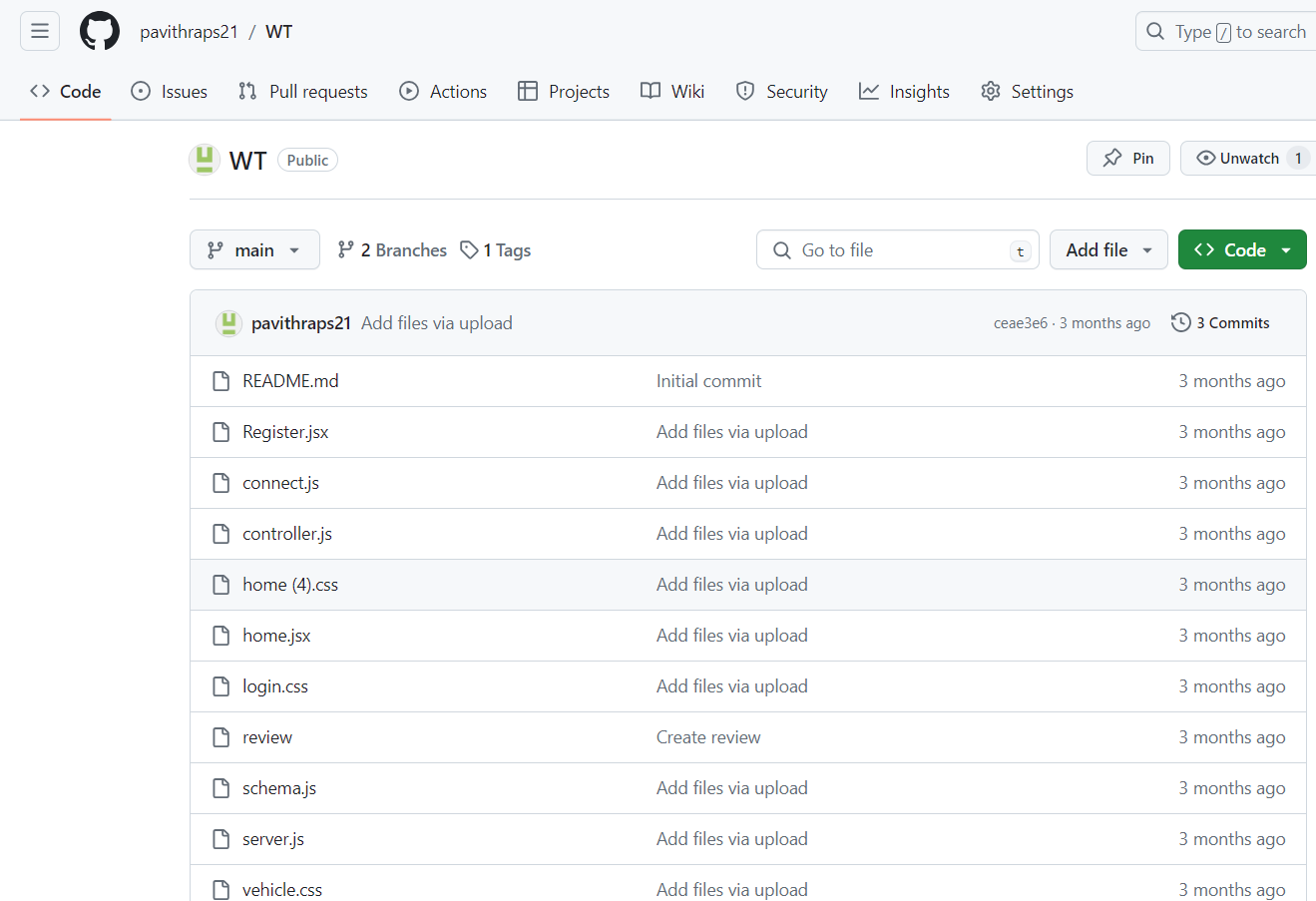
# chmod +x <hook\_name>

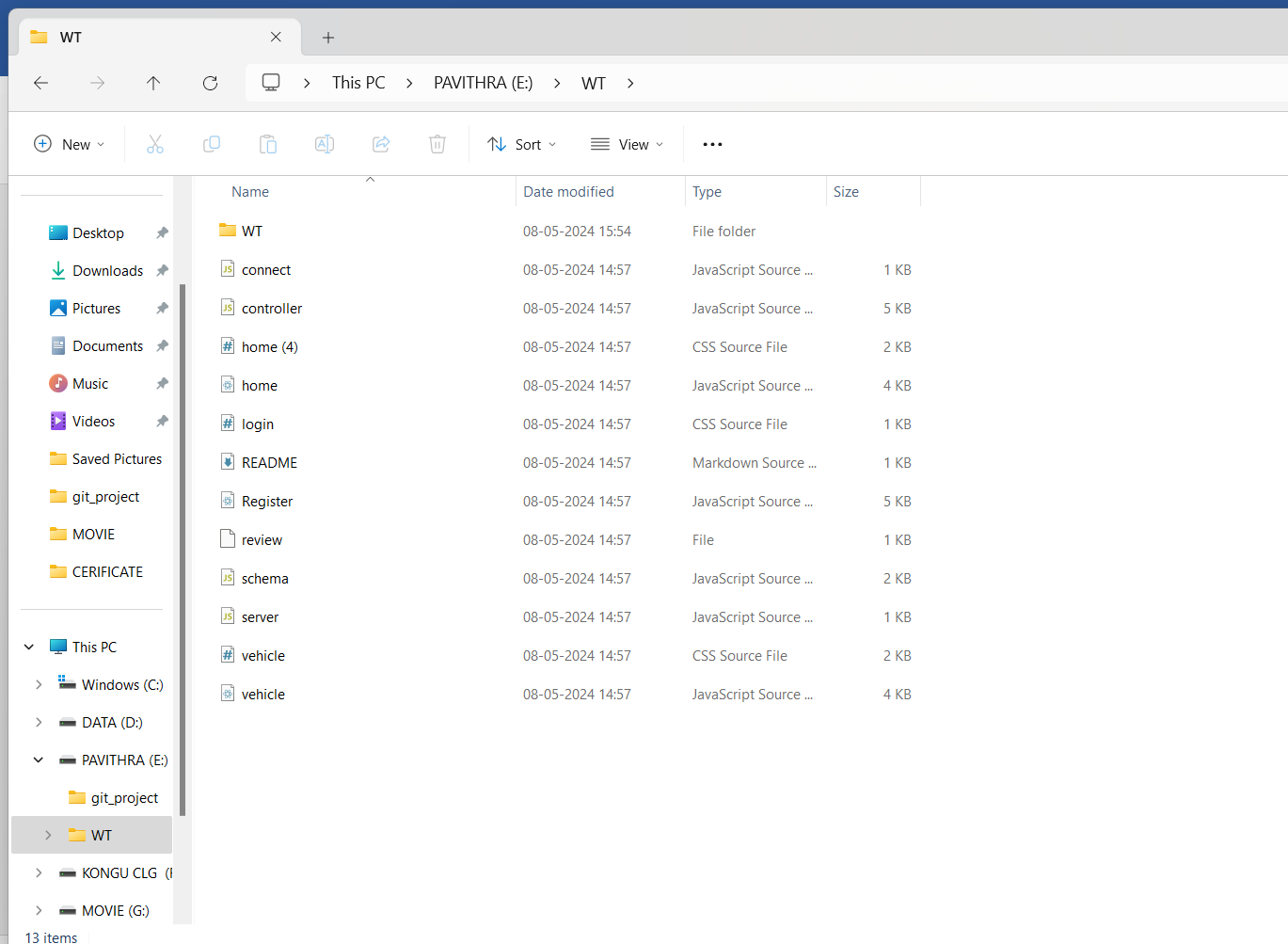




1. **Graph gitk --all**





****