**VCS Task**

**Task Description:**

Establish a new directory, populate it with script files, initiate an empty repository on GitHub, convert the local directory into a Git repository, and link it to GitHub for pushing the code into the repository.Perform merge, rebase, stash commands in following github repo.

**Techstacks needs to be used :**

* Shell (AWS, WSL, Vbox)
* Git (Gitbash)

**How do I submit my work?**

* Push all your work files to GitHub (Code & O/P screenshot images must).
* Submit your URLs in the portal.

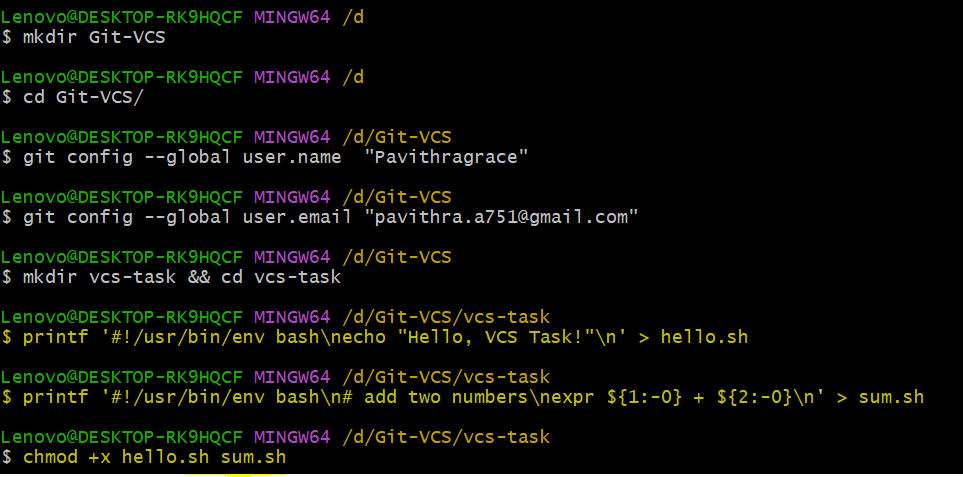
**Terms and Conditions?**

* You agree to not share this confidential document with anyone.
* You agree to open-source your code (it may even look good on your profile!). Do not mention our company’s name anywhere in the code.
* We will never use your source code under any circumstances for any commercial purposes; this is just a basic assessment task.

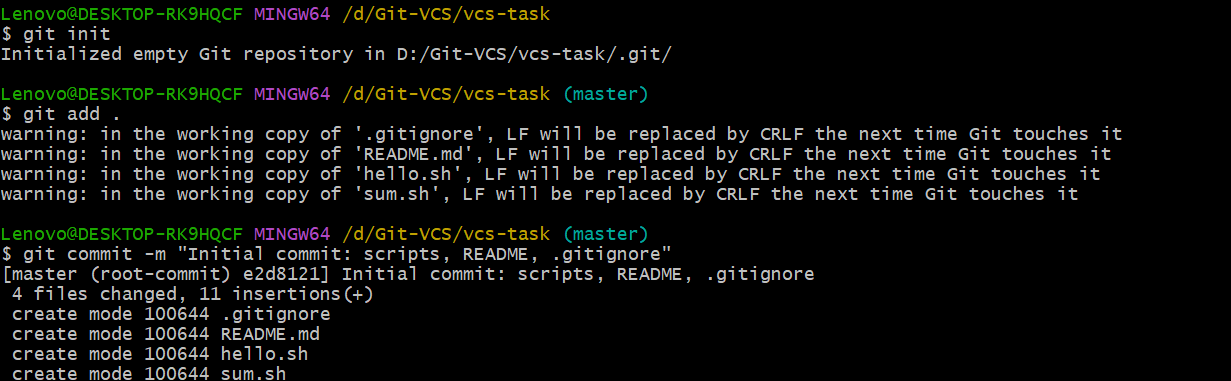
**NOTE:** Any violation of Terms and conditions is strictly prohibited. You are bound to adhere to it.

**Solution:**

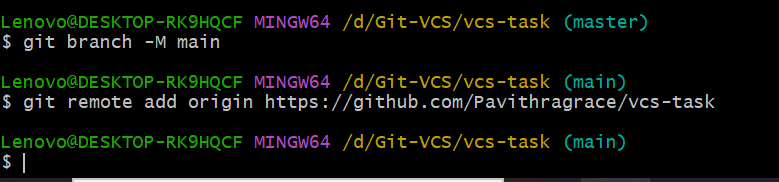
**Created a new project directory and added shell scripts (hello.sh, sum.sh).**

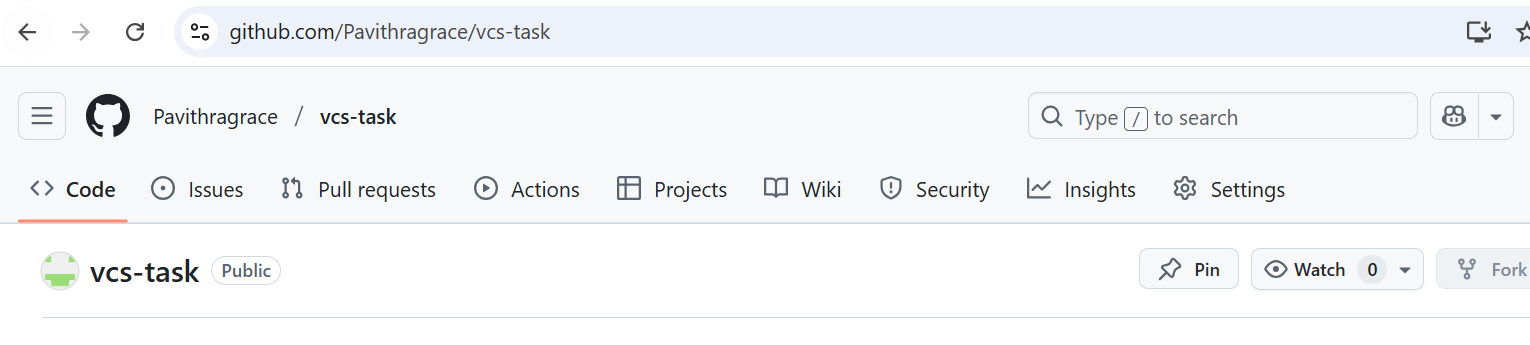


**Initialized a local Git repository and made the first commit.**



**Created an** empty **GitHub repository and linked my local repo as origin**



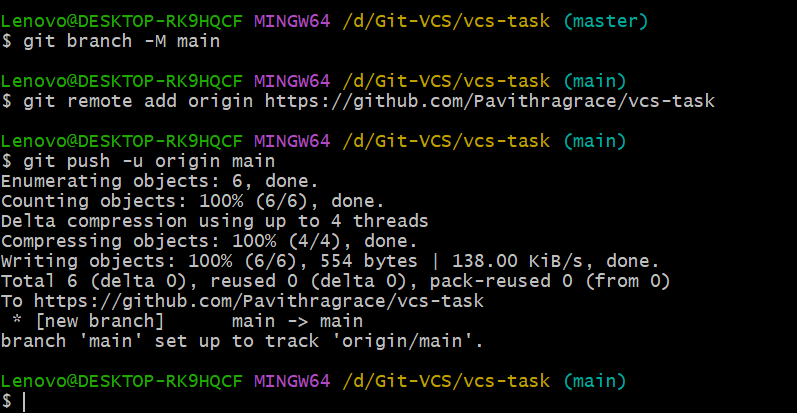


**Connect local to GitHub & push**

**git branch -M main**

**git remote add origin https://github.com/Pavithragrace/vcs-task**

**git push -u origin main**

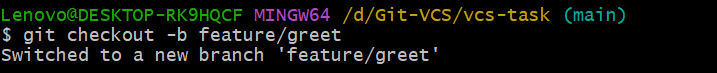


**Did a merge demo**

**Created a feature branch, change a file, merge back**

**# make a branch**

**git checkout -b feature/greet**

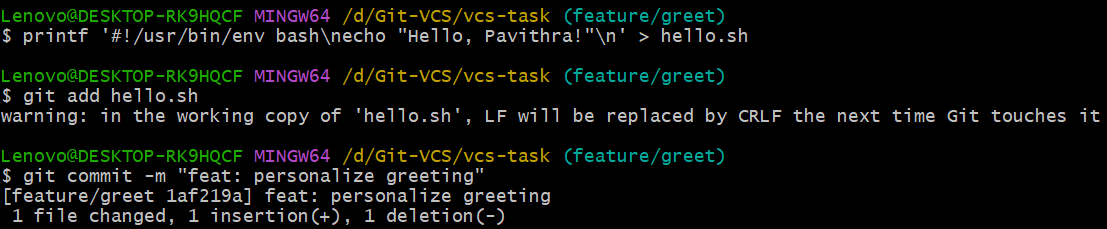


**# edit hello.sh to print your name**

**printf '#!/usr/bin/env bash\necho "Hello, Pavithra!"\n' > hello.sh**

**git add hello.sh**

**git commit -m "feat: personalize greeting"**

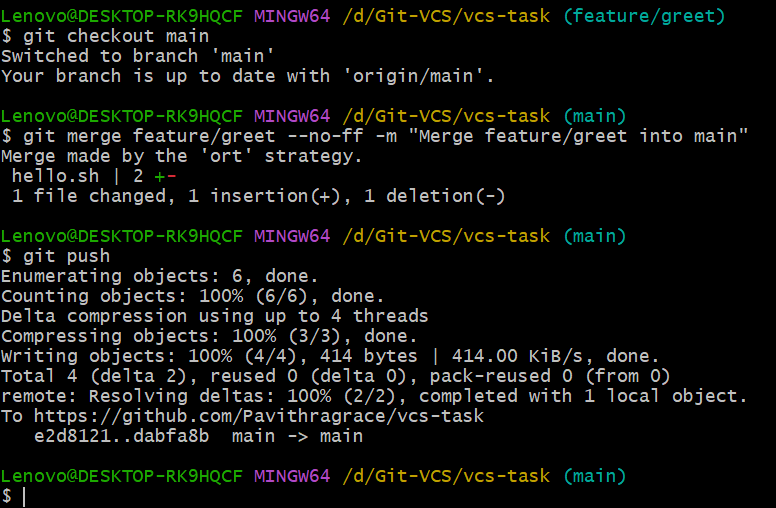


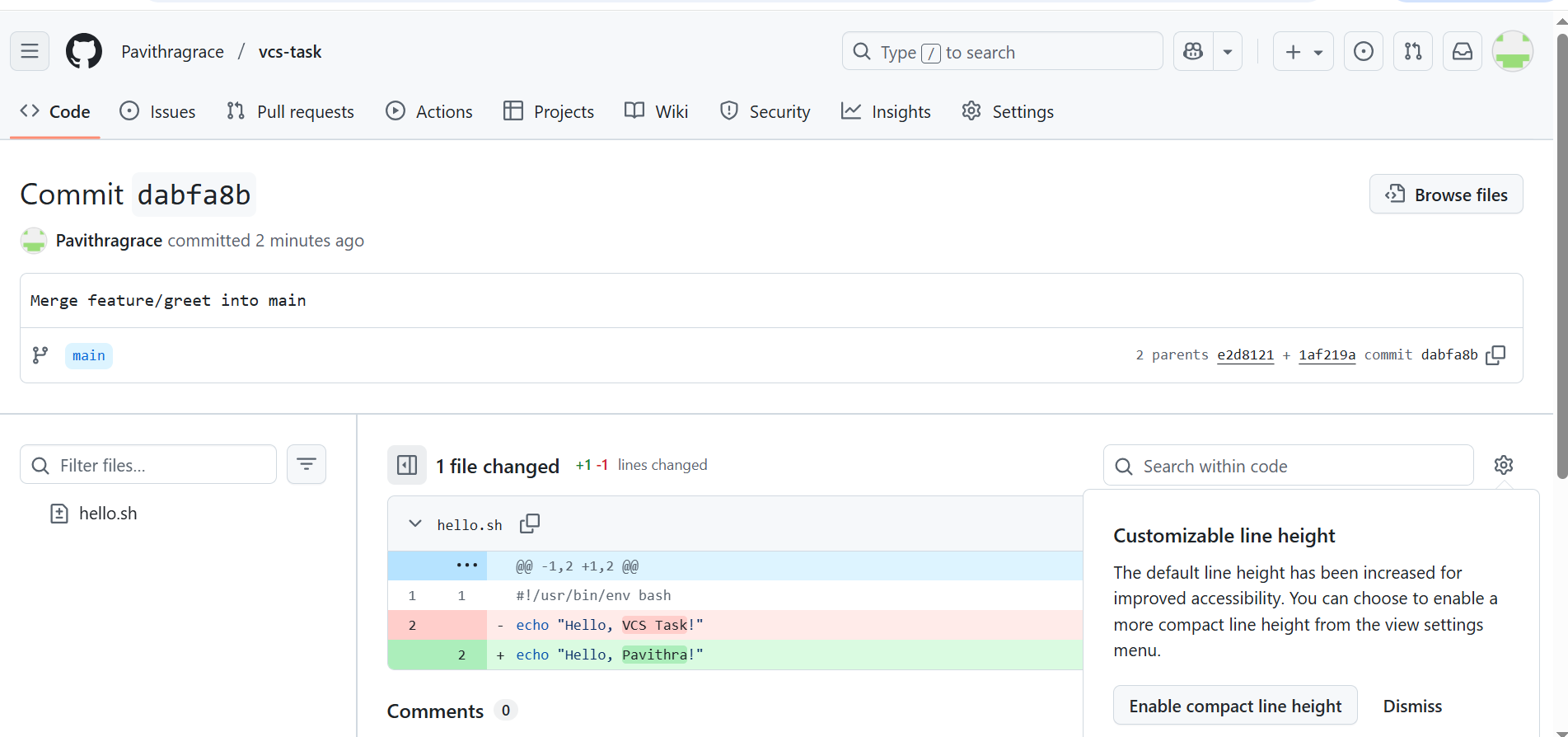
**# go back to main and merge**

**git checkout main**

**git merge feature/greet --no-ff -m "Merge feature/greet into main"**

**git push**





**Did a rebase demo**

**Create another branch and rebase it onto the latest main:**

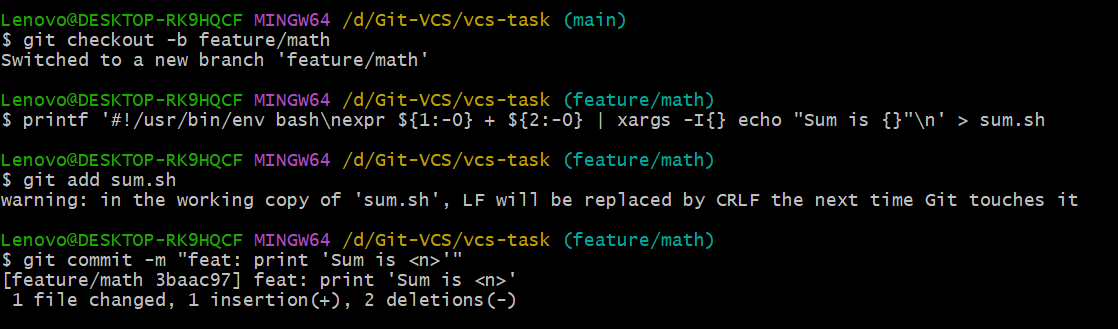
**git checkout -b feature/math**

**# change sum.sh to show a message**

**printf '#!/usr/bin/env bash\nexpr ${1:-0} + ${2:-0} | xargs -I{} echo "Sum is {}"\n' > sum.sh**

**git add sum.sh**

**git commit -m "feat: print 'Sum is <n>'"**



**# simulate new work on main so rebase is meaningful**

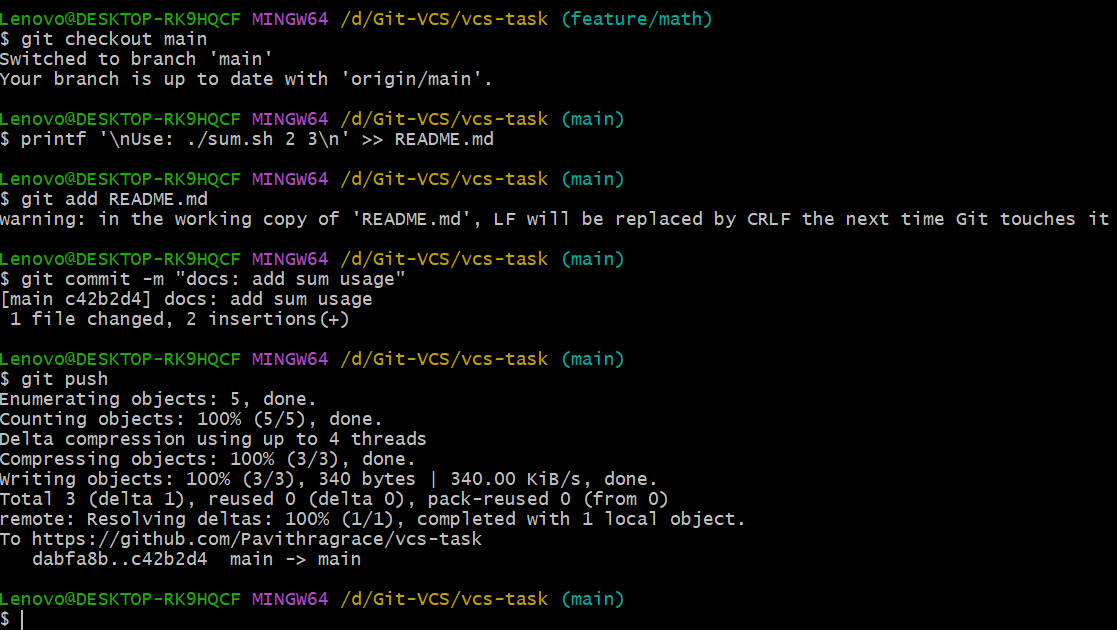
**git checkout main**

**printf '\nUse: ./sum.sh 2 3\n' >> README.md**

**git add README.md**

**git commit -m "docs: add sum usage"**

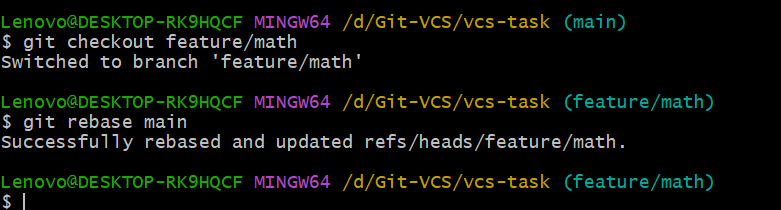
**git push**



**# rebase feature onto updated main**

**git checkout feature/math**

**git rebase main**

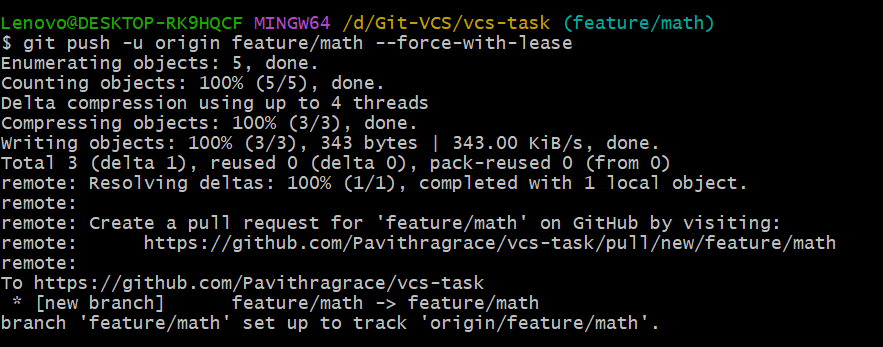


**# if rebase stops for conflicts, fix files, then:**

**# git add <fixed-files>; git rebase –continue**

**# push the rebased branch (force-with-lease is the safe way)**

**git push -u origin feature/math --force-with-lease**

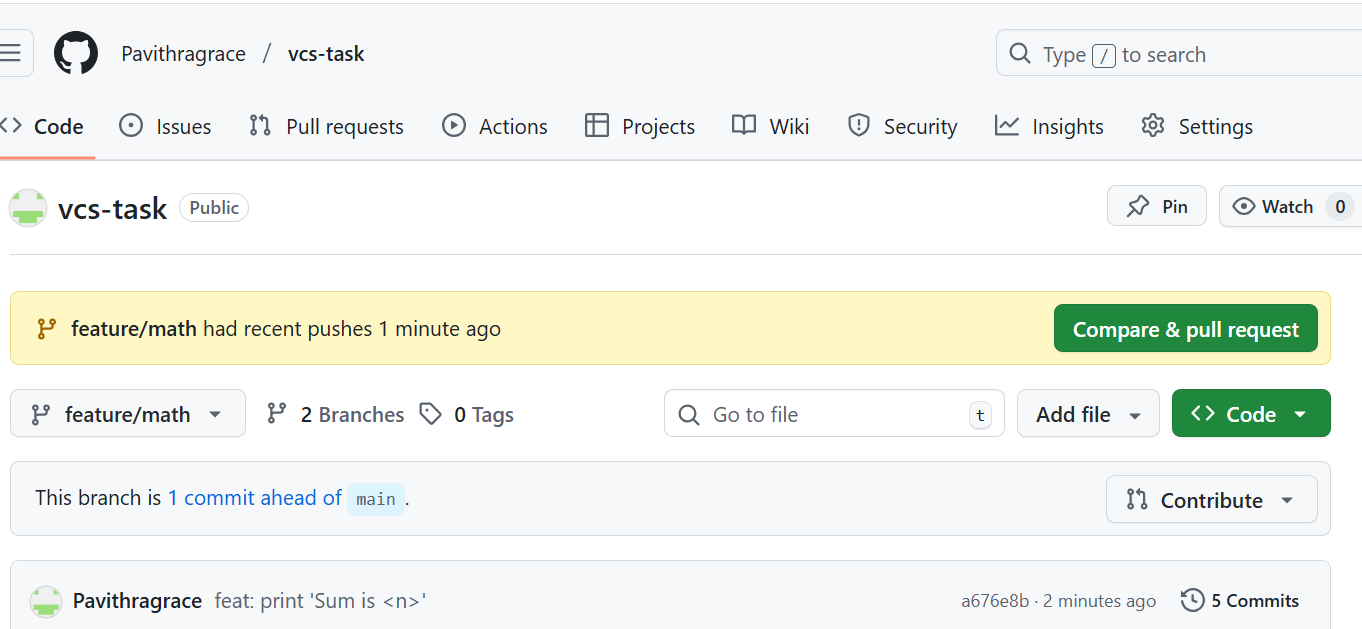


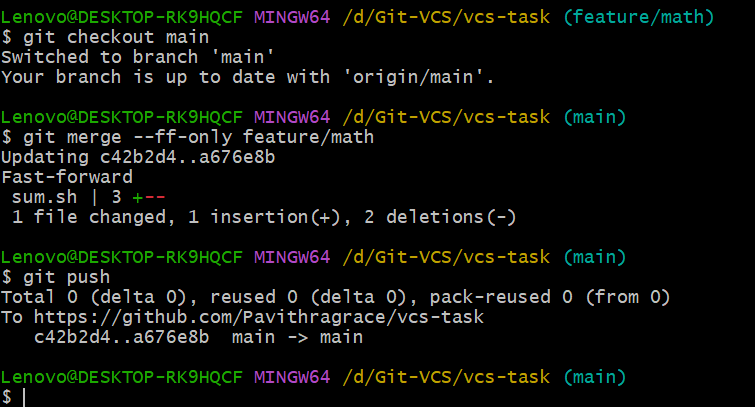
**# merged it cleanly now**

**git checkout main**

**git merge --ff-only feature/math**

**git push**



****

**Did a** stash **demo**

Shown save dirty work, switch away, then restore:

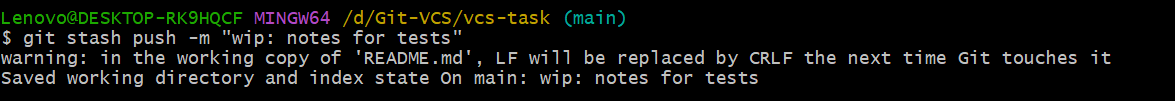
**# make a quick uncommitted change**

**echo "# TODO: add tests" >> README.md**



**# save it without committing**

**git stash push -m "wip: notes for tests"**



**# switch branches / do something else**

**git checkout -b hotfix/readme-typo**

**sed -i 's/Demo repo/demo repository/g' README.md || perl -i -pe 's/Demo repo/demo repository/g' README.md**

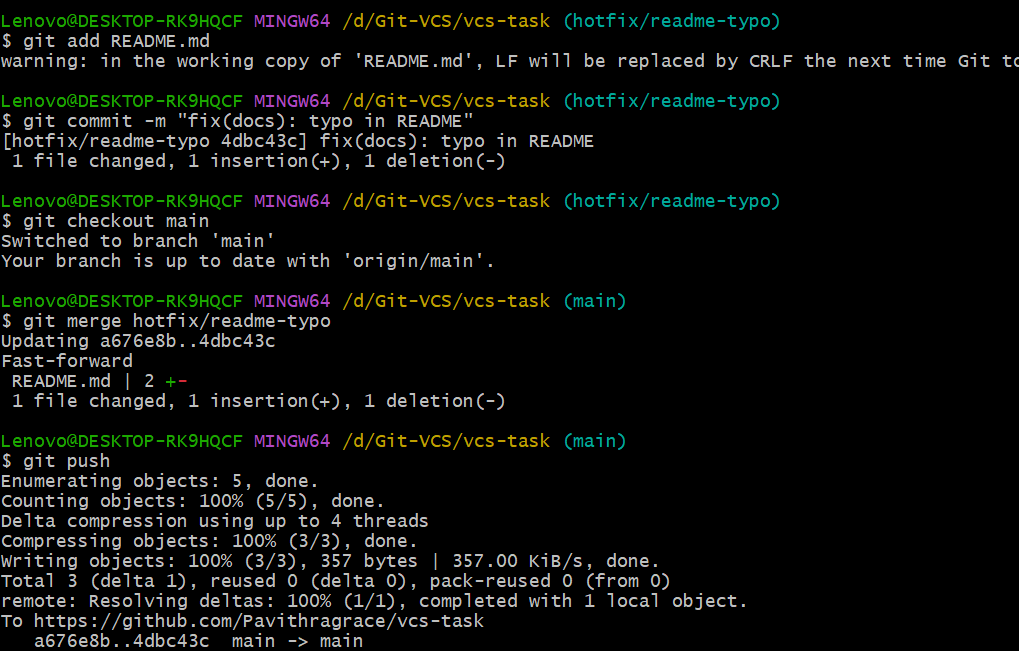
**git add README.md**

**git commit -m "fix(docs): typo in README"**

**git checkout main**

**git merge hotfix/readme-typo**

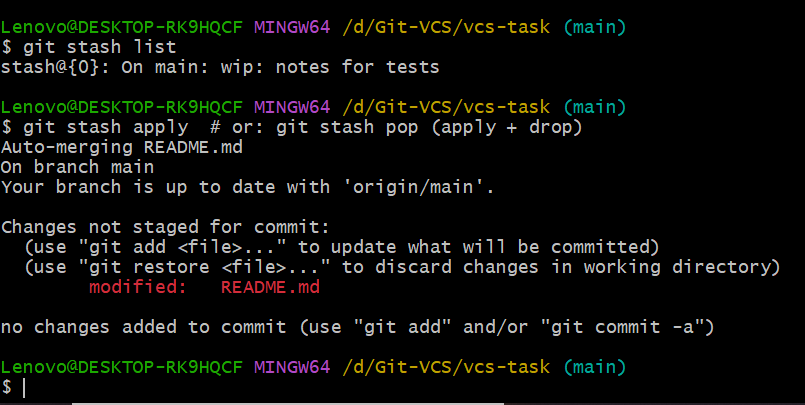
**git push**



**# come back and restore your stashed work**

**git stash list**

**git stash apply**



**Verify history (nice graph)**

