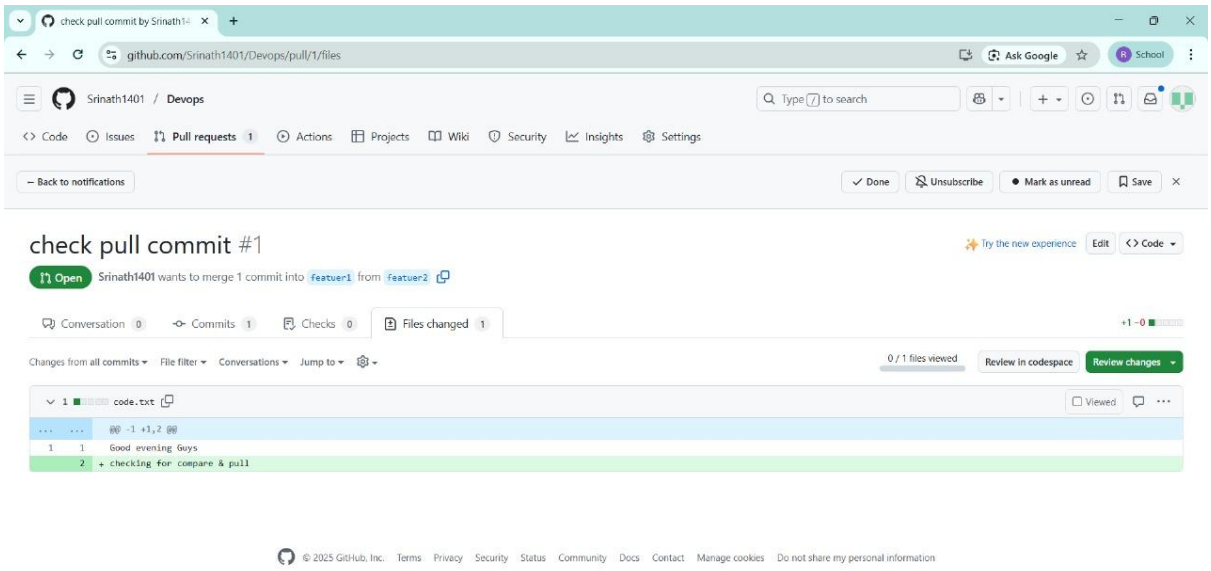
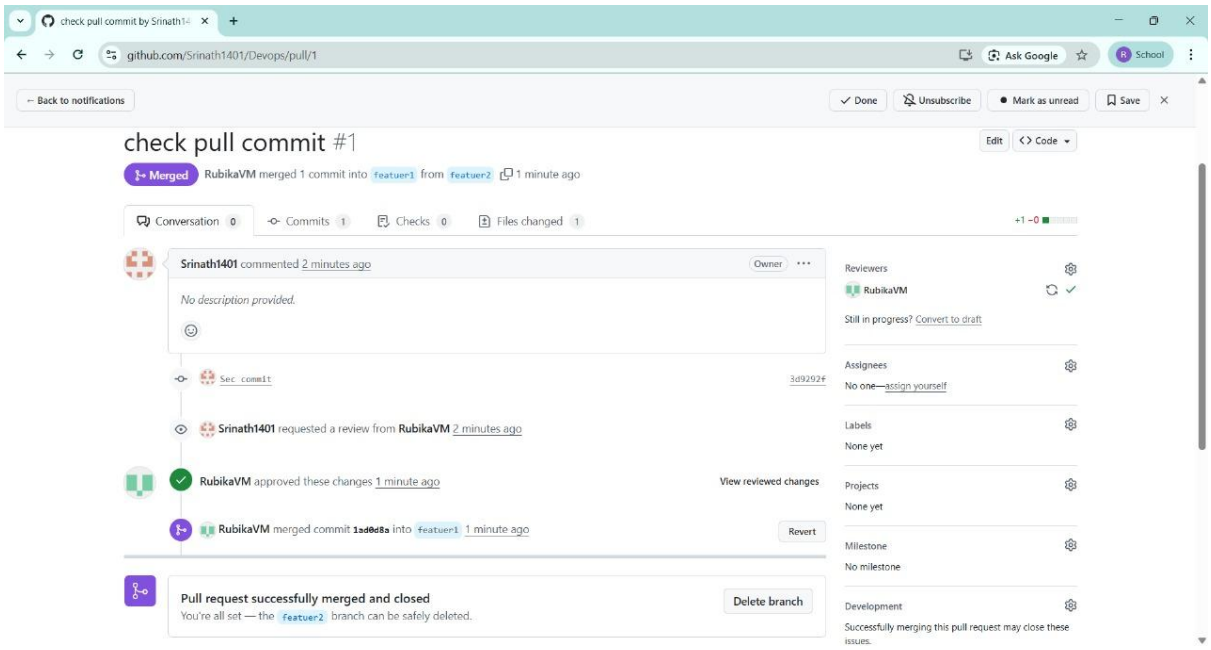
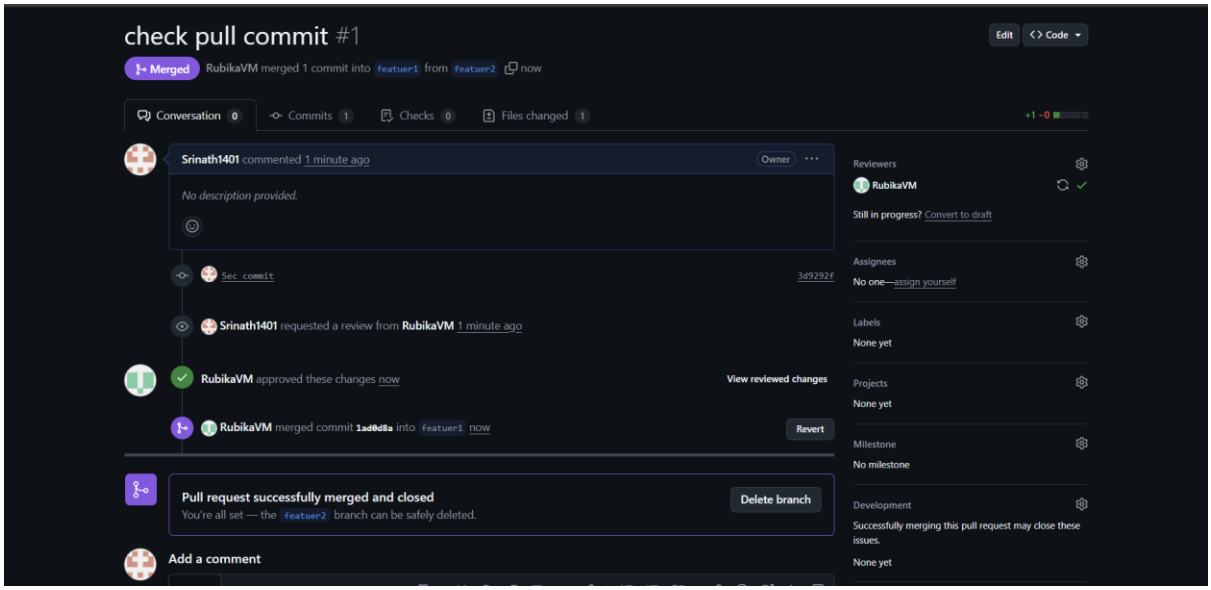
Task – 4

Git Pull Request:

A Git Pull Request is a feature used in platforms like GitHub to ask others to review and merge your code changes into another branch, usually the main branch. After creating a separate branch and pushing your work, you open a pull request to show what changes you made, why you made them, and to get approval from your team.




check pull commit by Srinath14
github.com/Srinath1401/Devops/pull/1
Ask Google
School
← Back to notifications
Done
Unsubscribe
Mark as unread
Save
check pull commit #1
Edit
Code
Merged
RubikaVM merged 1 commit into featuer1 from featuer2 1 minute ago
Conversation 0
Commits 1
Checks 0
Files changed 1
+1 −0
Srinath1401 commented 2 minutes ago
Owner
No description provided.
Sec commit
3d9292f
Srinath1401 requested a review from RubikaVM 2 minutes ago
RubikaVM approved these changes 1 minute ago
View reviewed changes
RubikaVM merged commit 1ad0d8a into featuer1 1 minute ago
Revert
Pull request successfully merged and closed
You're all set — the featuer2 branch can be safely deleted.
Delete branch
Reviewers
RubikaVM
Still in progress? Convert to draft
Assignees
No one—assign yourself
Labels
None yet
Projects
None yet
Milestone
No milestone
Development
Successfully merging this pull request may close these issues.


check pull commit #1
Edit
Code
Merged
RubikaVM merged 1 commit into featuer1 from featuer2 now
Conversation 0
Commits 1
Checks 0
Files changed 1
+1 −0
Srinath1401 commented 1 minute ago
Owner
No description provided.
Sec commit
3d9292f
Srinath1401 requested a review from RubikaVM 1 minute ago
RubikaVM approved these changes now
View reviewed changes
RubikaVM merged commit 1ad0d8a into featuer1 now
Revert
Pull request successfully merged and closed
You're all set — the featuer2 branch can be safely deleted.
Delete branch
Add a comment
Reviewers
RubikaVM
Still in progress? Convert to draft
Assignees
No one—assign yourself
Labels
None yet
Projects
None yet
Milestone
No milestone
Development
Successfully merging this pull request may close these issues.
None yet

Git reset –soft:

Resets only the commit history; it keeps all your changes in the staging area, so your files remain ready to commit again.

```
SRINATH@Leo MINGW64 ~/devops/Devops (featuer2)
$ git log --oneline
3d9292f (HEAD -> featuer2, origin/featuer2) Sec commit
a19f50b (origin/featuer1, featuer1) First commit
9de2dd7 (origin/main, origin/HEAD, main) Initial commit

SRINATH@Leo MINGW64 ~/devops/Devops (featuer2)
$ git reset --soft a19f50b

SRINATH@Leo MINGW64 ~/devops/Devops (featuer2)
$ git log --oneline
a19f50b (HEAD -> featuer2, origin/featuer1, featuer1) First commit
9de2dd7 (origin/main, origin/HEAD, main) Initial commit
```

Git reset --mixed:

(the default reset) also moves the commit history backward but removes files from the staging area while keeping your actual changes safely in the working directory.

```
SRINATH@Leo MINGW64 ~/srinath01/Devops (main)
$ git add .
warning: in the working copy of 'code1.txt', LF will be replaced by CRLF the nex
t time Git touches it

SRINATH@Leo MINGW64 ~/srinath01/Devops (main)
$ git reset --mixed

SRINATH@Leo MINGW64 ~/srinath01/Devops (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        code1.txt
```

Git reset --hard:

is the strongest reset—it not only moves the commit history back but also deletes all changes from both the staging area and the working directory, making your project look exactly like the chosen commit.