Jayanthi. M

Assignment

Assignment 12/08

Task 1 : Create a feature branch from main, make a few commits, and then rebase it onto an updated main branch.

Task 2 : Make a mistake in a commit and use git revert to undo the changes without deleting commit history.

Task3 : Pick a specific commit from another branch and apply it to your current branch.

Task 1 : Create a feature branch from main, make a few commits, and then rebase it onto an updated main branch.

From main branch we need to create one feature-branch and we need to checkout to feature-branch

**git checkout -b feature-branch**

then adding some files to feature-branch

**echo "Line 1 from feature branch" >> file1.txt**

**git add file1.txt**

**git commit -m "Add file1.txt with line 1"**

**echo "Line 2 from feature branch" >> file1.txt**

**git add file1.txt**

**git commit -m "Add line 2 to file1.txt"**

**A computer screen shot of a computer program

AI-generated content may be incorrect.**

**git checkout master**

**echo "Update from master branch" >> master.txt**

**git add master.txt**

**git commit -m "Update master.txt in master branch"**

**A computer screen shot of a computer program

AI-generated content may be incorrect.**

then we are checkout to feature-branch and rebase master

**git checkout feature-branch**

**A computer screen shot of a black screen

AI-generated content may be incorrect.**

**git rebase master**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**feature-branch is now successfully rebased on top of master.**

Task 2 : Make a mistake in a commit and use git revert to undo the changes without deleting commit history.

Making an mistake in a commit

**echo "Mistaken change" >> file1.txt**

**git add file1.txt**

**git commit -m "Add mistaken content"**

**git log**

**A screen shot of a computer

AI-generated content may be incorrect.**

and use git revert to undo the changes without deleting commit history.

**git revert 7262be2a284ff0702179aba5d4e513f85670e81a**

**A screenshot of a computer

AI-generated content may be incorrect.**

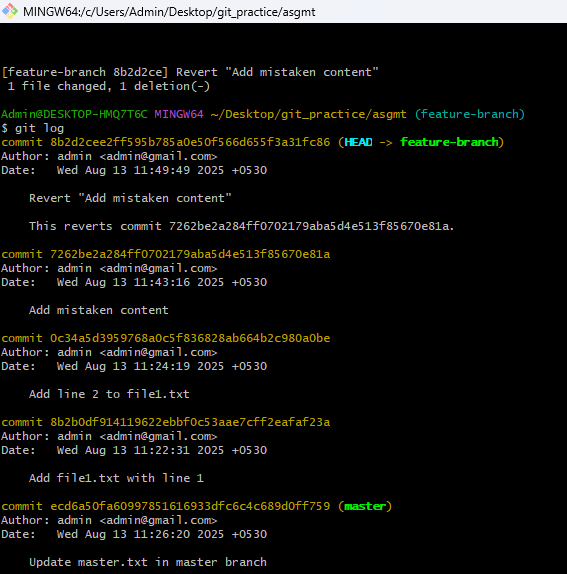
**Now I need to save and quit**

**Esc:wq!**

**git log**

**new commit id created above the 7262be2a284ff0702179aba5d4e513f85670e81a**

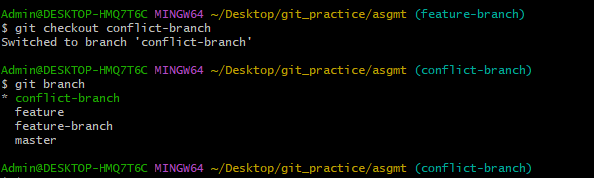
**new commit id is 8b2d2cee2ff595b785a0e50f566d655f3a31fc86**

****

Task3 : Pick a specific commit from another branch and apply it to your current branch.

To do this task I am checking out to conflict-branch using below command and I can use git cherry-pick <commitID>

**git checkout conflict-branch**



I am adding a new file to conflict-branch

echo "Special commit for cherry-pick" >> special.txt

git add special.txt

git commit -m "Special commit from conflict-branch"

A screenshot of a computer screen

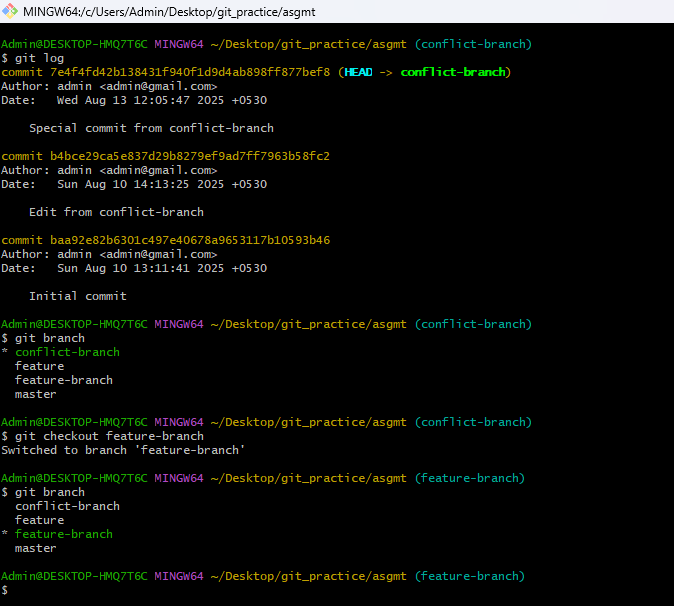
AI-generated content may be incorrect.

Again I need to switchback to feature-branch to make cherry-pick from conflict-branch

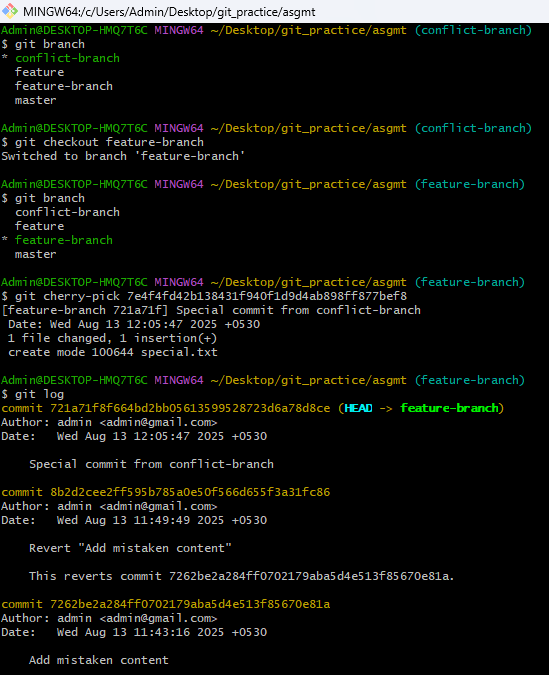
Before that I need to check and get the commitID to cherry-pick using

**git log**

**git checkout feature-branch**

****

**git cherry-pick 7e4f4fd42b138431f940f1d9d4ab898ff877bef8**

****

**the chosen commit from conflict-branch is applied on top of feature-branch.**