Jayanthi M

Assignment

Assignment 09/08\*

Task 1: Create a new branch, make some changes, commit them, and merge it back into

the main branch.

Task 2:- Task: Create a merge conflict intentionally by editing the same part of a file in two

branches and resolve it manually

Task 1: Create a new branch, make some changes, commit them, and merge it back into

the main branch.

We need to create a git repository and initializing git repository and creating a branch using below command

**mkdir central\_repo**

**git init**

**echo "Initial commit" > file.txt**

**git add file.txt**

**git commit -m "Initial commit"**

**A computer screen with text

AI-generated content may be incorrect.**

a.1) Create a new branch, make some changes, commit them, and merge it back into

the main branch.

To create a new branch we make use of

git branch <branch\_name>

ex: **git branch feature**

make changes to feature branch

**echo "This is a new feature" >> feature.txt**

**git add feature.txt**

to commit the changes to new branch

**git commit -m "Added feature.txt with new feature"**

****

merge it back into the main branch.

**git merge feature**

**A black screen with yellow text

AI-generated content may be incorrect.**

Task 2:- Task: Create a merge conflict intentionally by editing the same part of a file in two

branches and resolve it manually

For Task **2** (merge conflict)

We have **three branches**:

* master (main branch)
* feature
* conflict-branch

1. **we start editing a file in one branch** (say feature).
2. **we then edit the *same* part of the same file in the other branch** (conflict branch).
3. When you try to merge them — Git sees both branches touched the same lines → conflict.

Step 1 — Switch to master – because I am in feature branch now so

**git checkout master**

Step 2 — Create branch named conflict-branch

**git branch conflict-branch**

Step 3 - switch to conflict-branch from main-branch

**git checkout conflict-branch**

Step 4 - Make changes in conflict-branch

Edit a file (or create one) — make sure to pick a file that also exists in master and will be edited there too.

**echo "Edit from conflict-branch" > conflict.txt**

**git add conflict.txt**

**git commit -m "Edit from conflict-branch"**

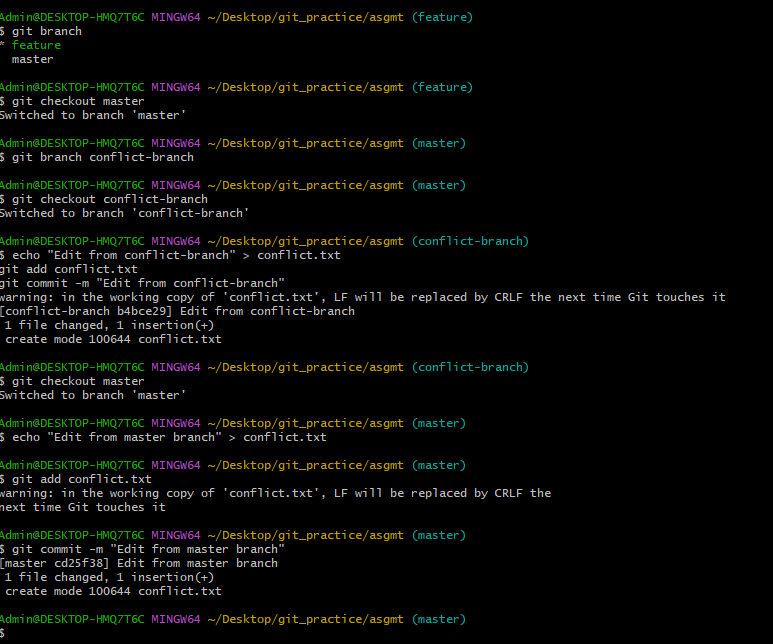
Step 5 — Switch to master and edit the same part

**git checkout master**

**echo "Edit from master branch" > conflict.txt**

**git add conflict.txt**

**git commit -m "Edit from master branch"**

****

Step 6— Merge and trigger conflict

**git merge conflict-branch**

**Now Git will show a merge conflict.**

**A black screen with white text

AI-generated content may be incorrect.**

After editing manually, the merge conflict

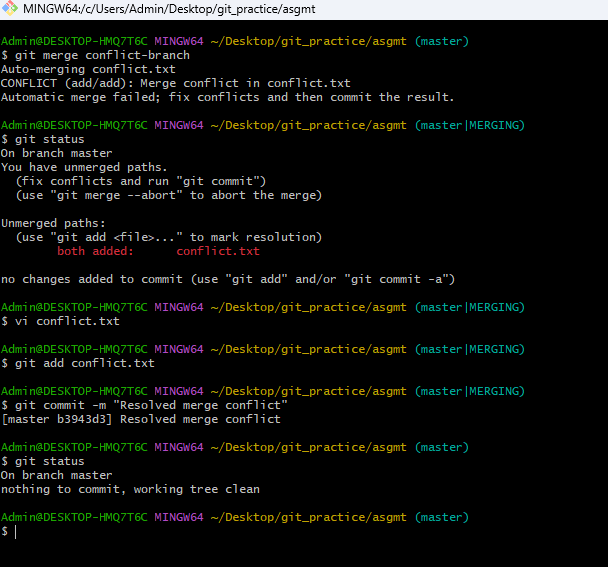
A black screen with white text

AI-generated content may be incorrect.

Adding files after solving merge conflict

**git add conflict.txt**

**git commit -m "Resolved merge conflict"**

****