**NAME: AMAN AGRAWAL**

**ROLL NO: G24AIT172**

**SUB: ML**-**OPs**

**TASK : ASSIGNMENT-1**

GitHub Repository link :- <https://github.com/Aman007Agg/MLOps-Jan2025>

#### Report Structure for GitHub Assignment: 1. Introduction:

"This assignment involves using Git and GitHub to manage source code repositories, handle merge conflicts, and practice basic Git commands for version control."

#### 2. Steps Followed:

* **Step 1**:Created a new public repository MLOps-Jan2025 on GitHub.
* **Step 2**: Cloned the repository to my local system using the command git clone git@github.com:Aman007Agg/MLOps-Jan2025.git.
* **Step 3**: Created a new file example.txt and committed the changes with the message "Added example.txt file".
* **Step 4**: Created two branches (G24AIT171 and G24AIT173), made changes to example.txt on each branch.
* **Step 5**: Merged branches and handled merge conflicts.
* **Step 6**: **Deleting the branch** (git branch -d <branch\_name>).
* **Step 7**: **Recovering the branch** using git reflog and git checkout -b.
* **Step 8**: **Tagging the commit** and pushing it using git push --tags.

#### 3. Screenshots:

* **Screenshot 1: git log --graph --oneline before the merge (showing your commit history).**
* **Screenshot 2: After resolving the merge conflict (showing the final commit history).**
* **Screenshot 3: Push confirmation after pushing the resolved changes to GitHub.**

#### 4. Challenges Faced:

* **"I initially encountered a merge conflict while merging the G24AIT171 branch into main, which required manual conflict resolution."**
* **"Another challenge was ensuring the correct branch was checked out before committing changes."**

#### 5. Solutions:

* **"I resolved the merge conflict by manually editing example.txt and committing the resolved file."**
* **"I double-checked my branch before committing and pushed all changes to GitHub successfully."**

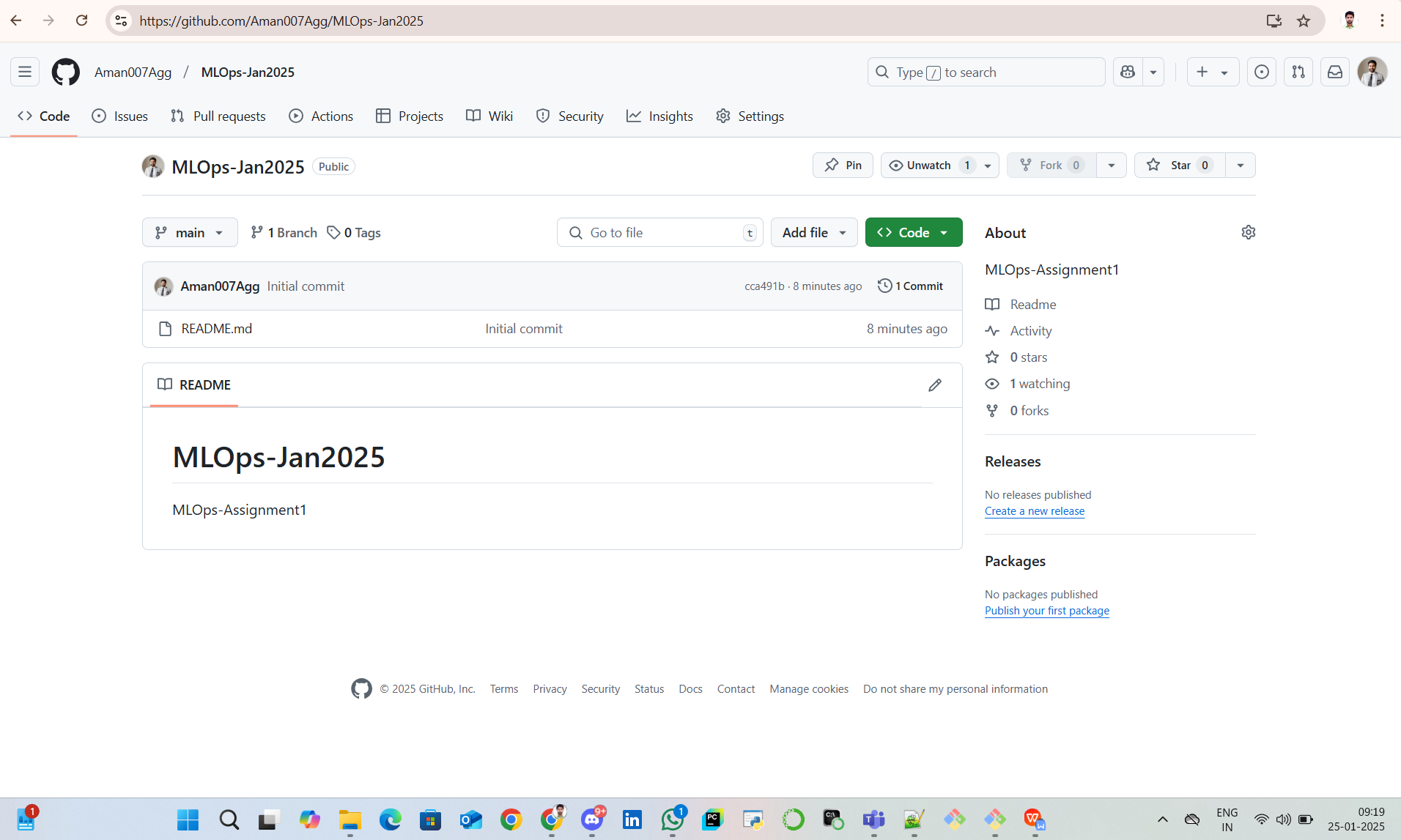
#### 6. Conclusion:

* **"Through this assignment, I gained hands-on experience with using Git for version control, including branching, merging, conflict resolution, and pushing changes to GitHub."**

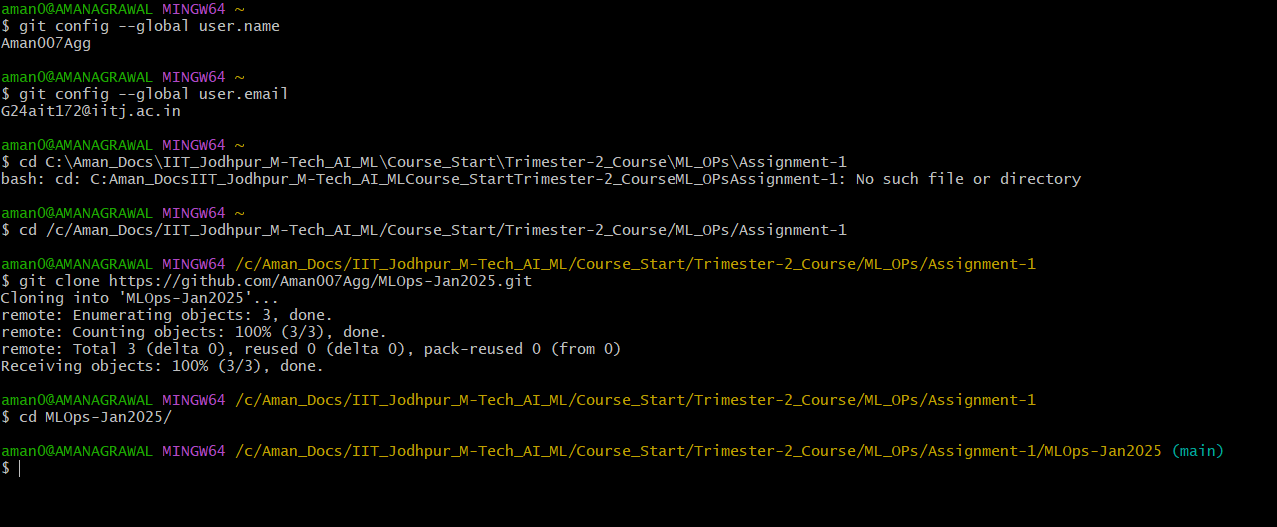
#### 7. GitHub Repository Link:

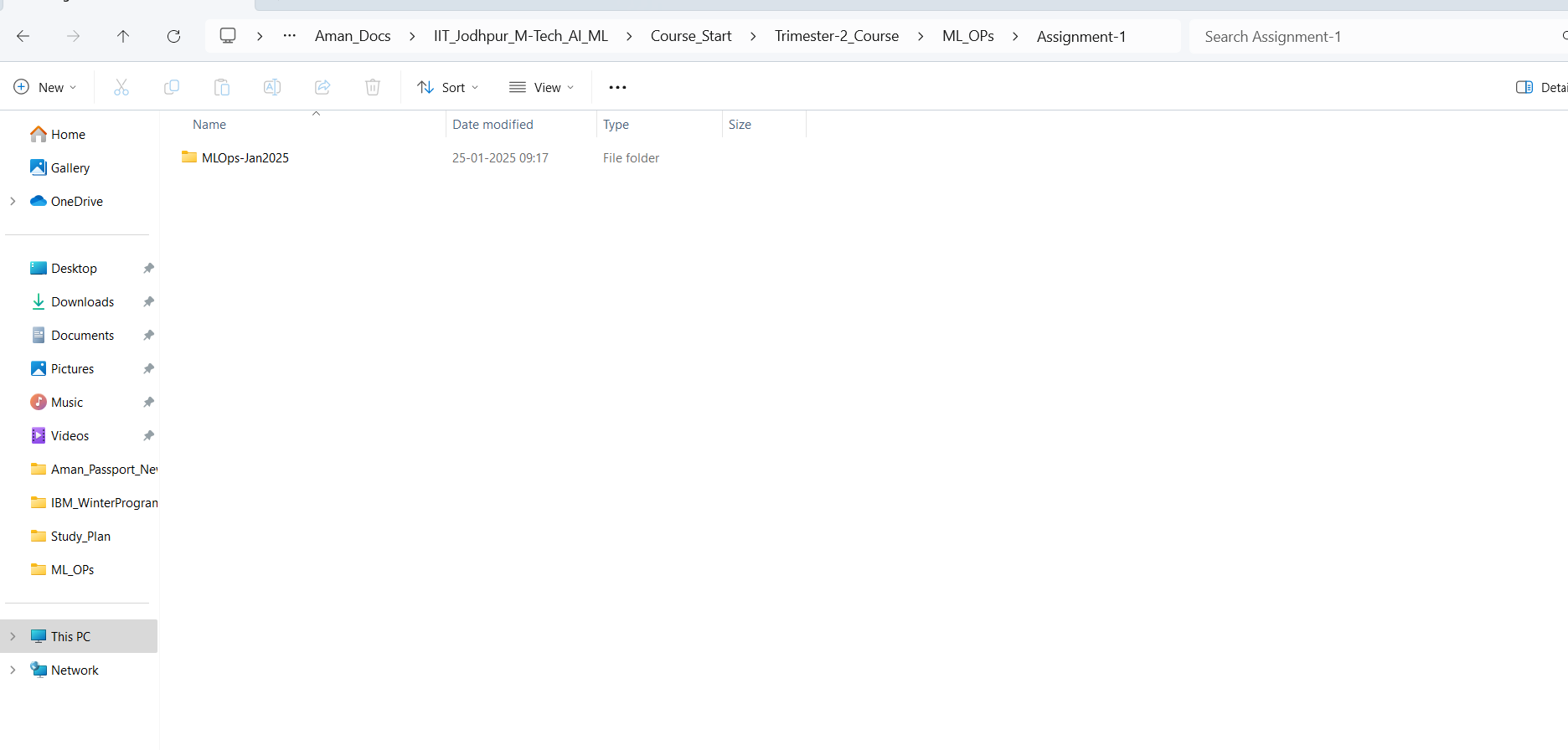
* **GitHub repository link:**  <https://github.com/Aman007Agg/MLOps-Jan2025>

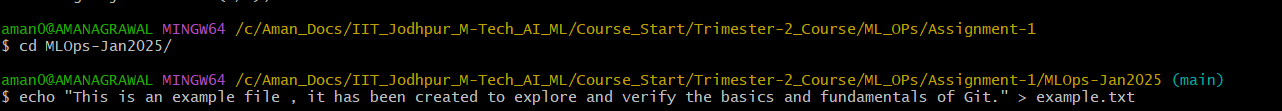
1. You need to create a new public repository on GitHub, named MLOps-Jan2025.

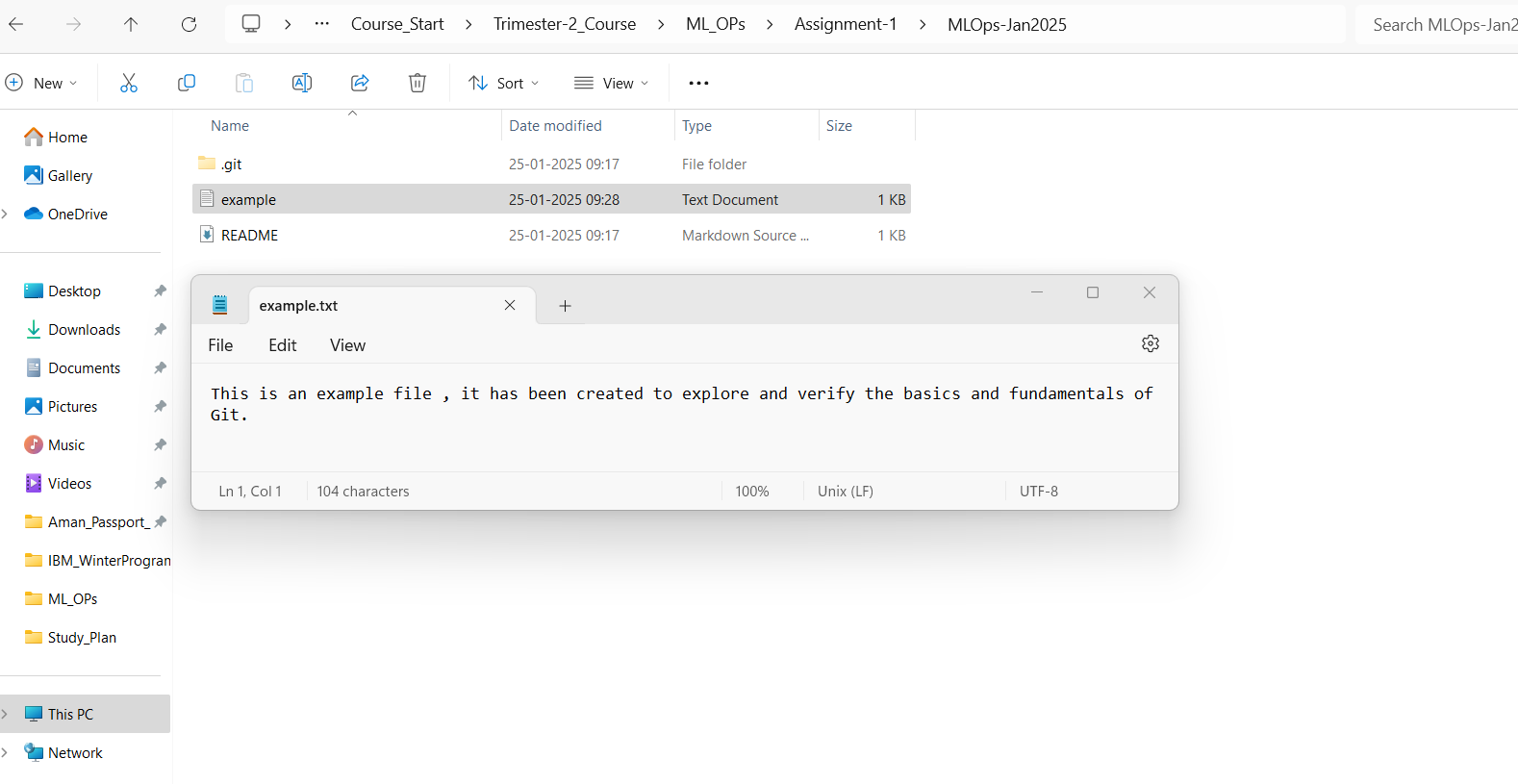


1. Clone this repository to your local system.

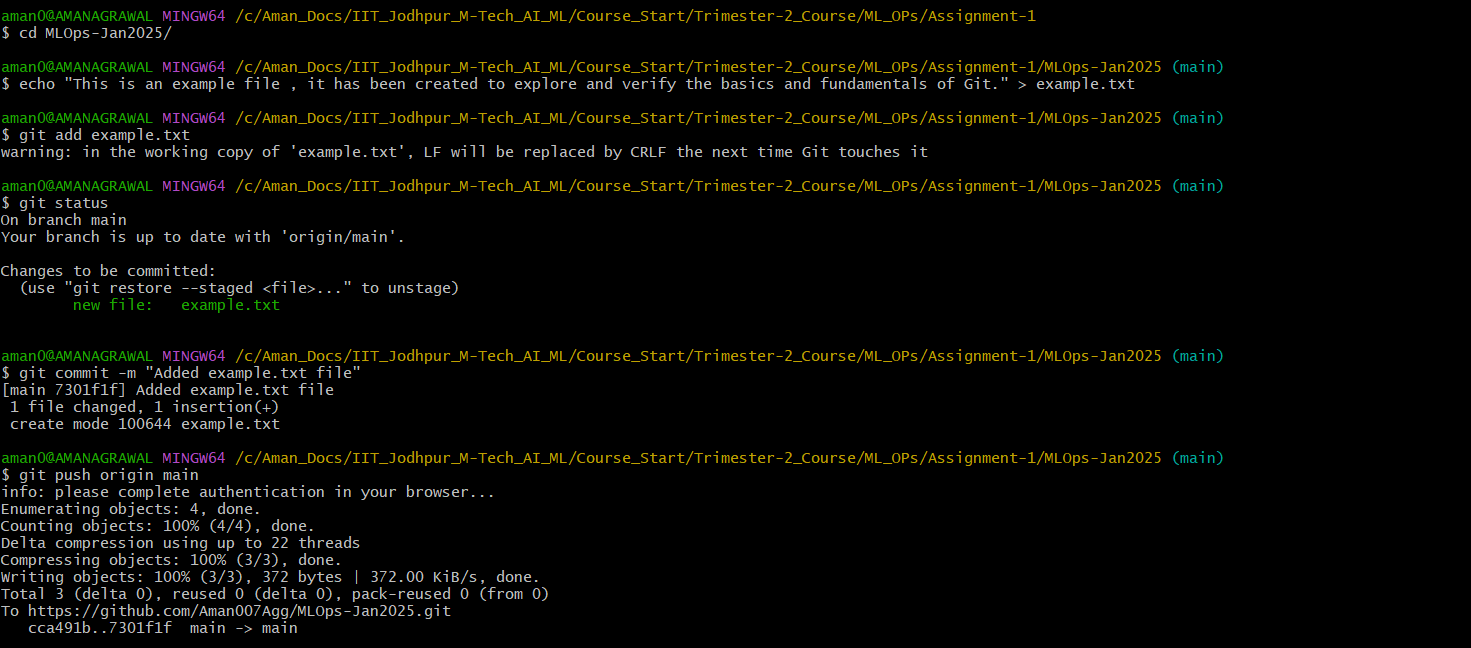


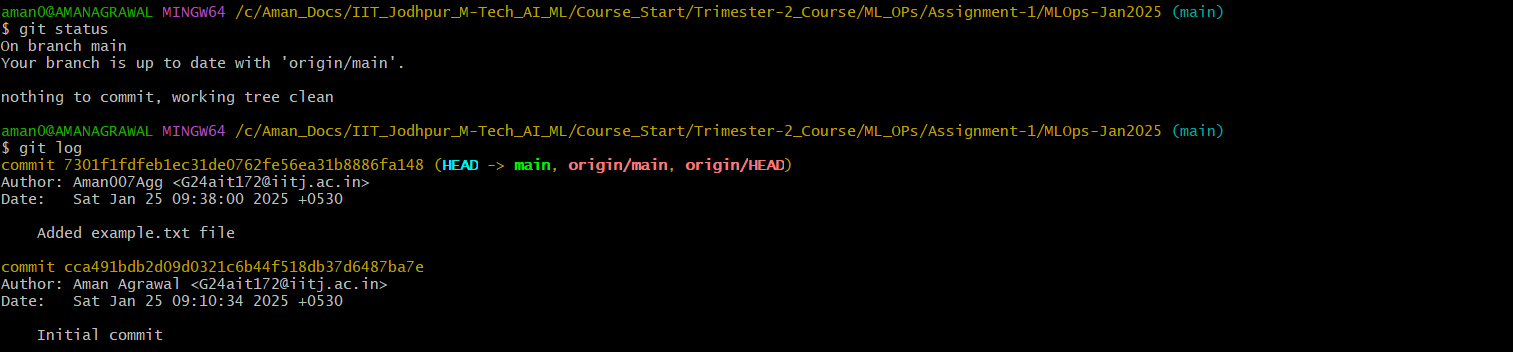


1. Add a file named example.txt to the repository, and commit the changes with a proper commit message.
2. Create a new file named example.txt:  
     
   



1. Stage the file for commit,and Commit the file with a proper message, and Push the changes to GitHub.

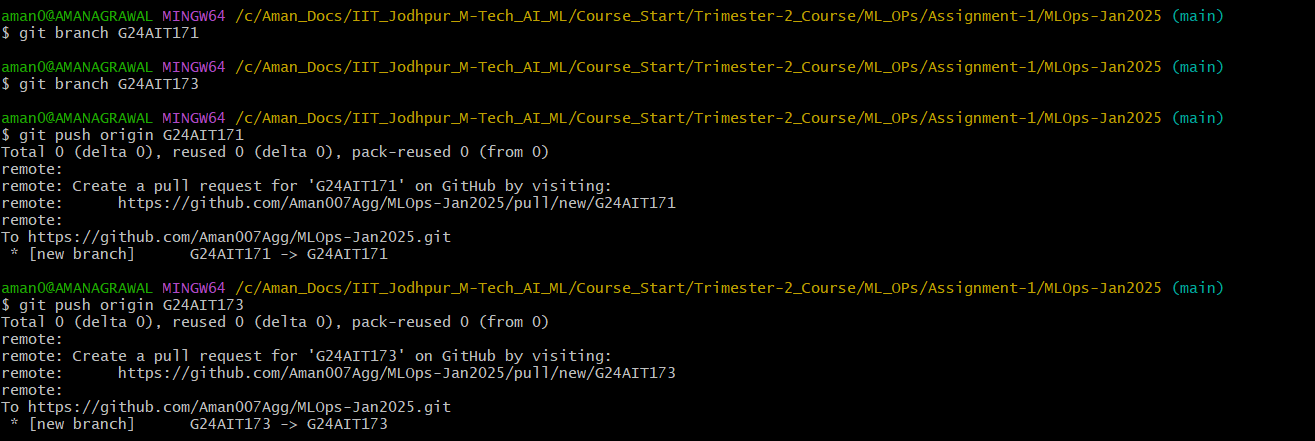


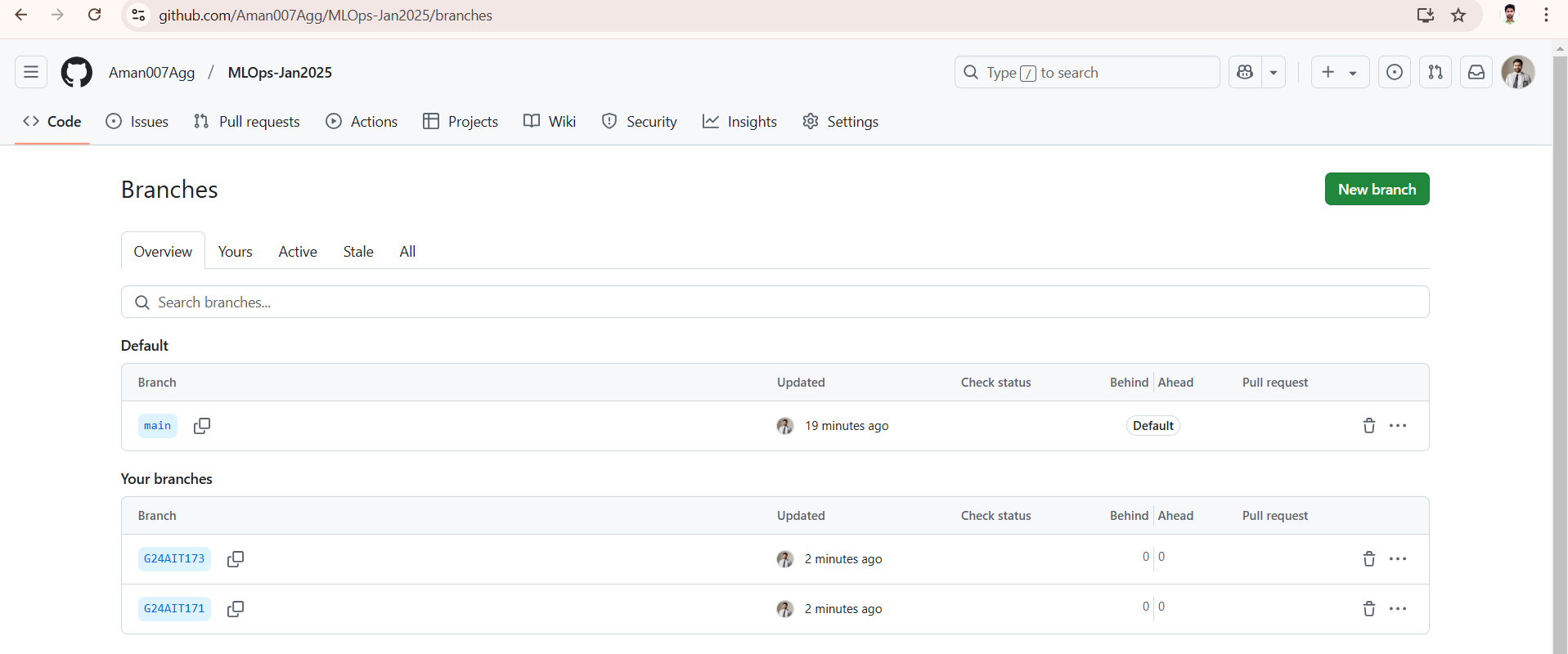


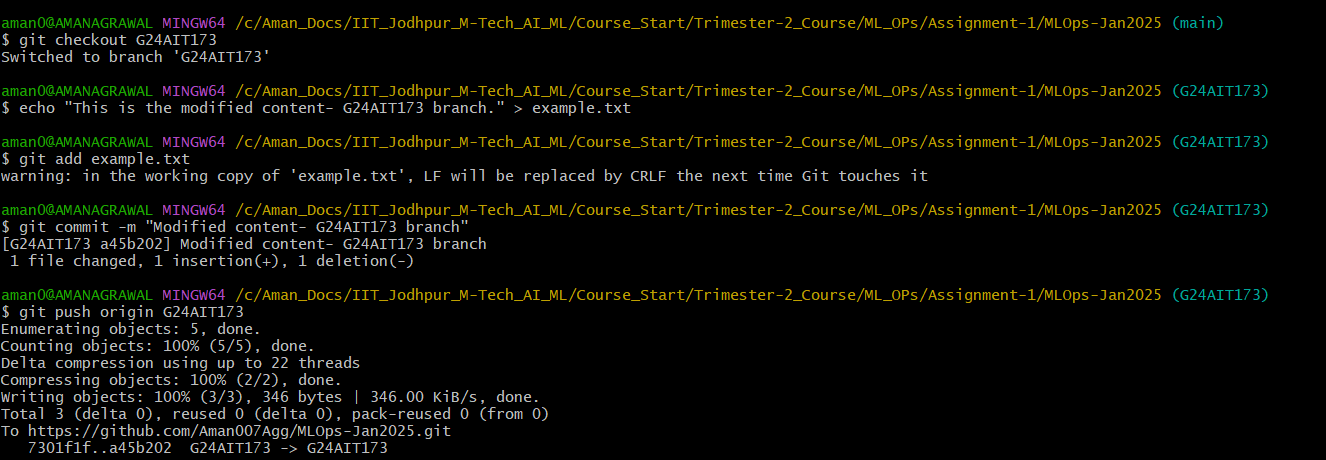
1. Create two new branches. If your roll number is M\_\_\_X, name one branch M\_\_\_X-1 and the other M\_\_\_X+1 (e.g., for roll number M\_\_\_001, you would create branches M\_\_\_000 and M\_\_\_002).

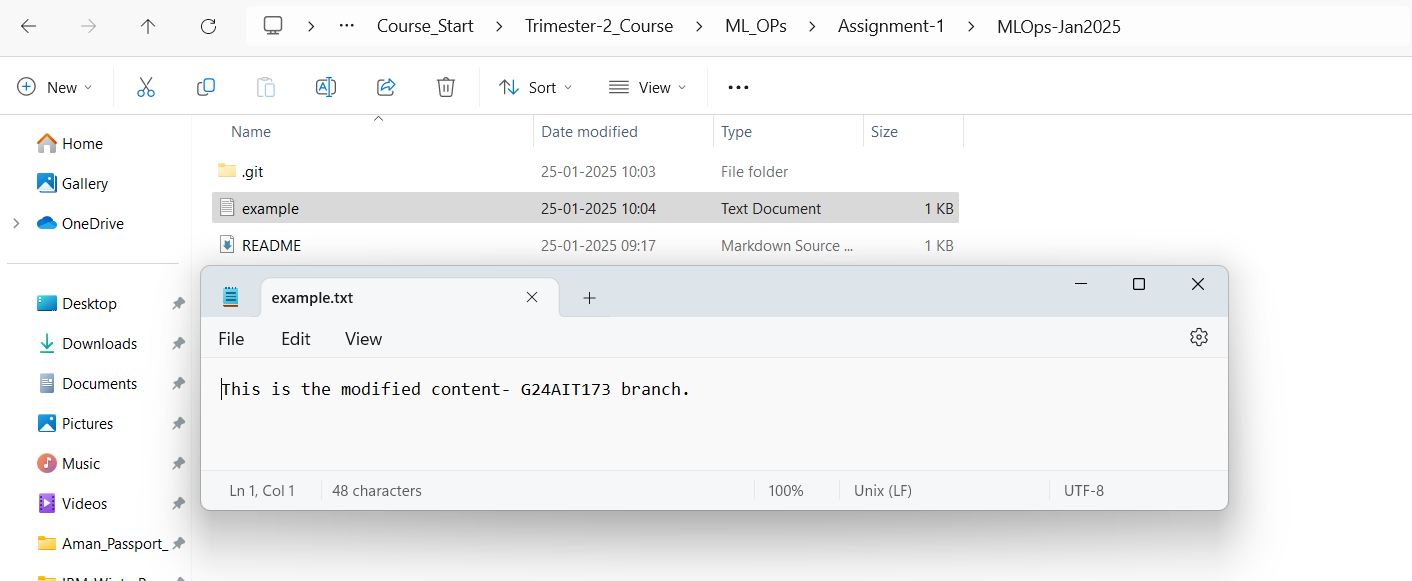
**Roll Number - G24AIT172**  
**Two branches have been created \* G24AIT171**

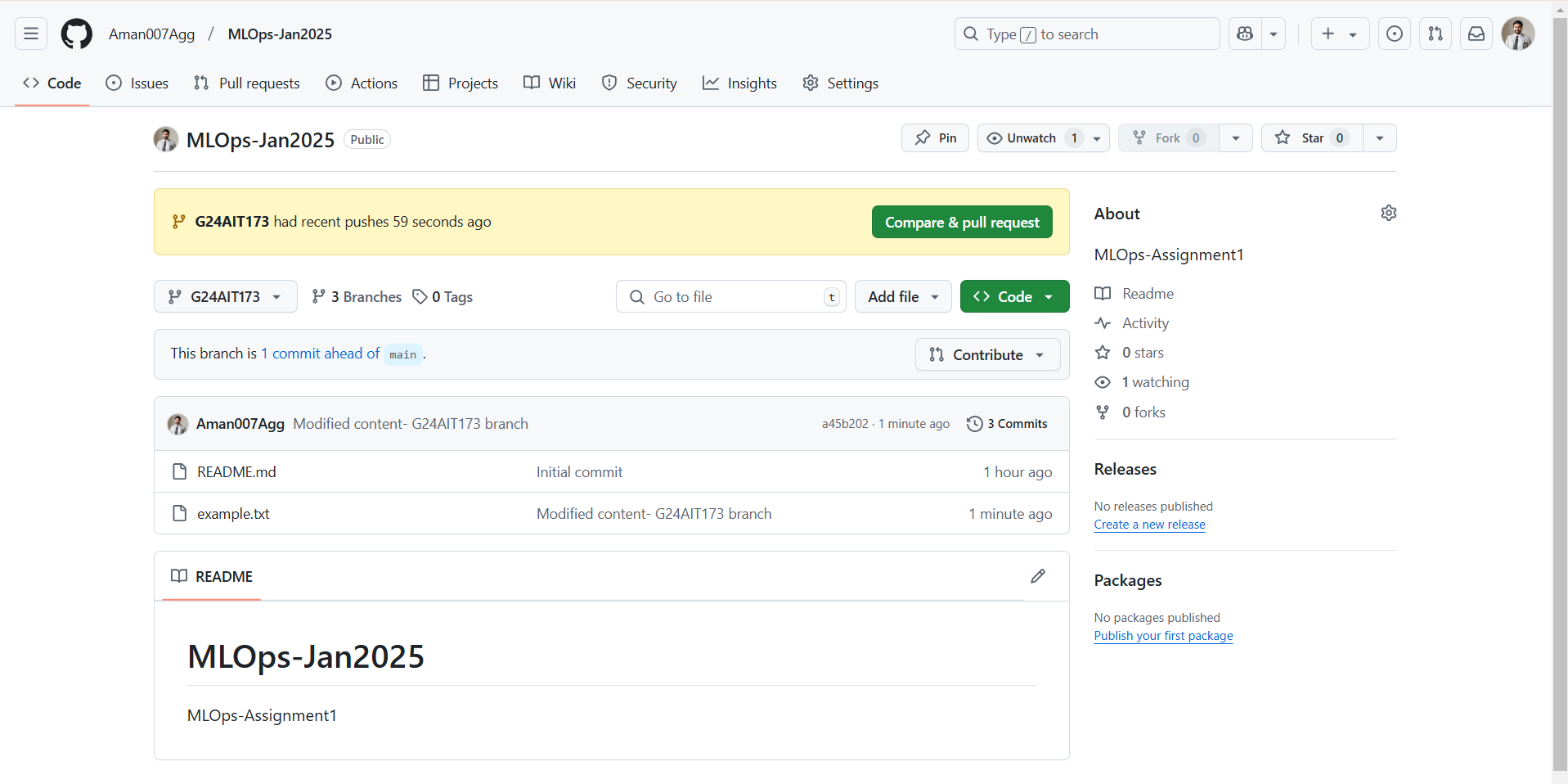
**\* G24AIT173**



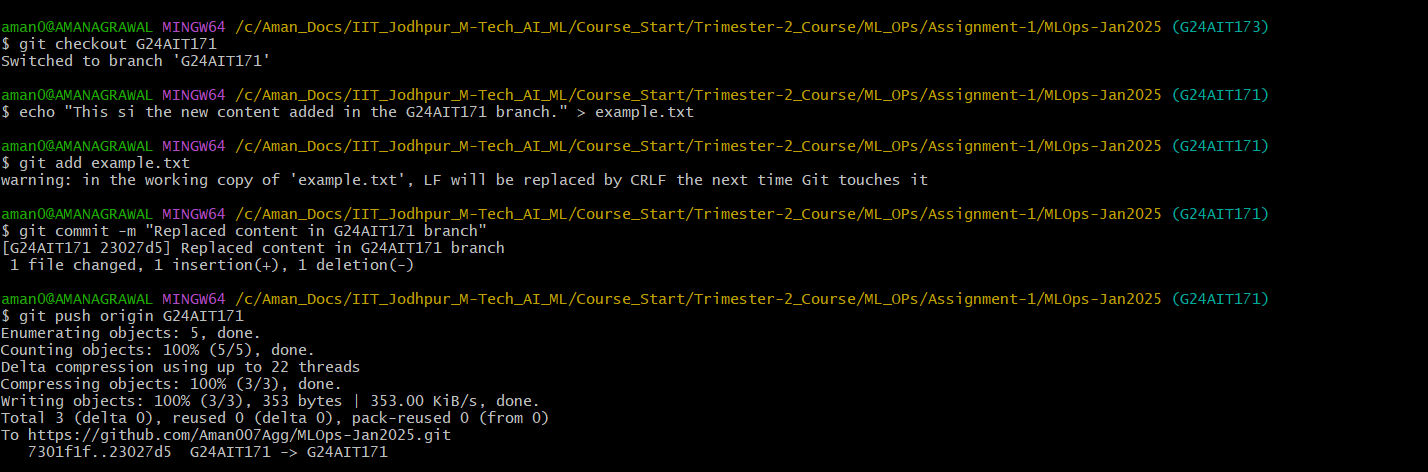
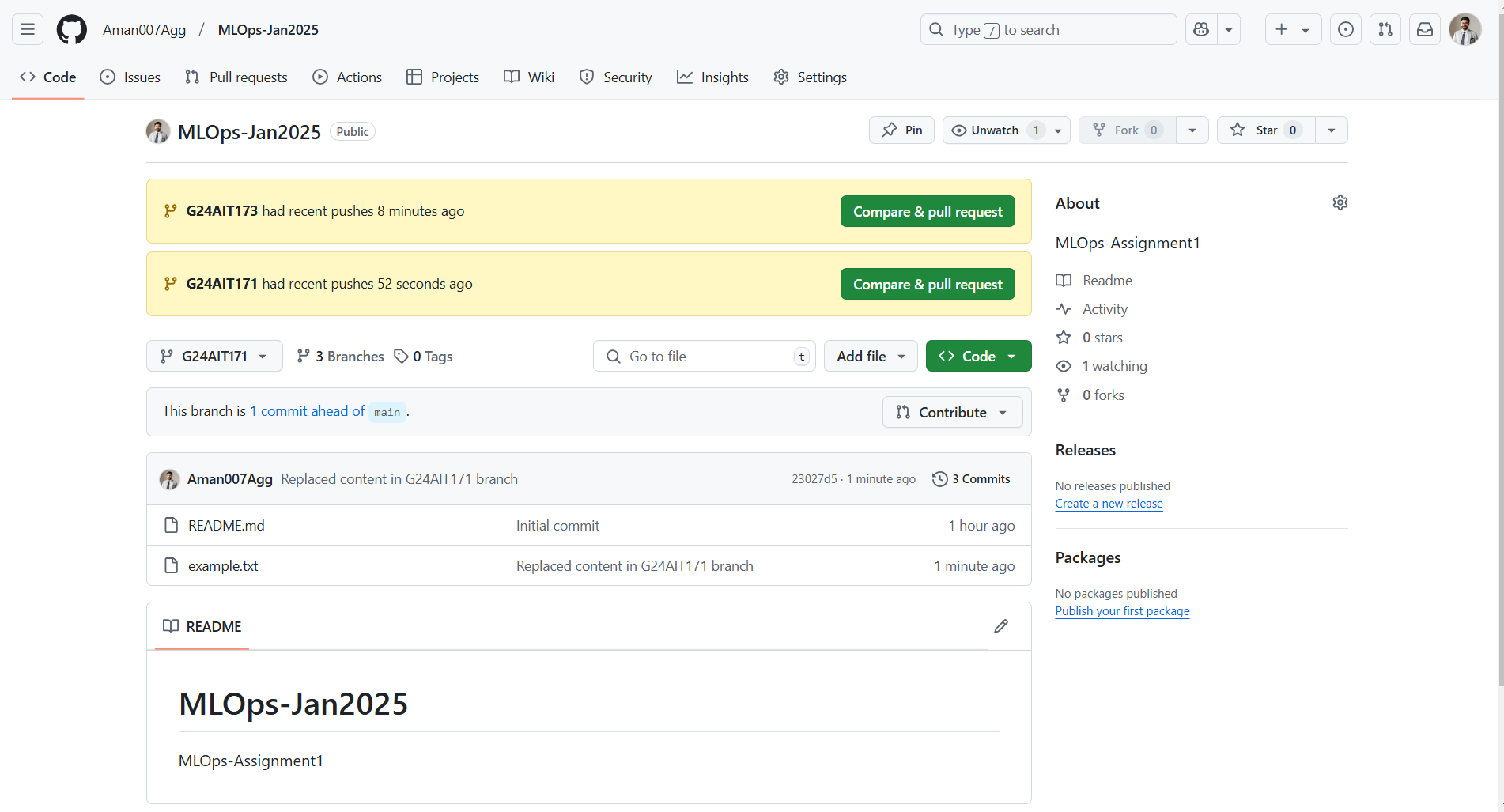


1. In the X+1 branch, modify the content of the existing file. In the X-1 branch, replace the content entirely. Commit these changes, then take a screenshot of the output of the following command: git log --graph --oneline and include it in your report.
2. Switch to the G24AIT173 branch,Modify example.txt,and Commit the changes, and Push the changes  
   

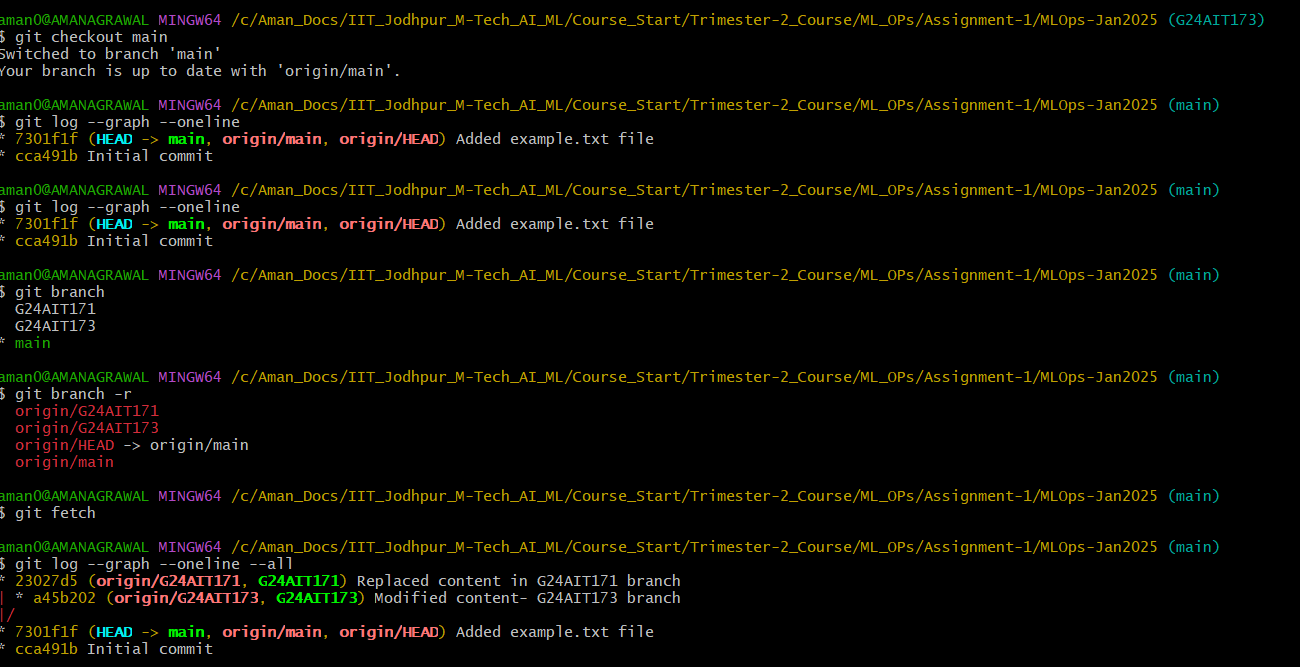


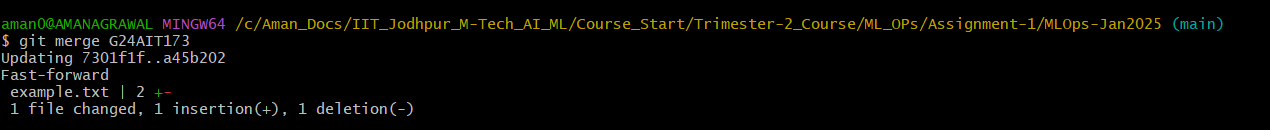
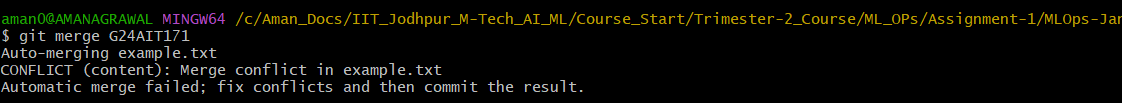
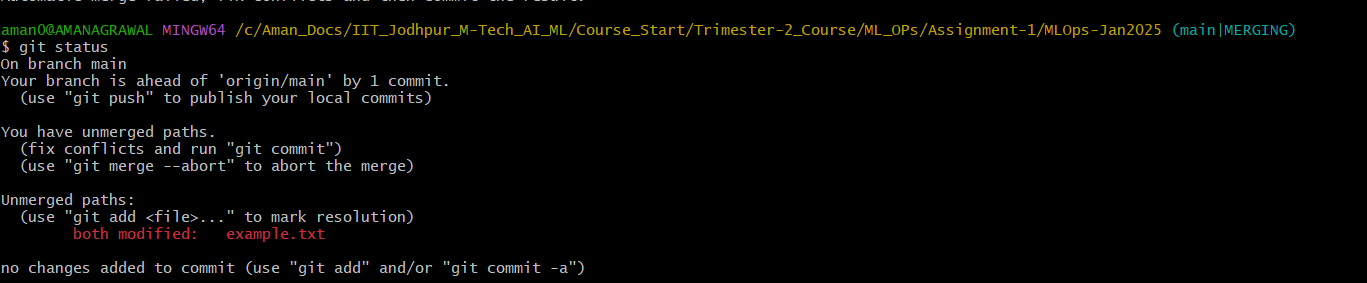
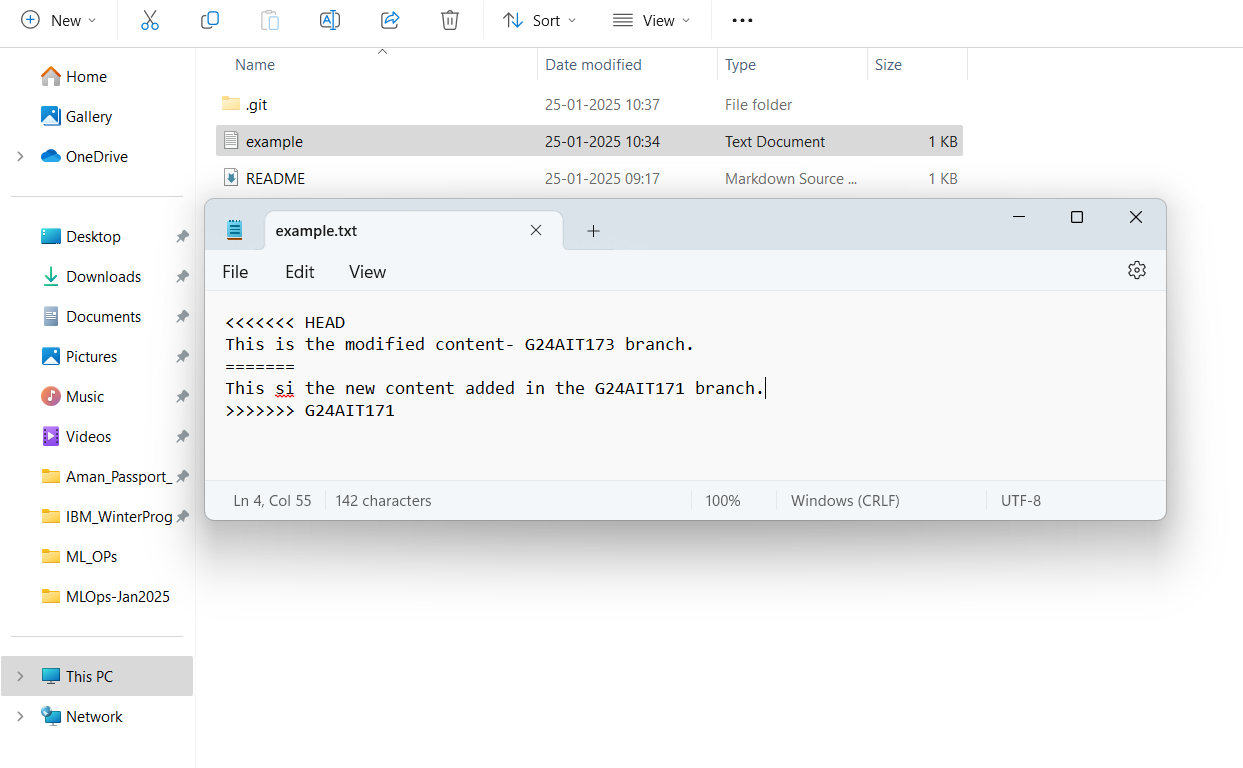
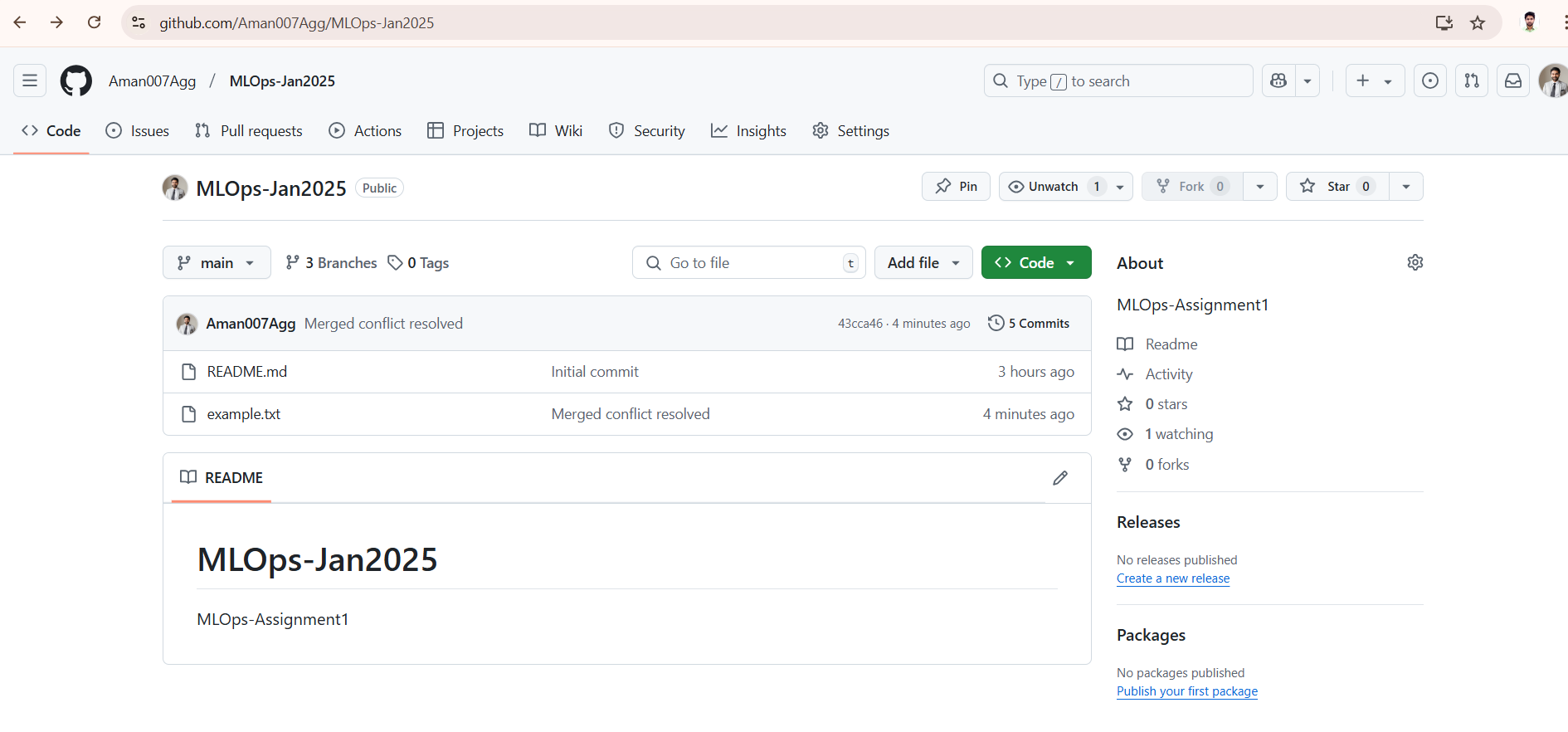
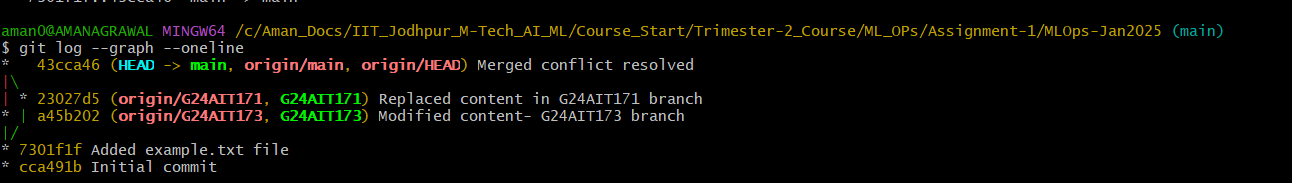


In the X-1 (G24AIT171) branch , replace the content entirely. Commit these changes  
  
Switch to the G24AIT171 branch ,Replace the content of example.txt entirely, Commit the changes,and Push the changes:

take a screenshot of the output of the following command: git log --graph --oneline and include it in your report

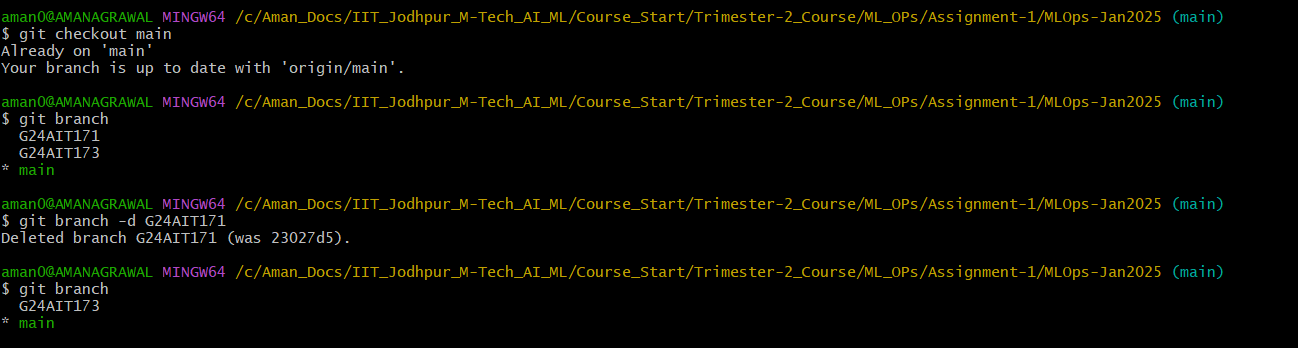
Switch back to the main branch,and View the commit graph  


F. Merge both branches into the main branch, resolving any conflicts if they arise. Take another screenshot of the output of the command: git log --graph --oneline and add it to your report.  
  
Merge G24AIT173 into main  
  
  
Merge G24AIT171 into main  
  
Note: Merge Failed - Merge conflict occurred  
  
Merge Conflict Occurred when trying to merge Branch G24AIT171 to main:  
  
  
  
  
Resolve the merge conflict - manually editing the text file, it is on you what changes you want to keep, you keep either of the changes or both as you wish, I will keep content from both the branches.  
  
  
  
screenshot of the output of the command: git log --graph --oneline  


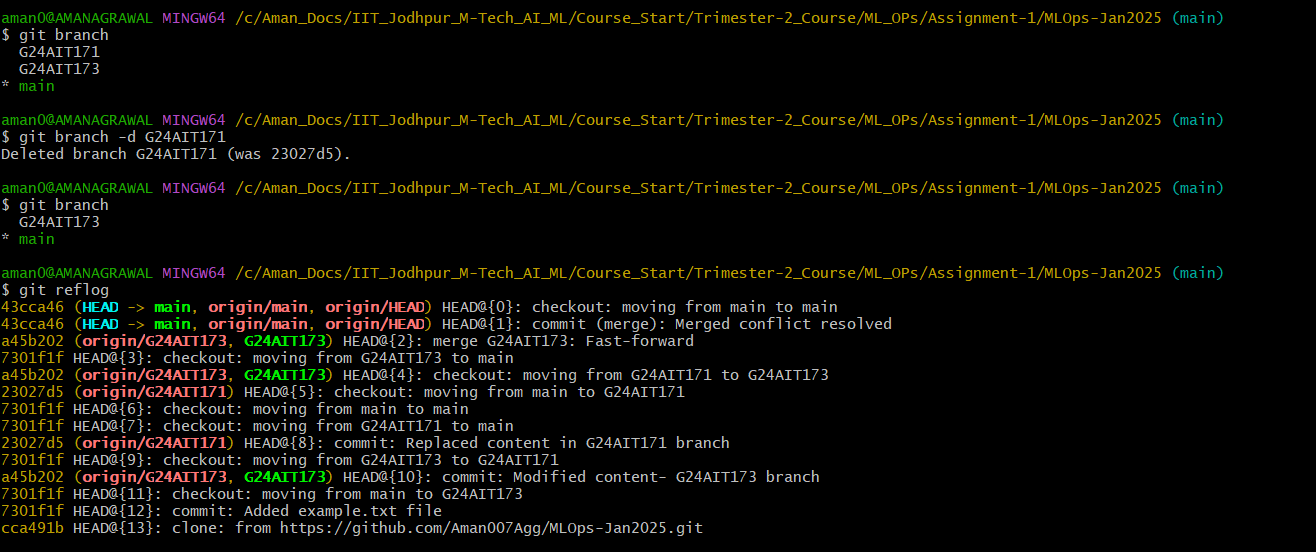
G. Push all changes to your GitHub repository

### H. Deleting a Branch- G24AIT171 :

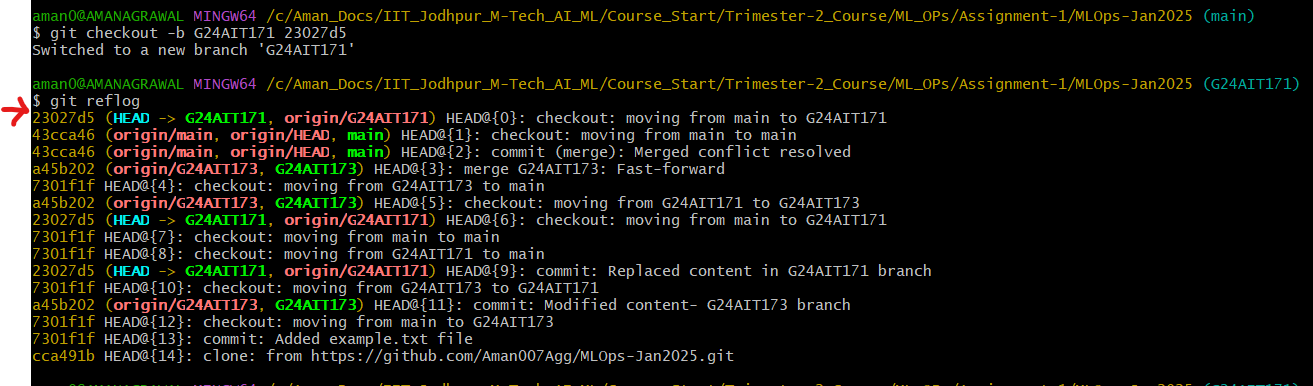
Note: “First, make sure you're on the main branch (or another branch, but not the one you're deleting).”

"I deleted the G24AIT171 branch using the command git branch -d G24AIT171. The branch was successfully removed, as shown in the screenshot."  


### I. Recovering the Deleted Branch- G24AIT171:

If you want to recover the deleted branch, you can use the reflog to find the commit hash from the branch before it was deleted.  
  


In our case it is point to hash commit   
 HEAD@{8}: commit: Replaced content in G24AIT171 branch

The commit hash associated with this entry (**23027d5**) is the commit you need to recover the branch.  
After finding the commit hash, you can create a new branch from that point.  
  
**Action: git checkout -b G24AIT171 23027d5**  
  
  
**The commit hash associated with this entry (23027d5) has now been recovered the branch G24AIT171**

### J. Tagging a Commit:

Tagging is useful for marking important commits. To add a tag to the commit you want to mark  
add a tag with a message , and push the tags to GitHub  
