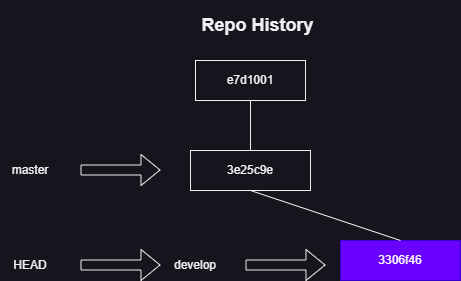
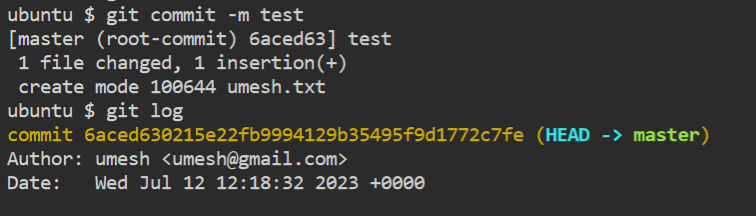
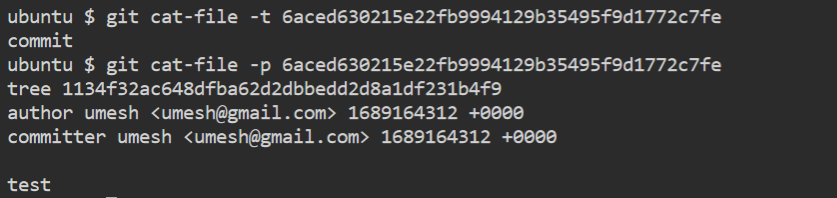
**How GIT Works**

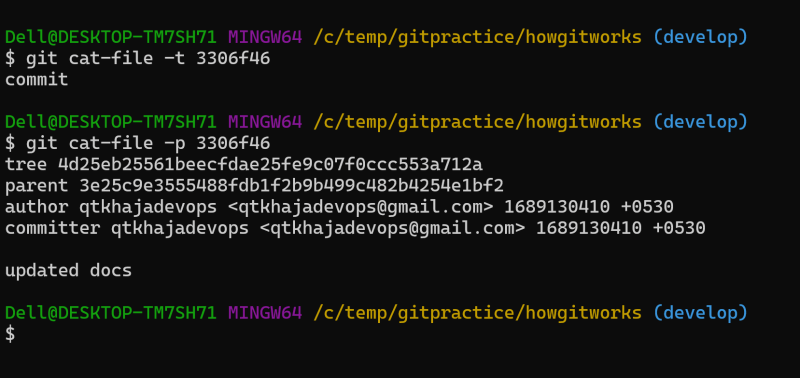
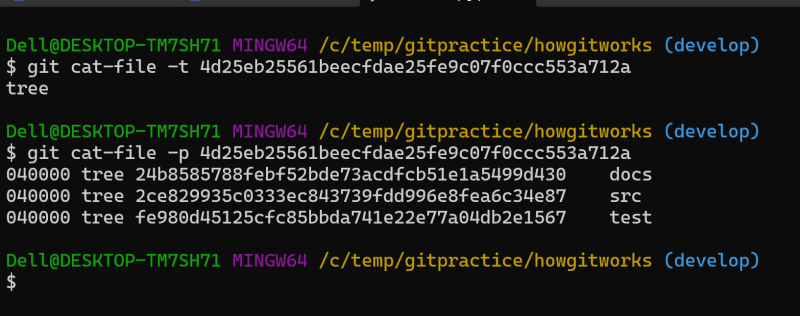
Current repository changes  


To understand we will be using plumbing commands****

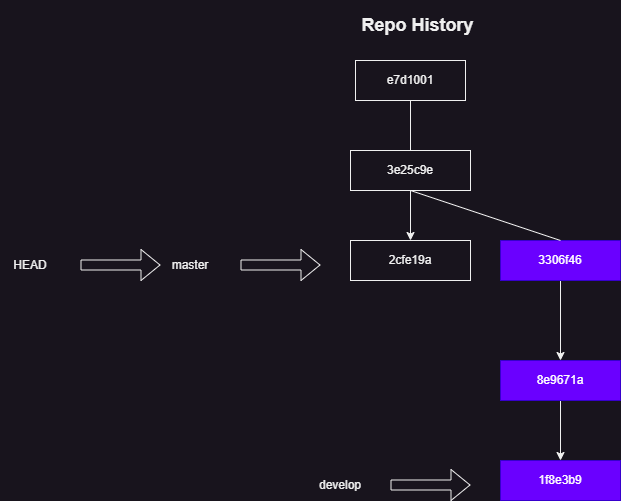
git cat-file -t <commit-id>

git cat-file -p <commit-id>



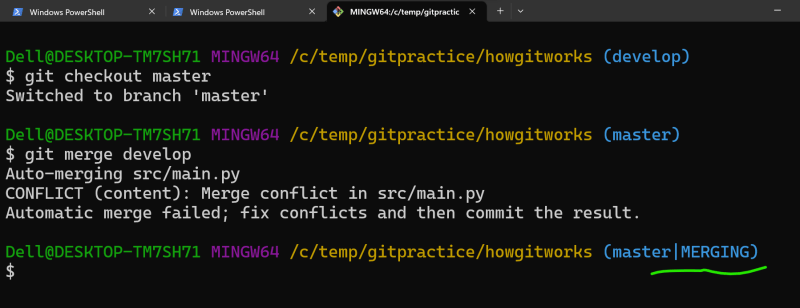
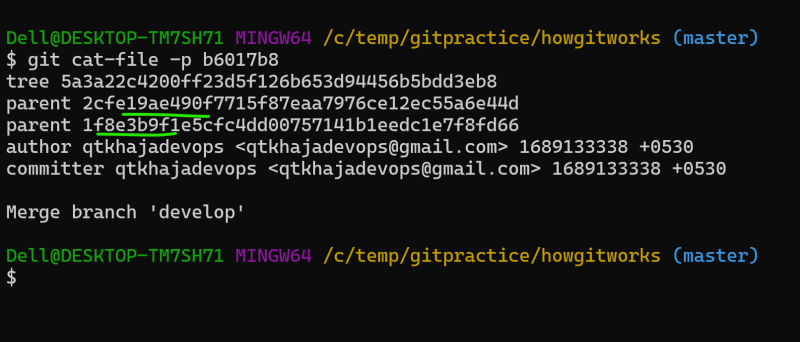
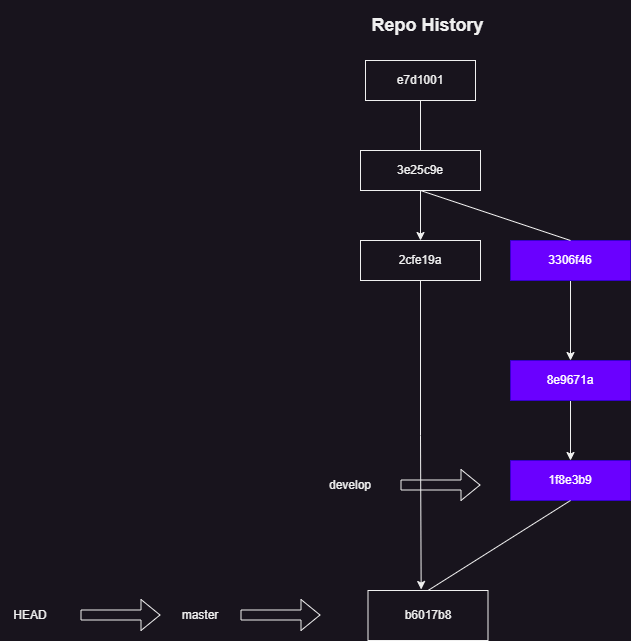
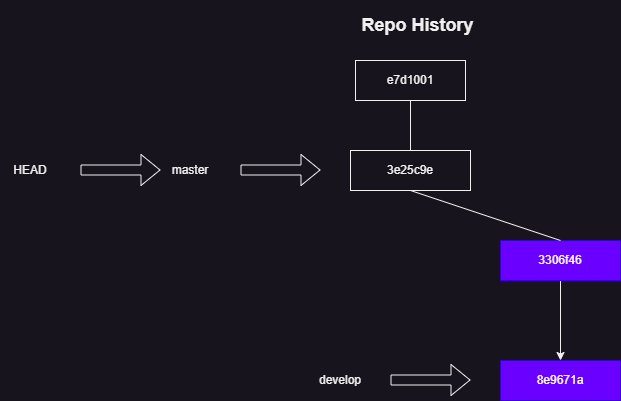
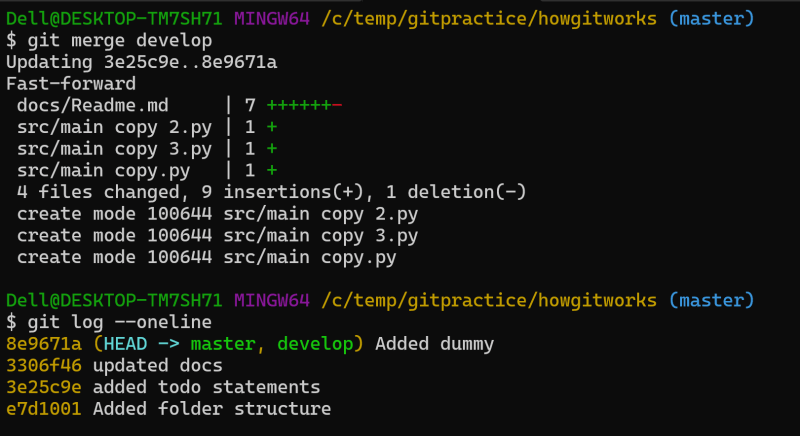
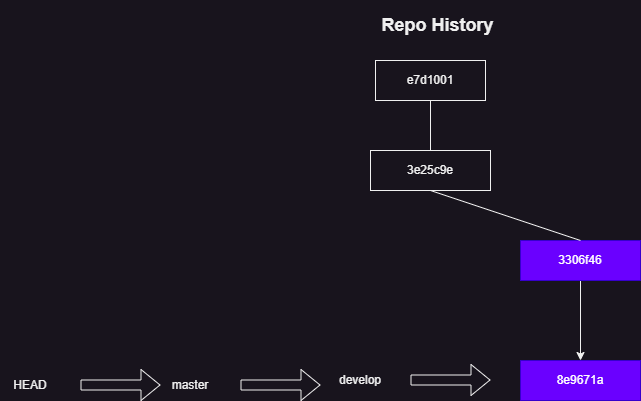
* Commit is an SHA-1 hash of
  + parent
  + author
  + message
  + date time
  + contents
* Git reference objects
  + branch
  + tag
* What is branch in git?
  + Branch is a reference object which points to some commit id, Branch points to a new commit id when a new commit happens
* What is tag in git?
  + tag is a reference object which points to some commit id and doesnot move with new commits
* Let’s figure from latest commit on develop branch 3306f46  
  
* tree in git represents folder/directory and blob/object represents file
* lets view contents of tree in latest commit  
  

**Merging Changes between Branches**

* Consider the following history  
  
* we need changes from develop branch into master branch
* now checkout to master and then execute merge

git checkout master

git merge develops

* When git cannot merge changes due to different content in same lines it will raise conflicts.  
  
* After fixing conflicts, add the changes and commit them which leads to creation of new commit with two parents  
    
  
* This is called as three-way merge
* Consider the following  
  
* We need to have all the changes in develop branch into master  
    
  

Note: If two file has same contain then they would be storing once only-

