

Week 8

1.Git-HOL

Gitbash:

- `cd ~/GitDemo`
- `git init`
- `git config --global init.defaultBranch master`
- `echo "Hello, I am Learning Git!" > welcome.txt`
- `git status`
- `cat welcome.txt`
- `git add welcome.txt`
- `git commit -m "Added welcome.txt file"`
- `git log`
- `git push origin master`
- `git branch --set-upstream-to=origin/master`
- `git status`
- `git push`
- `git add <file-name>`
- `git commit -a`
- `git status`

Gitlab:

Recent activity
Last 30 days

Merge requests created
0

Issues created
0

Members added
1

Subgroups and projects

Shared projects

Shared groups

Inactive

🔍

Search (3 character minimum)

Q

Name

🔽

🔒

G

GitDemo

🔒

★ 0

19 minutes ago

G

GitDemo

🔒

🔔

☆ Star

0

🍴 Fork

0

⋮

🔽 master

gitdemo

+

Find file

Code

⋮

🌐

Modified the welcome.txt

username authored 4 minutes ago

ff8257fd

🔒

History

Name	Last commit	Last update
<div>welcome.txt</div>	Modified the welcome.txt	4 minutes ago

Project information

↗ 1 Commit

🔗 1 Branch

🏷 0 Tags

💾 241 B Project Storage

+ Add README

+ Add LICENSE

+ Add CHANGELOG

+ Add CONTRIBUTING

+ Enable Auto DevOps

+ Add Kubernetes cluster

+ Set up CI/CD

+ Add Wiki

+ Configure Integrations

🔽 master

gitdemo / welcome.txt

🔒

welcome.txt

Find file

Blame

Edit

⋮

🌐

Modified the welcome.txt

username authored 5 minutes ago

ff8257fd

🔒

History

👤 Code owners

Assign users and groups as approvers for specific file changes.

Learn more.

Manage branch rules

welcome.txt

60 B

🔒

📄

📄

📄

1

Hello, I am learning Git!

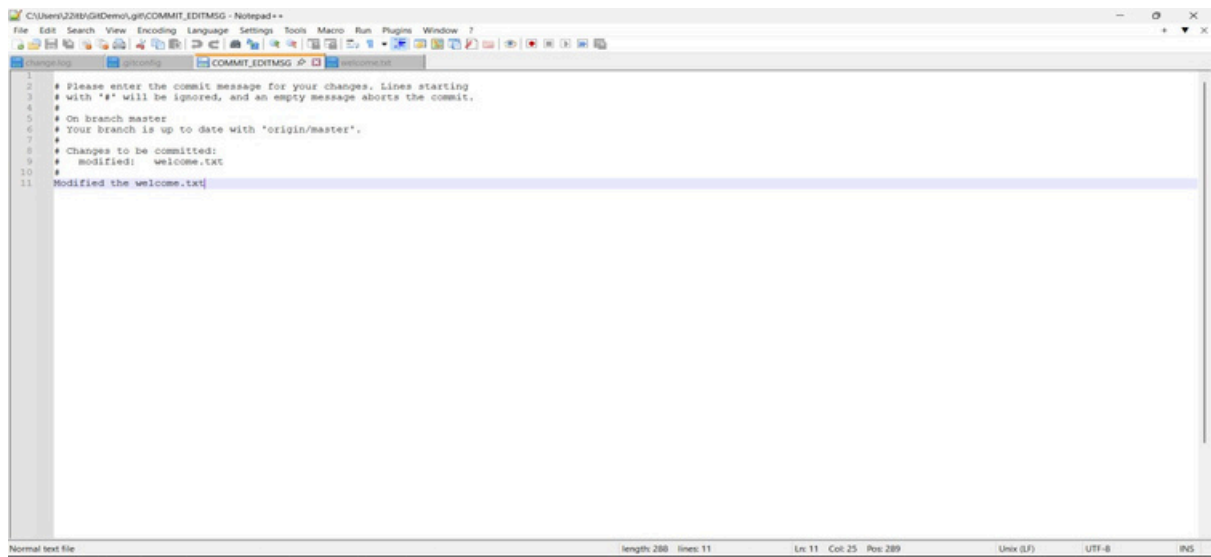
2

3

Welcome to the Version Control !

4

Notepad++



The screenshot shows the Notepad++ application window titled "C:\Users\J228th\GitDemo\git\COMMIT_EDITMSG - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and ?. The toolbar contains various icons for file operations and editing. The text area displays a commit message template with the following content:

```
1
2 # Please enter the commit message for your changes. Lines starting
3 # with '#' will be ignored, and an empty message aborts the commit.
4 #
5 # On branch master
6 # Your branch is up to date with 'origin/master'.
7 #
8 # Changes to be committed:
9 #   modified:   welcome.txt
10 #
11 Modified the welcome.txt
```

The status bar at the bottom indicates "Normal text file", "length 288 lines 11", "Ln 11 Col 25 Pos 289", "Unix (LF)", "UTF-8", and "ANSI".

2. Git-HOL

Gitbash:

- `git rm--cached`
- `rm .gitignore`
- `rm welcome.txt`
- `mkdir log`
- `echo "Fix: ignore log files and folders" >> .gitignore`
- `git add .`
- `git commit -m "Fix: ignore log files and folders"`
- `git push origin master`
- `git status`

Gitlab:

The screenshot shows the GitLab web interface for a project named 'GitDemo'. The top navigation bar includes the project name, a search bar, and buttons for 'Star', 'Fork', and 'History'. The main content area displays the commit history for the 'master' branch. The commit message is 'Fix: ignore log files and folders' by 'username' 3 minutes ago. The commit hash is 'dce4ed45'. Below the commit message is a table with columns 'Name', 'Last commit', and 'Last update'. The table lists four files: 'log', '.gitignore', 'debug.log', and 'welcome.txt'. The right sidebar shows project information, including '3 Commits', '1 Branch', '0 Tags', and '881 B Project Storage'. It also includes links to 'Add README', 'Add LICENSE', 'Add CHANGELOG', 'Add CONTRIBUTING', 'Enable Auto DevOps', 'Add Kubernetes cluster', 'Set up CI/CD', 'Add Wiki', and 'Configure Integrations'. At the bottom, it shows 'Created on August 07, 2025'.

Name	Last commit	Last update
log	Fix: ignore log files and folders	4 minutes ago
.gitignore	Added .gitignore to ignore log files and...	8 minutes ago
debug.log	Fix: ignore log files and folders	4 minutes ago
welcome.txt	Modified the welcome.txt	2 hours ago

master

gildedemo / debug.log

debug.log

Find fileBlameEdit

Fix: ignore log files and folders

username authored 5 minutes ago

dce4ed45History

Code ownersAssign users and groups as approvers for specific file changes. Learn more.

Manage branch rules

debug.log19 B

1This is a log file

2

master

gildedemo / log

log

+LockFind fileCode

Fix: ignore log files and folders

username authored 5 minutes ago

dce4ed45History

Code ownersAssign users and groups as approvers for specific file changes. Learn more.

Manage branch rules

Name	Last commit	Last update
..		
errors.log	Fix: ignore log files and folders	5 minutes ago

master

gildedemo / log / errors.log

errors.log

Find fileBlameEdit

Fix: ignore log files and folders

username authored 5 minutes ago

dce4ed45History

Code ownersAssign users and groups as approvers for specific file changes. Learn more.

Manage branch rules

errors.log10 B

1error log

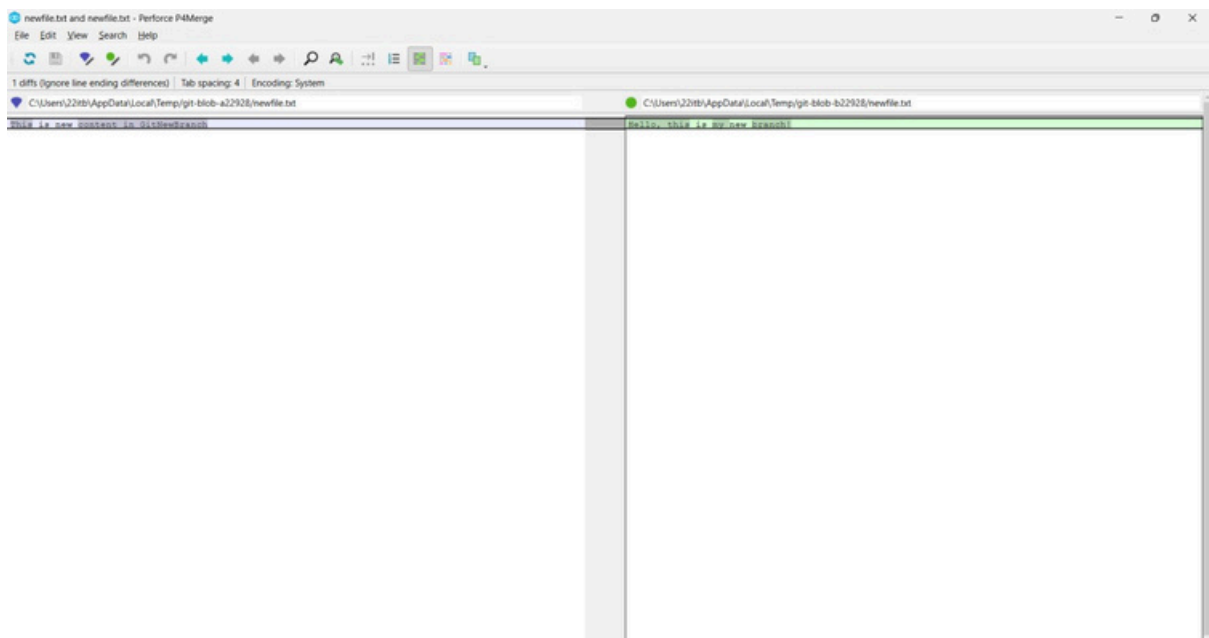
2

3. Git-HOL

Gitbash:

- gitbranch
- echo "Hello, this is my new branch!" > newfile.txt
- git add newfile.txt
- git commit -m "Added newfile.txt in gitnewbranch"
- git status
- git checkout master
- git diff
- echo "This is new content in GitNewBranch" > newfile.txt
- git add newfile.txt
- git commit -m "Added new content in GitNewBranch"
- git checkout master

View the changes by p4merge:



After resolving the conflict the new branch is deleted:

Deleted the new branch that is “GitNewBranch”:

Gitbash:

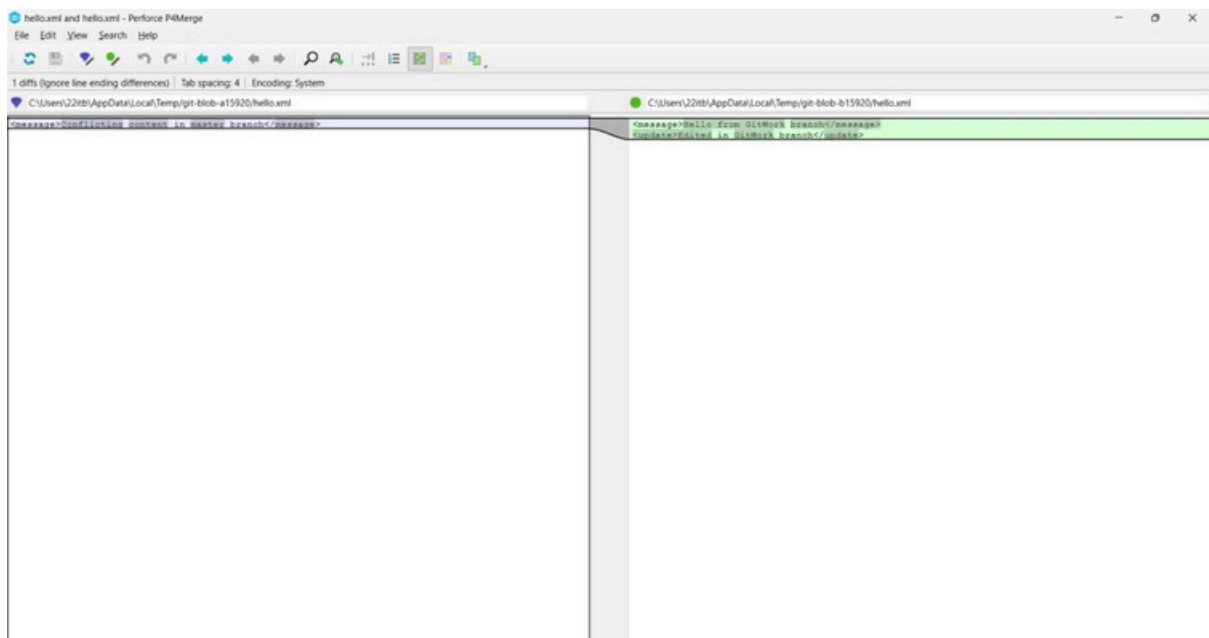
- git status
- git checkout GitNewBranch
- git diff master..GitNewBranch
- git log --oneline --graph --decorate
- git diff master..GitNewBranch
- git branch -D GitNewBranch
- git status

4. Git-HOL

Gitbash:

- `gitstatus`
- `echo "updated/edited in GitWork branch" > hello.xml`
- `git add hello.xml`
- `git commit -m "Added and updated hello.xml"`
- `git checkout master`
- `git commit -m "Master: Added conflicting hello.xml"`
- `git log --oneline --graph --decorate`
- `git diff master..GitWork`
- `git diff master..GitWork --stat`
- `git config --global diff.tool vimdiff`
- `git difftool master..GitWork`

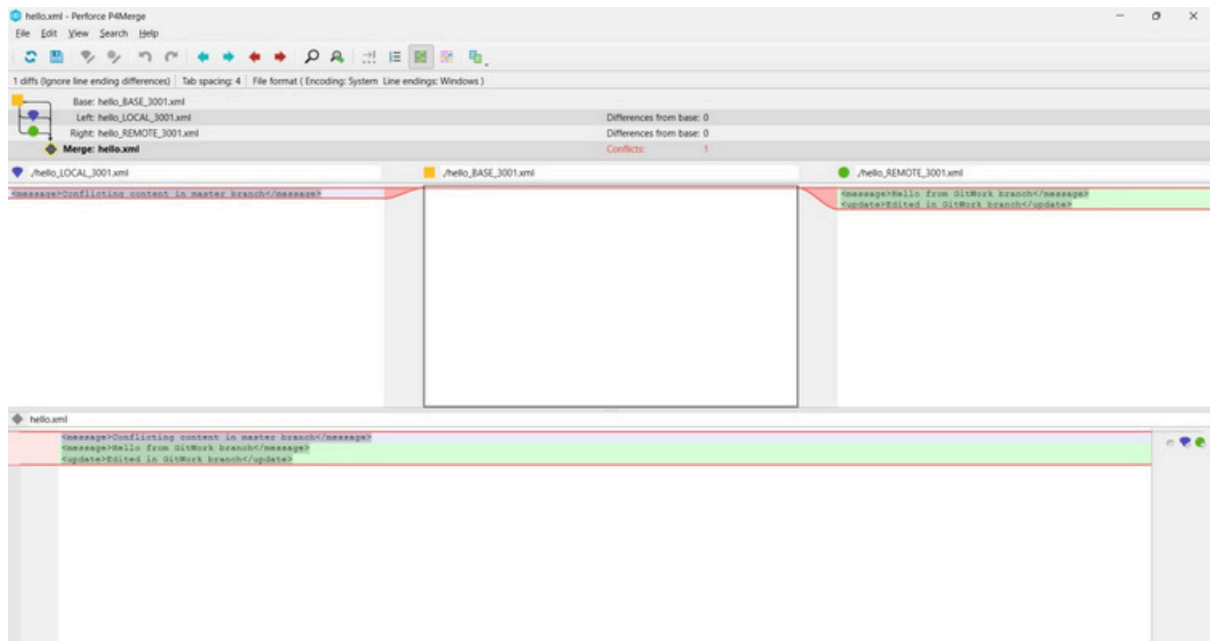
P4merge:



Merging Conflict

Gitbash:

- ls
- git mergetool



Pushed to master branch

Gitbash:

- git add hello.xml
- git commit -m "Resolved merge conflict in hello.xml"

After merging to master branch, the GitWork branch is deleted:

Gitbash:

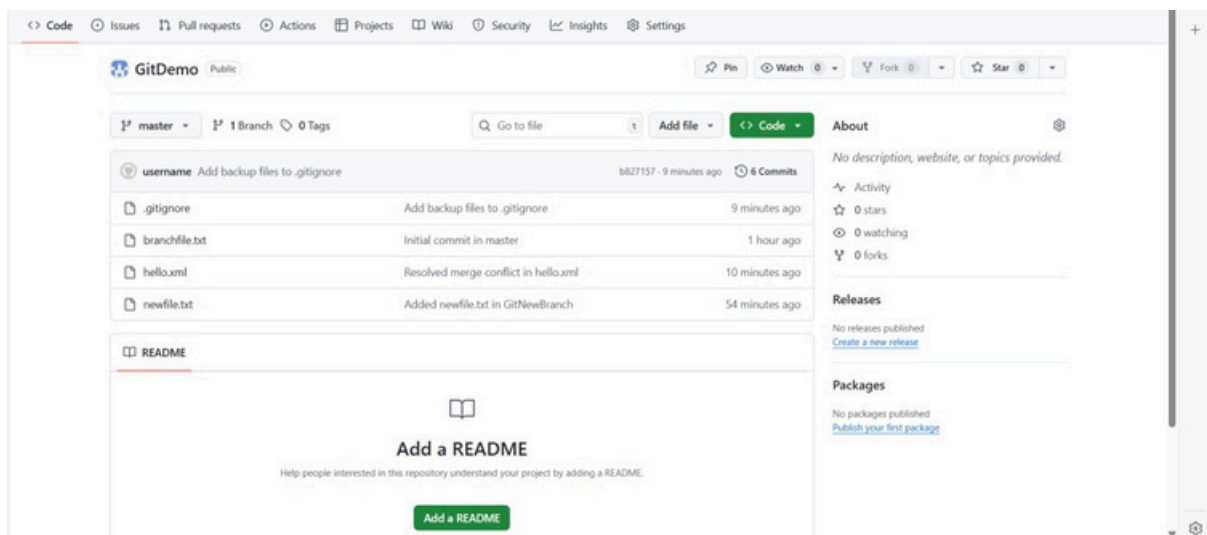
- git log --oneline --graph --decorate

5. Git-HOL

Gitbash:

- git branch
- git pull origin master
- git remote -v
- git remote add origin https://github.com/Babuparsanth27/GitDemo.git
- git remote -v
- git fetch https://github.com/Babuparsanth27/GitDemo.git
- git push --set-upstreamoriginmaster

Pushed the code from local to github:



Changes are displayed

