Experiment No. 1 Date 11/2/25 Page No 1

Name of Experiment: Git and GitHub-1

A1) Demonstrate & create project in local and remote repository using GitBash and GitHub and apply init, status, log, add, commit, push, config, clone and reset commands on repository.

Git Init command:-

This command is used to create a local repository

$git init

To add remote repository to local git repository:-

$ git remote add origin https://github.com/Yashwanth1304/devops.git

Git add command:-

This command is used to add one or more file to staging area

$ git add .

```
student@DBMS33 MINGW64 ~/desktop/devops
$ git init
Initialized empty Git repository in C:/Users/student/Desktop/devops/.git/
```

```
student@DBMS33 MINGW64 ~/desktop/devops (master)
$ git remote add origin https://github.com/Yashwanth1304/devops.git
```

Experiment No.....Date.....Page No 2

Name of Experiment.....

Git config command:-

This command configures the user. The Git config is the first & necessary command used on the Git command line

```
$ git config --global user.name "Yashwanthraidu"
$ git config --global user.email "yashwanthraidu@gmail.com"
```

Git commit command :-

git used to commit staged files with a message

```
$ git commit -m "version1"
```

git push command:-

It is used to upload local repository content to a remote repository.

```
$ git push origin master
```

```
student@DBMS33 MINGW64 ~/desktop/devops (master)
$ git config --global user.email "yashwanthnnaidu@gmail.com"
```

```
student@DBMS33 MINGW64 ~/desktop/devops (master)
$ git commit -m "version1"
[master (root-commit) 944db07] version1
 2 files changed, 1 insertion(+)
 create mode 100644 hello.docx
 create mode 100644 hi.txt
```

```
$ git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 20 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 8.68 KiB | 2.89 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/Yashwanth1304/devops/pull/new/master
remote:
To https://github.com/Yashwanth1304/devops.git
 * [new branch]      master -> master
```

Experiment No.....Date.....Page No 3

Name of Experiment.....

Init status command:-
The status command is used to display the state of working directory & the staging area. It also lists the files that you've changed & those you still need to add or commit

$ git status

Init log command:-
This command is used to check the commit history
$ git log

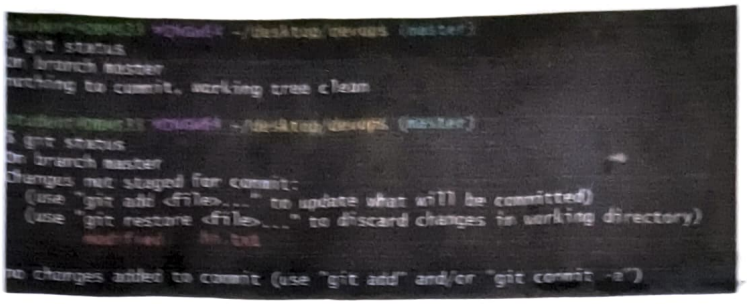
Init reset:
The term reset stands for undoing changes.

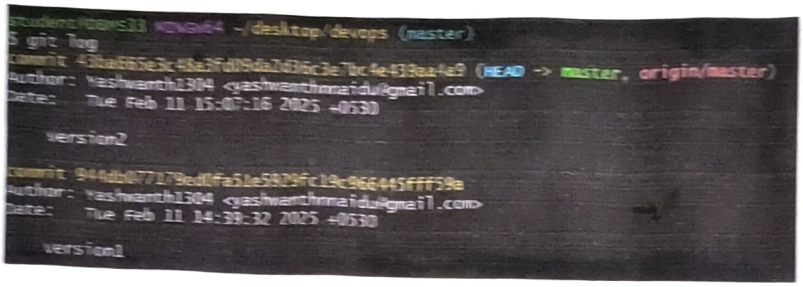
$git reset --hard id

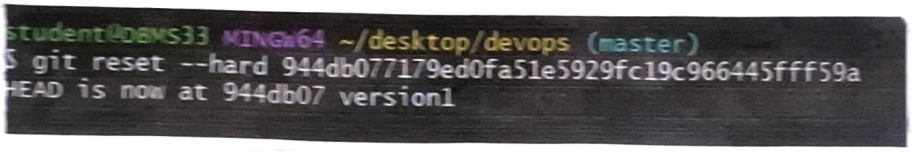
Init clone:-
This command is used to make a copy of a repo from an existing url

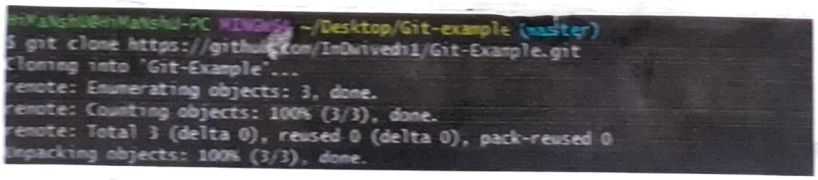
$git clone https://github.com/Vaishnavanthi304/devops.git

25/2/25


$ git status
On branch master
nothing to commit, working tree clean
$ git status
On branch master
Changes not staged for commit:
(use "git add <file>..." to update what will be committed)
(use "git restore <file>..." to discard changes in working directory)
no changes added to commit (use "git add" and/or "git commit -a")


$ git log
commit 944db077179ed0fa51e5929fc19c966445fff59a (HEAD -> master, origin/master)
Author: Yashwanth1304 <yashwanthnaidu@gmail.com>
Date: Tue Feb 11 15:07:16 2025 +0530
version2
commit 944db077179ed0fa51e5929fc19c966445fff59a
Author: Yashwanth1304 <yashwanthnaidu@gmail.com>
Date: Tue Feb 11 14:39:32 2025 +0530
version1


student@DBMS33 MINGW64 ~/desktop/devops (master)
$ git reset --hard 944db077179ed0fa51e5929fc19c966445fff59a
HEAD is now at 944db07 version1


~/Desktop/Git-example (master)
$ git clone https://github.com/InDwivedi1/Git-Example.git
Cloning into 'Git-Example'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.

Experiment No. 2 Date 18/2/25 Page No 4

Name of Experiment Git and GitHub - II

A2) Demonstrate to create a project in remote repository & apply fork, merge, diff, merge conflict, branch & pull request concepts on repository using Github.

Git Fork:

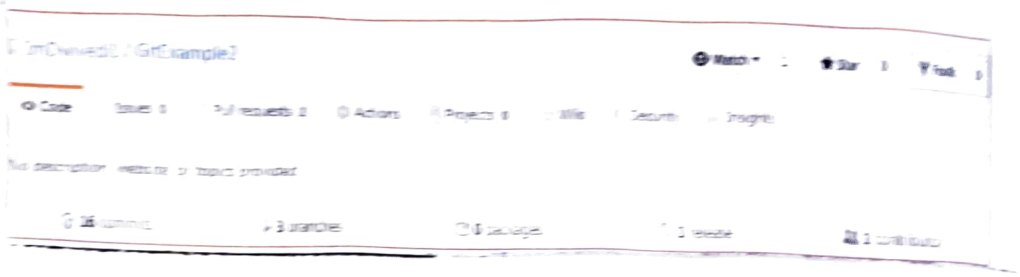
A fork is a rough copy of a repo. Forking a repository allows you to freely test & debug without affecting the original project

Steps:

1. Login to the GitHub account.
2. Find the GitHub repository which you want to fork
3. Click the fork button on the upper right side of the repository's page.

Git Branching: Branches allows for working on seperate features without affecting the main branch

$git checkout -b branch1
(or)
$ git branch branch1
$ git checkout branch1
This creates & switches to new branch called branch1



```
student@DBMS30 MINGW64 ~/desktop/devops (master)
$ git checkout -b branch1
Switched to a new branch 'branch1'
```

Experiment No..... Date..... Page No 5

Name of Experiment.....

Git merge:-

The git merge command facilitates you to take the data created by git branch & integrate them into single branch.

$ git merge master

Git Diff:-

It compares the different versions of data source. The version control system stands for working with a modified version of file. So the diff command is useful tool for working with Git

$ git diff branch1

Git merge conflict:-

When two branches are trying to merge & both are edited at the same time & in the same file, Git won't be able to identify which version is to take for changes. Such a situation is called merge conflict

$ git merge branch1

```
student@DBMS30 MINGW64 ~/desktop/devops (branch1)
$ git merge master
Already up to date.

student@DBMS30 MINGW64 ~/desktop/devops (branch1)
$ git switch master
M       lab.txt
Switched to branch 'master'
```

```
student@DBMS30 MINGW64 ~/desktop/devops (master)
$ git diff branch1
warning: in the working copy of 'lab.txt', LF will be replaced by CRLF the next time Git touches it
diff --git a/lab.txt b/lab.txt
index ce01362..2967aca 100644
--- a/lab.txt
+++ b/lab.txt
@@ -1 +1 @@
-hello
+hello hiii
```

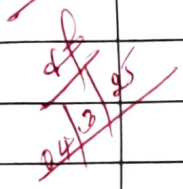
```
student@DBMS30 MINGW64 ~/desktop/devops/conflict (main)
$ git merge branch1
Auto-merging vvce.txt
CONFLICT (content): Merge conflict in vvce.txt
Automatic merge failed; fix conflicts and then commit the result.
```

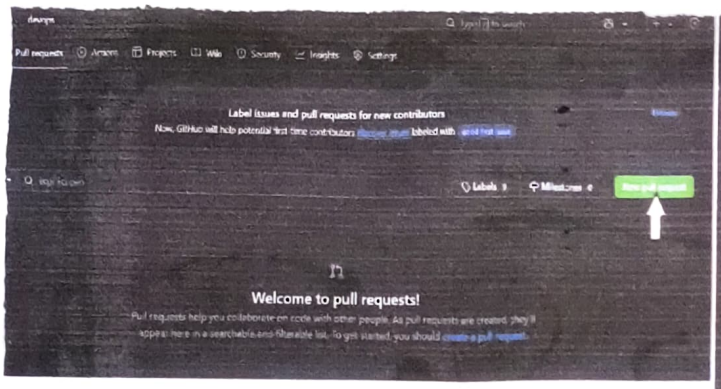
Experiment No..... Date..... Page No 6

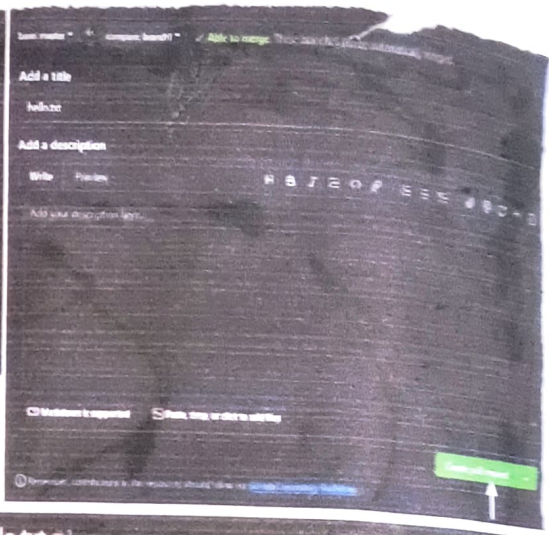
Name of Experiment.....

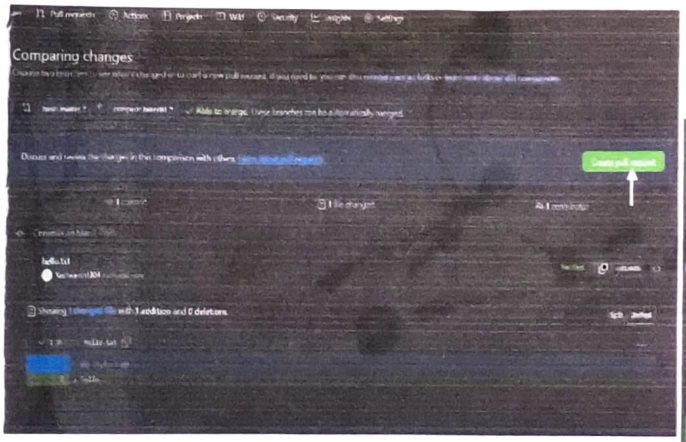
Git pull:-

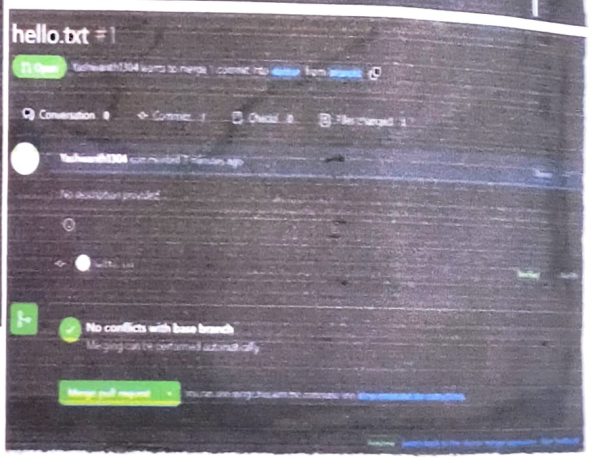
It fetches & merges changes from the remote do your working directory or in GitHub/GUI merge the sub branche with a main branch. By




Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Label issues and pull requests for new contributors
Now, GitHub will help potential first time contributors
Labels
Milestones
New pull request
Welcome to pull requests!
Pull requests help you collaborate on code with other people. As pull requests are created, they'll appear here in a searchable and filterable list. To get started, you should create a pull request.


Add a title
hello.txt
Add a description
Write
Preview
Create pull request


Comparing changes
Able to merge. These branches can be automatically merged.
Create pull request
1 file changed
hello.txt
1 addition and 0 deletions


hello.txt #1
Open
Conversation
Commits
Checks
Files changed
No description provided.
No conflicts with base branch
Merging can be performed automatically.
Merge pull request

Experiment No. A3 Date 4/3/25 Page No 7

Name of Experiment Jenkins

A3. Demonstrate the process of integration github repository with Jenkins to automate the project extension in CI/CD pipeline.

Download and install Jenkins

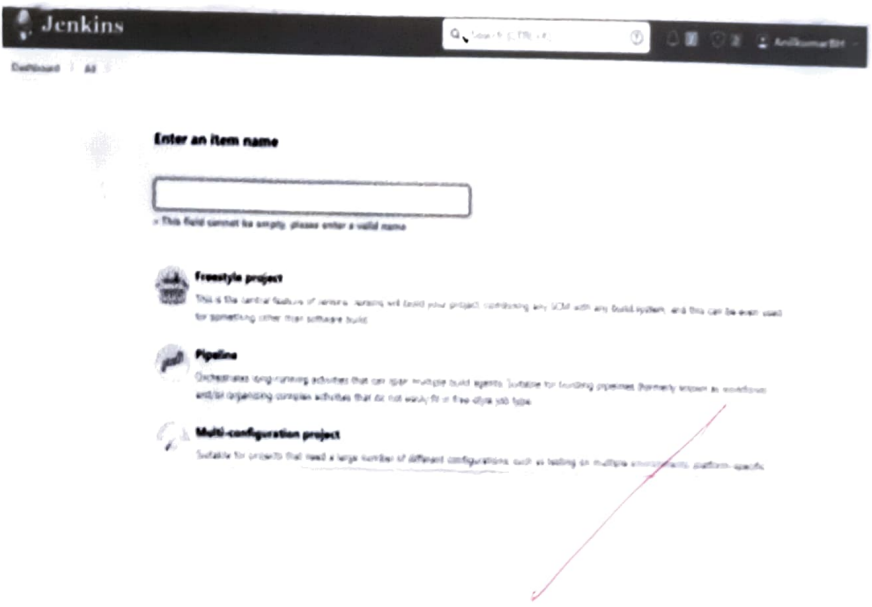
Download Jenkins. Under OS. click on windows.
* After the file is downloaded, unzip it. Click on the folder & install it. select "finish" once done

Run Jenkins on localhost 8081
* open browser & type "localhost:8081"
* Enter the credentials & log in.

Jenkins server interface
* New items allows you to create a new project
* Build history shows the status of your builds.
* Manage system deals with the various configurations of the system.

Configuration of Jenkins:

1. Create a new project & provide name for project. choose "Freestyle project"


Jenkins
Dashboard
All
Enter an item name
» This field cannot be empty, please enter a valid name
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific

2. Click on newly ~~added~~ created project and move to configuration. Under source code Management provide git repository

3. Provide branch name where project is located

4. Under Build triggers select Poll SCM and configure the schedule according to your project

5. In build steps choose "Execute windows batch command" and provide command for execution

6. Apply and Save configuration.

7. Click on "Build Now" for detailed output click on "console output" & verify the process. Finally build status will be displayed according to your project implementation

Source Code Management

None

Git ?

Repositories ?

Repository URL ?

https://github.com/AnilBiluvala/Demo.git

Credentials ?

- none -

Add

Branches to **build** ?

Branch Specifier (blank for 'any') ?

*/main

Build Triggers

Trigger builds remotely (e.g., from scripts) ?

Build after other projects are built ?

Build periodically ?

GitHub hook trigger for GITScm polling ?

Poll SCM ?

Schedule ?

* * * * *

Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *" to poll once per hour

Build Steps

Execute Windows batch command ?

Command

See the list of available environment variables

python first.py

Advanced