| **Name : Parth Kadam** | **Class/Roll No. : D11AD/27** | **Grade :** |
| --- | --- | --- |

**Title of Experiment : To Perform various GIT operations on local and Remote repositories using GIT Cheat-Sheet.**

**Objective of Experiment : To be aware of different Version Control tools like GIT, CVS or Mercurial**

**Outcome of Experiment : To obtain complete knowledge of the ―version control system to effectively track changes**

**augmented with Git and GitHub**

**Description / Theory :**

**Local Repository Operations**:

Initialization:

git init: Initializes a new Git repository in the current directory.

Cloning:

git clone <repository\_url>: Creates a local copy of a remote repository.

Adding Changes:

git add <file>: Stages changes for the next commit.

git add . or git add -A: Stages all changes.

Committing:

git commit -m "Commit message": Records staged changes to the local repository.

Branching:

git branch <branch\_name>: Creates a new branch.

git checkout <branch\_name>: Switches to a different branch.

git merge <branch\_name>: Merges changes from one branch to another.

Viewing Changes:

git status: Shows the status of changes.

git log: Displays commit history.

git diff: Shows changes between commits.

Undoing Changes:

git reset <file>: Unstages changes.

git revert <commit>: Reverts a commit.

git reset --hard <commit>: Discards changes and resets to a specific commit.

**Remote Repository Operations:**

Adding a Remote:

git remote add <remote\_name> <repository\_url>: Adds a remote repository.

Fetching:

git fetch <remote\_name>: Retrieves changes from a remote repository.

Pulling:

git pull <remote\_name> <branch\_name>: Fetches changes and merges them into the current branch.

Pushing:

git push <remote\_name> <branch\_name>: Pushes local changes to a remote repository.

git push -u <remote\_name> <branch\_name>: Sets up tracking for the branch.

Branch Operations:

git push <remote\_name> --delete <branch\_name>: Deletes a remote branch.

git remote prune <remote\_name>: Removes references to remote branches that have been deleted.

Collaboration:

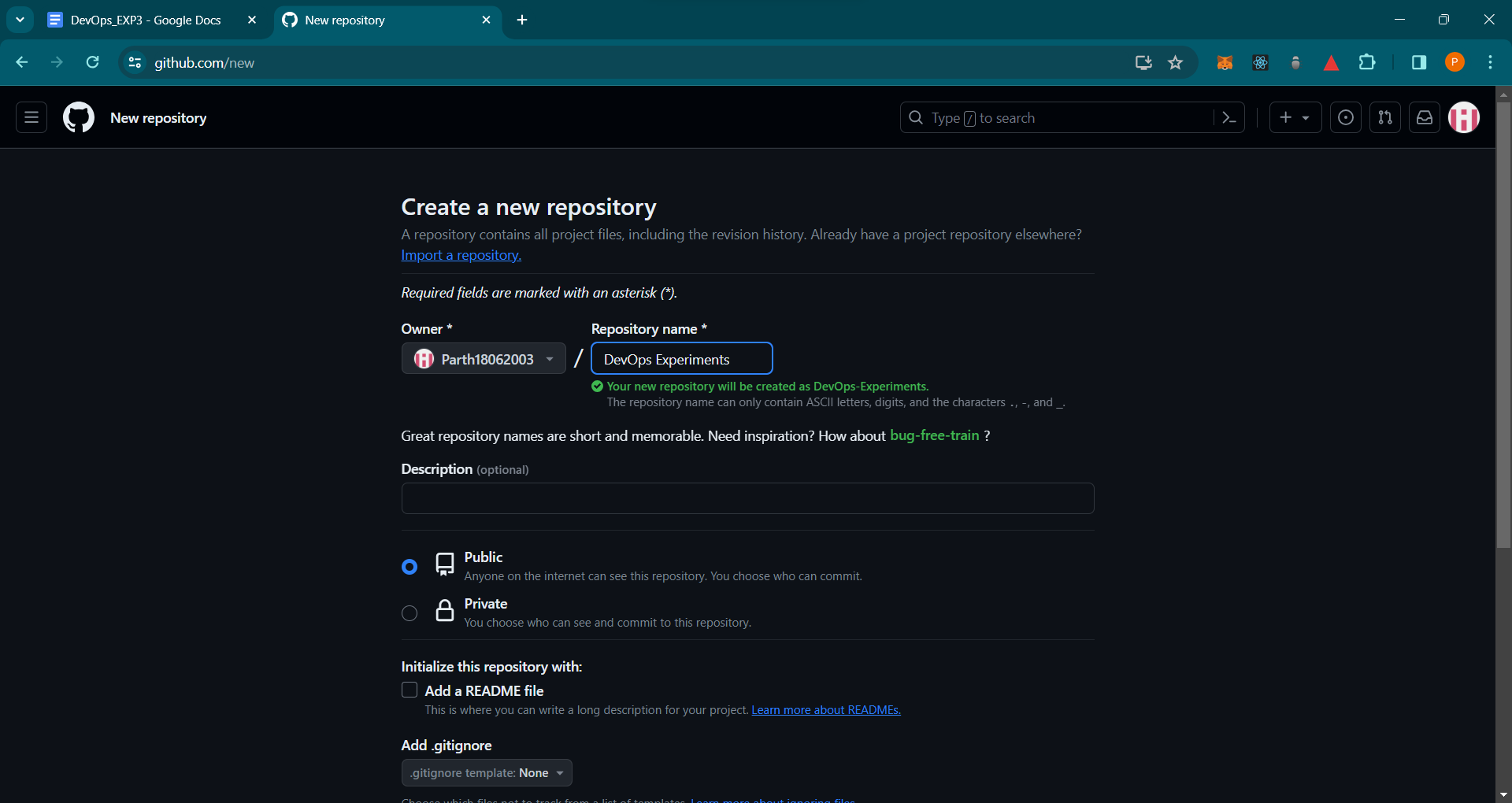
Forking and Pull Requests: Common in open-source projects.

git clone <forked\_repository\_url>: Clones a forked repository.

git pull-request: Creates a pull request to merge changes.

**Program :**

1)Create a GitHub repository

****

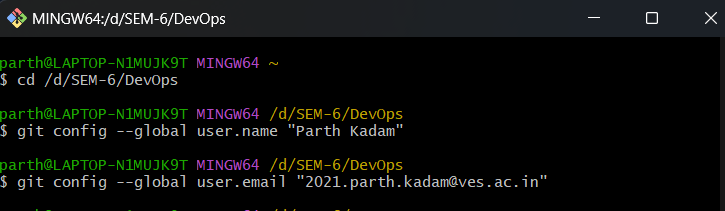
2)SETUP Configuring user information

1. git config --global user.name “[firstname lastname]” :- set a name that is

identifiable for credit when reviewing version history .

2. git config --global user.email “[valid-email]” :-set an email address that will be associated with each history marker,

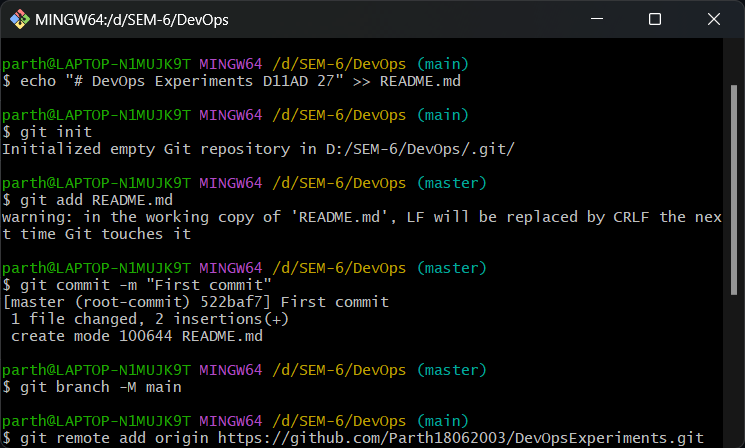
3. git config --global color.ui :- auto set automatic command line coloring for Git for easy reviewing.

****

3)SETUP & INIT:

1. git init :- initialize an existing directory as a Git repository.

2. git clone [url] :- retrieve an entire repository from a hosted location via URL

****

4)STAGE & SNAPSHOT:

1. git status:- show modified files in the working directory, staged for your next commit.

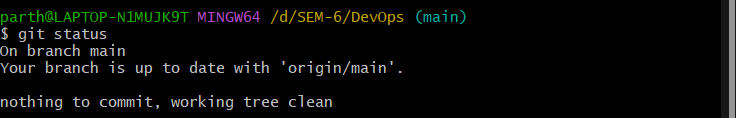
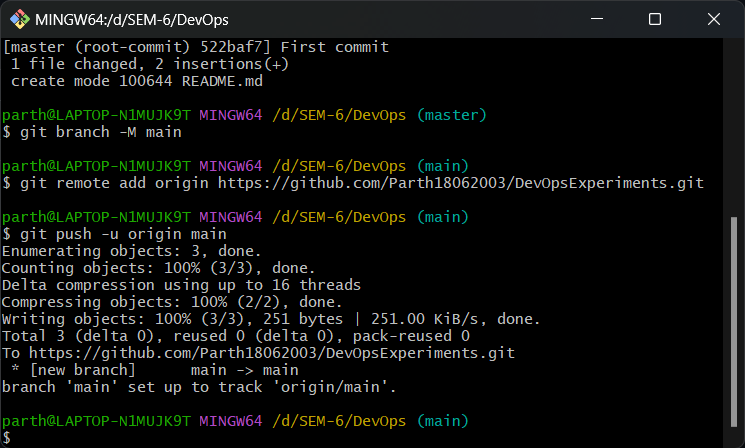
2. git add [file]:- add a file as it looks now to your next commit (stage).

3. git reset [file]:- unstage a file while retaining the changes in the working directory.

4. git diff:- diff of what is changed but not staged.

5. git diff --staged :- diff of what is staged but not yet committed.

6. git commit -m “[descriptive message]”:- commit your staged content as a new commit snapshot

****

5)BRANCH & MERGE :

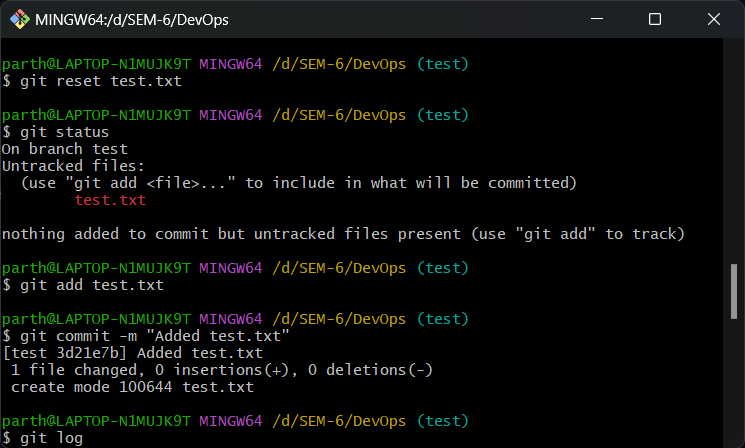
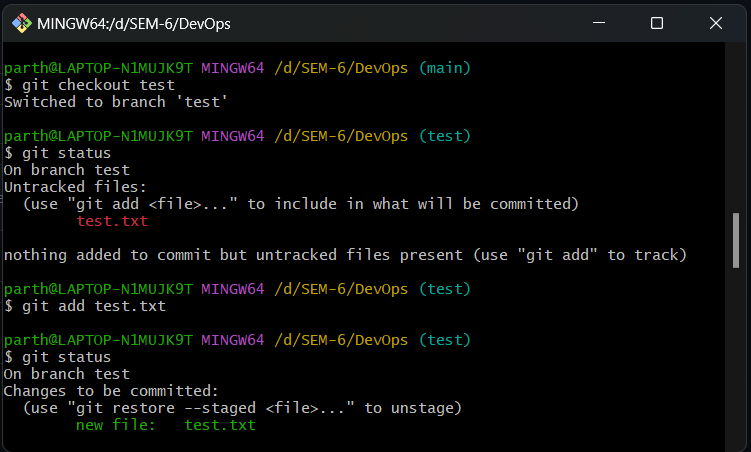
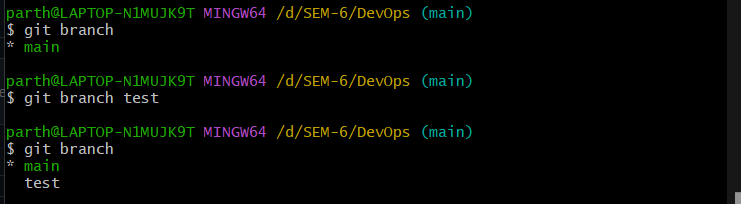
1. git branch list your branches. a \* will appear next to the currently active branch.

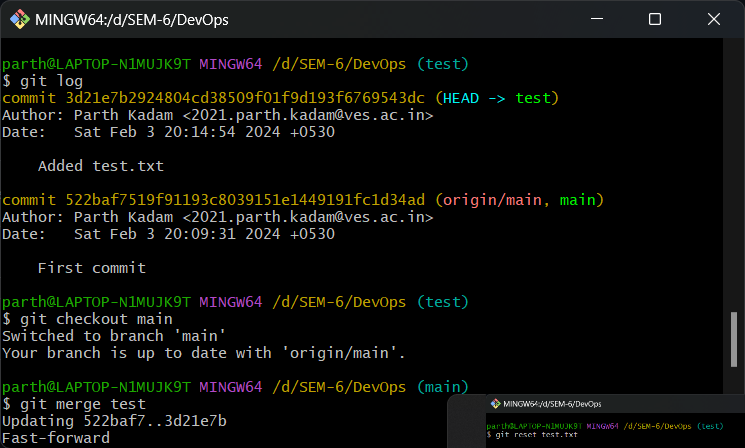
2. git branch [branch-name] create a new branch at the current commit.

3. git checkout switch to another branch and check it out into your working directory.

4. git merge [branch] merge the specified branch’s history into the current one.

5. git log show all commits in the current branch’s history

****

****

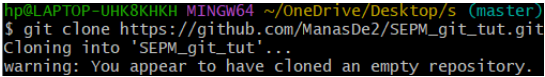
SHARE & UPDATE

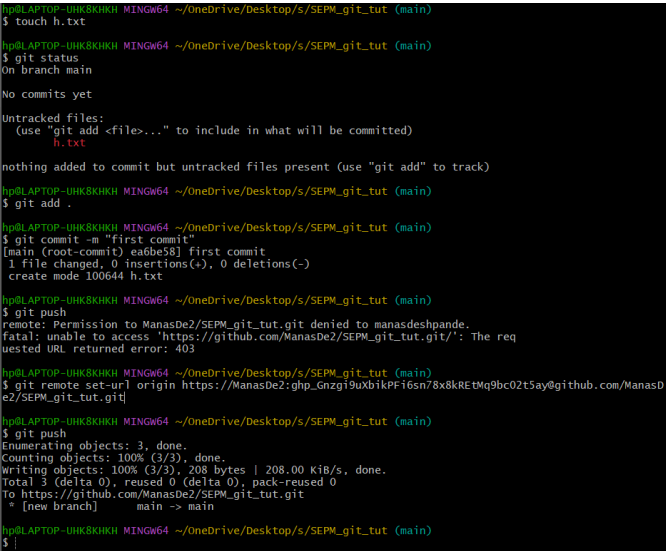
Retrieving updates from another repository and updating local repos

1. git remote add [alias] [url] :- add a git URL as an alias

2.git push [alias] [branch] :-Transmit local branch commits to the remote repository

branch

****

****

**Results and Discussions :** Thus we have successfully performed various Git operations on local and remote repositories using Git cheatsheet