ABES Engineering College
Estd. 2000
College Code-032

GIT AND GIT HUB

Copyright © 2021, ABES Engineering College

General Guideline

© (2024) ABES Engineering College.

This document contains valuable confidential and proprietary information of ABESEC. Such confidential and proprietary information includes, amongst others, proprietary intellectual property which can be legally protected and commercialized. Such information is furnished herein for training purposes only. Except with the express prior written permission of ABESEC, this document and the information contained herein may not be published, disclosed, or used for any other purpose.

Table of Content

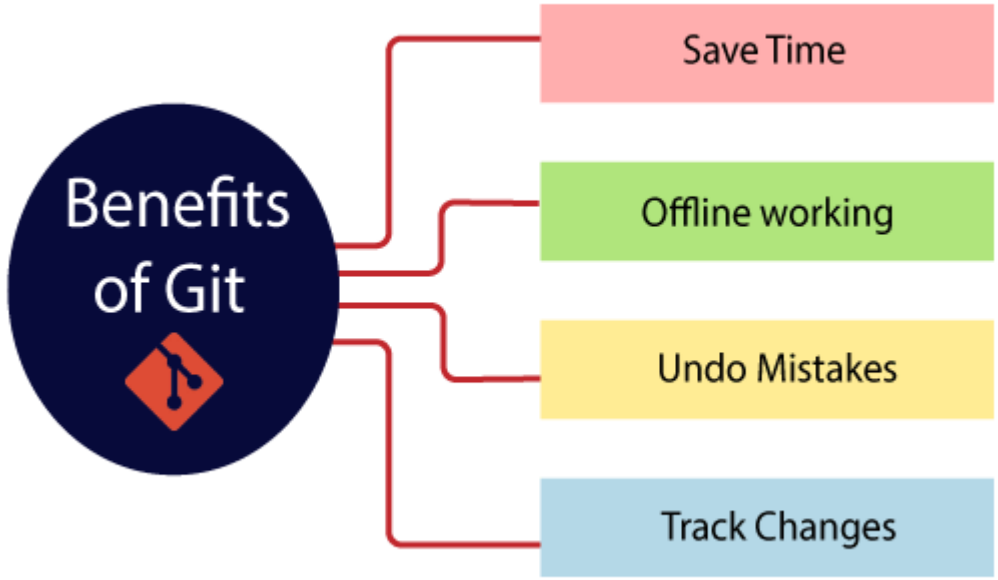
- Git Introduction
- Git Hub Introduction
- Git Setup
- Git Branching
- Git Stashing
- Git Merging

About Git

- git is a distributed version control system
- git was developed by the linux community in 2005 to help manage the development of linux itself
- Git is foundation of many services like **GitHub** and **GitLab**, but we can use Git without using any other Git services.
- Git can be used **privately** and **publicly**

Copyright © 2021, ABES Engineering College

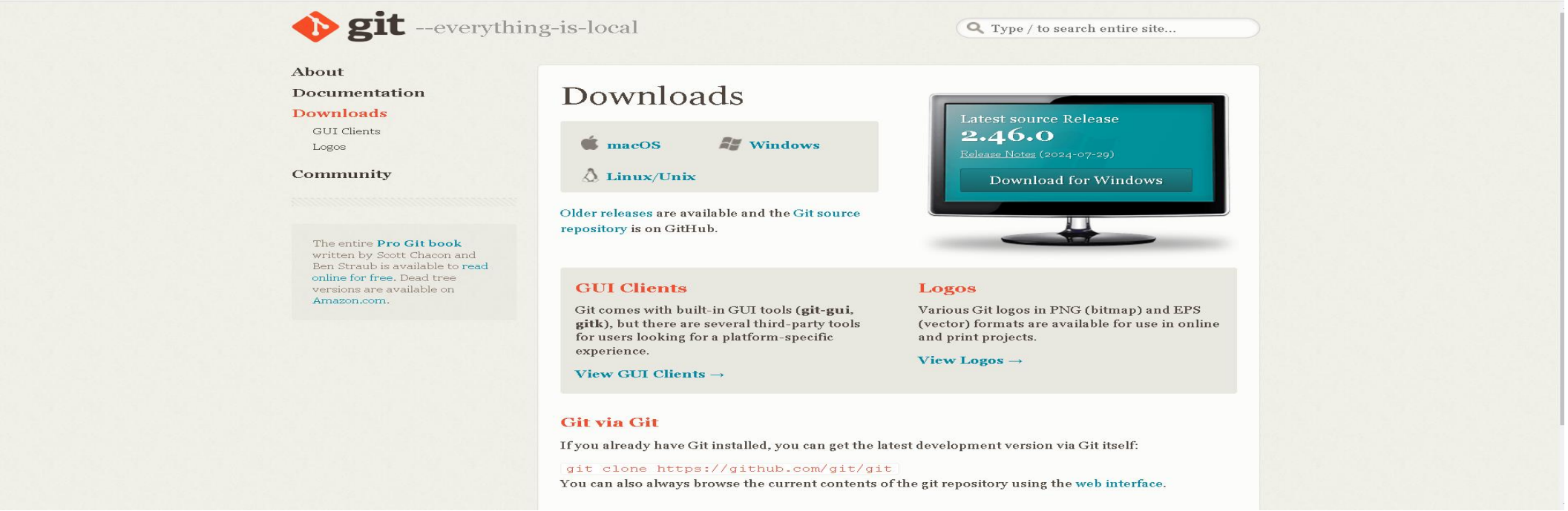
Benefits of Git



Copyright © 2021, ABES Engineering College

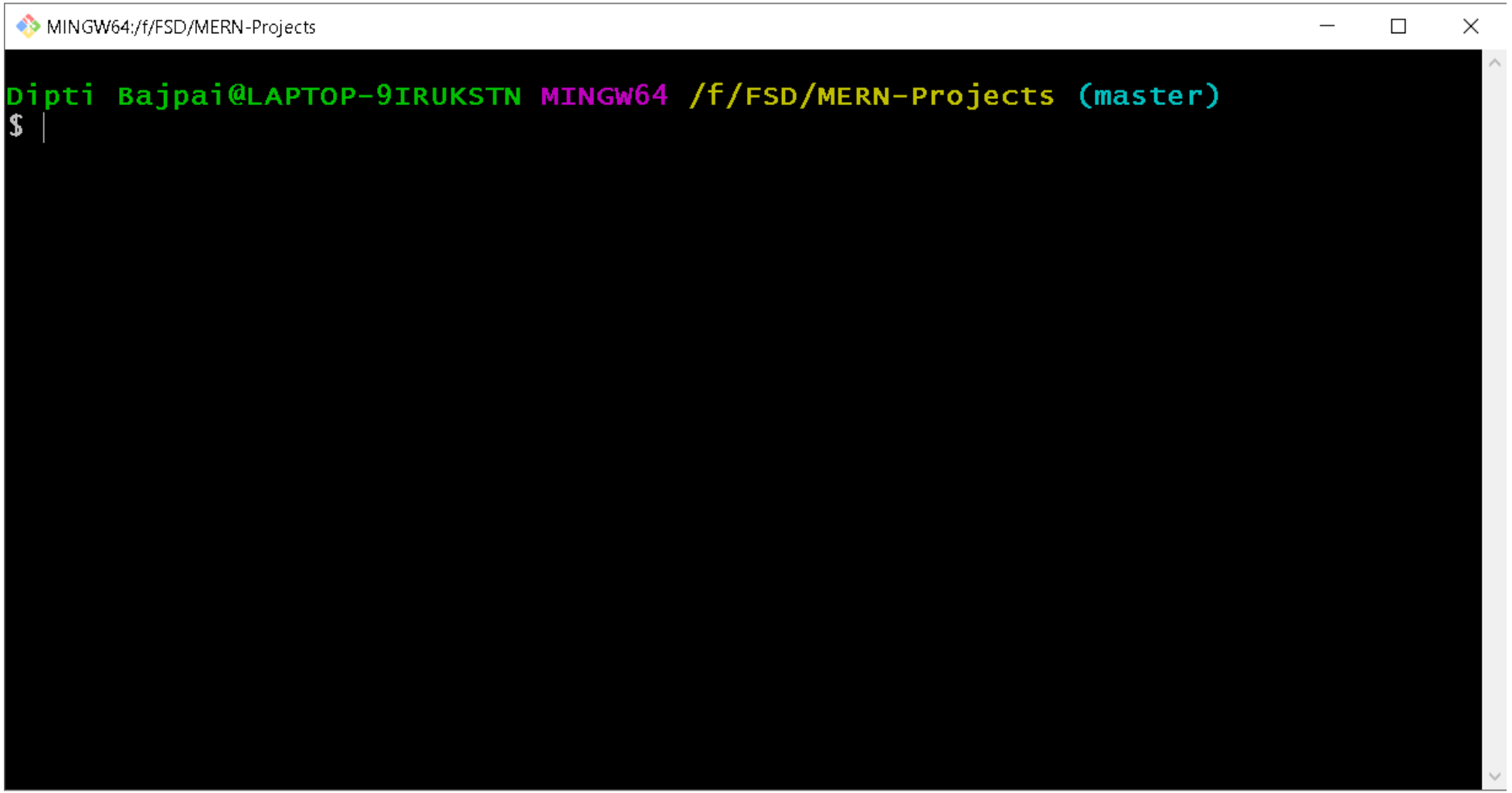
Download Git

- Download git from https://git-scm.com
- On Mac OS X: first install homebrew: Details at http://brew.sh



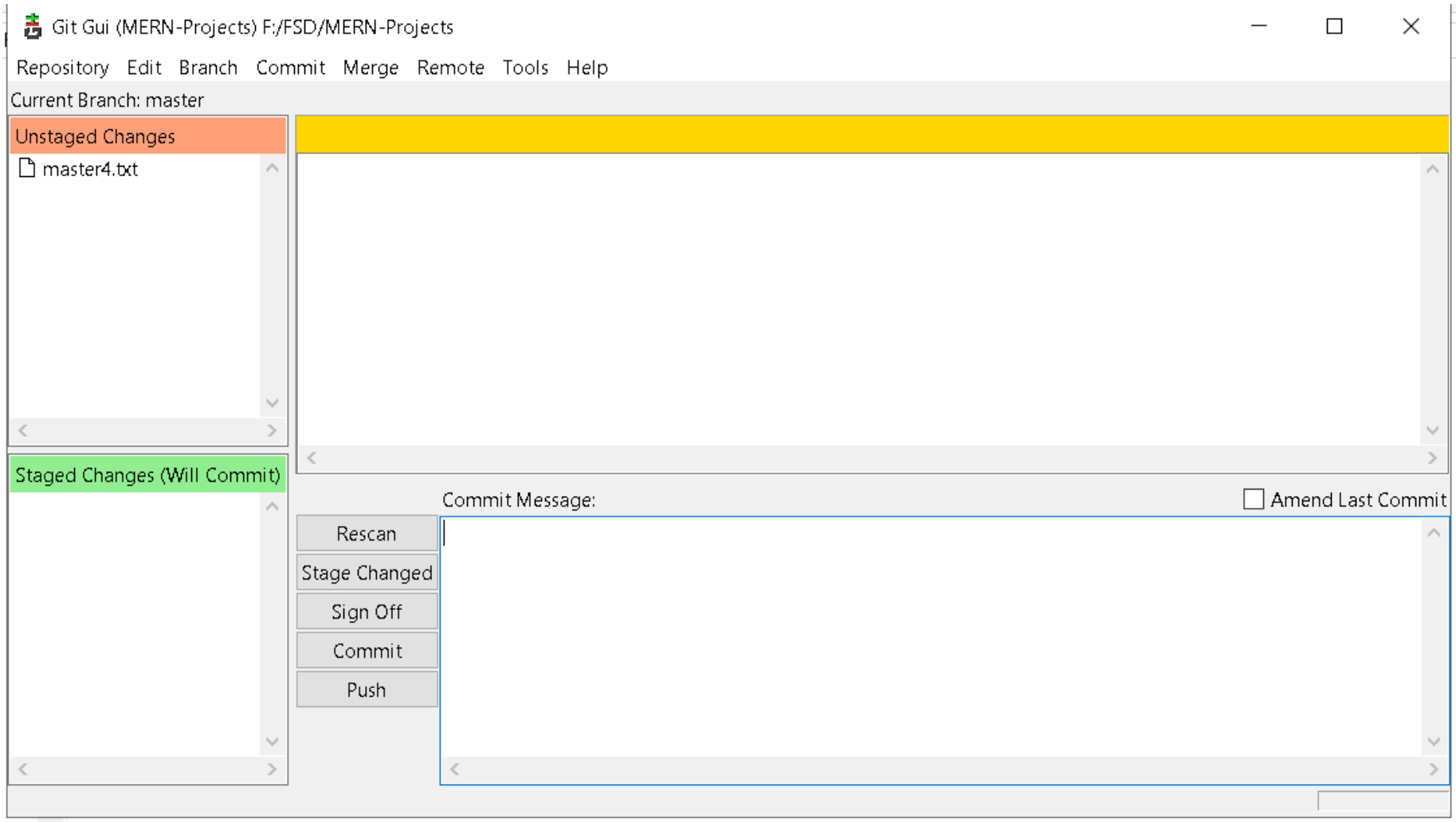
Copyright © 2021, ABES Engineering College

Git bash



Copyright © 2021, ABES Engineering College

Git gui



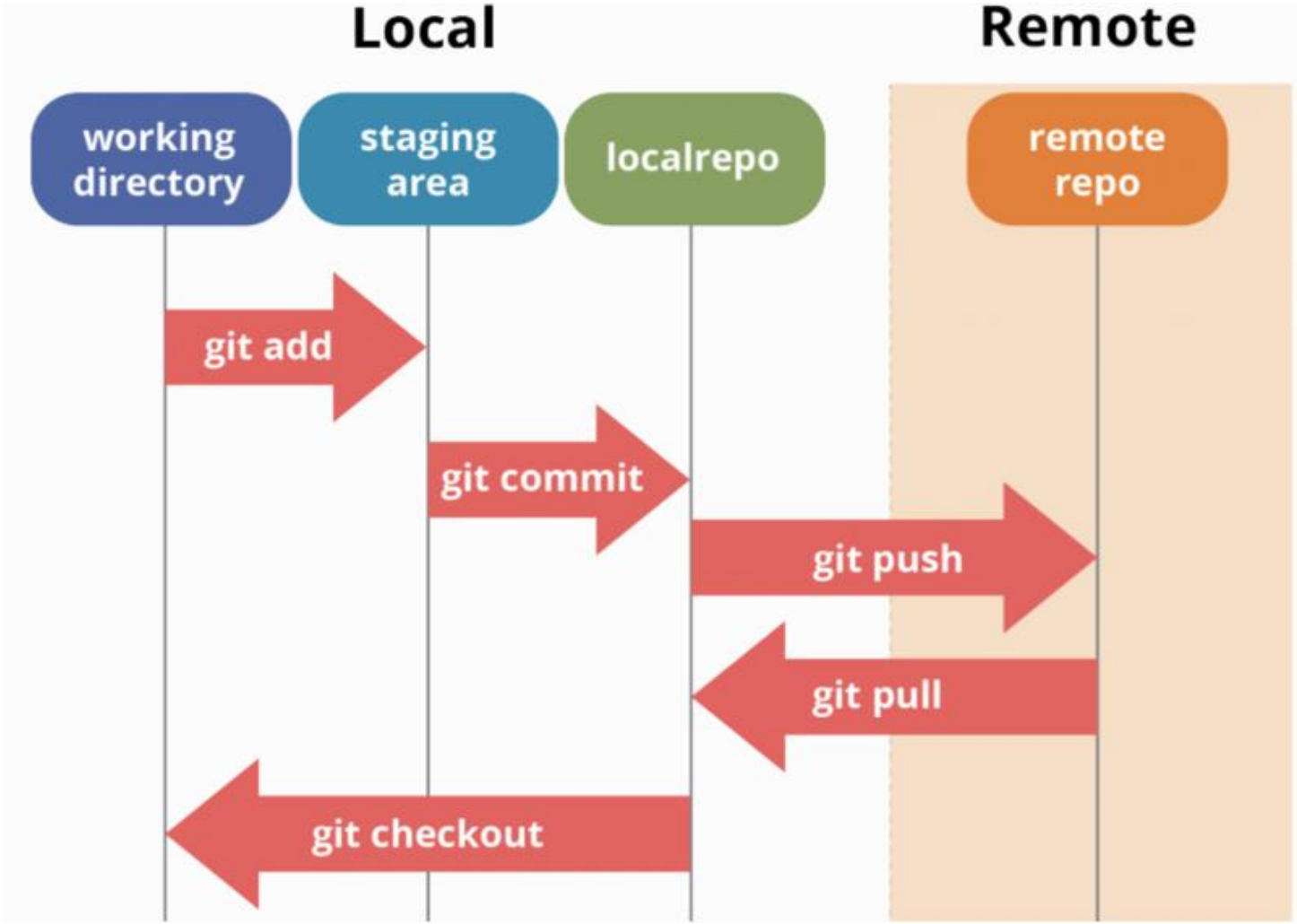
Copyright © 2021, ABES Engineering College

Git Hub

- GitHub is a Git repository hosting service.
- It offers both distributed version control and source code management (SCM) functionality of Git.
- GitHub also facilitates with many of its features, such as access control and collaboration.
- It provides a Web-based graphical interface.

Copyright © 2021, ABES Engineering College

How git works



Copyright © 2021, ABES Engineering College

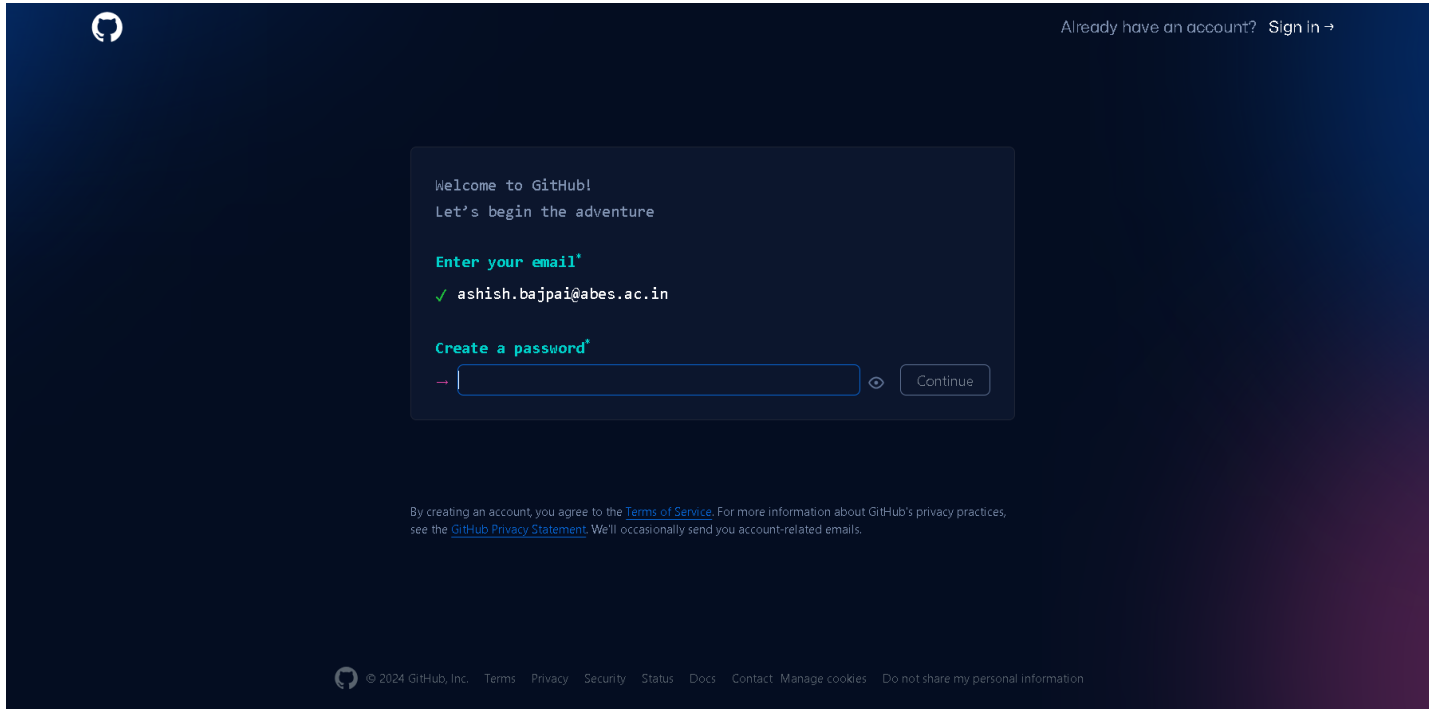
Git Hub Features

- Collaboration
- Integrated issue and bug tracking
- Graphical representation of branches
- Git repositories hosting
- Project management
- Team management
- Code hosting
- Track and assign tasks

Copyright © 2021, ABES Engineering College

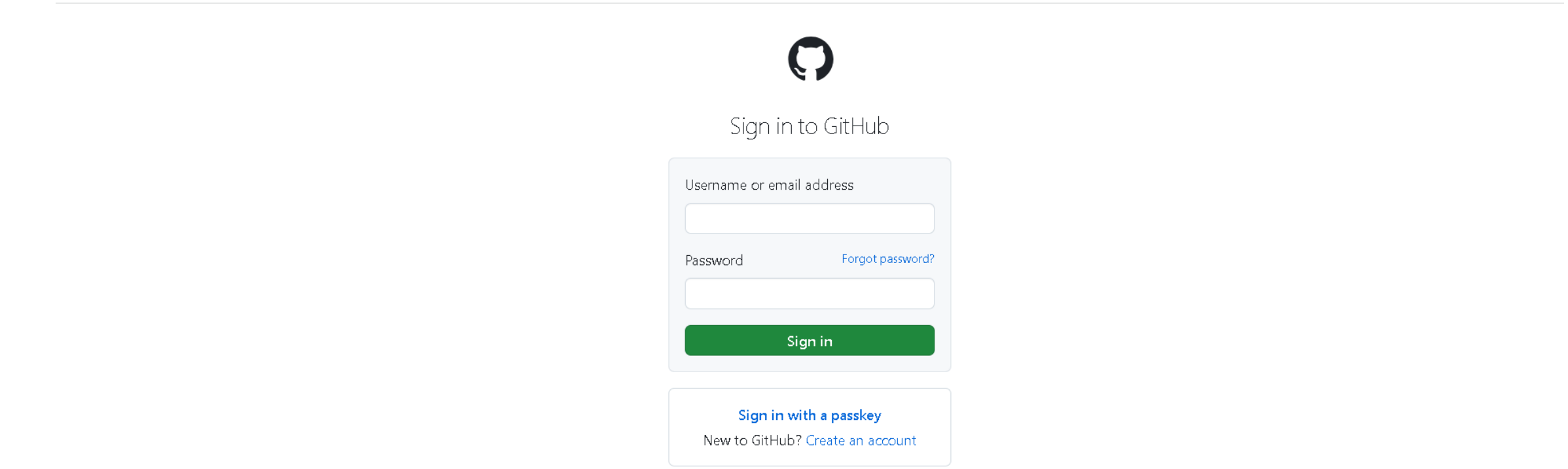
Steps to create GitHub Account

- Step-1Navigate to https://github.com/.
- Step-2 Click signup
- Step-3 verify your email address



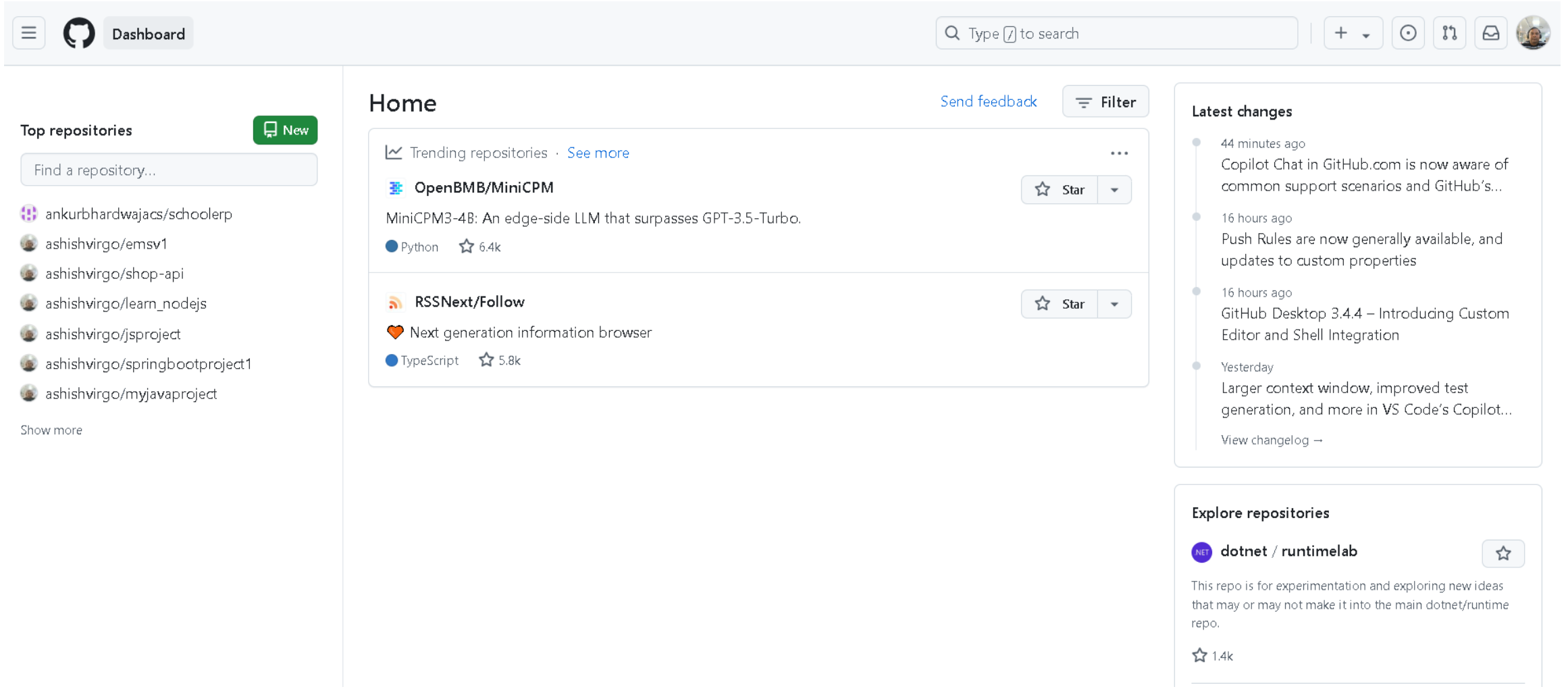
Copyright © 2021, ABES Engineering College

Git Hub Account



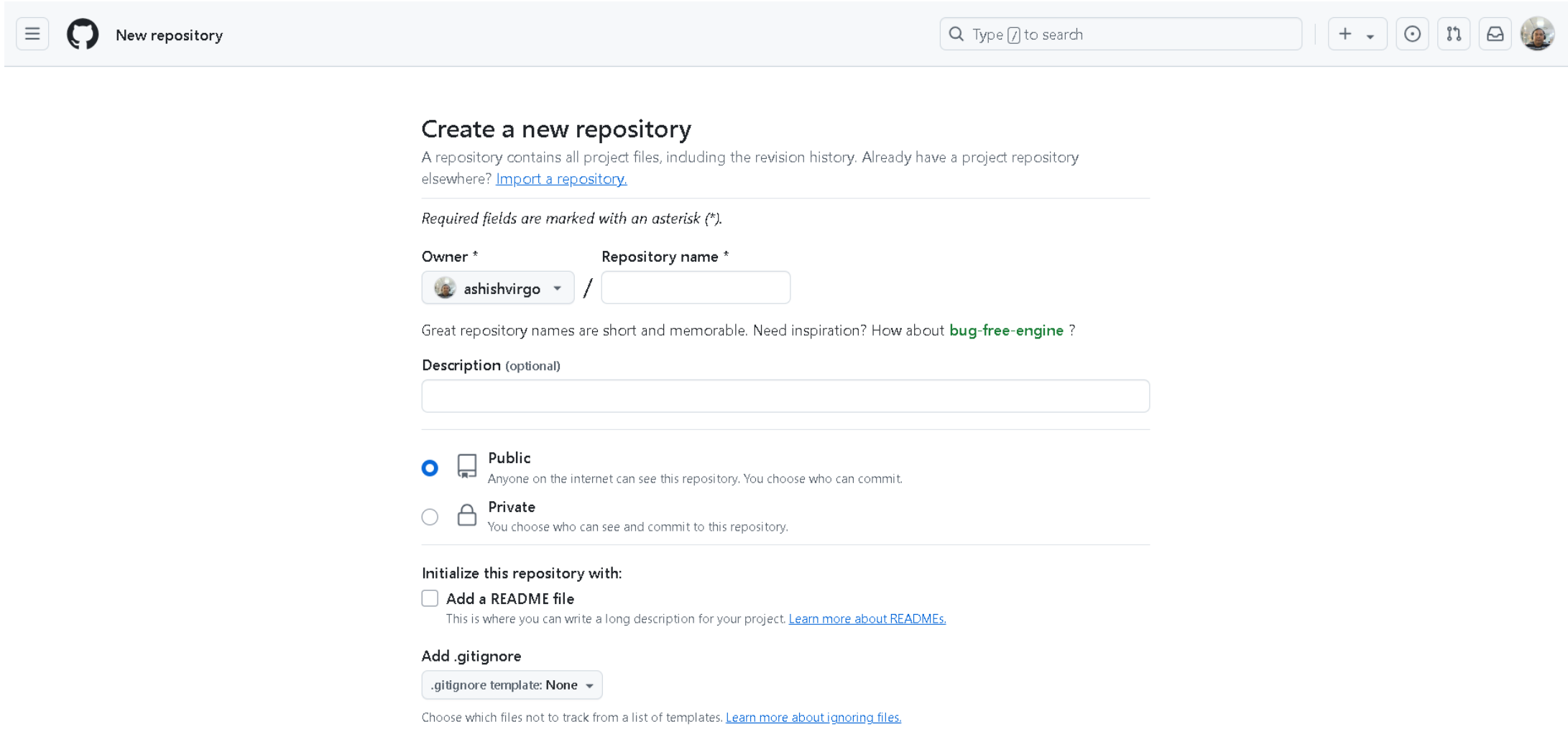
Copyright © 2021, ABES Engineering College

Git Hub Repositories



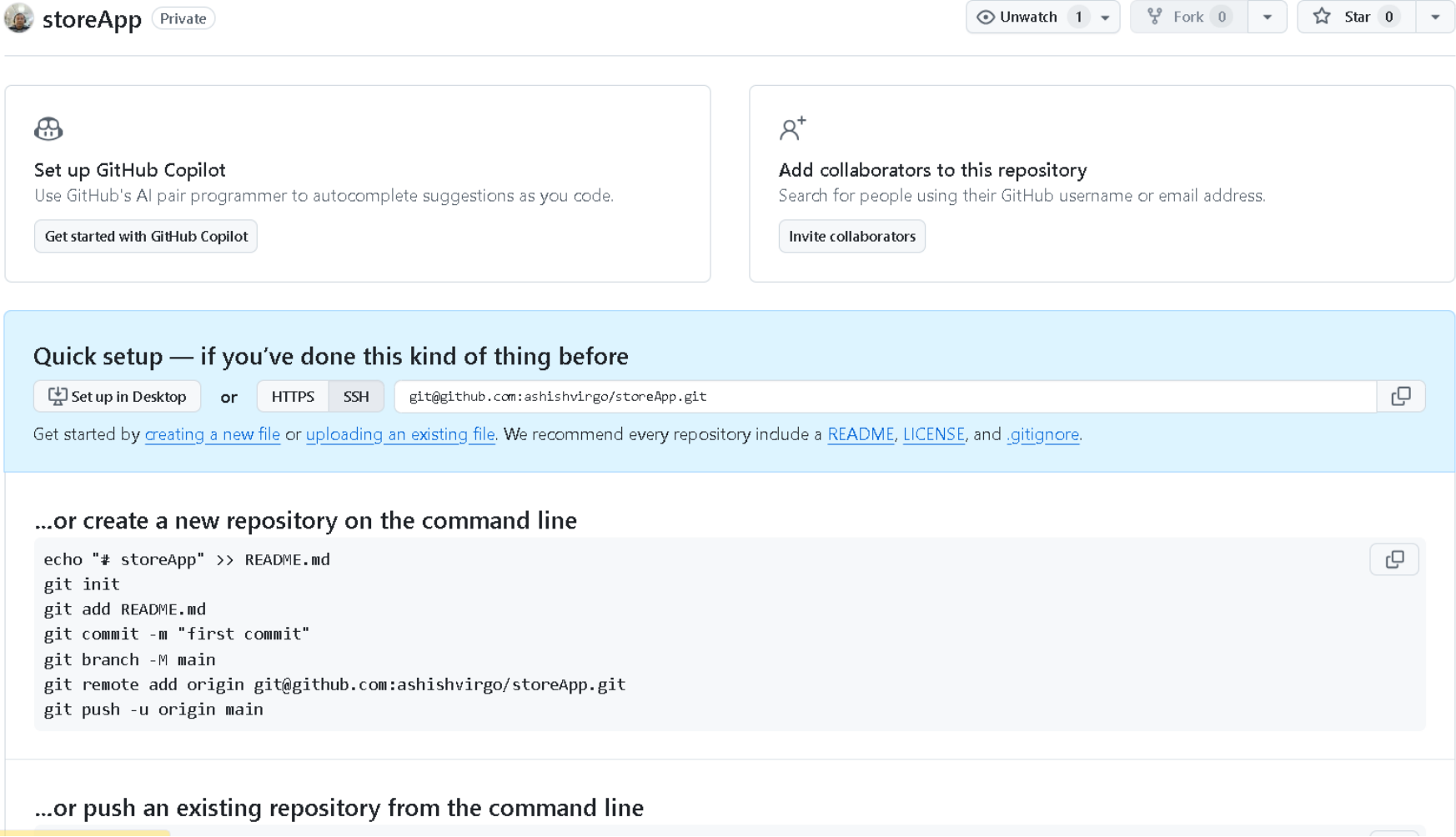
Copyright © 2021, ABES Engineering College

Create New Repository



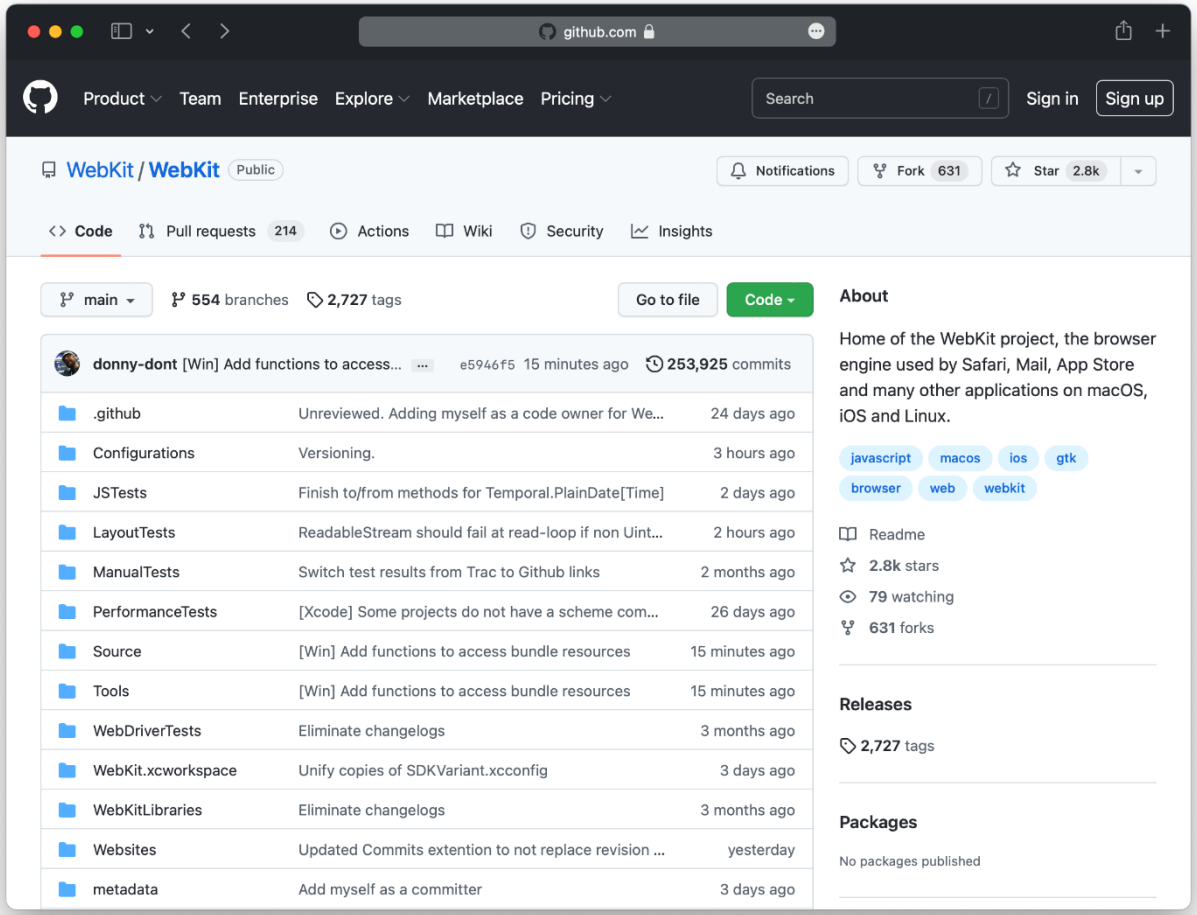
Copyright © 2021, ABES Engineering College

Add collaborators



Copyright © 2021, ABES Engineering College

Repository files



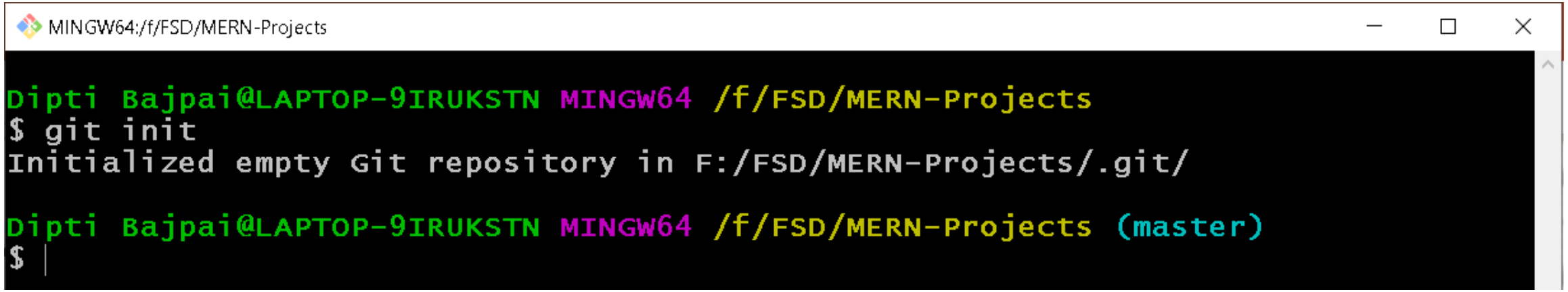
Copyright © 2021, ABES Engineering College

Git vs GitHub

1. It is a software	1. It is a service
2. It is installed locally on the system	2. It is hosted on Web
3. It is a command line tool	3. It provides a graphical interface
4. It is a tool to manage different versions of edits, made to files in a git repository	4. It is a space to upload a copy of the **Git** repository
5. It provides functionalities like Version Control System Source Code Management	5. It provides functionalities of Git like VCS, Source Code Management as well as adding few of its own features

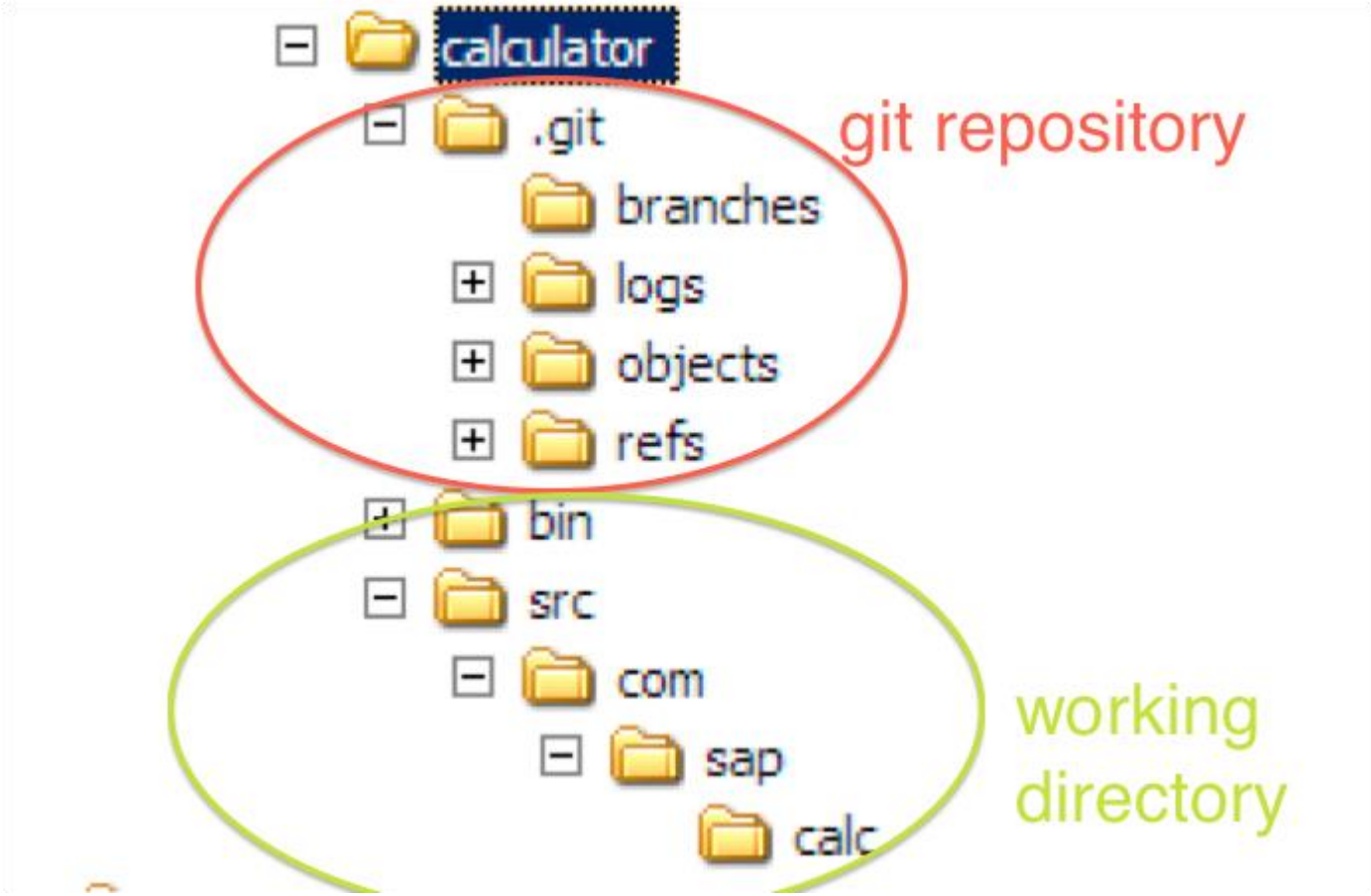
Copyright © 2021, ABES Engineering College

Git Initialized



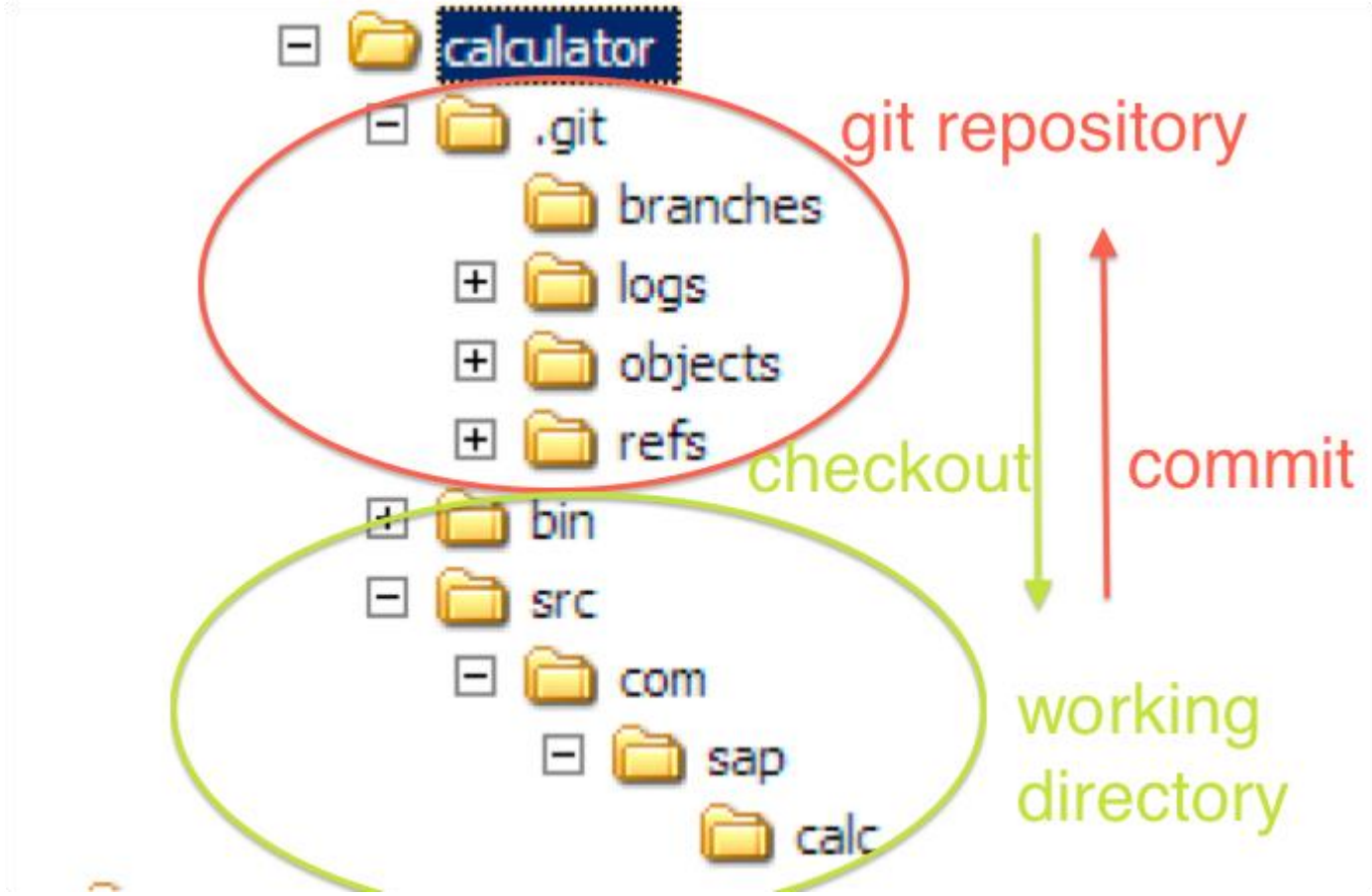
Copyright © 2021, ABES Engineering College

Git Concept: Storage



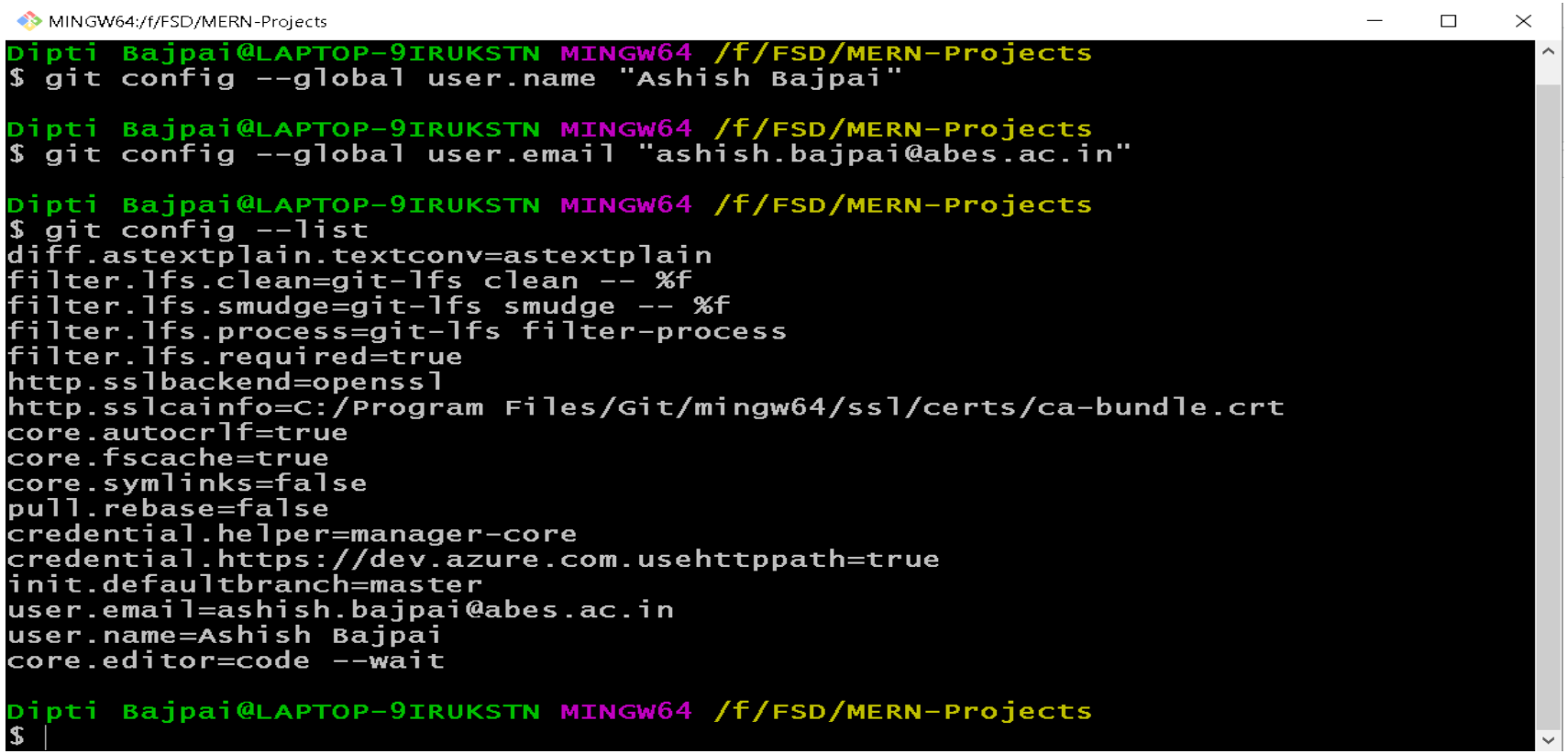
Copyright © 2021, ABES Engineering College

Git Concept: Storage



Copyright © 2021, ABES Engineering College

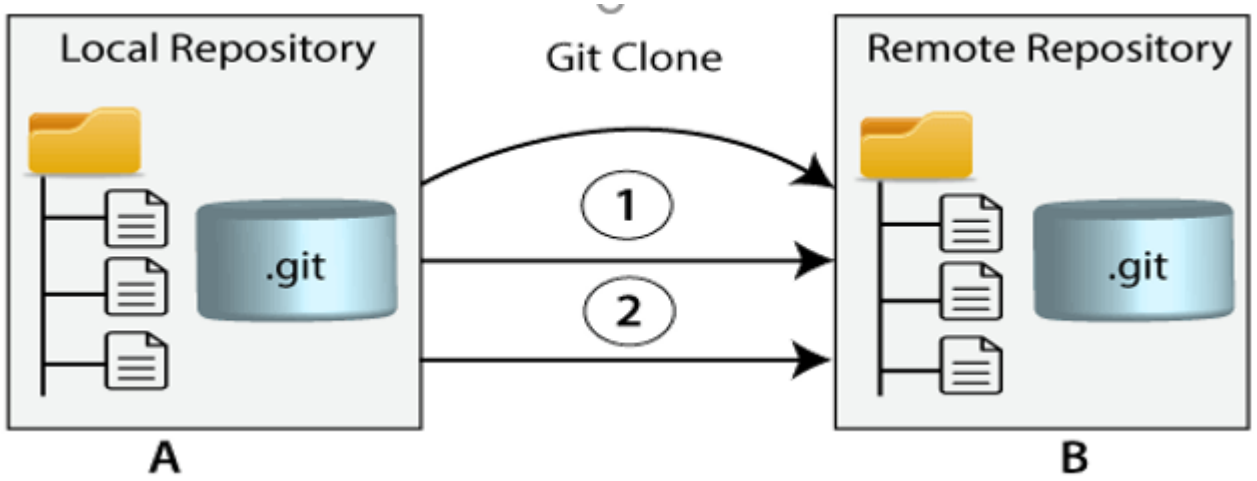
Git Setup



Copyright © 2021, ABES Engineering College

Git Push

- Git push origin <branch_name>



Git Remote

git remote add origin <repo_link>

```
$ git remote add origin git@github.com:ashishvirgo/mernApp.git
fatal: not a git repository (or any of the parent directories): .git

Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-PROJECTS
$ cd MernApp/

Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-PROJECTS/MernApp
$ git init
Initialized empty Git repository in F:/FSD/MERN-PROJECTS/MernApp/.git/

Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-PROJECTS/MernApp (master)
$ git remote add origin git@github.com:ashishvirgo/mernApp.git

Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-PROJECTS/MernApp (master)
$ git remote
origin

Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-PROJECTS/MernApp (master)
$ git remote -v
origin  git@github.com:ashishvirgo/mernApp.git (fetch)
origin  git@github.com:ashishvirgo/mernApp.git (push)
```

Copyright © 2021, ABES Engineering College

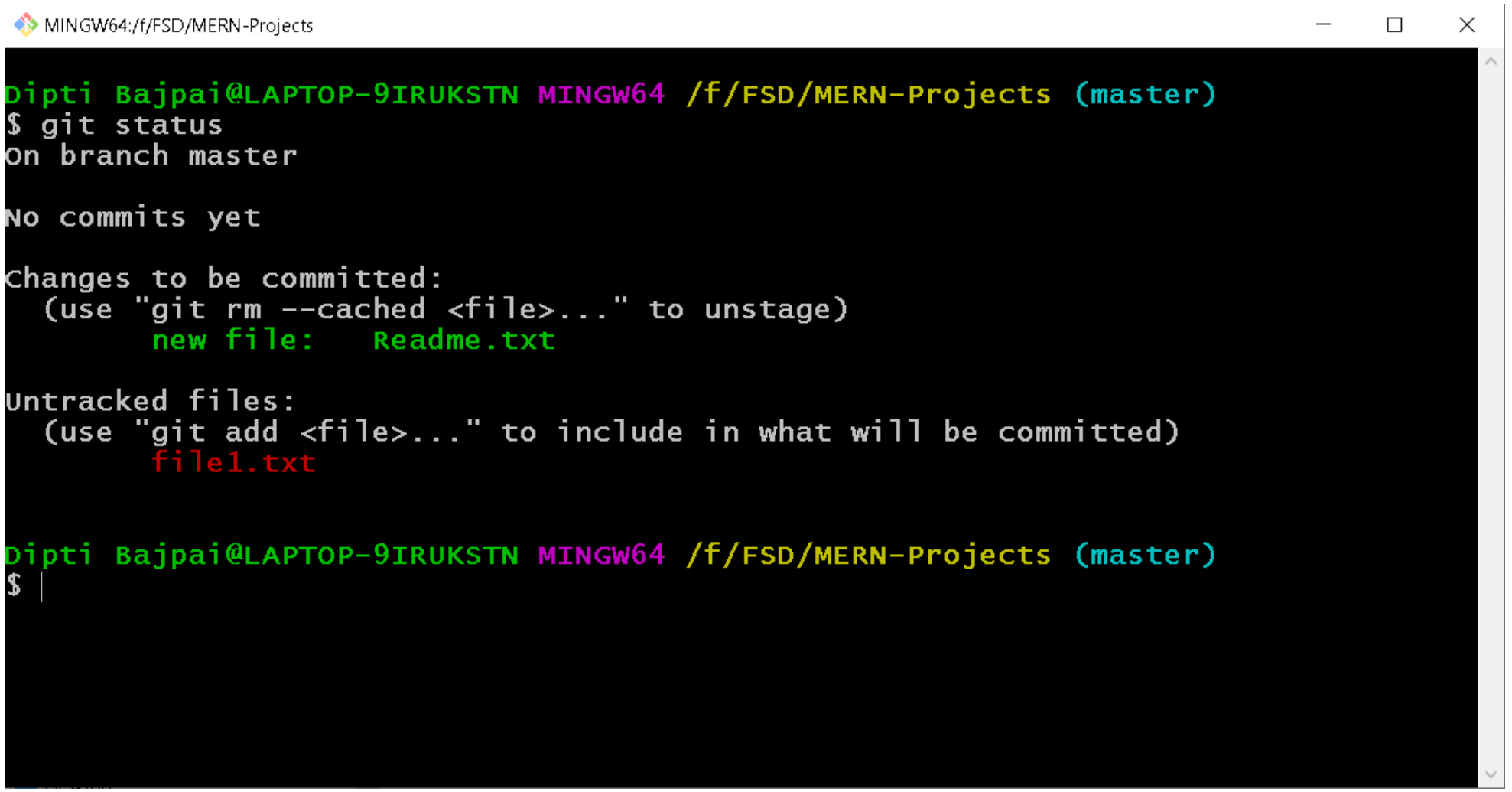
Git Pull

- Pull is a combination of two different commands
  fetch and merge
  git pull origin <branch_name>

  Git Fetch
  fetch gets all the change history of a tracked branch/repo.
  on your local Git, fetch updates to see what has changed on GitHub

Copyright © 2021, ABES Engineering College

Git Status



Copyright © 2021, ABES Engineering College

Add file from work are to staging area

```
Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (master)
$ git add file1.txt

Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Readme.txt
        new file:   file1.txt


Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (master)
$
```

Copyright © 2021, ABES Engineering College

Git commit and log

```
Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (master)
$ git commit -m "my first commit"
[master (root-commit) bd5f57c] my first commit
 2 files changed, 1 insertion(+)
 create mode 100644 Readme.txt
 create mode 100644 file1.txt

Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (master)
$ git log
commit bd5f57c7979ce66c2717834eec225ec1f7c255bb (HEAD -> master)
Author: Ashish Bajpai <ashish.bajpai@abes.ac.in>
Date:   Tue Sep 10 09:46:17 2024 +0530

    my first commit
```

Copyright © 2021, ABES Engineering College

Git Branching

View branch list

git branch

Create New Branch

git branch <new_branch_name>

Switch Branch

git checkout <old_branch_name>

Create a Branch and Switch to it

git checkout –b <new_branch_name>

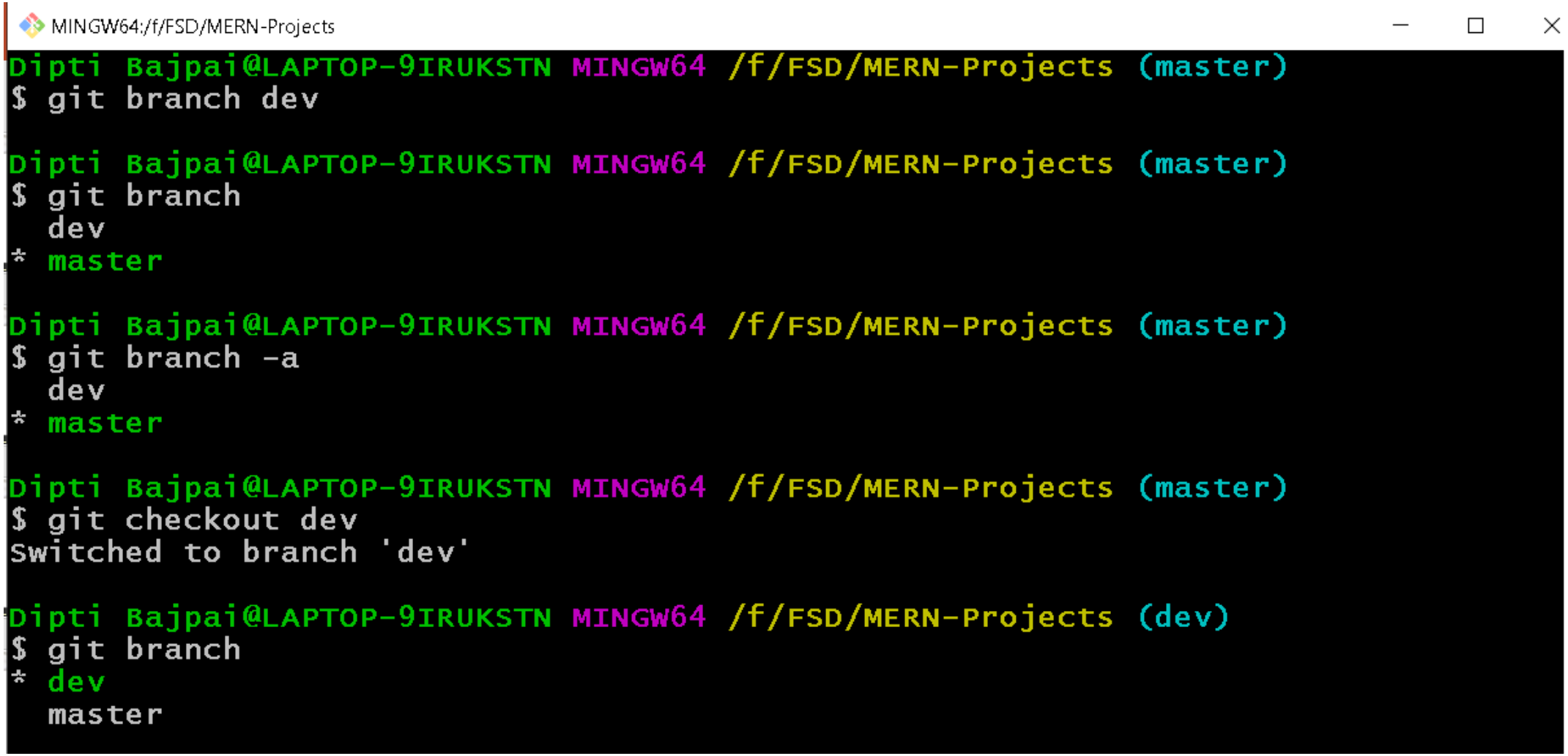
Delete Branch

git branch –d <branch_name>(if it has already been fully merged in its upstream branch)

git branch –D <branch_name>

Copyright © 2021, ABES Engineering College

Git Branching(cont ...)



Copyright © 2021, ABES Engineering College

Delete branch

```
Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (dev)
$ git branch
* dev
  master

Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (dev)
$ git branch -d dev
error: Cannot delete branch 'dev' checked out at 'F:/FSD/MERN-Projects'

Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (dev)
$ git checkout master
Switched to branch 'master'

Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (master)
$ git branch -d dev
Deleted branch dev (was bd5f57c).

Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (master)
$ git branch
* master
```

Copyright © 2021, ABES Engineering College

Git merging

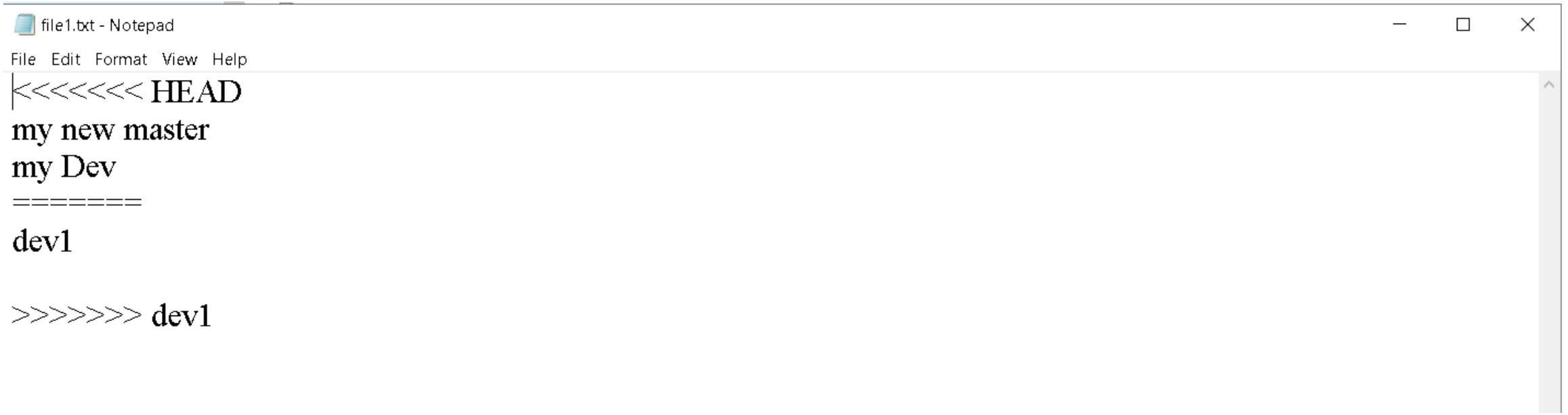
- Git merge <branch_name>

After merge push code on remote repo

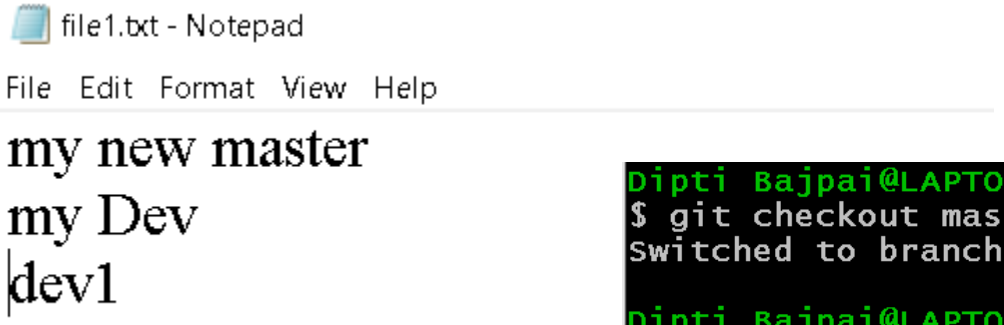
```
Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (master)
$ git merge dev1
Merge made by the 'ort' strategy.
 dev1_file1.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 dev1_file1.txt
```

Copyright © 2021, ABES Engineering College

Conflict


file1.txt - Notepad
File Edit Format View Help
<<<<<<< HEAD
my new master
my Dev
=======
dev1
>>>>>>> dev1

Resolve conflict manually




```
Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (dev1)
$ git checkout master
Switched to branch 'master'

Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (master)
$ git merge dev1
Auto-merging file1.txt
CONFLICT (content): Merge conflict in file1.txt
Automatic merge failed; fix conflicts and then commit the result.

Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (master|MERGING)
$ git add file1.txt

Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (master|MERGING)
$ git commit -m "updated file1.txt"
[master d6bc9d4] updated file1.txt

Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (master)
$
```

Copyright © 2021, ABES Engineering College

Git Stashing

git stash

git stash list

git stash pop

git stash apply stash@{n}

```
Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (dev)
$ git stash
Saved working directory and index state WIP on dev: 1ca2162 added two new files

Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (dev)
$ git status
On branch dev
nothing to commit, working tree clean

Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (dev)
$ git stash list
stash@{0}: WIP on dev: 1ca2162 added two new files

Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (dev)
$ git stash apply stash@{0}
On branch dev
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   dev3.txt
```

Copyright © 2021, ABES Engineering College

Discard changes in working directory

```
Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    Readme.txt
        deleted:    dev1.txt
        deleted:    dev2.txt
        deleted:    dev3.txt
        modified:   file1.txt
        deleted:    master1.txt
        deleted:    master2.txt

no changes added to commit (use "git add" and/or "git commit -a")

Dipti Bajpai@LAPTOP-9IRUKSTN MINGW64 /f/FSD/MERN-Projects (master)
$ git restore .
```

Copyright © 2021, ABES Engineering College

Reference

GitHub Docs(https://docs.github.com/en/get-started)

Thank You

Copyright © 2021, ABES Engineering College