Subject Name: **Source Code Management**

Session: **2022-23**

Department: **DCSE**

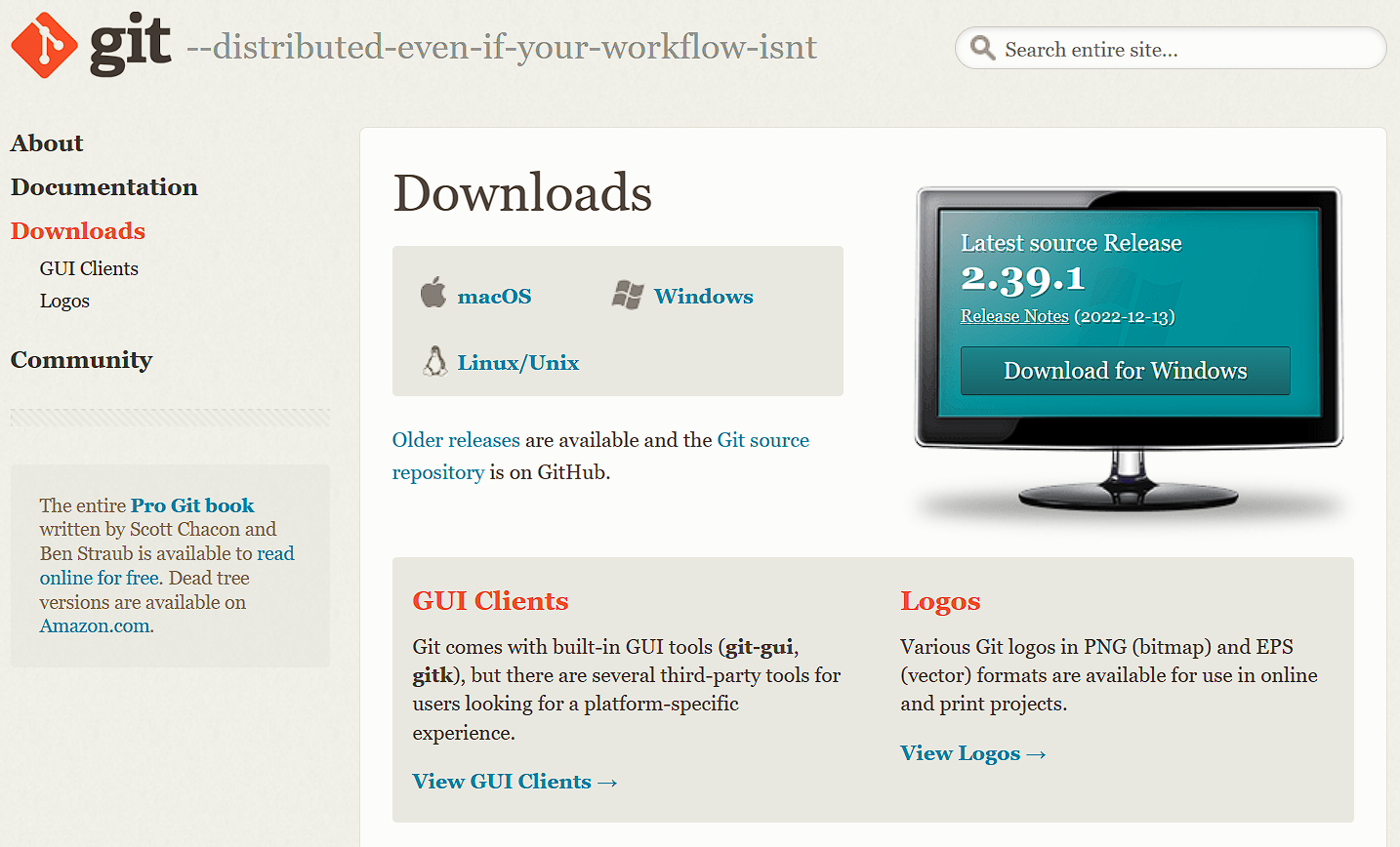
****

|  |  |  |
| --- | --- | --- |
| **Submitted By:**  Ishaan Singla  2210992582  G8-A |  | **Submitted To:**  Dr. Neha Sharma |

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Program Title** | **Page No.** | **Remarks** |
| 1 | Setting up of Git Client | 1 |  |
| 2 | Setting up GitHub Account | 5 |  |
| 3 | Generate logs | 6 |  |
| 4 | Create and visualizing branches | 7 |  |
| 5 | Git life cycle description | 8 |  |

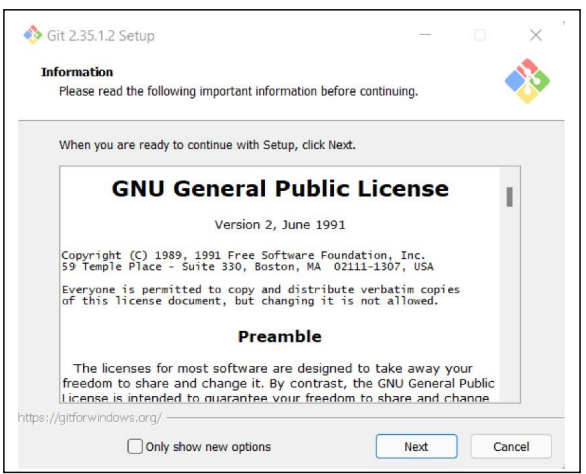
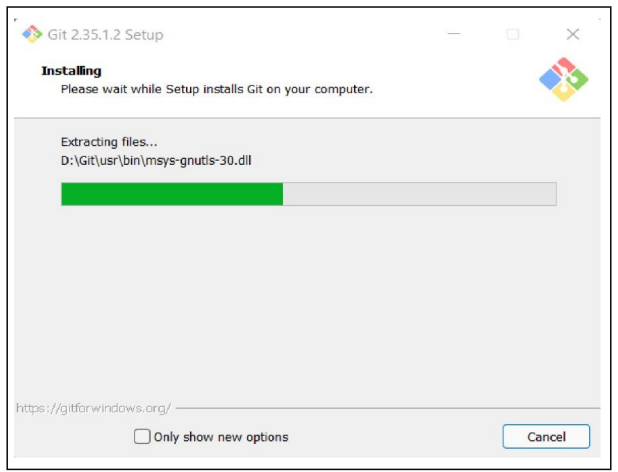
# Practical 1 Setting of Git client

* First go to <https://git-scm.com/downloads>
* After clicking downloads choose your OS  
  
* After selecting os follow the instructions and download the file based on your system type , for windows 🡪

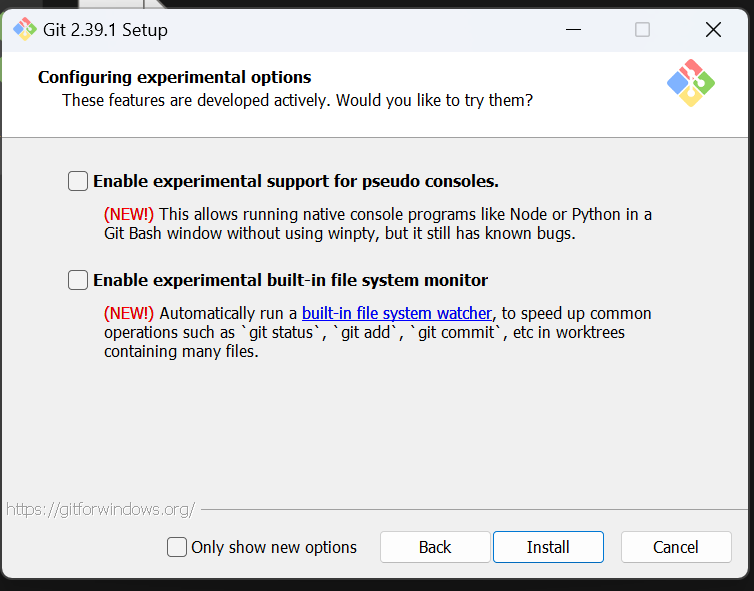


* For windows the package will be downloaded , run the file and click Next next next….. until the install button appears , there’s no need to change the settings since its already set to its best settings during installations.

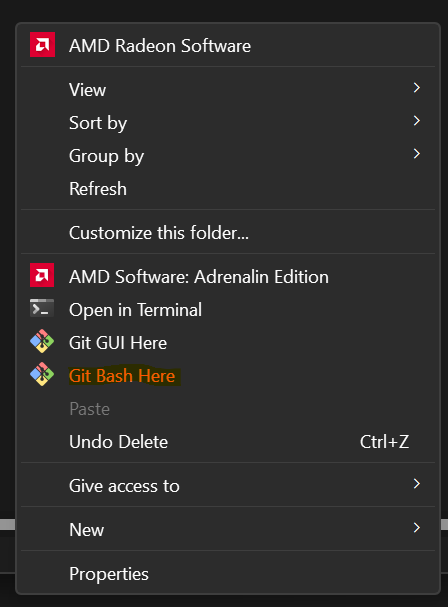
Page | 1

****

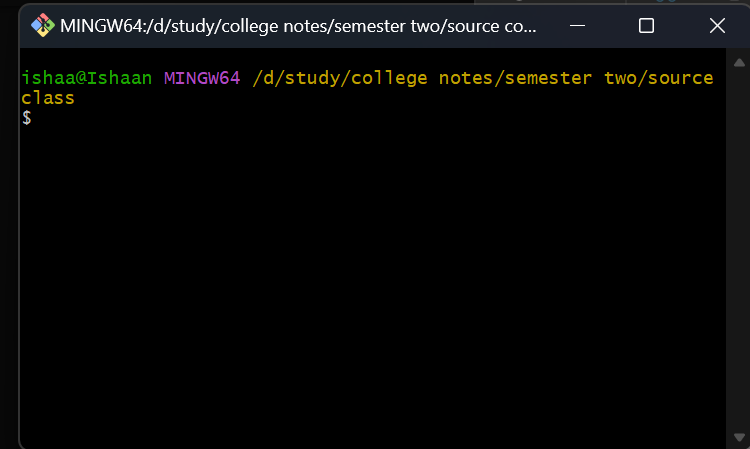
Page | 2

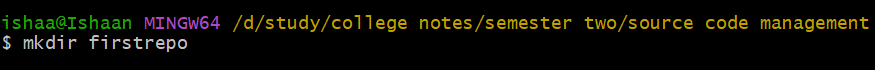


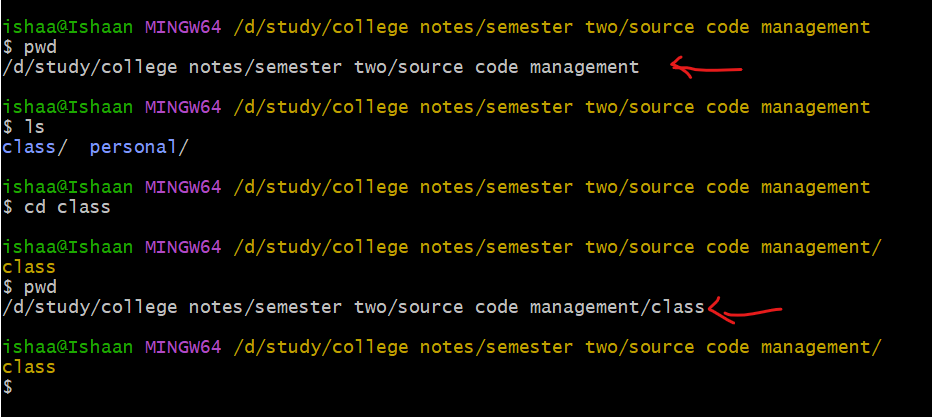
* For opening git go to any folder , right click
* Click “**Git bash here”** shown in Screenshot
* A terminal windows will be opened



Page | 3



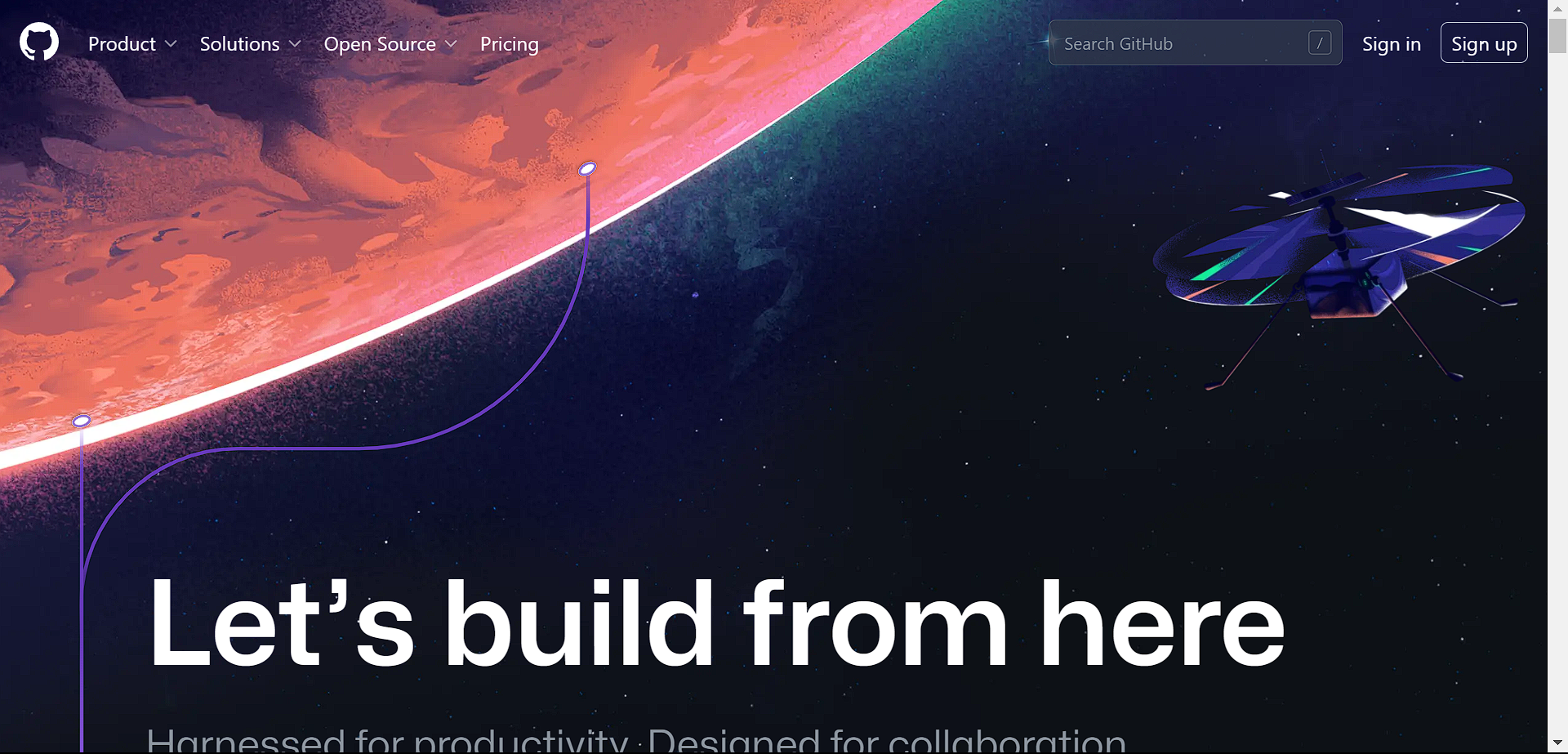
* Creating directory 🡪
  + mkdir (folder name) 🡪 it will create a folder for git repository
* **Basic Commands for Git 🡪**

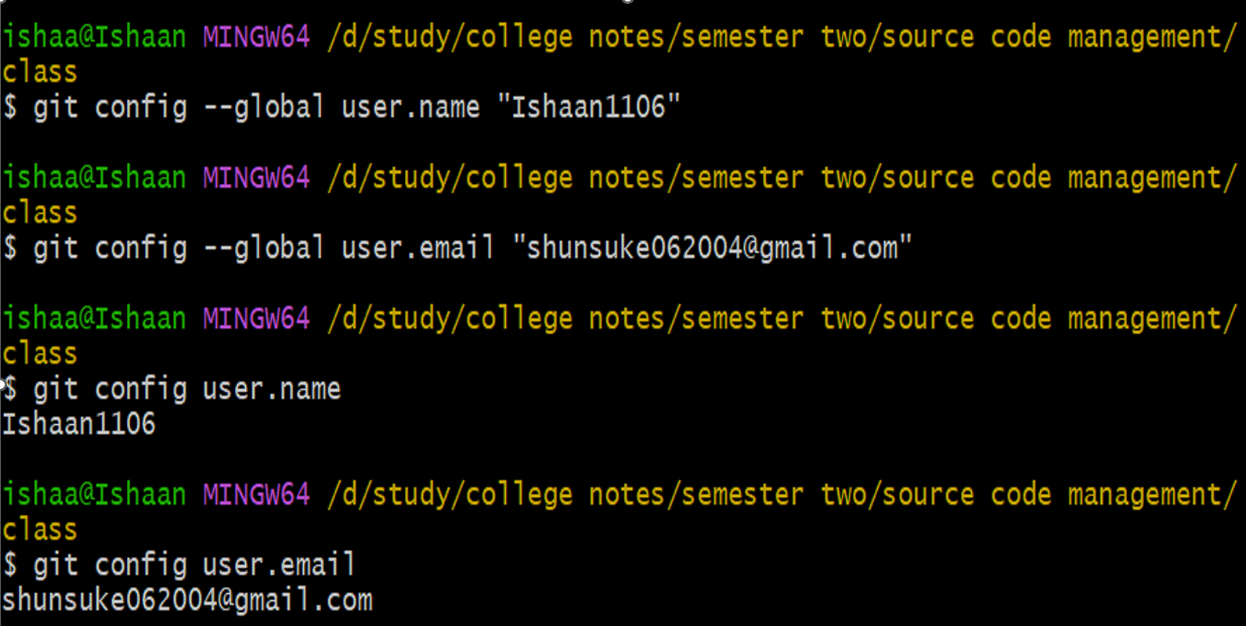
1. Pwd 🡪 Displays the current working directory in git (present working directory)
2. Cd (path) 🡪 used to jump from one path to another
3. ls 🡪 shows all the files in pwd

Page | 4

**Piratical 2**  
**Setting of GitHub account**

* + - * For setting up account for to <https://github.com/>



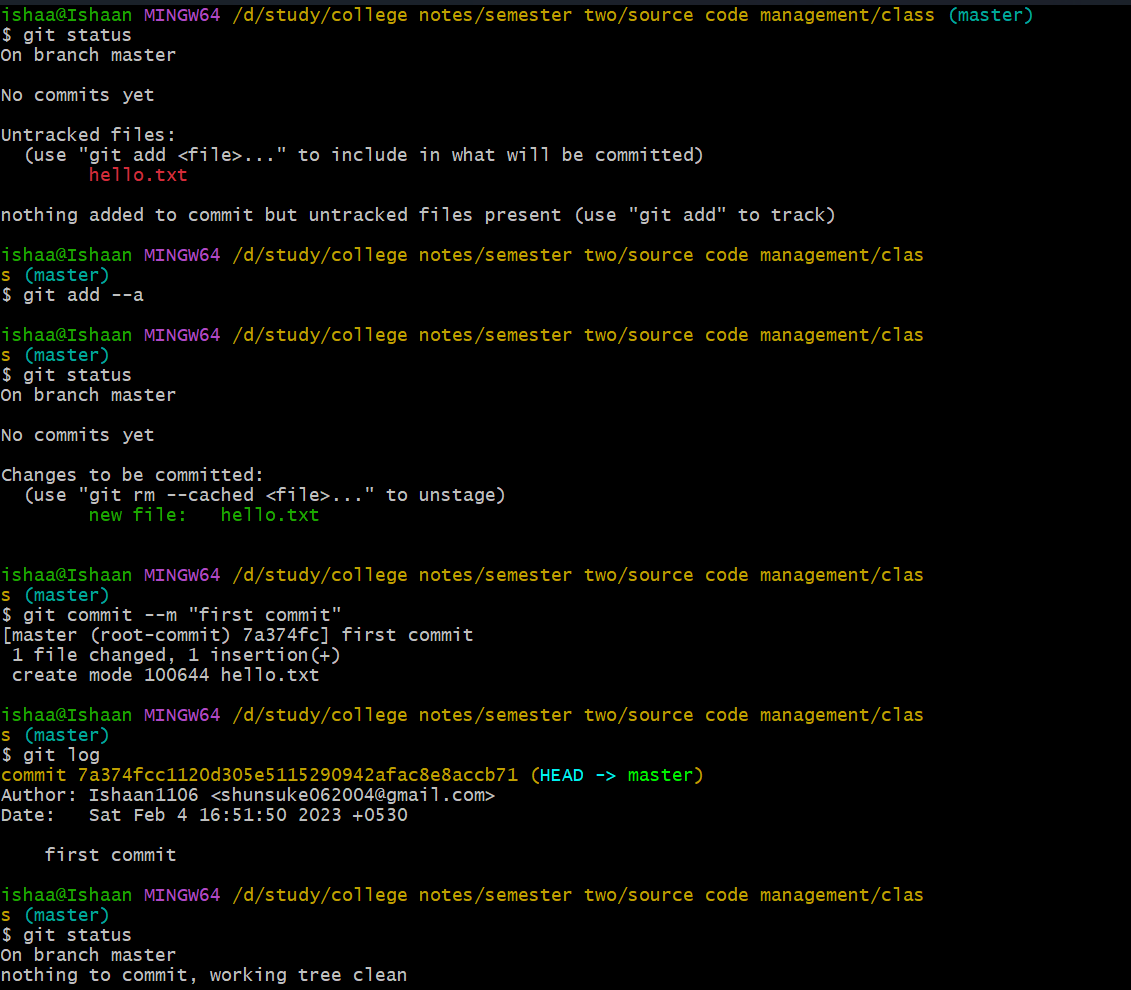
* For making new account click on “sign up” or if you already have account click “Sign in”
* When done opening github account
* Now for setting githhub account in “git bash”
  + Open git bash
  + Enter commands
    - git config --global user.name “(your githhub username)” 🡪 setting username in git
    - git config --global user.email “(your githhub email)” 🡪 setting email in git {run 1st and 2nd together }
    - git config user.name 🡪 will display the current user name
    - git config user.email 🡪 will display the current email

Page | 5

Piratical 3  **Generate logs**

* logs are the detailed list of an application information, system performance, or user activities.
* For generating logs first we need to do changes
  1. Creating a git repository (.git file) , enter **git init** in git bash.
  2. **touch** (file name) 🡪 for creating a file in present working directory. (ex :- touch hello.txt)
  3. open the file and write something in it or edit however you like
  4. git status 🡪 shows if files are in the staging area or not

(red means files aren’t , green means files are [but not commited yet ])

* 1. git add --a 🡪 add all the files in staging area (in .git folder)
  2. now run “git status” and it will be green now
  3. git commit --m “(enter something)” 🡪 snap shot of git tree and contents of the file in the repository  
     **(when files are committed git status will display 🡪** On branch master nothing to commit, working tree clean since there’s nothing left to commit
  4. git log 🡪displays when and by who the file was changed after being committed

Page | 6

Piratical | 4

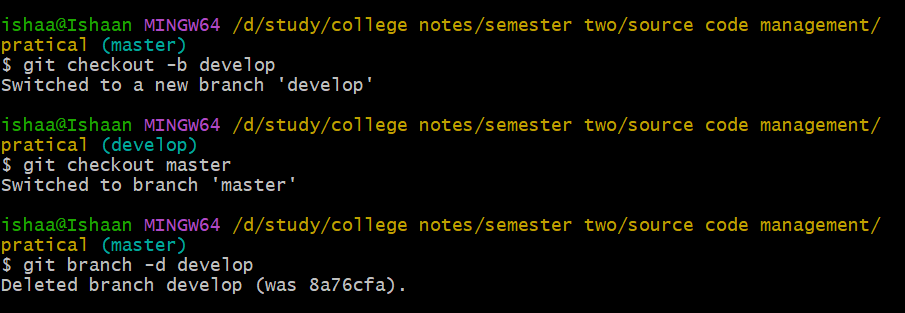
Creates & visualizing branches

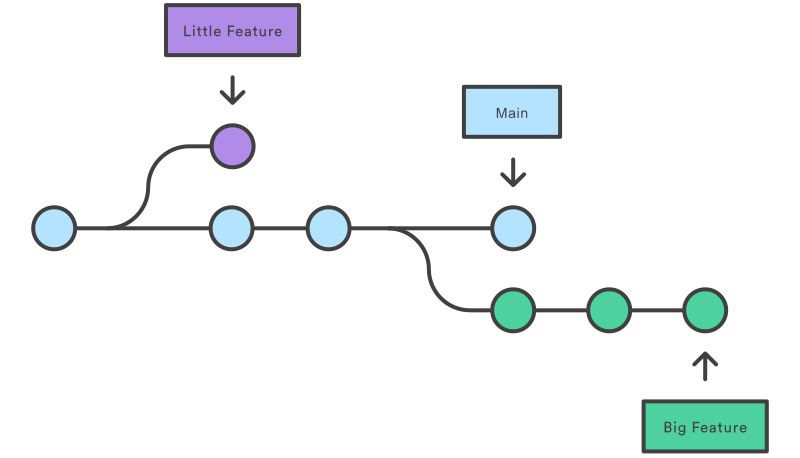
A branch is a new/separate version of the main repository.

The main branch is called the master branch. We can make branches

out of the main branch

commands for creating branch *🡪*

* For creating new branch use 🡪 “git checkout -b (branch-name)“
* For switching branch 🡪 “git checkout (branch names)”
* For deleting branch 🡪 “git branch -d (branch name)”
* An imagine visualising branches in git



Page | 7

Piratical 5

Git lifecycle description

Files in a git project have various stages like creation , modification

Refactoring and deletion and so on. When project is under git

version control system, they are present in tree major Git states

**Stages in GIT Life Cycle 🡪**

Files in a Git project have various stages like Creation, Modification, Refactoring, and Deletion and so on. Irrespective of whether this project is tracked by Git or not, these phases are still prevalent. However when a project is under Git version control system, they are present in three major Git states in addition to these basic ones. Here are the three Git states:

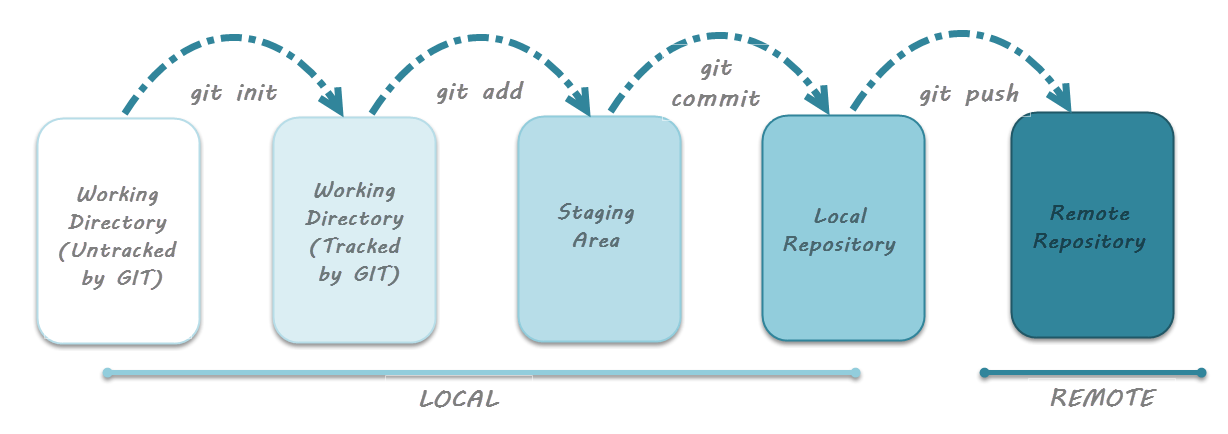
* 1. Working directory
  2. Staging area
  3. Git directory

1. ***WORKING DIRECTORY 🡪***  
   Consider a project residing in your local system. This project may or may not be tracked by Git. In either case, this project directory is called your Working directory.  
   ***Working directory is the directory containing hidden .git folder.***
2. ***Staging Area 🡪***

While we're in the working directory, we select the files that have to be tracked by Git. ***Why do we need to this? Why don't we track everything in the project?*** That's because some files in the project like ***class files, log files, result files and temporary  
data files are dynamically generated***. It doesn't make sense to track the versions of these files. ***Whereas the source code files, data files, configuration files and other project artifacts contain the business logic of the application***. These files are to be tracked by  
Git are thus needs to be added to the staging area.  
(In other words, staging area is the playground where you group, add and organize the files to be committed to Git for tracking their versions.)

Page | 8

1. **Git directory 🡪**Git directory is the database where metadata about project files' history will be tracked.  
     
   **the further part is repository**  
   A git repository is the .git/ folder inside a project. This repository tracks all changes made in to file in your project, building a history over time
   1. **Local Repository 🡪** The local repository is a git repository that is stored on your computer
   2. **Remote Repository 🡪** The remote repository is a git repository that is stored on some remote computer or server

****

Page | 9