Experiment -1.1

Student Name: Hitashi **UID: 22BDO10039**
Branch: CSE devops **Section/Group:22BCD1-A**
Semester: 4th **Date of Performance:17.01.24**
Subject Name: Git and Github **Subject Code:22CSH-293**

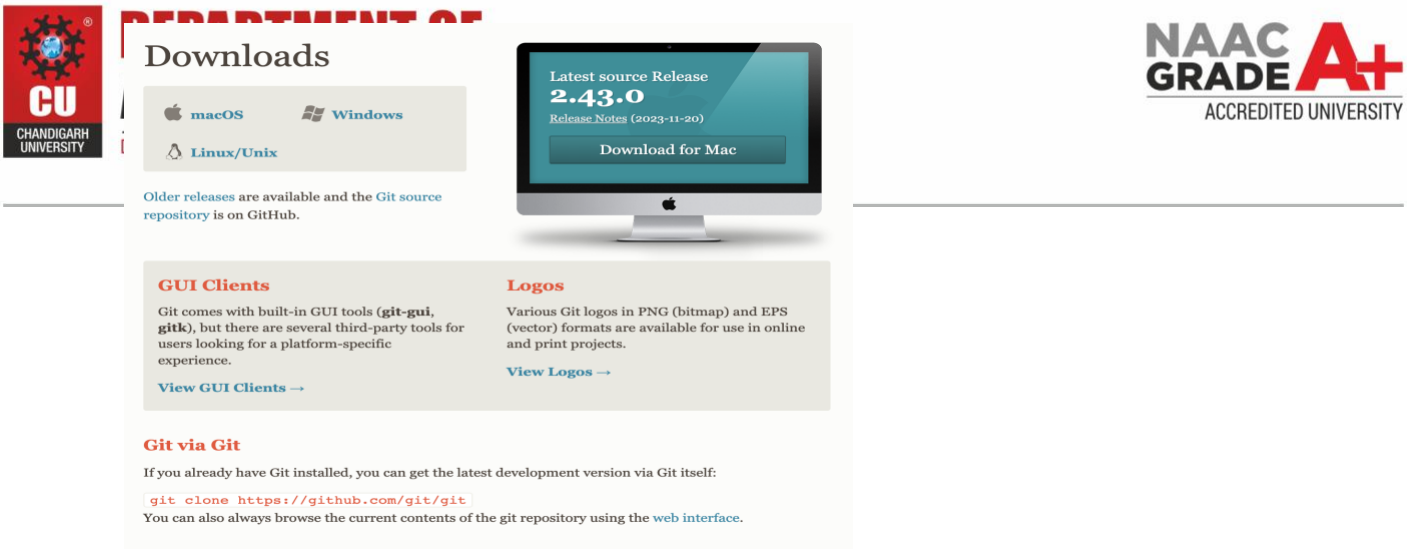
1. Aim/Overview of the practical: Install Git and creating repository

2. Task to be done: Download Git for Mac OS. And, to make repositories.

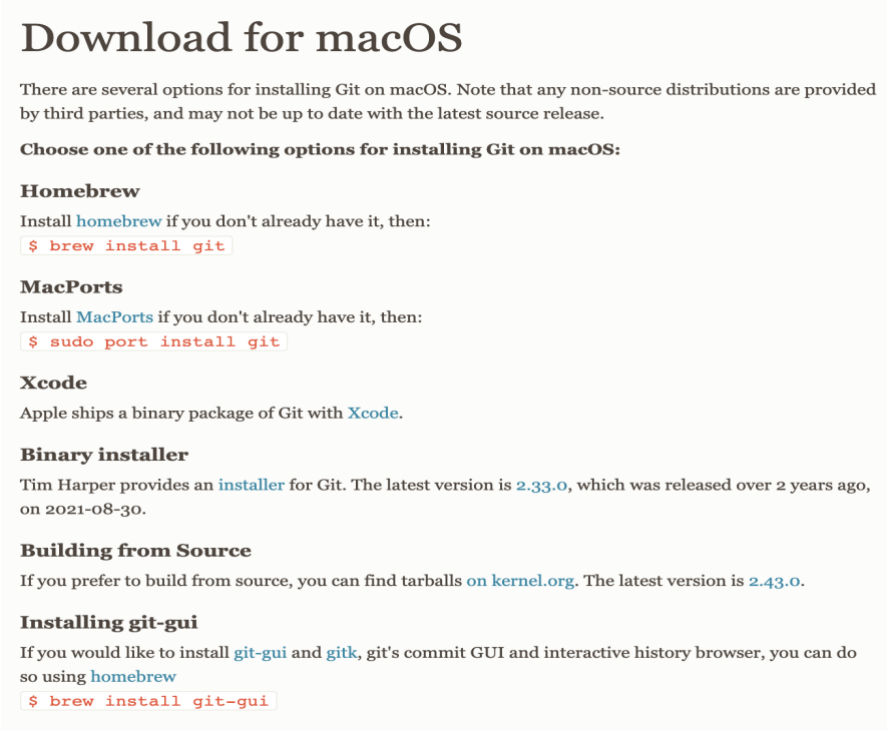
3. Steps for experiment/practical:

1)Browse to the official Git website: https://git-scm.com/downloads.

2) Click the **download link for Mac and allow the download to complete. Or you can simply download it from**



3) Browse to the download location (or use the download shortcut in your browser). Double-click the file to extract and launch the installer.Now select the type to install.



4) Allow the app to make changes to your device by clicking Yes on the User Account Control dialog that opens. Follow the installation wizard and configure Git to suit your development needs. If you are new to version control systems, the best option would be to leave the default settings.

5). Click **Install** and type in your password if necessary.

6) Confirm once again by clicking **Install Software.** With this, you have finished setting up Git on your Mac. Move on to the next step of configuring Git.

7) Review the GNU General Public License, and when you're ready to install, click **Next**.

Git for Mac Installer

The easiest way to install Git on a Mac is via the stand-alone installer:

1. Download the latest Git for Mac installer.

2. Follow the prompts to install Git.

3. Open a terminal and verify the installation was successful by typing `git --version`:

4. Configure your Git username and email using the following commands, replacing Emma's name with your own. These details will be associated with any commits that you create:

```
himanshi@Himanshis-MacBook-Pro ~ % git version
himanshi@Himanshis-MacBook-Pro ~ % git config --global user.name
"Hitashikankran"
himanshi@Himanshis-MacBook-Pro ~ % git config --global user.email
"hitashi2401@gmail.com"
himanshi@Himanshis-MacBook-Pro ~ % git config --list
```

Configure GitHub Credentials

Configure your local Git installation to use your GitHub credentials by entering the following:

git config --global user.name "github_username"

git config --global user.email "email_address"

Note: Replace **github_username** and **email_address** with your GitHub credentials.

Clone a GitHub Repository

Go to your repository on GitHub. In the top right above the list of files, open the **Clone or download** drop-down menu.

Copy the **URL for cloning over HTTPS**.

Switch to your PowerShell window, and enter the following:

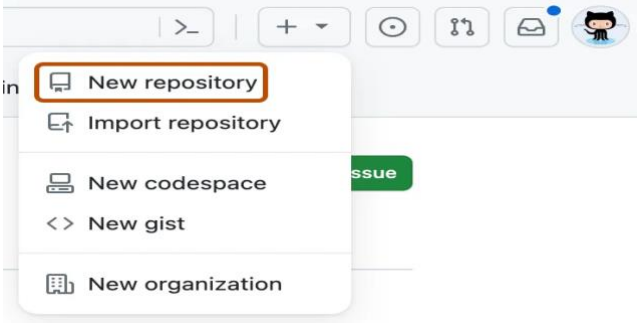
git clone repository_url

2.)Creating a new repository

You can create a new repository on your personal account or any organization where you have sufficient permissions.

Creating a new repository from the web UI

1.) In the upper-right corner of any page, select + then click New repository.

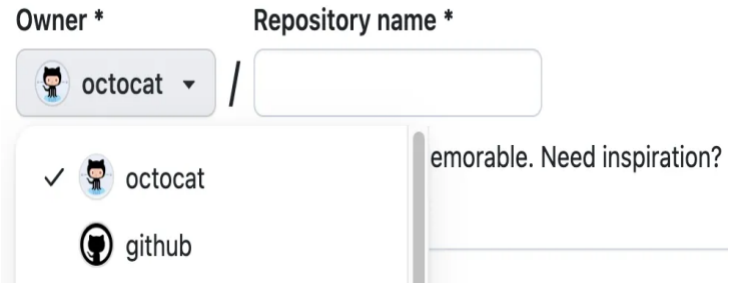


2.) Optionally, to create a repository with the directory structure and files of an existing repository, select the Choose a template dropdown menu and click a template repository. You'll see template

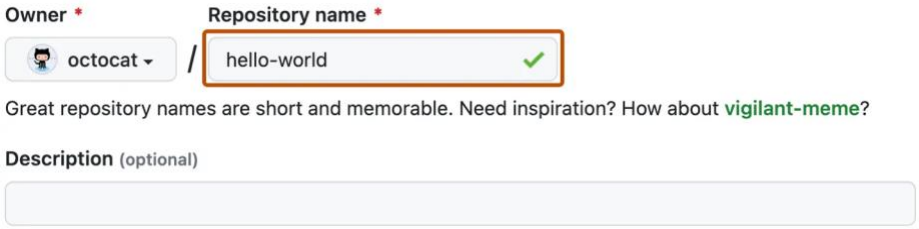
repositories that are owned by you and organizations you're a member of or that you've used before. For more information, see "Creating a repository from a template."

3.) Optionally, if you chose to use a template, to include the directory structure and files from all branches in the template, and not just the default branch, select **Include all branches.**

4.) Use the **Owner** dropdown menu to select the account you want own the repository.



5.) Type a name for your repository, and an optional description.



6.) Choose a repository visibility. For more information, see "About repositories."

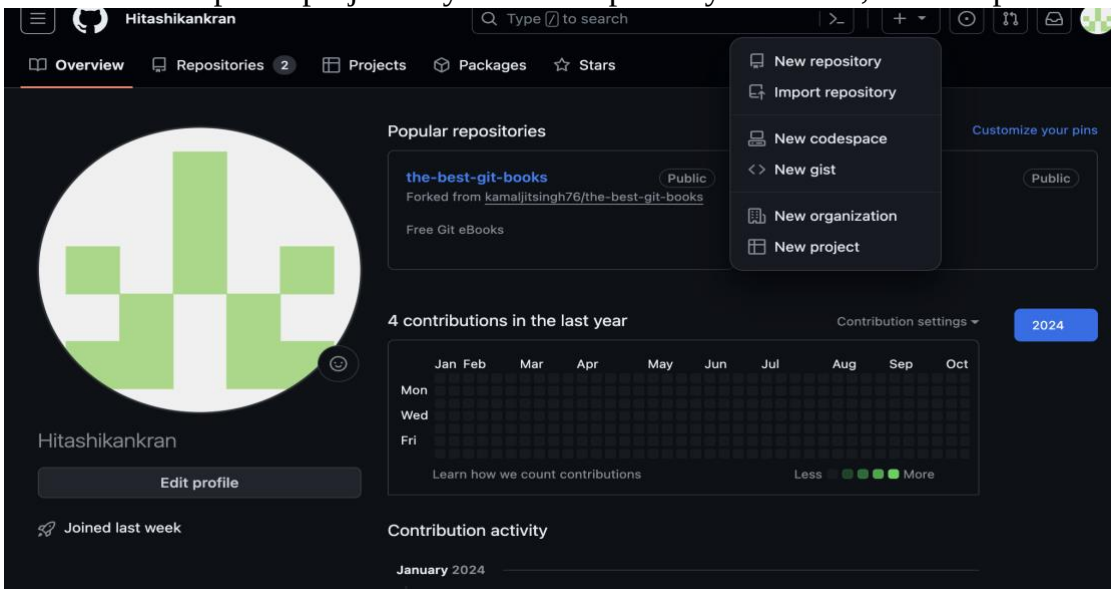
7.) If you're not using a template, there are a number of optional items you can pre-populate your repository with. If you're importing an existing repository to GitHub, don't choose any of these options, as you may introduce a merge conflict. You can add or create new files using the user interface or choose to add new files using the command line later. For more information, see "Importing an external Git repository using the command line," "Adding a file to a repository," and "Addressing merge conflicts."

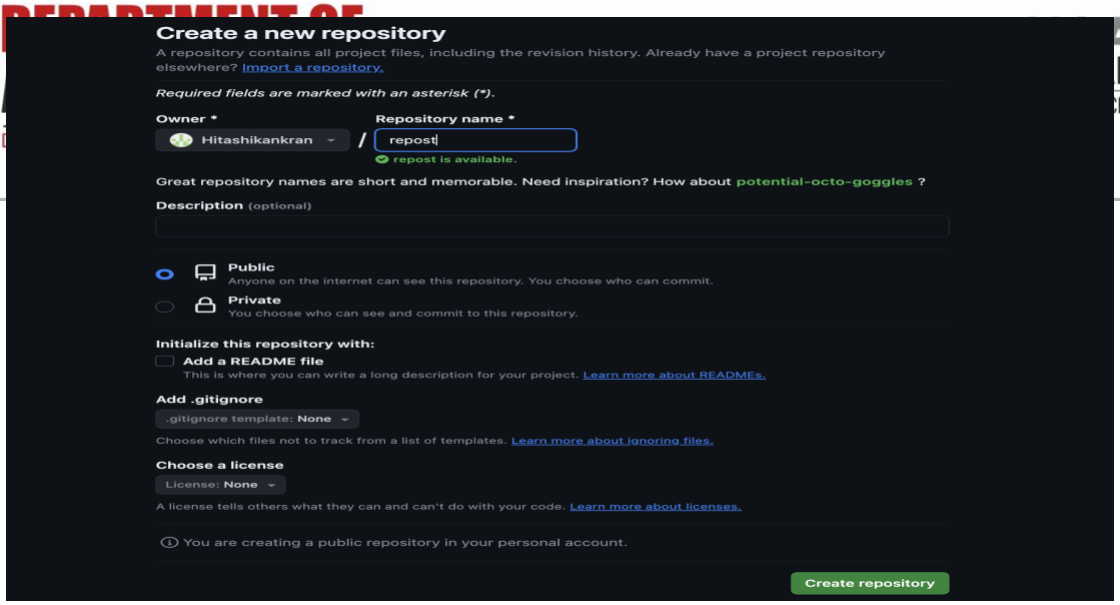
a. You can create a README, which is a document describing your project. For more information, see "About READMEs."
b. You can create a .gitignore file, which is a set of ignore rules. For more information, see "Ignoring files."
c. You can choose to add a software license for your project. For more information, see "Licensing a repository."

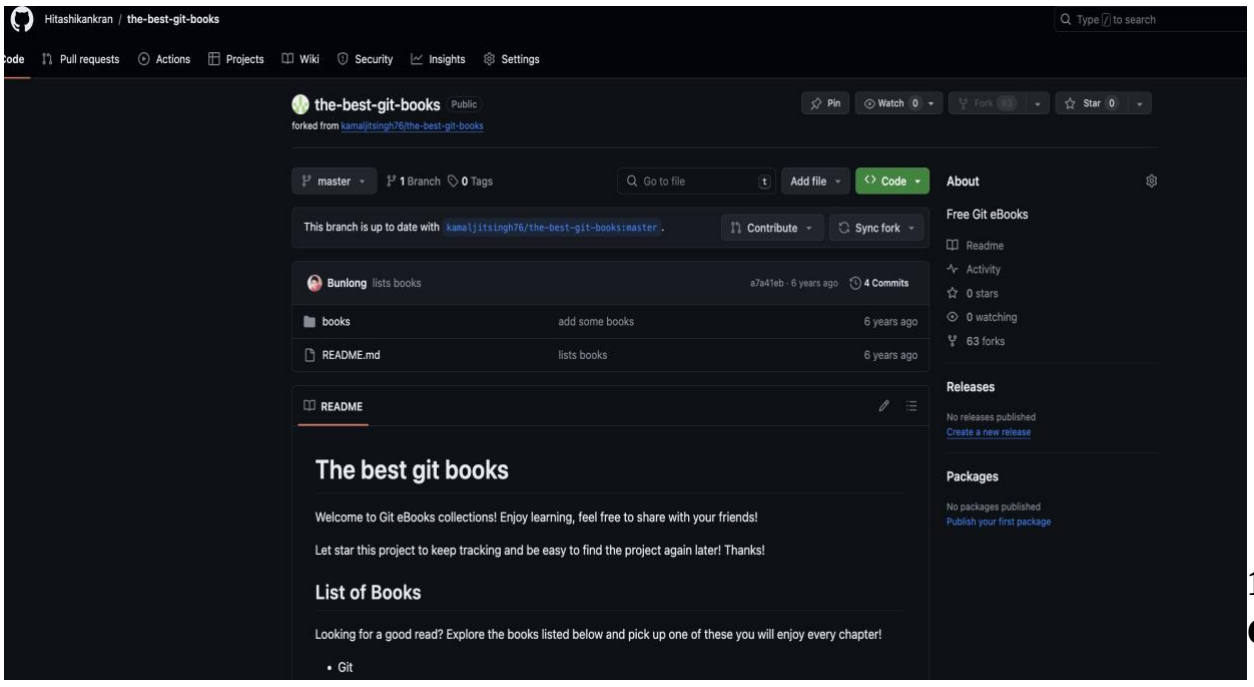
8.) Optionally, if the personal account or organization in which you're creating uses any GitHub Apps from GitHub Marketplace, select any apps you'd like to use in the repository.

9.) Click Create repository.

10.) At the bottom of the resulting Quick Setup page, under "Import code from an old repository", you can choose to import a project to your new repository. To do so, click Import code.

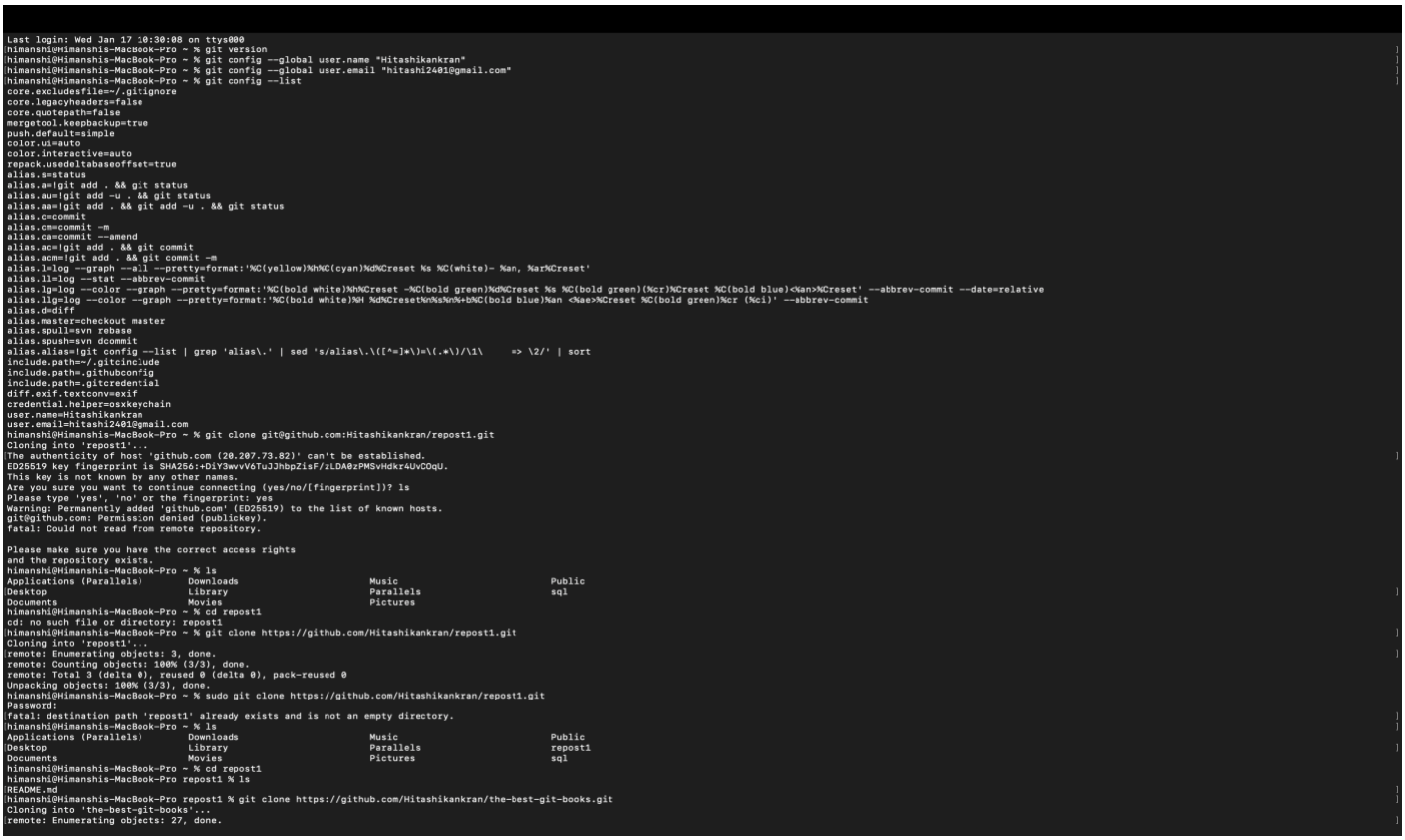



Create a new repository
A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.
Required fields are marked with an asterisk (*).
Owner *
Hitashikankran
Repository name *
repost
repost is available.
Great repository names are short and memorable. Need inspiration? How about potential-octo-goggles ?
Description (optional)
Public
Anyone on the internet can see this repository. You choose who can commit.
Private
You choose who can see and commit to this repository.
Initialize this repository with:
Add a README file
This is where you can write a long description for your project. Learn more about READMEs.
Add .gitignore
.gitignore template: None
Choose which files not to track from a list of templates. Learn more about ignoring files.
Choose a license
License: None
A license tells others what they can and can't do with your code. Learn more about licenses.
You are creating a public repository in your personal account.
Create repository


Hitashikankran / the-best-git-books
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
the-best-git-books
Public
forked from kamaljitsingh76/the-best-git-books
Pin
Watch 0
Star 0
master
1 Branch
0 Tags
Go to file
Add file
Code
About
Free Git eBooks
This branch is up to date with kamaljitsingh76/the-best-git-books:master.
Contribute
Sync fork
Readme
Activity
0 stars
0 watching
63 forks
Bunlong lists books
a7a41eb · 6 years ago
4 Commits
books
add some books
6 years ago
README.md
lists books
6 years ago
README
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
The best git books
Welcome to Git eBooks collections! Enjoy learning, feel free to share with your friends!
Let star this project to keep tracking and be easy to find the project again later! Thanks!
List of Books
Looking for a good read? Explore the books listed below and pick up one of these you will enjoy every chapter!
Git

11. Graphs

(If Any): Image /Soft copy of graph paper to be attached here



Learning outcomes (What I have learnt):

1. Learnt about GitHub
2. Learnt about Git.
3. Learnt about various git commands that can be applied on Git Bash.
4. Learnt about repositories.

5. Learnt about how to pull request and push.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			