First we have to go learn.epam.com/start

Then login

1st course is **version control with git**

2nd course **java basic**

**What is version control?**

VC, also known as source control, is a software that tracks and manages changes to file over time. Allows revisit earlier versions of the file, compare changes between version undo changes, etc.

**Main Features:**

Track change across multiple files.

Compares version of a project.

VCS (version control system) is a software tool that helps track changes made to files and directories over time. Some key features of VCS include:

* Versioning: VCS keeps track of every version of the files and directories it manages, allowing users to revert to previous versions if necessary.
* Branching and merging: VCS allows users to create multiple branches of a project, allowing multiple users to work on the same project simultaneously without interfering with each other's work.
* Collaboration: VCS allows multiple users to collaborate on the same project by sharing changes and merging their work together.
* Conflict resolution: VCS helps users resolve conflicts that arise when merging changes from multiple sources.
* Audit trails: VCS keeps a history of all changes made to the files and directories it manages, allowing users to see who made which changes and when.
* Backup and restore: VCS provides a backup of all versions of the files and directories it manages, which can be restored in case of data loss or corruption.
* Remote access: Some VCS support remote access, allowing users to access their files and directories from anywhere, and collaborate with others in different locations.

**Centralized vcs**

A centralized version control system (VCS) is a type of version control system in which all users access a single, central repository to check out, check in, and manage files. This central repository is typically located on a server that can be accessed by all members of a team. Examples of centralized VCS include Subversion (SVN) and Perforce.

**Distributed vcs**

A distributed version control system (DVCS) is a type of version control system that allows multiple users to work on a project simultaneously, and also allows users to work offline. Unlike centralized version control systems, which rely on a single central repository, each user in a DVCS has a full copy of the entire repository, including the entire project history. This allows for more flexibility and autonomy for users, as well as improved performance and reliability. Some popular examples of DVCS include Git and Mercurial.

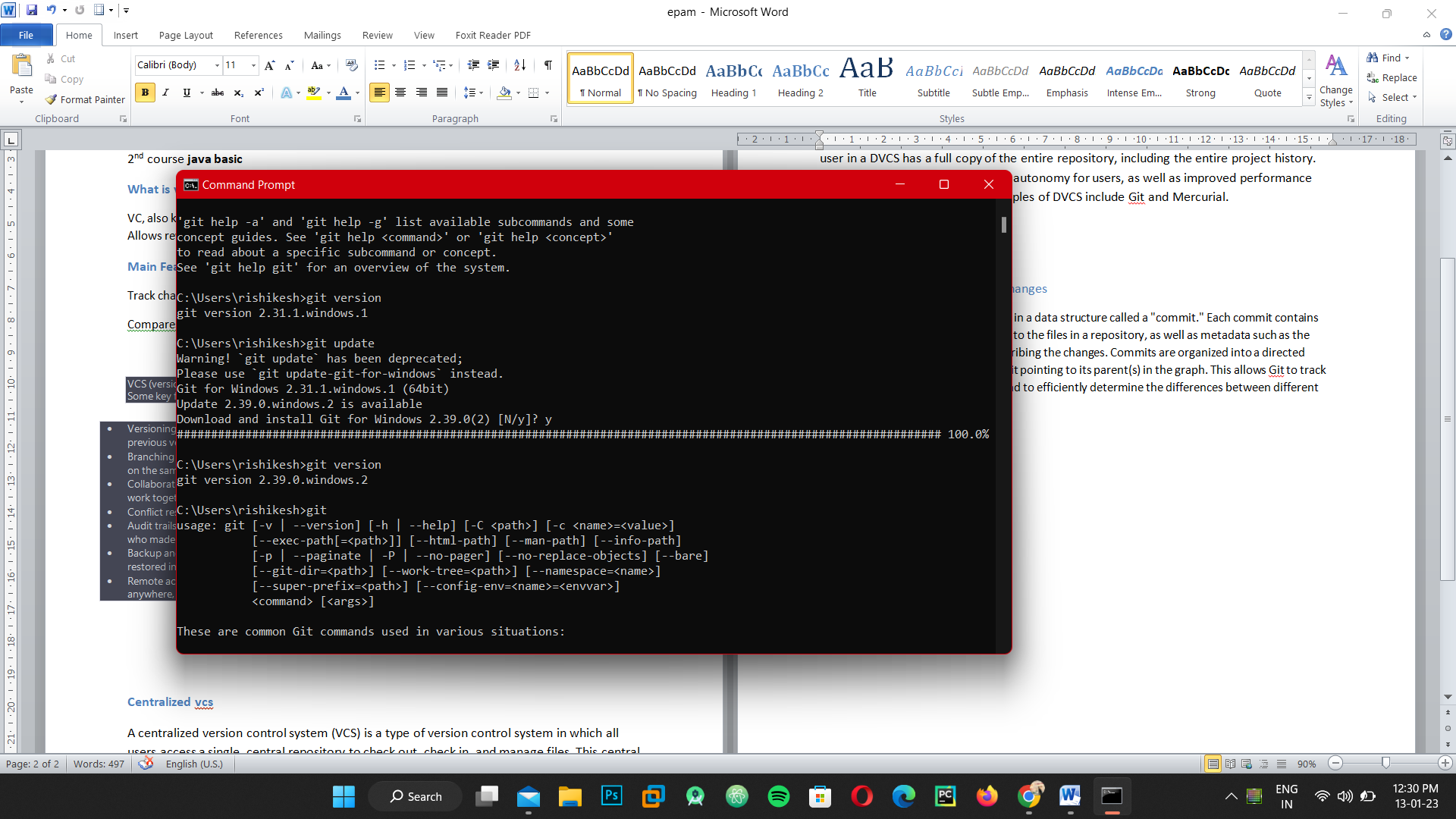
How git store information about changes

Git stores information about changes in a data structure called a "commit." Each commit contains information about the changes made to the files in a repository, as well as metadata such as the author, the date, and a message describing the changes. Commits are organized into a directed acyclic graph (DAG), with each commit pointing to its parent(s) in the graph. This allows Git to track the entire history of the repository and to efficiently determine the differences between different versions of the files.

**Installing git**

Git version

Git update



**Git config --list**

**Git config --global user.name=rishikesh**

**Git config –global user,name “rishikesh”**

**Git config –global user.email “rishikeshsinghntv@gmial.com**

**Git init [folder]**

**Mkdir helloworld**

**Cd helloworld**

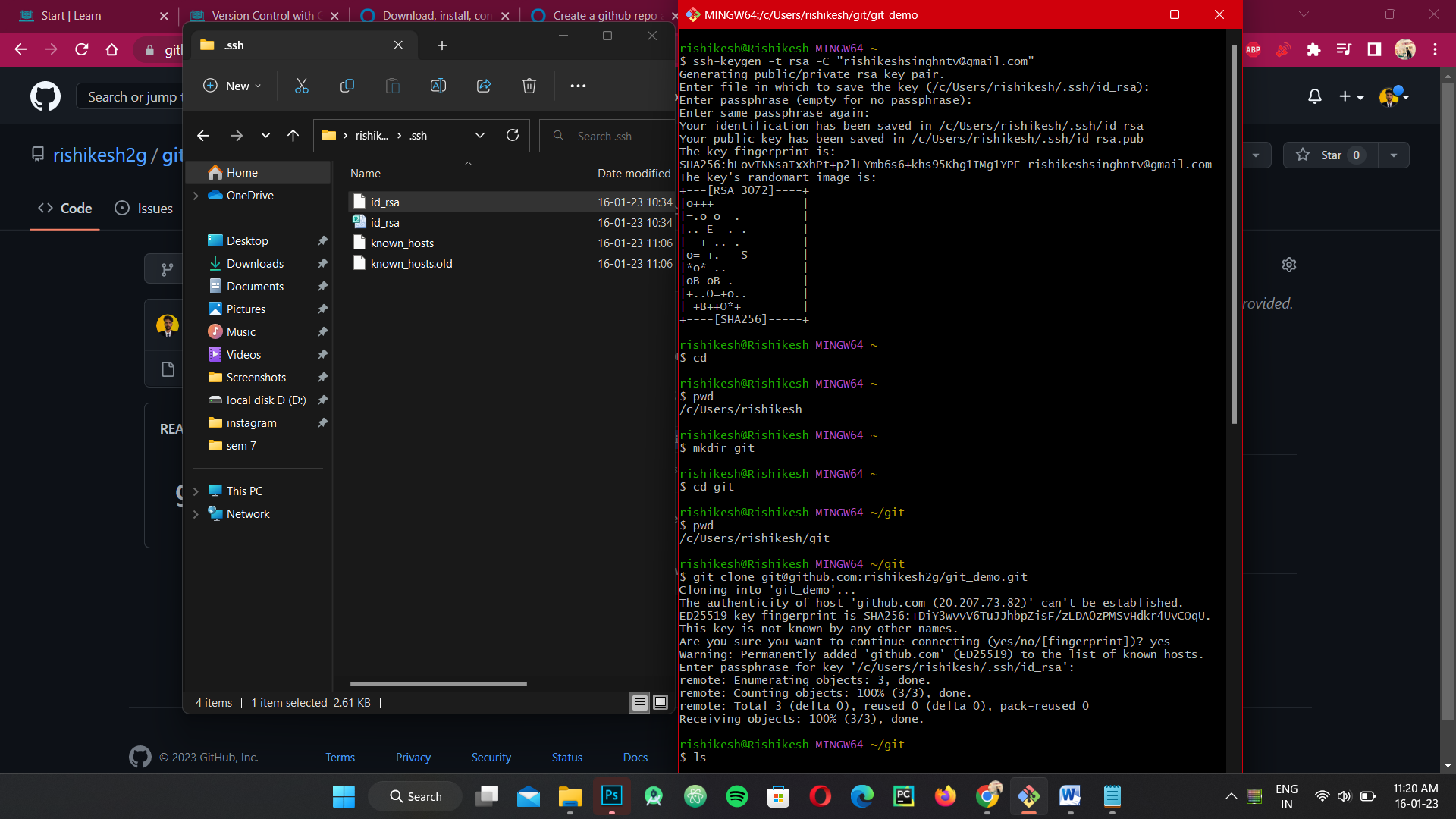
**Git init**

**For ssh keygen**

**ssh-keygen –t rsa –C “rishikeshsinghntv@gmail.com**

**And then create password**

**Go to fill explorer in user there a file known as ssh go there and copy public key and past it on github shh section.**



**To creating git new repository and clone to your system**

**Creaat new repository in you github**

**Copy ssh link**

**Come to your gitbash**

**And make new directory by this command mkdir “name” //$mkdir git**

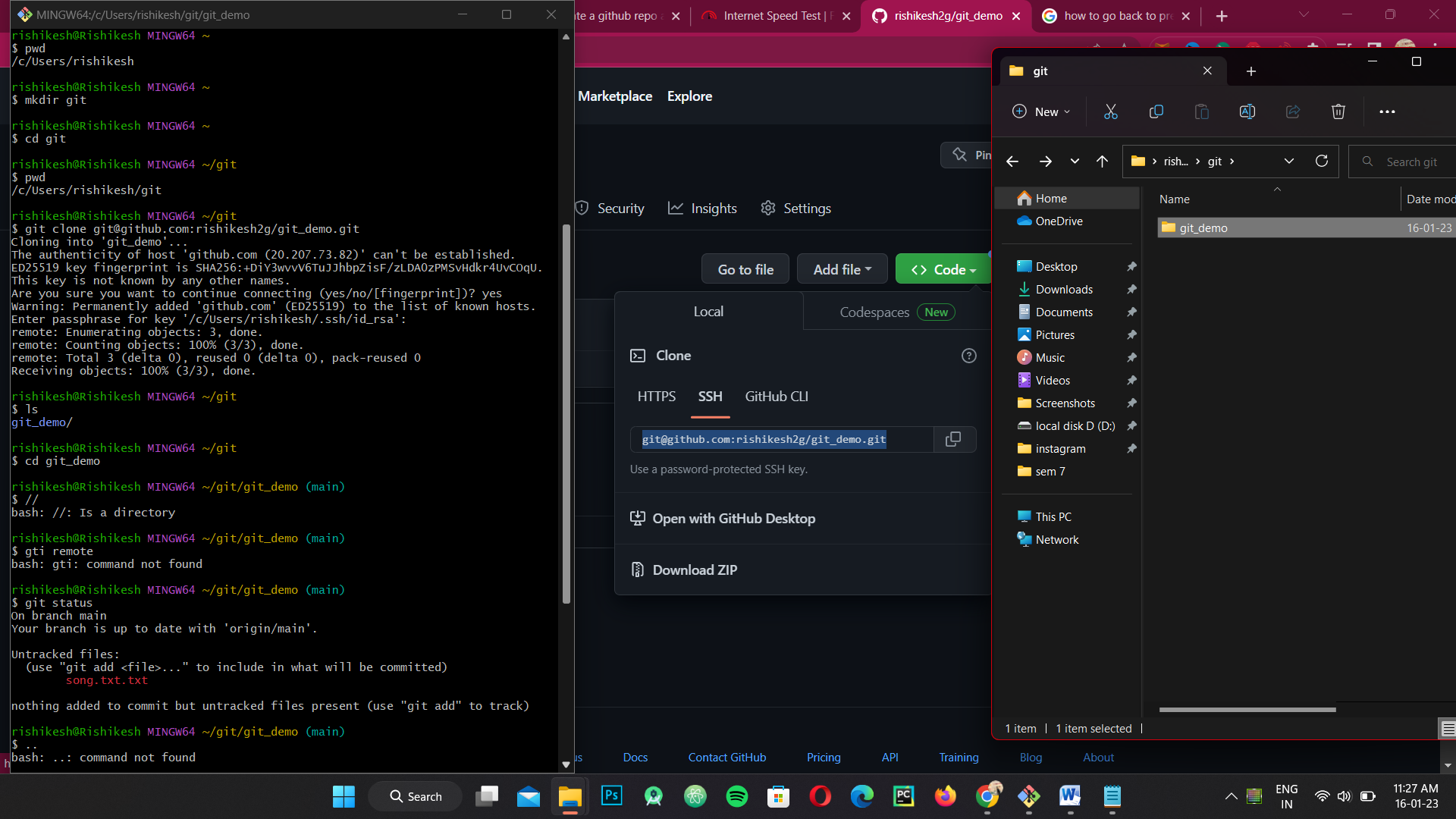
**Go to the directory $cd git**

**Now past the command which was copy on github ssh link**

**Command is :-**

**$git clone** [**git@github.com:rishikesh2g/git\_demo.git**](mailto:git@github.com:rishikesh2g/git_demo.git)

**The file come in your system that was created on github which was named as git\_demo**



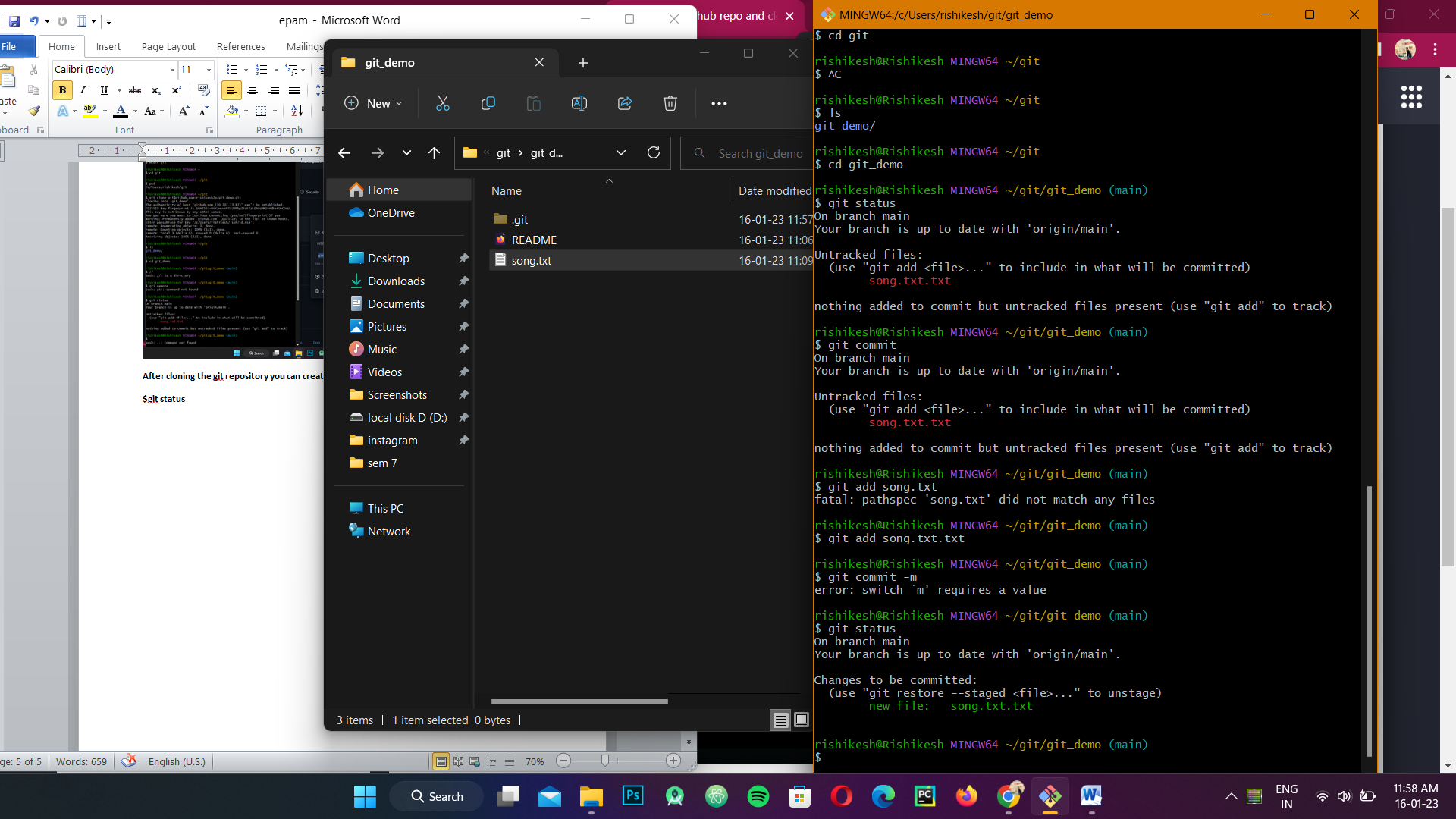
**After cloning the git repository you can create file on that folder and check the status on git bash**

**$git status**

**Where in repository which was clone from git hub create new file on there and check status of you fill**

**Like:-**

**$Git status**



**To commit your song.txt**

**Write a command :-**

**$git add song.txt**

**$git status**

**$git commit –m “add first 2 line in the song**

**$git log //is allow the know previous history**

**$git push**