- ***Introduction:***

  - ***Overview of Git and Github :***

    Git is a Software , It can be installed locally on the system. It is a distributed version control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

    GitHub is a Git repository hosting service that provides a web-based graphical interface. It is the world's largest coding community. Putting a code or a project into GitHub brings it increased, widespread exposure. Programmers can find source codes in many different languages and use the command-line interface, Git, to make and keep track of any changes.

    GitHub helps every team member work together on a project from any location while facilitating collaboration. You can also review previous versions created at an earlier point in time.

- ***About Repositories:***

  - A repository, or Git project, encompasses the entire collection of files and folders associated with a project, along with each file's revision history. The file history appears as snapshots in time called commits. The commits can be organized into multiple lines of development called branches. Because Git is a DVCS, repositories are self-contained units and anyone who has a copy of the repository can access the entire codebase and its history. Using the command line or other ease-of-use interfaces, a Git repository also allows for: interaction with the history, cloning the repository, creating branches, committing, merging, comparing changes across versions of code, and more.

- Through platforms like GitHub, Git also provides more opportunities for project transparency and collaboration. Public repositories help teams work together to build the best possible final product.

- ***Basic Git Commands :***

To use Git, developers use specific commands to copy, create, change, and combine code. These commands can be executed directly from the command line or by using an application like GitHub Desktop. Here are some common commands for using Git:

- **git init** initializes a brand new Git repository and begins tracking an existing directory. It adds a hidden subfolder within the existing directory that houses the internal data structure required for version control.

- **git clone** creates a local copy of a project that already exists remotely. The clone includes all the project's files, history, and branches.

- **git add** stages a change. Git tracks changes to a developer's codebase, but it's necessary to stage and take a snapshot of the changes to include them in the project's history. This command performs staging, the first part of that two-step process. Any changes that are staged will become a part of the next snapshot and a part of the project's history. Staging and committing separately gives developers complete control over the history of their project without changing how they code and work.
- **git commit** saves the snapshot to the project history and completes the change-tracking process. In short, a commit functions like taking a photo. Anything that's been staged with git add will become a part of the snapshot with git commit.

- **git status** shows the status of changes as untracked, modified, or staged.

- **git branch** shows the branches being worked on locally.

- **git merge** merges lines of development together. This command is typically used to combine changes made on two distinct branches. For example, a developer would merge when they want to combine changes from a feature branch into the main branch for deployment.

- **git pull** updates the local line of development with updates from its remote counterpart. Developers use this command if a teammate has made commits to a branch on a remote, and they would like to reflect those changes in their local environment.

- **git push** updates the remote repository with any commits made locally to a branch.

- ***How to use Git and Github :***

  - Create your GitHub account.
  - Create a repository or "repo" for short. This is where you store your code.
  - Build a file.
  - Make a commit. Whenever you create a file or change it, you create a Git commit to store the new version.
  - Connect your repo with your computer system.

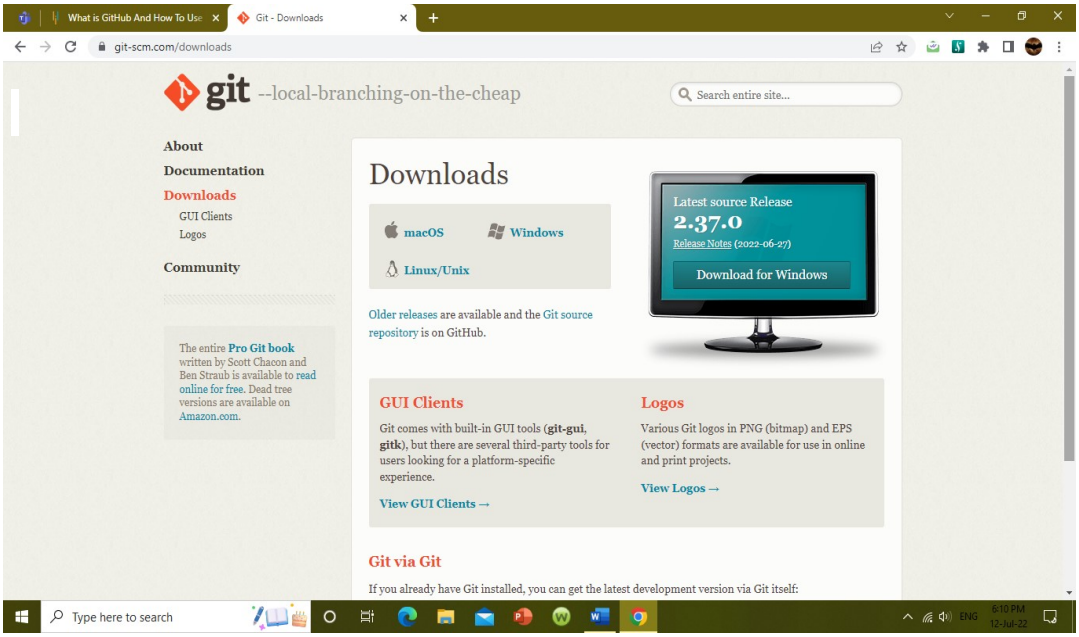
Creating and uploading a project on Github using Git bash and Git GUI :

Git is a distributed version control system for tracking changes in source code during software development . Git provides two entries when we install git in windows , i.e **Git Bash** and **Git Gui** .

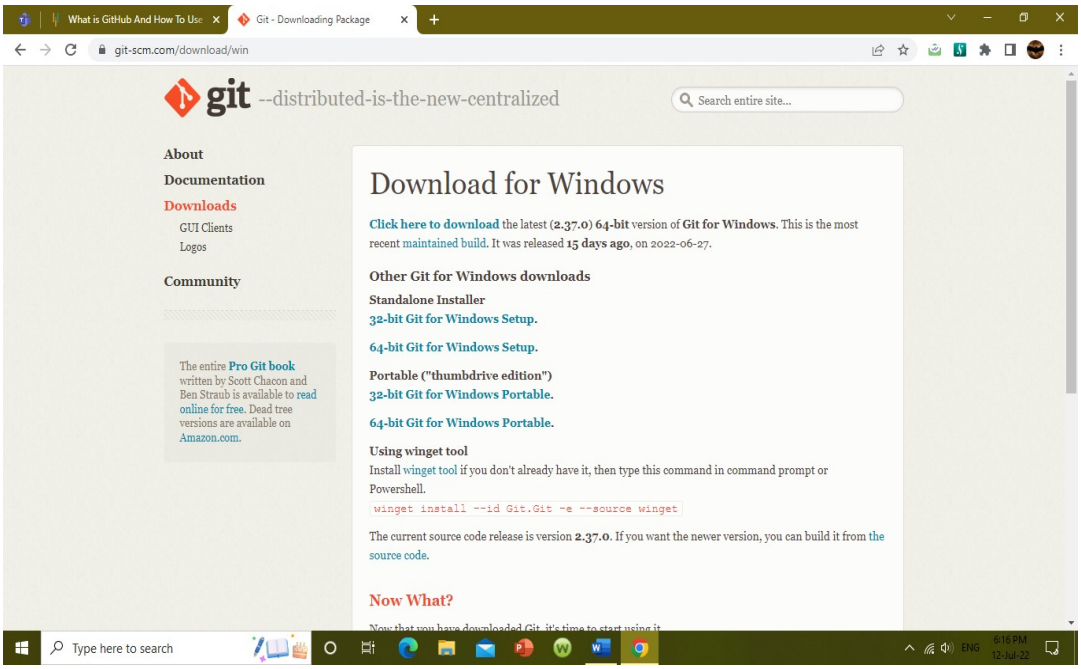
Git Bash , we can use git bash to clone the repository , and **Git Gui** is a Graphical User Interface focuses on allowing users to make changes to their repository by making new commits or we can use Git Gui to commit and push new changes to remote repo's at **GitHub .**

- ***Procedure :***

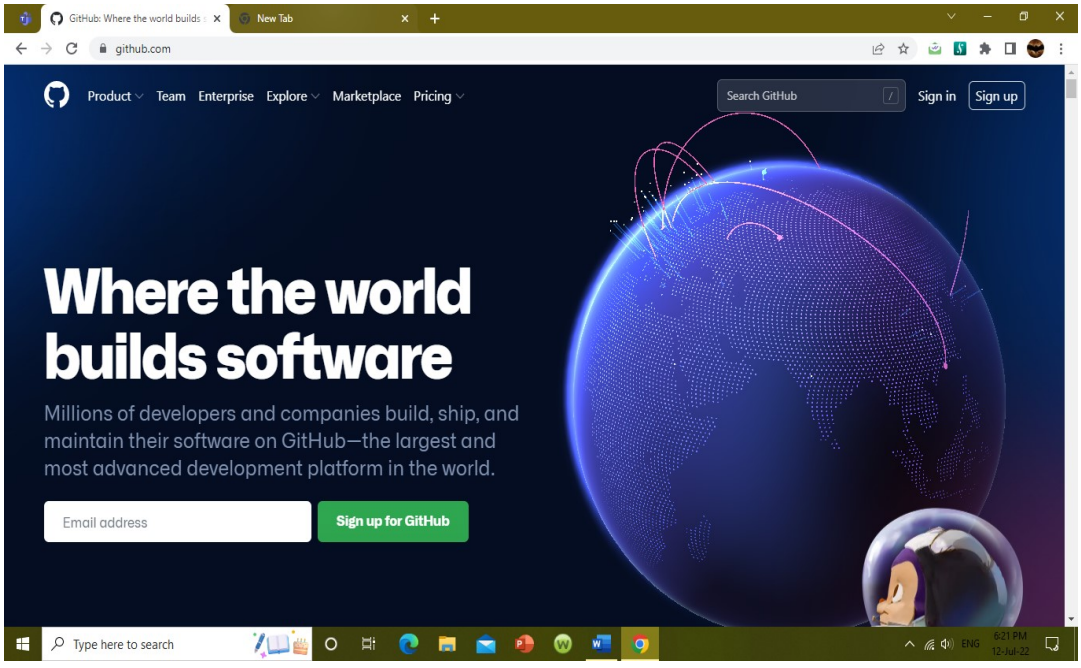
  - The first step is to install a Git for windows.
  - Type **Git download** in the browser and press enter and go on the Download -Git .
  - Or Go to the link "https://git-scm.com/downloads" .



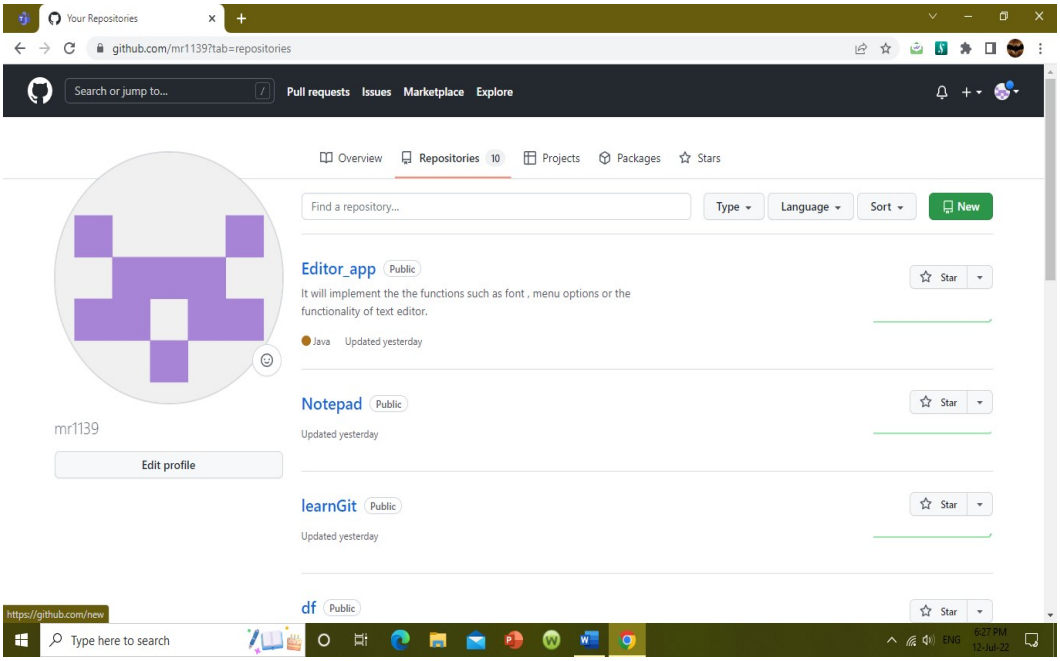
  - Then click on download "windows" , on the next page click on "Click here to download ".



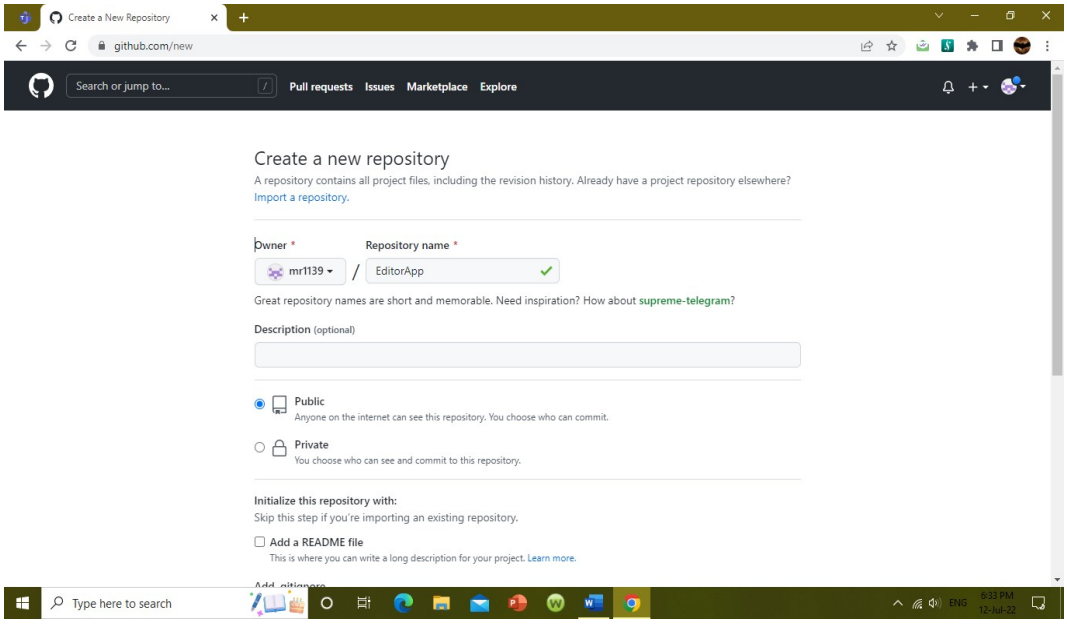
- Now , after we install git for windows , First of all we should have a repository to start with.
- Now we will use github .
- Go to the browser and type “GitHub”.
- After that click on the git-hub , and create an account in github .



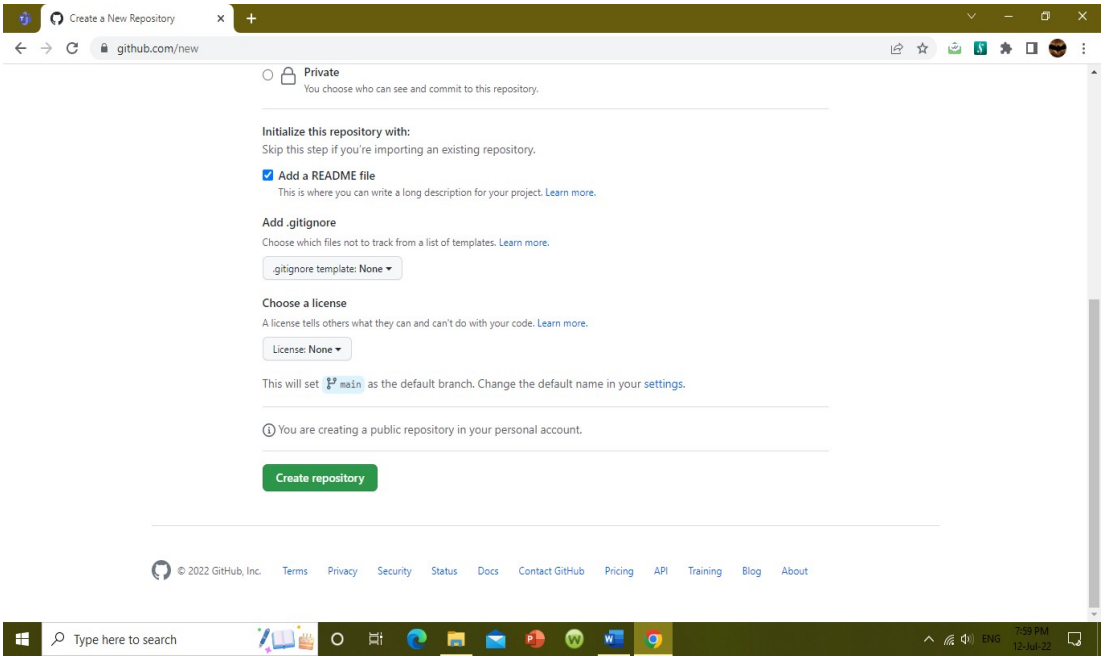
- Now for creating a repository in the github , go the tab repository , and create new by clicking on “new”.



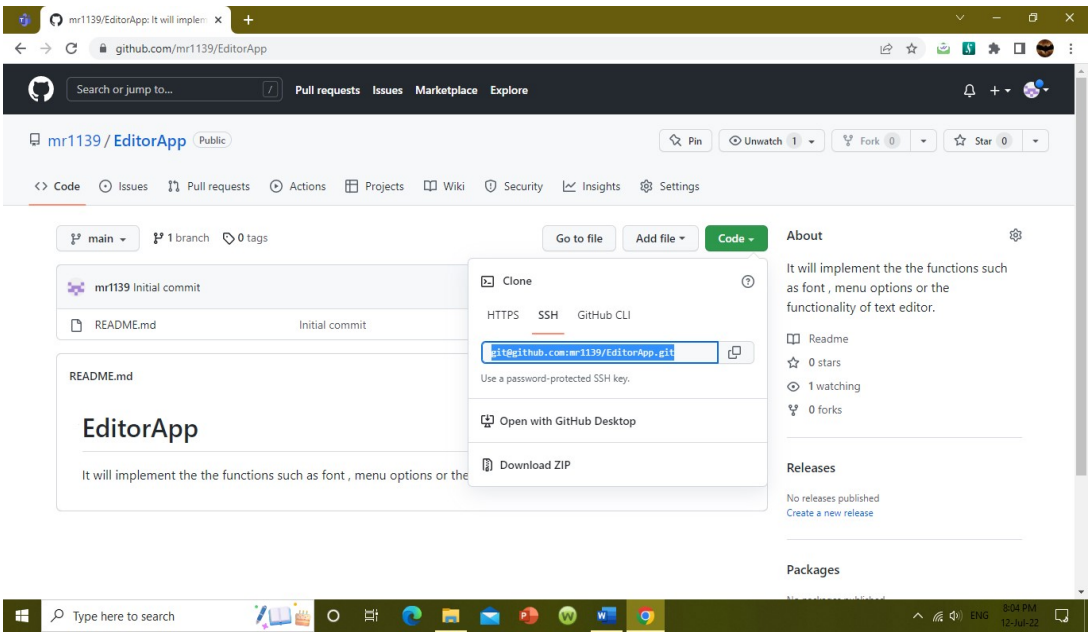
- And type name of repository as we are creating project for “editor -app” So , we have typed “EditorApp” , and give the description of the repo , tick “add read me file” .



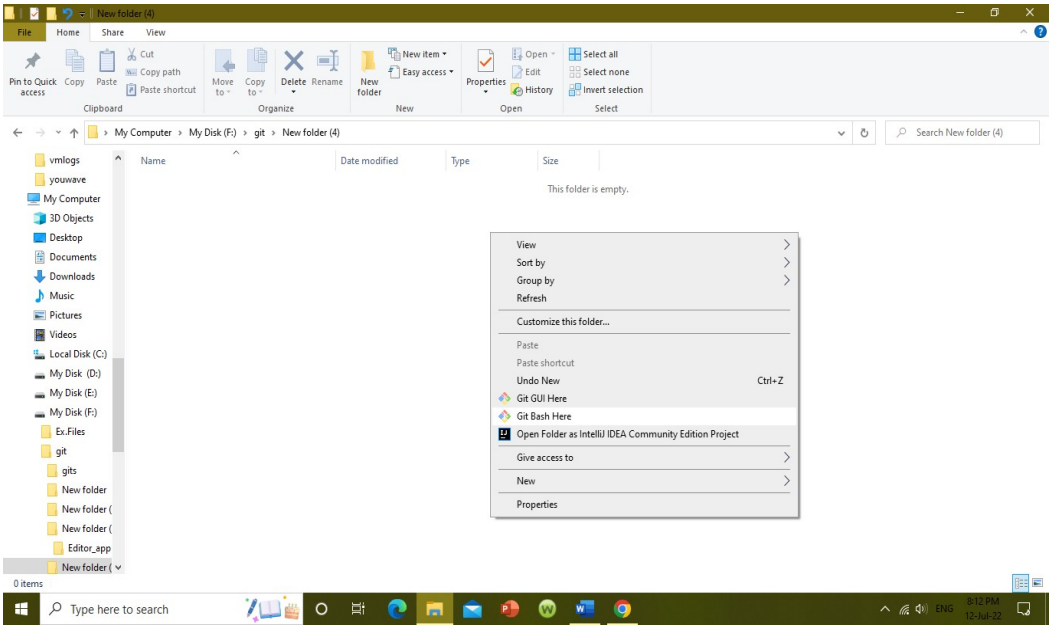
- And click on “create repository”.



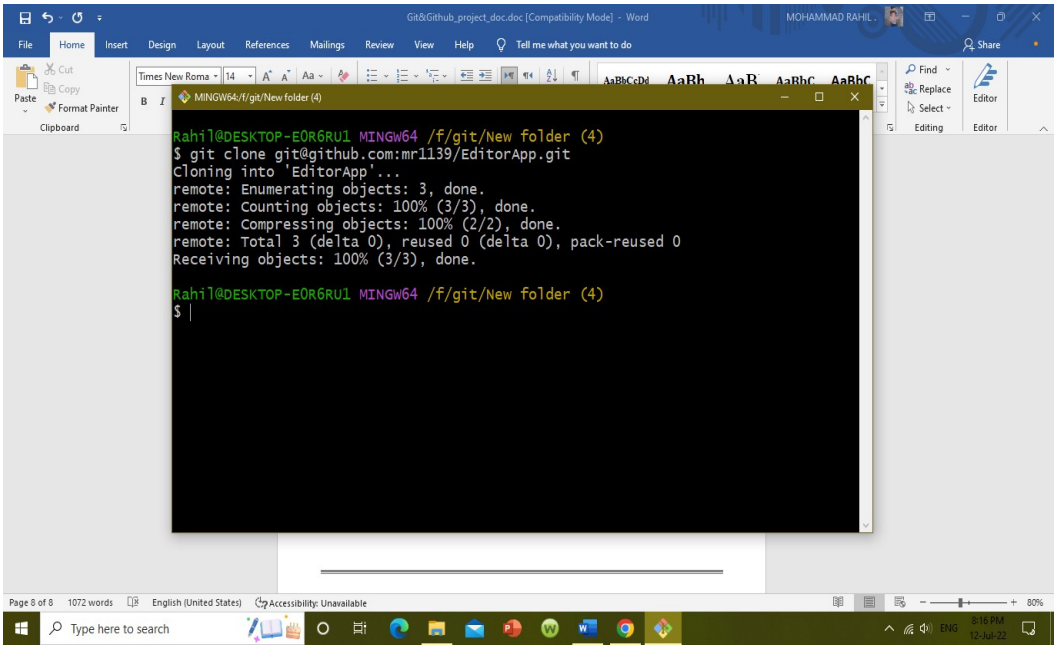
- After creating the repository we have got the its path in the “code” tab , for cloning the repo’s.
- https path , SSH path .



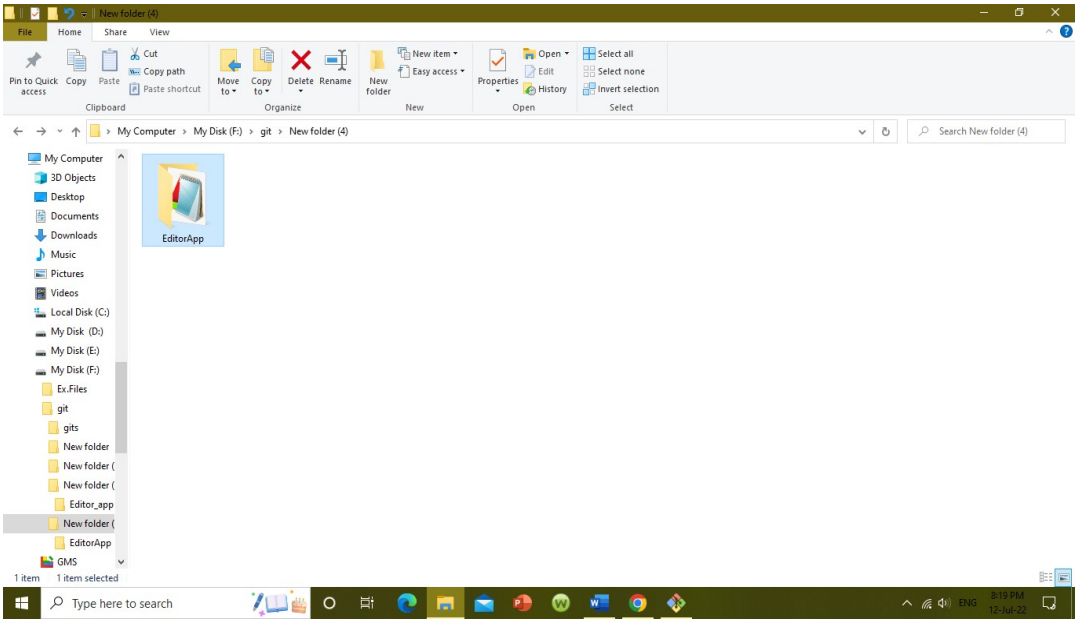
- now copy this SSH path for cloning the repo .
- copy the SSH path , and open the “Git Bash” in any empty folder by right clicking on the path and click on ” Git bash here ” .



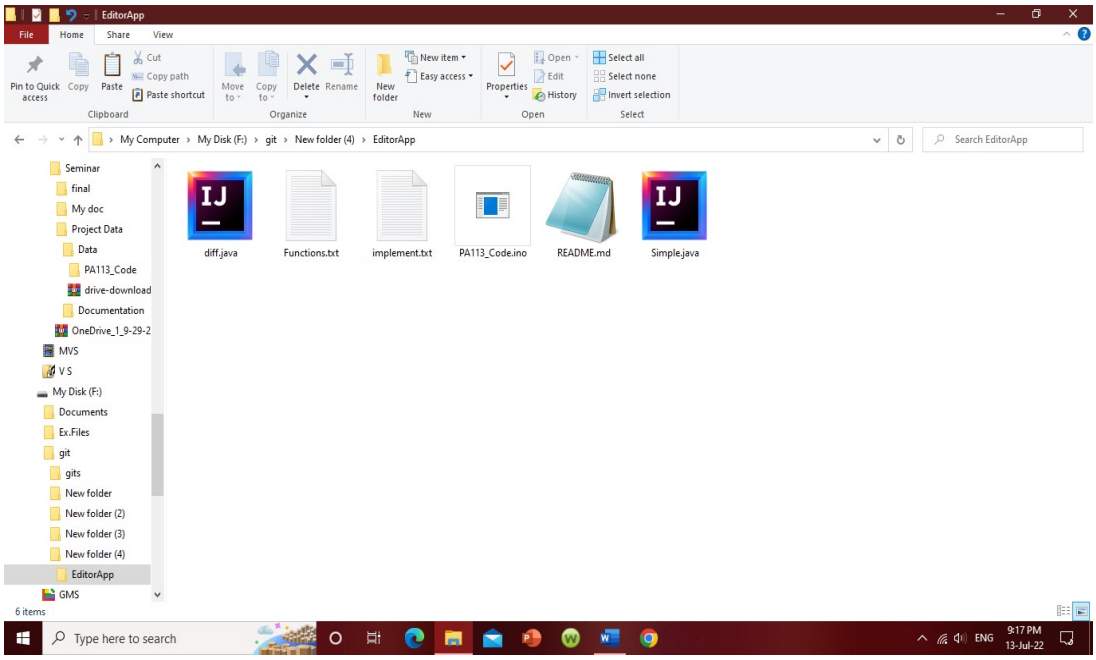
- On clicking “Git bash here”, the git bash command will open and then type the command “git clone ” and then its SSH path .
- And press enter



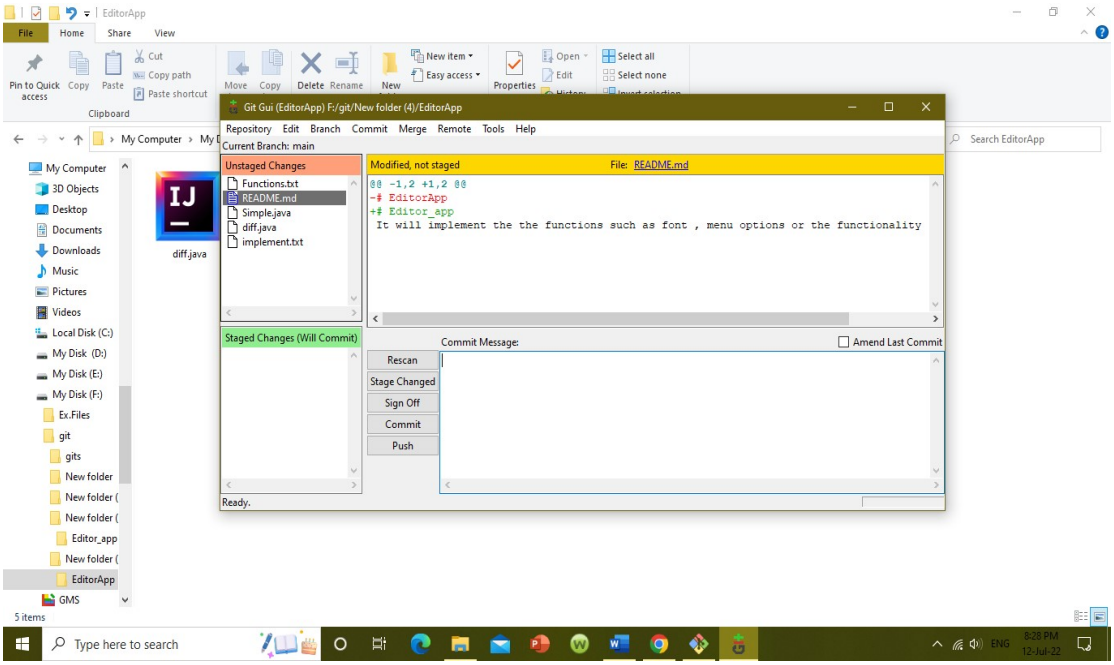
- So using this command we have got our repo in our local directory.



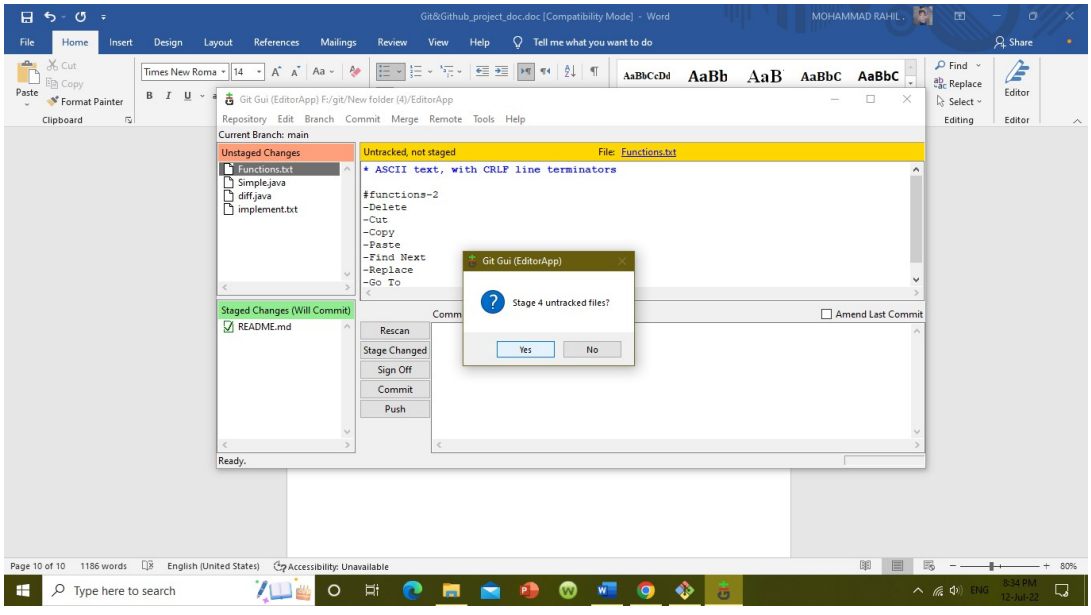
- Now we are adding our projects documents/file or code for it in the same folder .



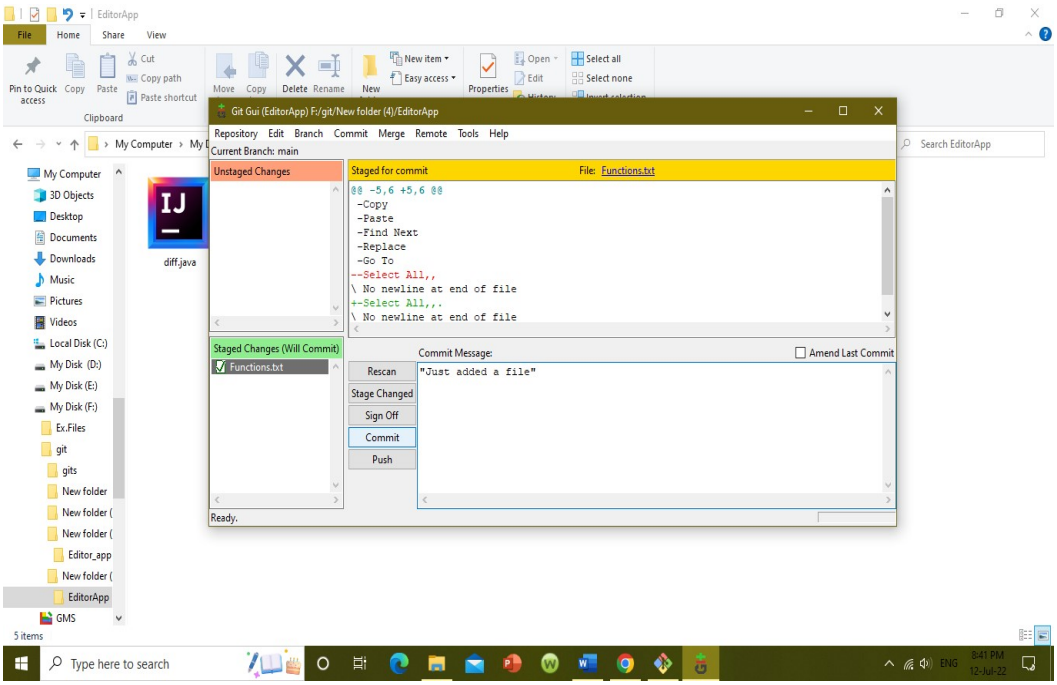
- Now we push all the file into our remote repo created in github .
- So , here is the best way using GUI (git gui) .
- Now on the same folder right click on the “git gui here”.
- After this it will open the Git Gui .



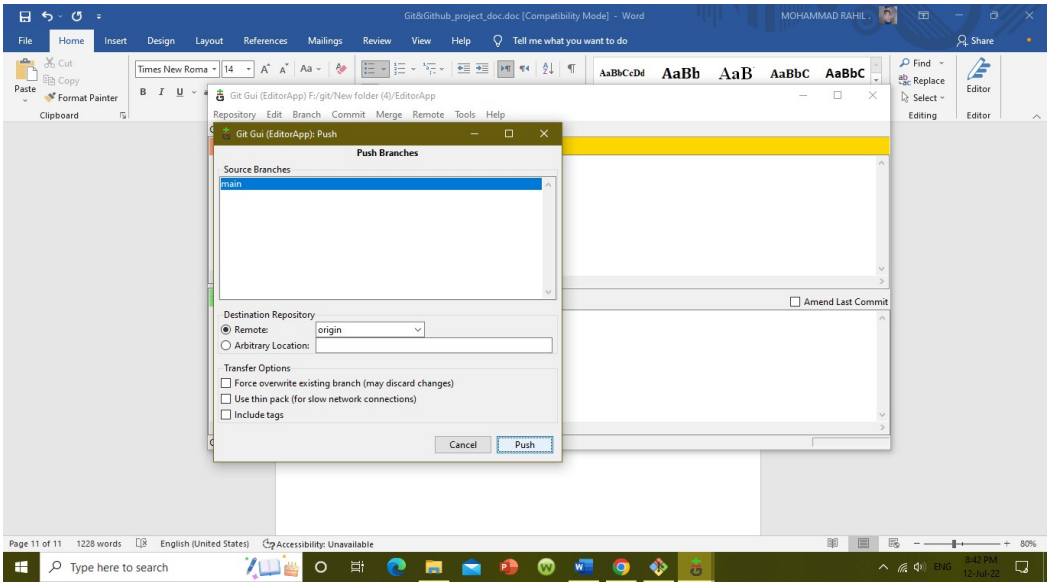
- Now here you can see unstaged changes , means git-gui says that these files/documents has been added here .
- So click on the documents and click on staged changes “staged 4 untracked files”.



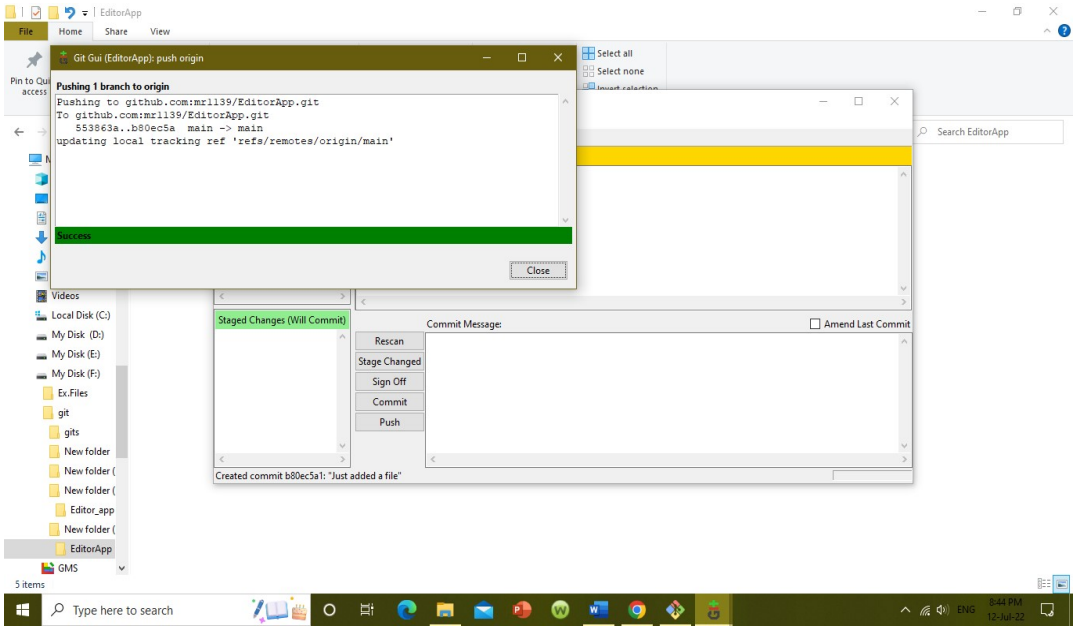
- No these are staged changes , these changes will add into commit.
- Now we will add our message as “Just added a file”.
- And then click at commit.
- And then we will click at push.



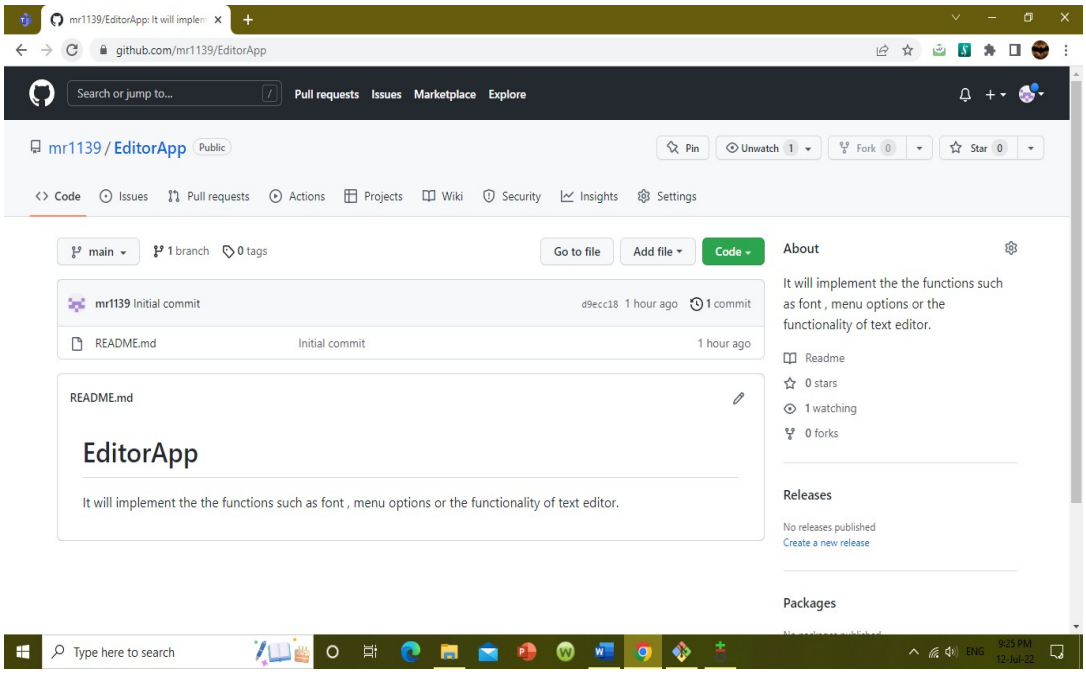
- So its branch is main , and click on “push ”.



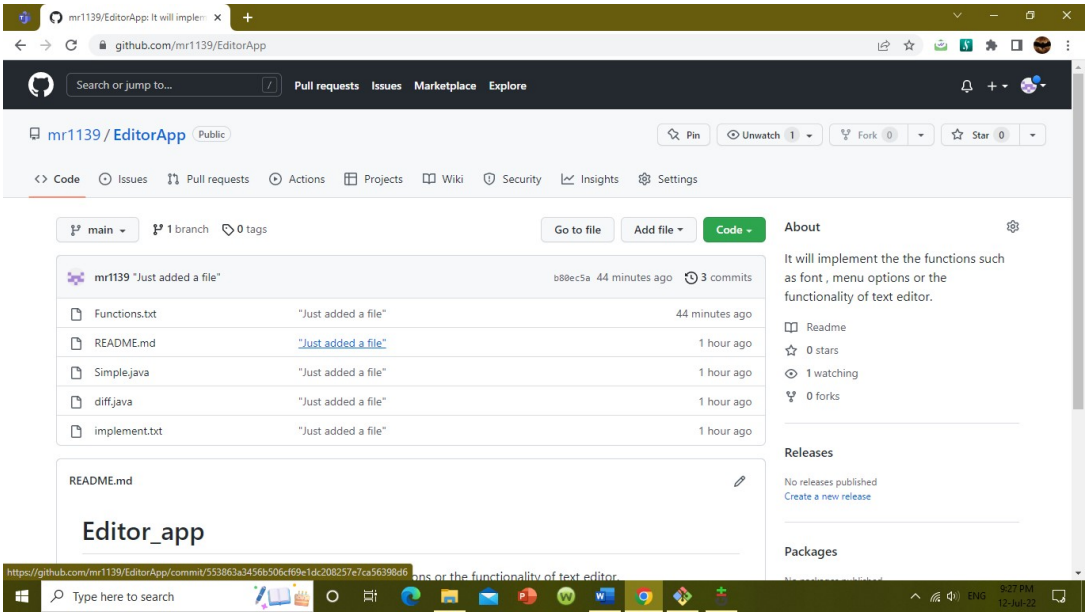
- As you can see after clicking on push its showing success , and then close it.



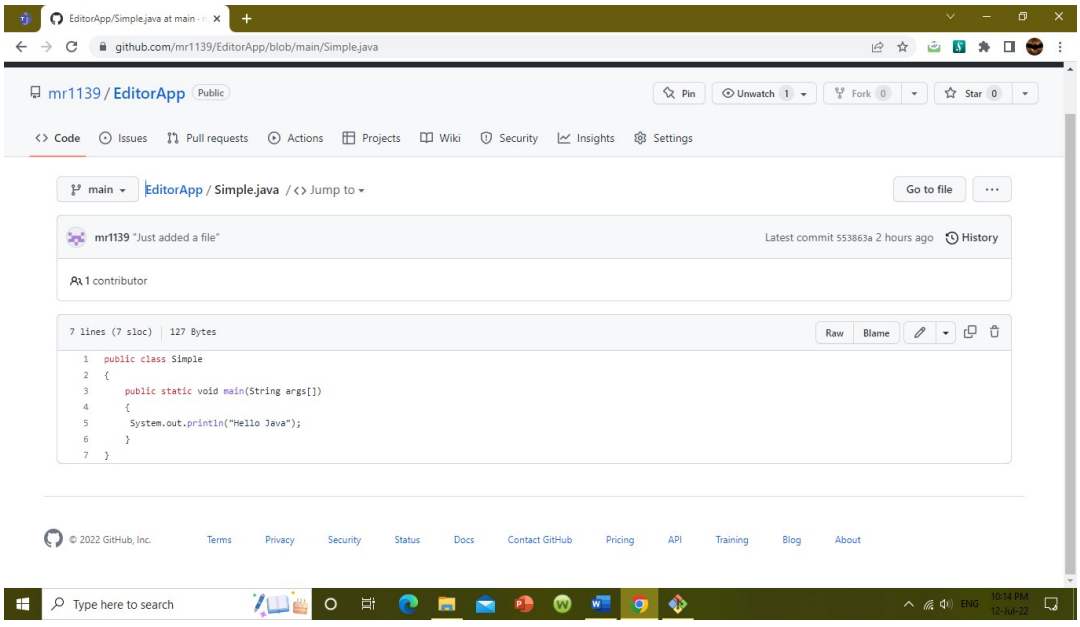
- After that go to our remote repo present in the github , and reload the --- page .

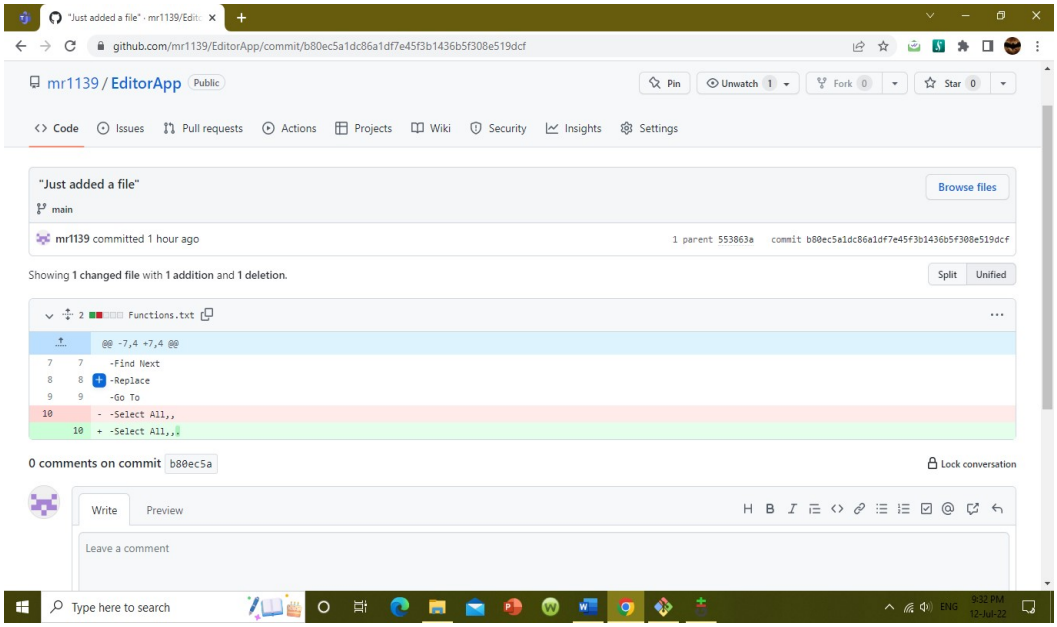


- As you can see here , all the files/documents has been added here in the github remote repo.
- Also we can see the commit message we have commit using git-gui .

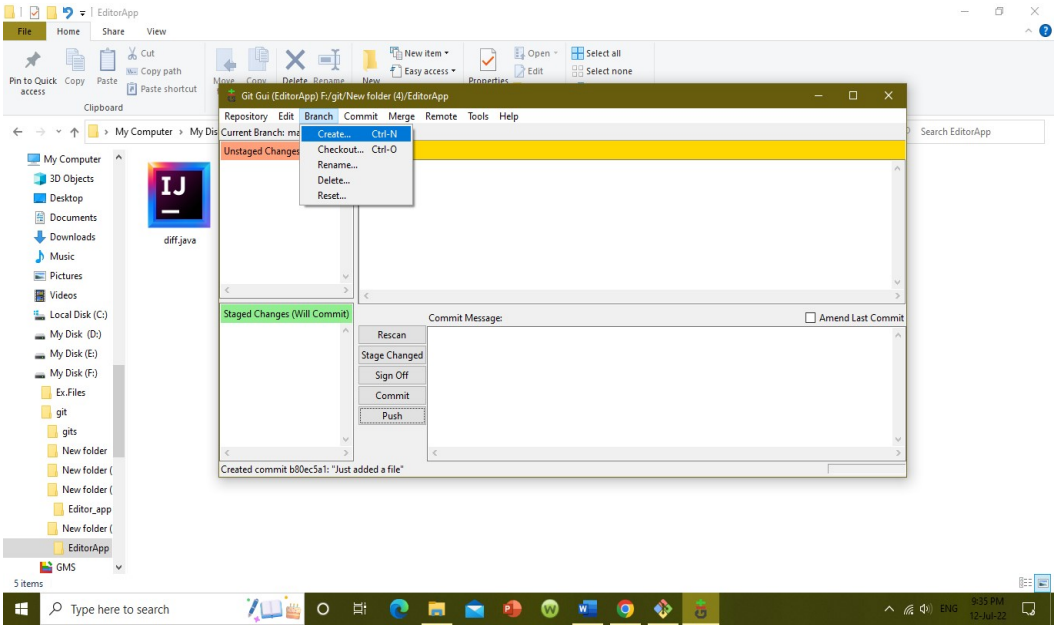


- Now we can see the changes in the github , whatever we have done using git-gui .

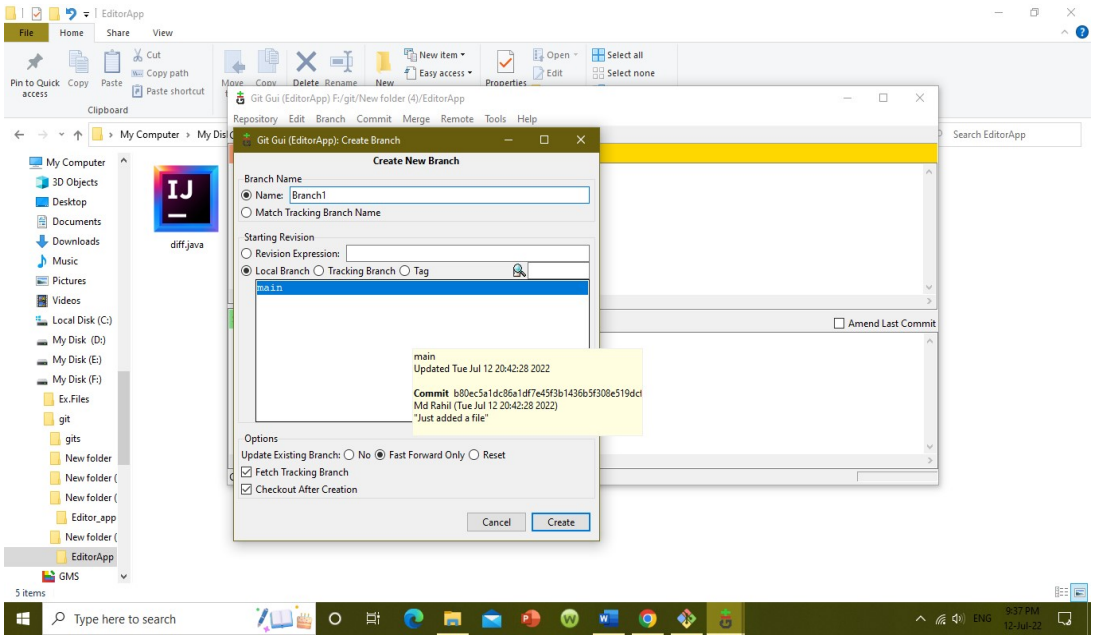




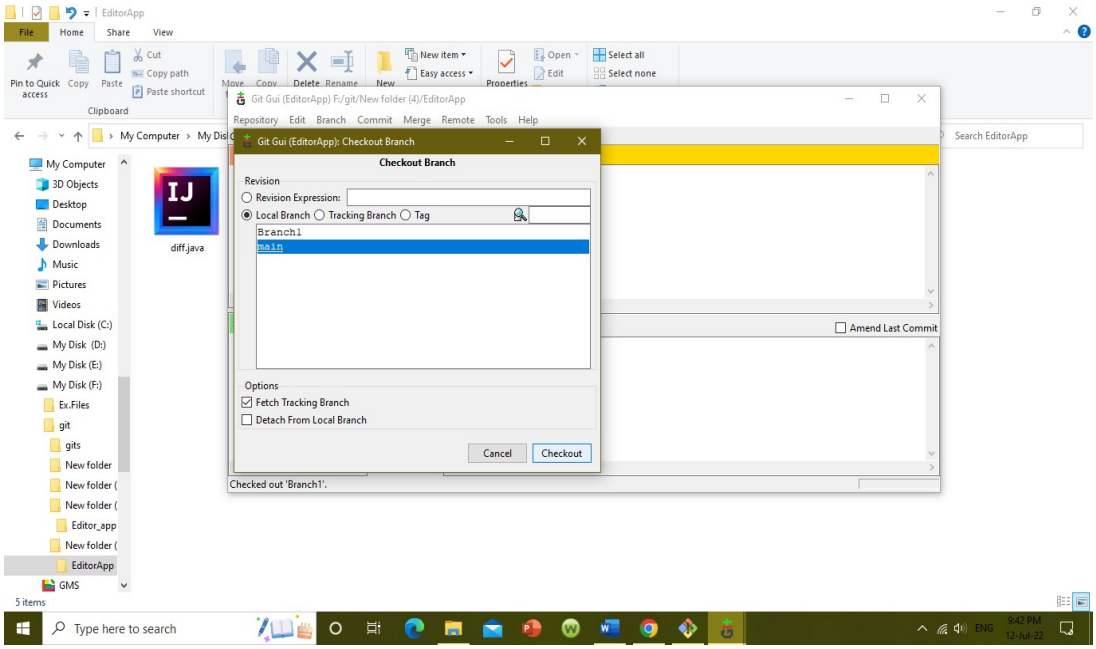
➢ Now If we want to create a branch and merge it using gui , we have to first create a branch by clicking on branch > create branch in the git-gui.

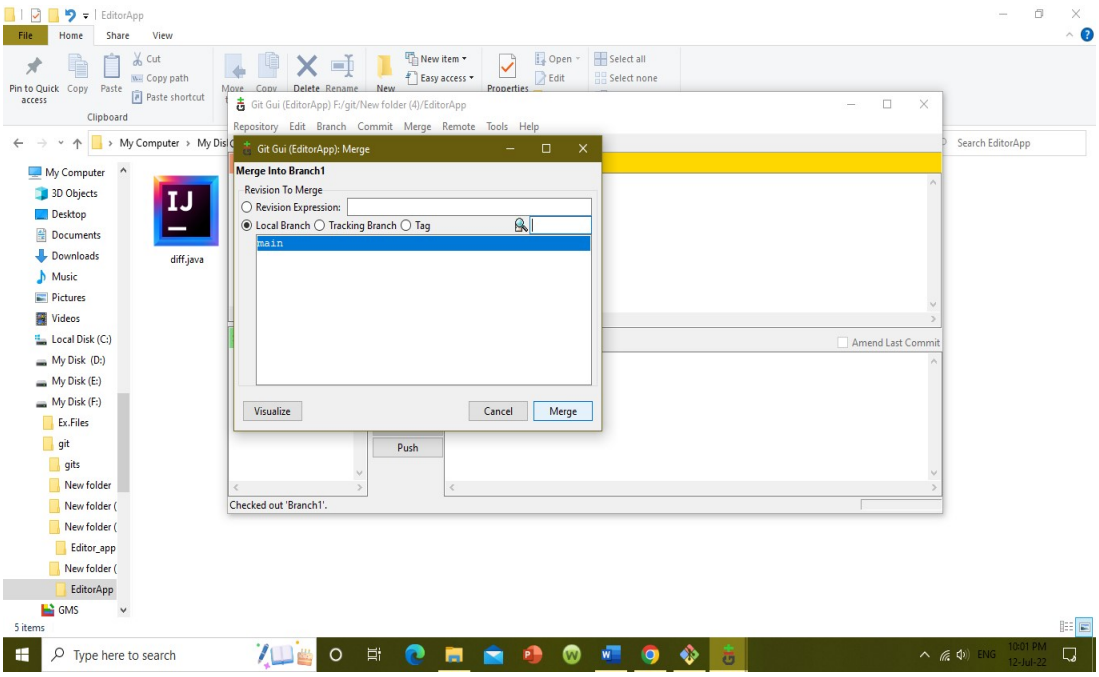


- So create a branch of any name let say “branch1”.

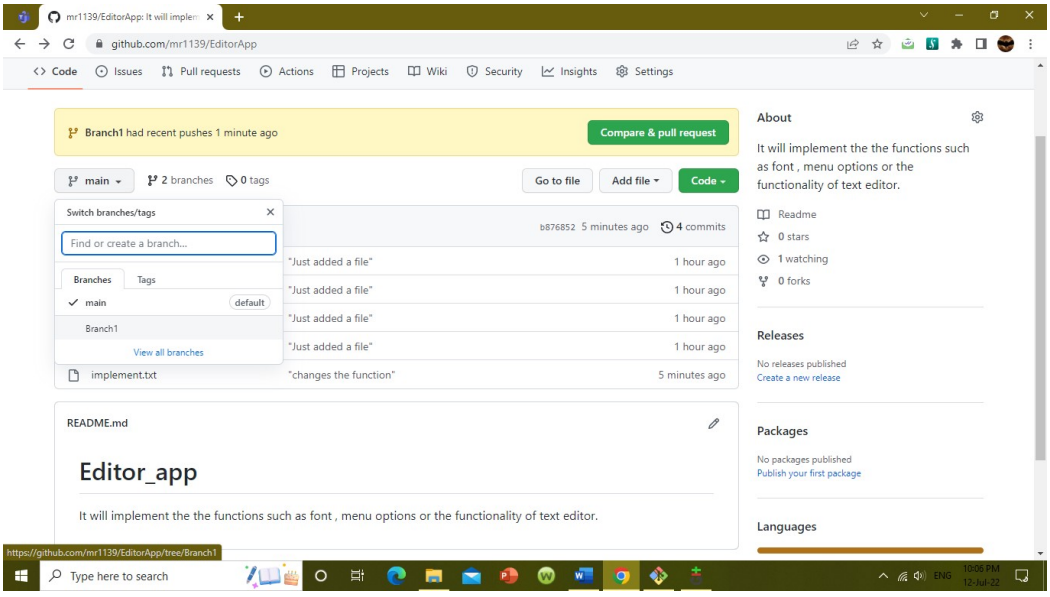


- you can see at the “current branch : branch1” , we are in the branch 1 .
- so this how we can create multiple branches .
- for merging we have to checkout the main branch and merge it with the branch1 , by clicking on the “local merge”.





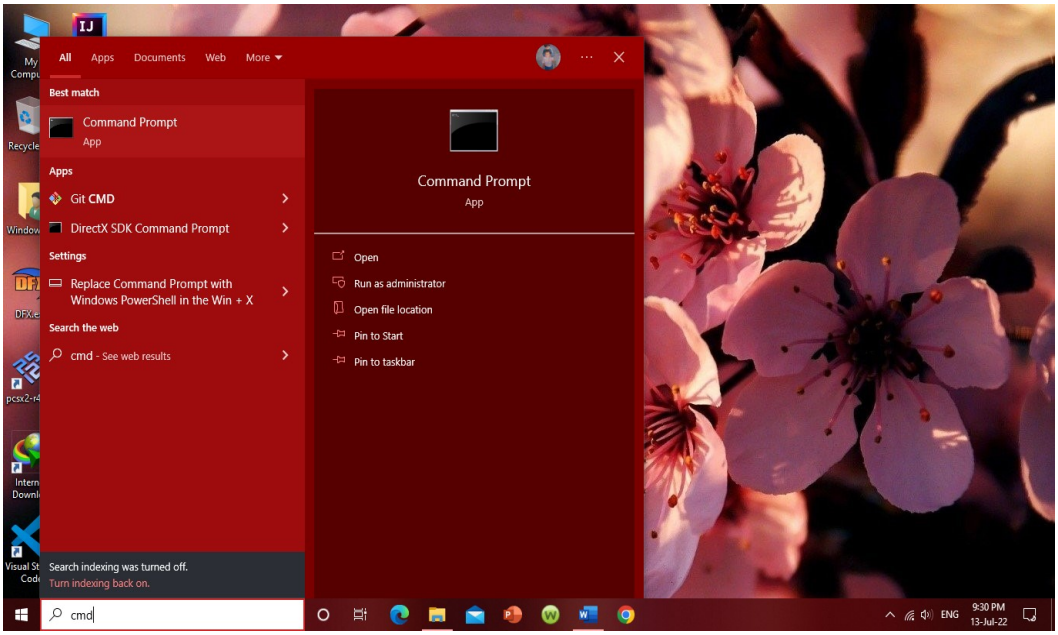
➢ As you can see in the github new branch i.e branch1 has been created .

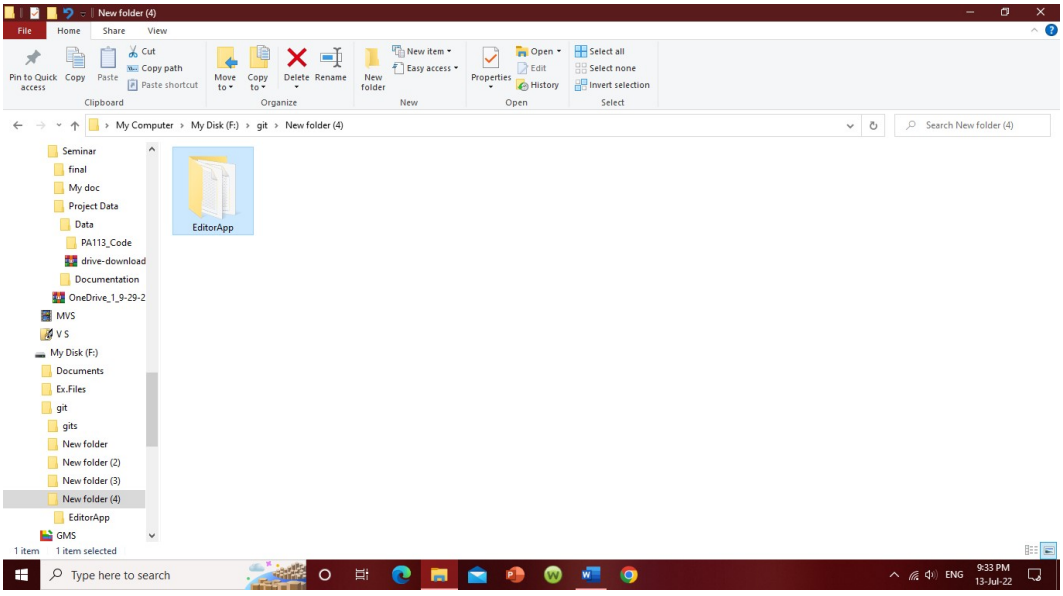


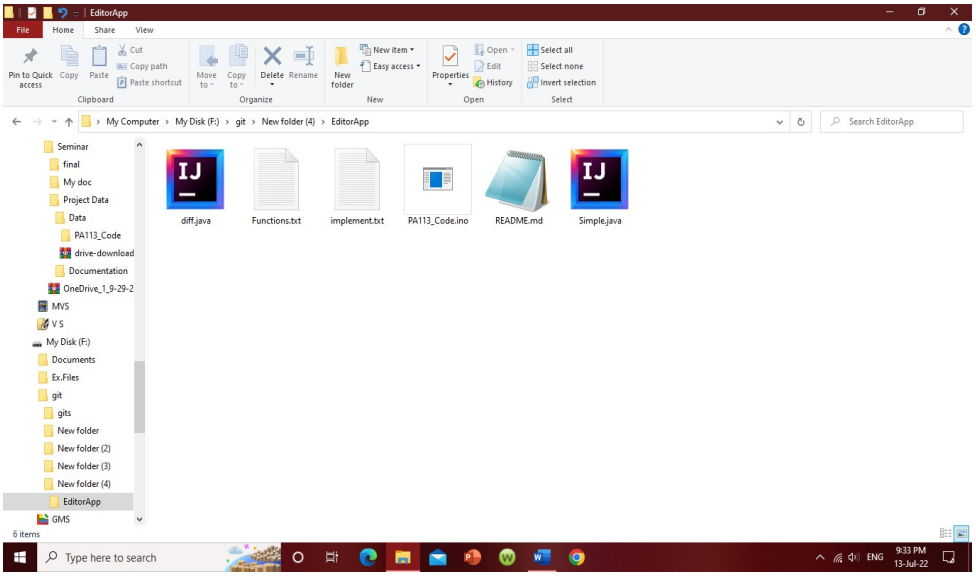
- And it has been merged into the main branch.
- So this is how we have created and uploaded our project using Git-Gui.

Using Command prompt (cmd):

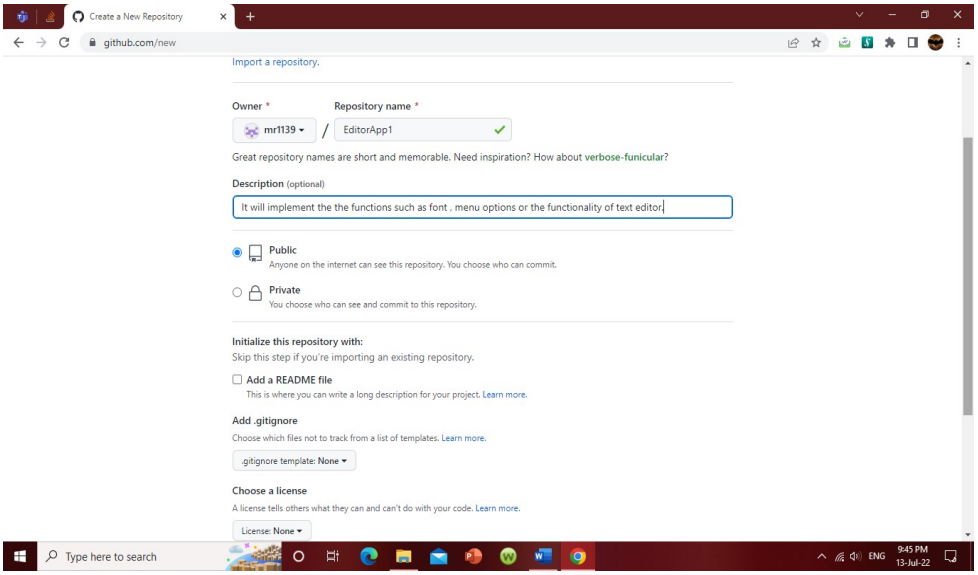
- So we are going to use command prompt .
- Click on windows button and search for “cmd” , and open it.
- Now we have our project files/documents in our local directory.



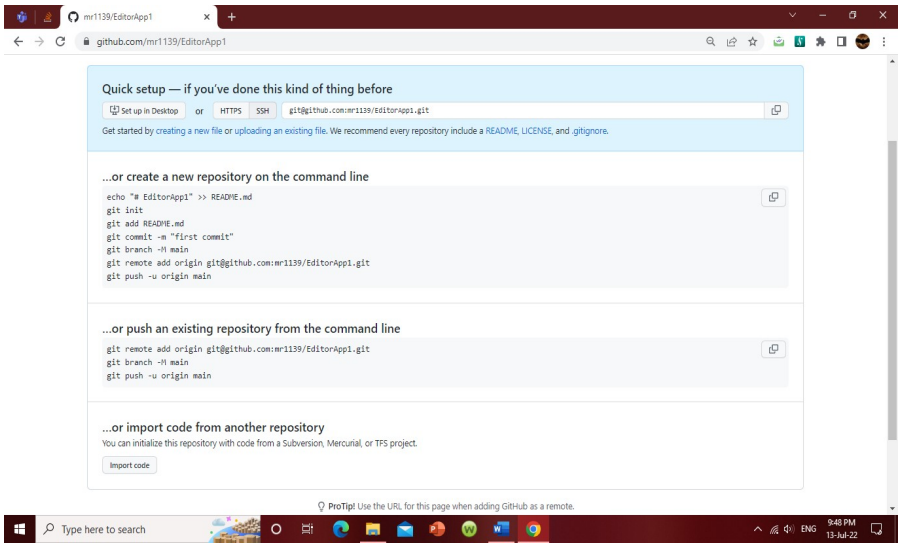




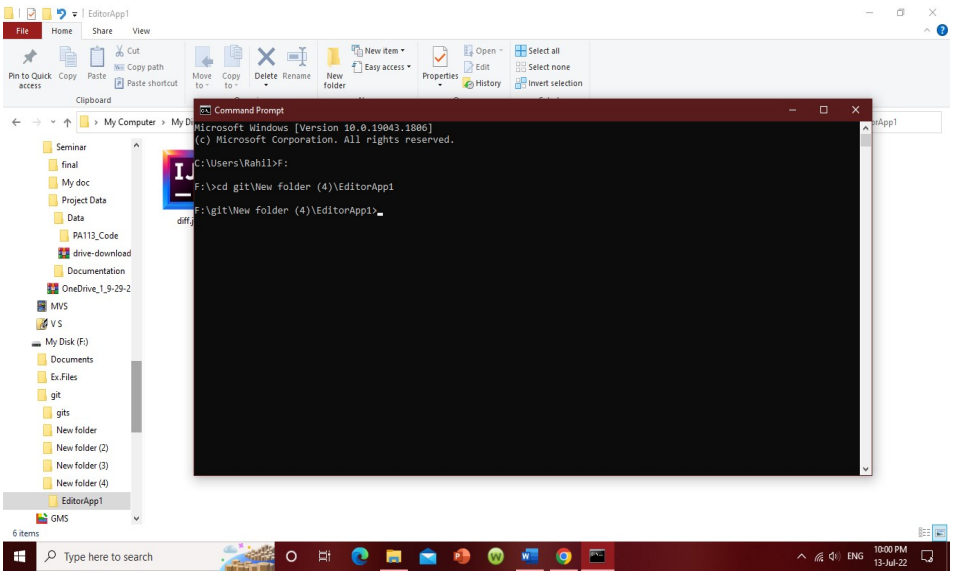
- Now we have to push/upload this project folder in to my github repository using command prompt .
- So for this we need to have github account .
- Once the github account is created simply go ahead and create repository .
- For creating repository in the github , go to “your repository” and click on “new” . create new repository.



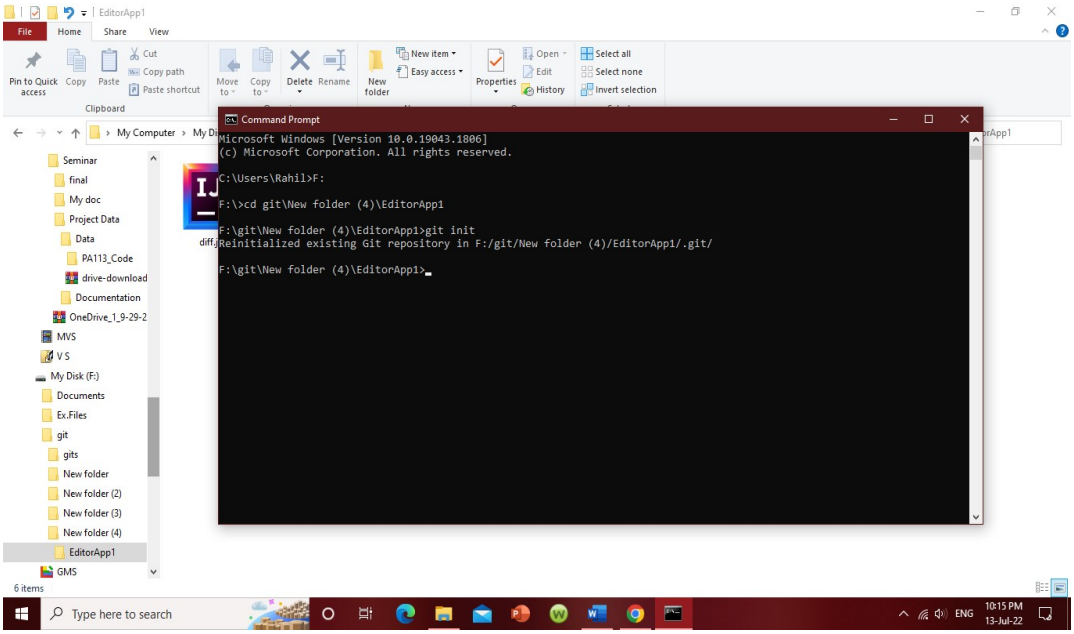
- now we will get the our repository path i.e in HTTPS or in SSH.
- As we have our project folder local directory , and open the command prompt.

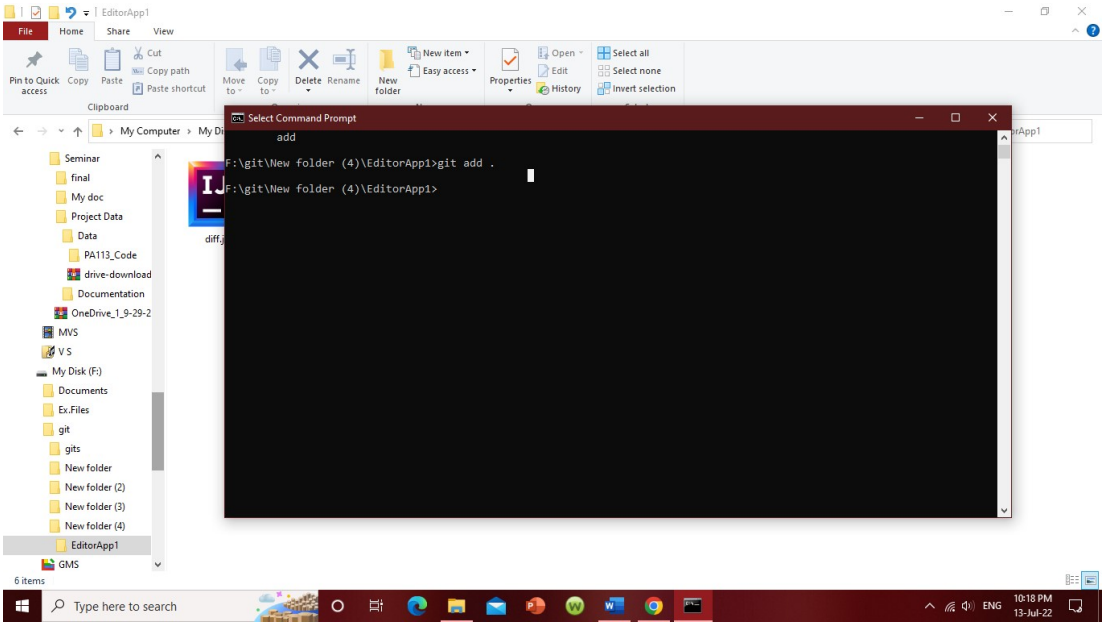


- Now open the command prompt and go to the directory where the project file/document is present using cd(change directory) command .

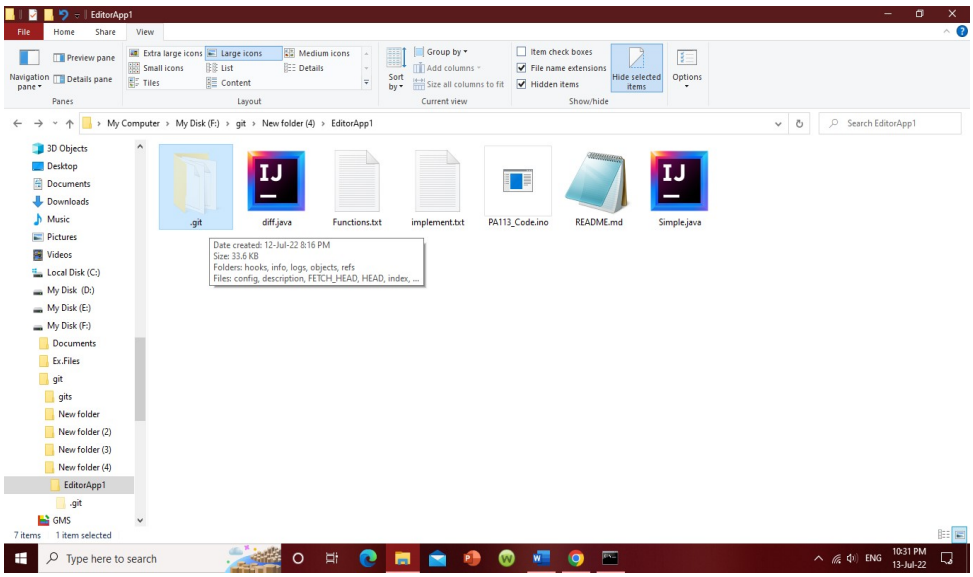


- Now you can see we are in our local directory , where our all file and documents are present.
- Now in this here we have to initialize empty git repository using the command "**git init**" .
- And after this we have add all the files and the documents in the repository using the command "**git add.**" Which will add the files and documents into our created local repository or we can say it will add a change in the working directory to the staging area.

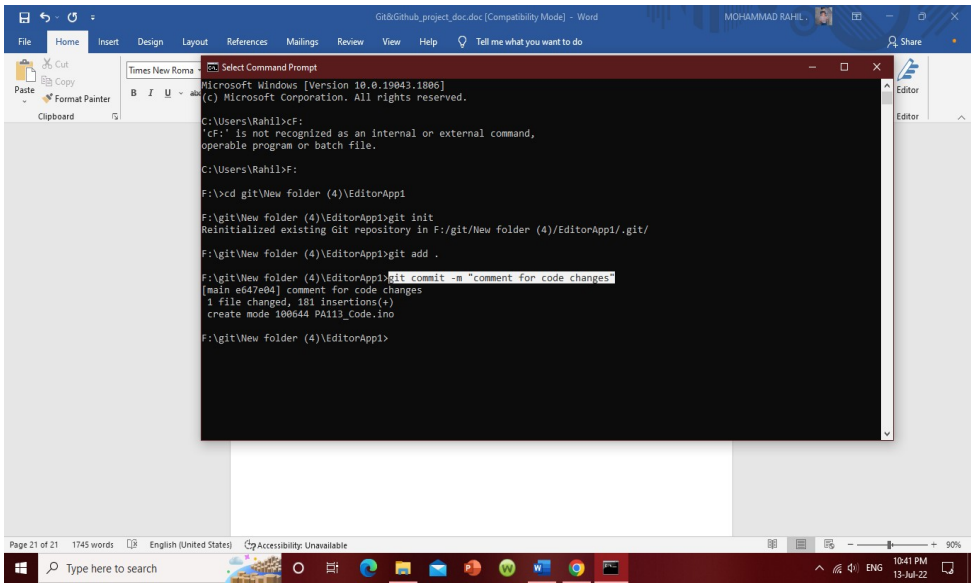




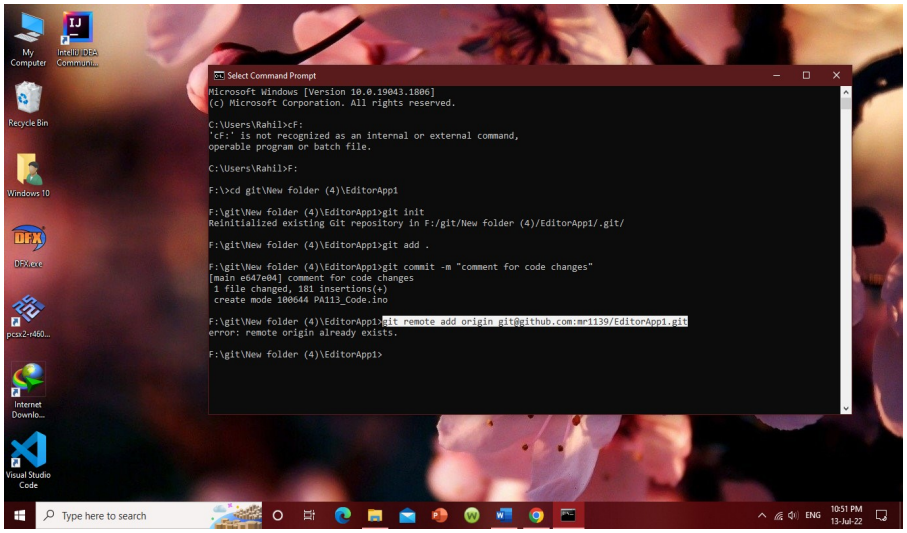
- Now you can see we have initializes the git repository into our existing project folder.



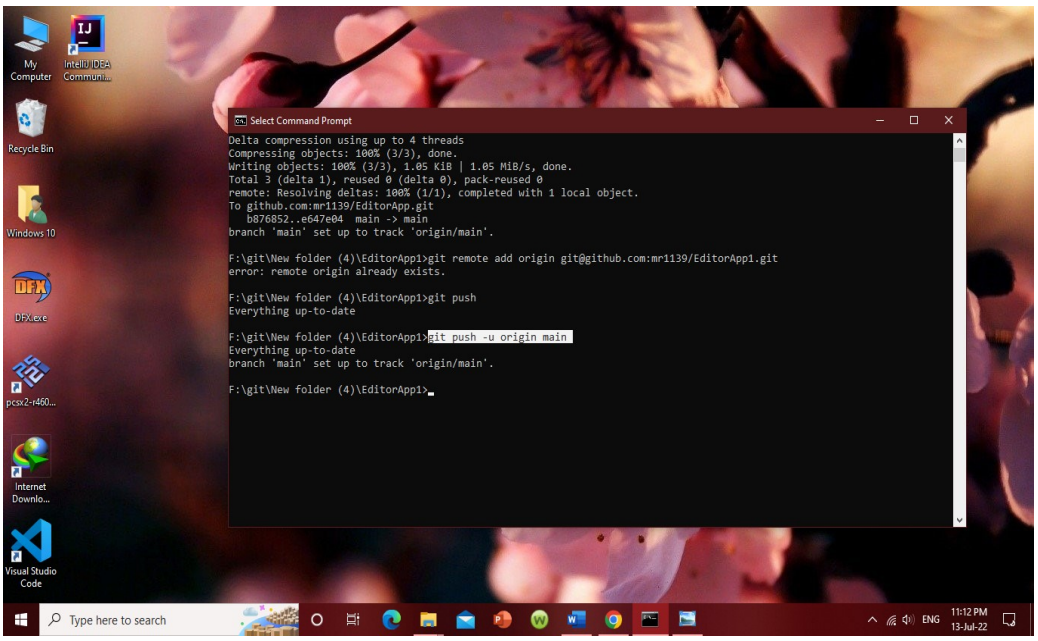
- Now we will use git commit command “git commit – m “message” , means adding commits keep track of our progress and changes as we work .



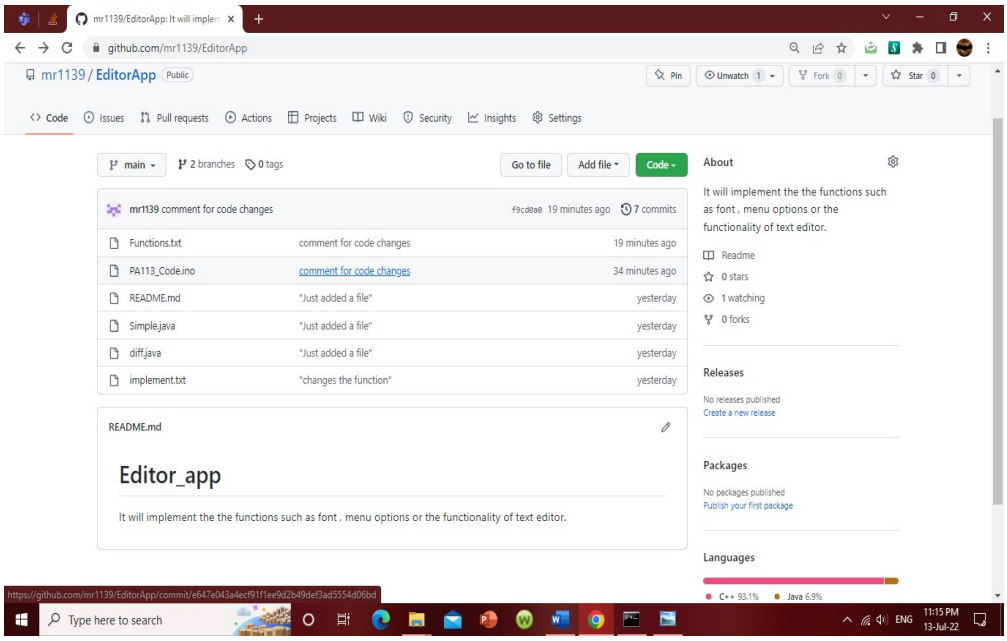
- Also we will use the command “git remote add origin HTTPS path” to add our remote repo.

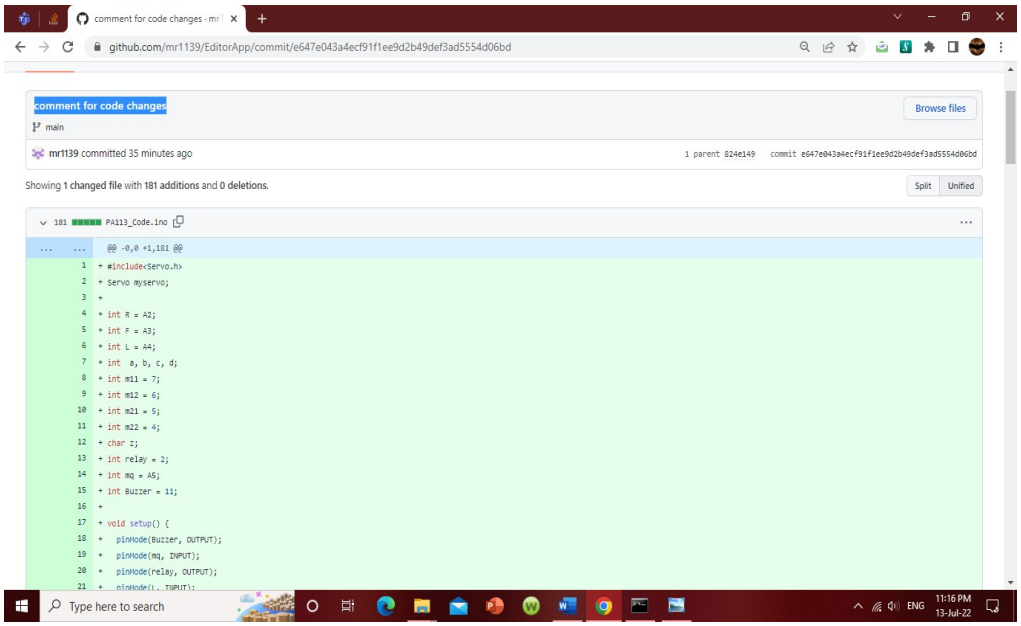


- after this we can use the command “git push -u origin main”
- It explicitly specifies to be pushed into a repository called origin.
- Git push origin is usually used only where there are multiple remote repository and you want to specify which remote repository should be used for the push .



- Now , in the git-hub if you refresh the page , you can see all the files / documents , code has been added there , and whatever the changes we have done it will showing in the commit message as “comment for code changes”





Conclusion :

So, the Git and GitHub provide fast and convenient ways to track projects, whether the project is by one individual or a team of software developers.

In this we have done with creating and uploading projects on Github using the Git Gui and by using the command prompt , both has been successfully completed .