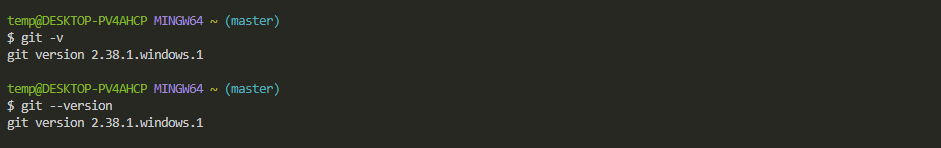
**GIT AND GITHUB ASSIGNMENT**

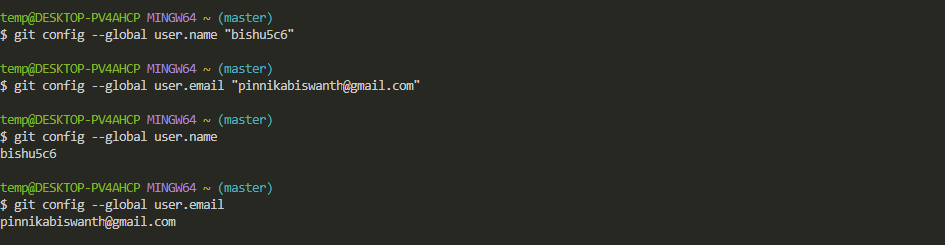
* To know the version of the git use these command
* Git –version or git -v



2.To set your global name and email in git:

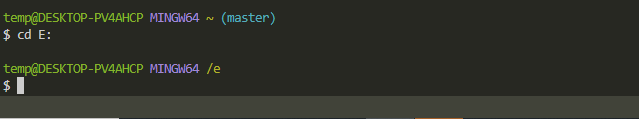
Git config –global user.name “yourname”

Git config –global user.email “youremail”



3.To create a project in a specific folder or location you want instead of default browser:

Cd foldername:



4.To create a new directory in that folder

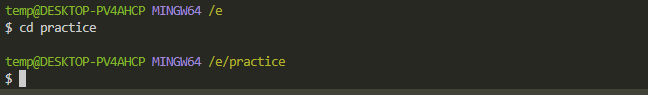
Mkdir practice

A picture containing text

Description automatically generated

5.To read that specific folder

Cd practice



6. to create a file in that folder

Touch filename.txt



7.To insert something in that file use vi editor

Vi filename.txt

Text

Description automatically generated

A picture containing text

Description automatically generated

8.Try to enter the test you want to print

After inserting the data

Press esc :wq to save and exit the file and :wq! To exit without saving the file..

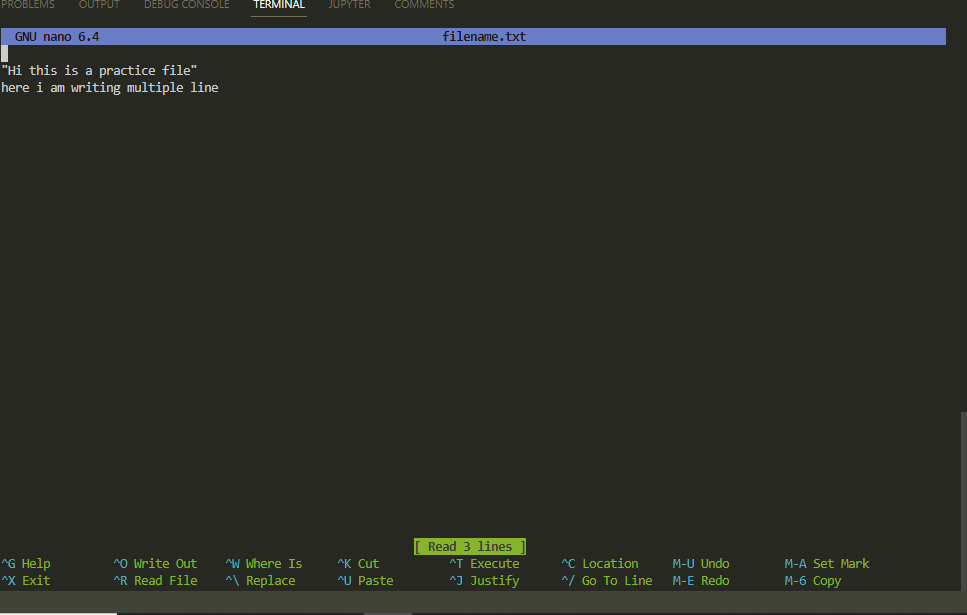
A picture containing graphical user interface

Description automatically generated

9.Another way to edit the file is :

Nano filename.txt

It will open like this:



Write something in that..

And press ctrl+x to close the file..

10.To go back to the previous working directory

Press cd ..

Keeping writing the command until u go back to the default working directory..

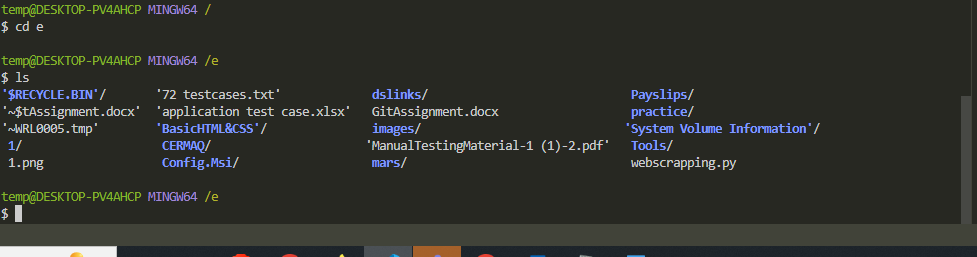
Graphical user interface, text

Description automatically generated

11. To get all the repos in that working directory..

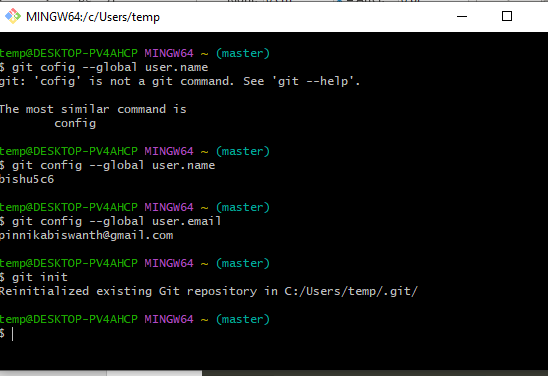
Press $ ls

It will give all the existing files in that repo..



12.After configuring user name and email you have initialize the git in it, To initialize use

Git init



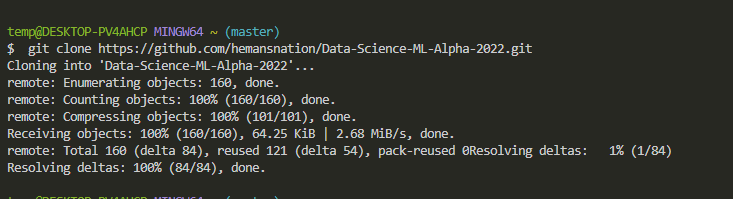
13. To know the about the branch that you are using

Git branch it will shows what kind of existing branch u are in.

Graphical user interface, text, website

Description automatically generated

14. To cloning a particular repository



15. to get the status of the git

Git status

Text

Description automatically generated

16. To unset the global user name or email..

Text

Description automatically generated

**TASK – 2**

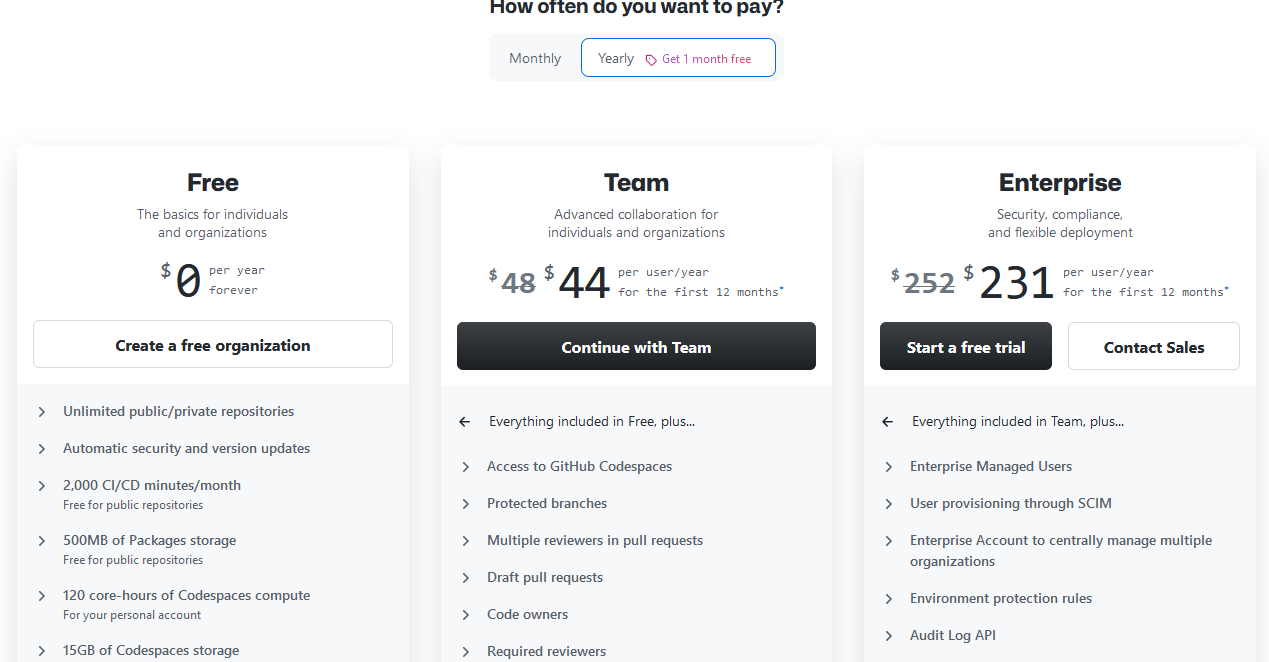
Creating an open source project in an organization with minimal permission for all the users to access it.

Click on the plus symbol there click on new organization

Graphical user interface, text, application, chat or text message

Description automatically generated

You will get a pop up like these:

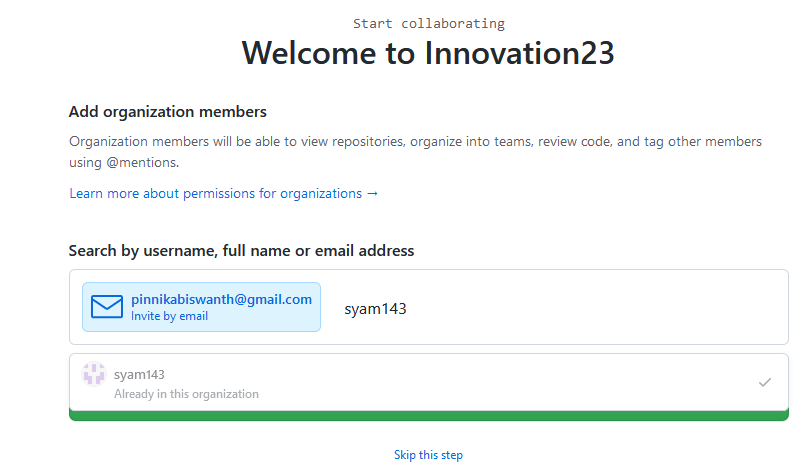


You will get a layout like these select one based on your interest..

Graphical user interface, text, application, email

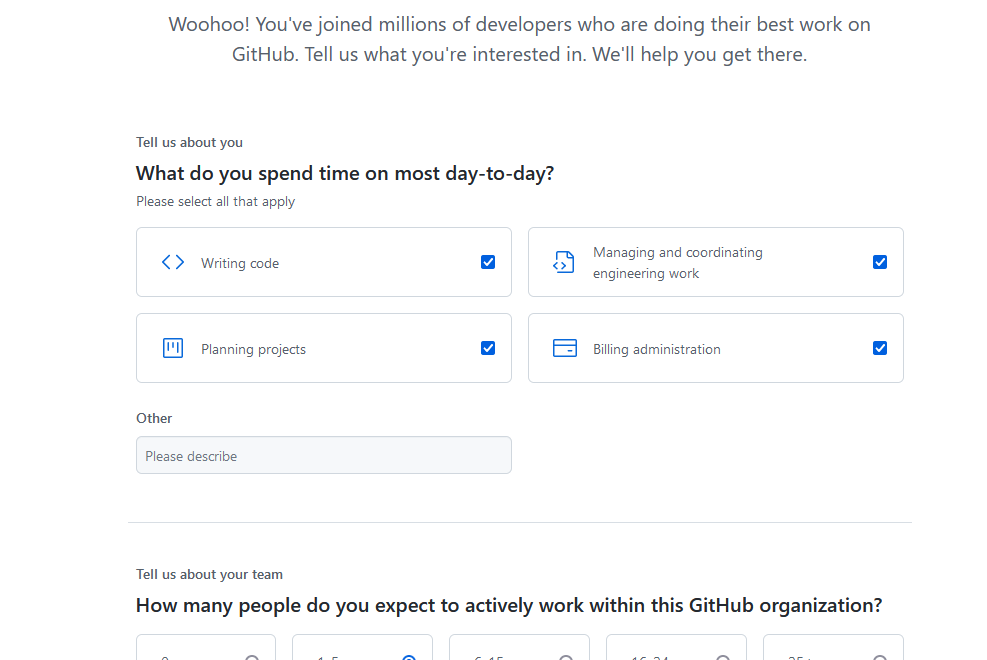
Description automatically generated

Set up your organization account like these and give permission.



Send an invitation to your friends who want to collaborate with them..

Give all the permissions you want to give and click on submit..



Here is my final created open source project:

<https://github.com/Innovation23>

TASK – 3

1.Creating a issues in your github repository

Go to your repository and open it, there u will see issue click on issues tab to create a issue and submit it.

It will be seen as below.

Graphical user interface, text, application, chat or text message

Description automatically generated

Click on new issue to create a issue

Graphical user interface, text, application, email

Description automatically generated

Type the issue and submit it

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

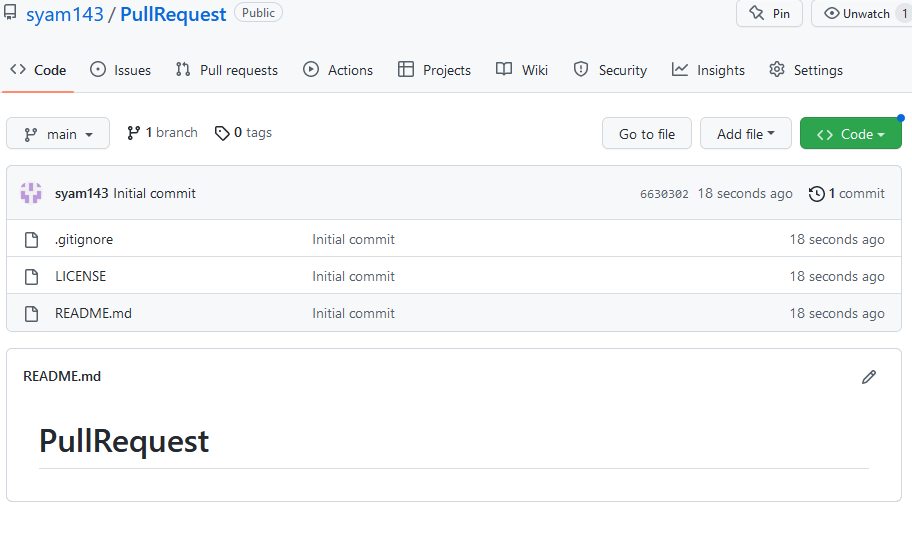
Description automatically generated

Issue created successfully you can close the issue and open the issue also

2.Raising, Merging and rejecting a Pull Request:

When two developers are working on a same project in an organization, they basically created their own code, after completion of their work they want to merge their code with master branch, this is where pull request comes into the scenario.

The developer who wants to merge the code with the master branch will raise a pull request. After raising a pull request the master branch side will check whatever the changes they made and merge with the master branch with the help of pull request.



I have created a repo named pull request which contains only one branch that is in main branch

Now I am creating another branch and named it as syam.

Graphical user interface, text, application, email

Description automatically generated

As we can see we have a created a branch name syam and we are in it.

In these branch I am opening readme file and doing some changes to it.

I have done some changes and committing it.

Graphical user interface, text, application, email

Description automatically generated

Now I am going to pull request to change it into the main branches.

Graphical user interface, text, application, email

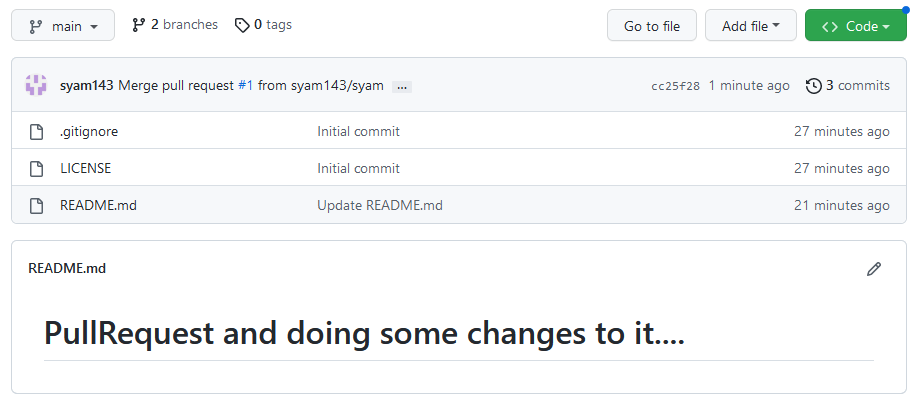
Description automatically generated

As we can see we are able to see the compare and pull request. Whenever we do changes in the one branch by clicking on the pull request you will automatically raise a pull request.

Graphical user interface, text, application, email

Description automatically generated

Raising a pull request on branch syam.



As we can see we successfully raise a pull request and deleted it.

Reject pull request with comment:

To reject a pull request under your repository goto pull request.

Graphical user interface, text, application, email

Description automatically generated

Now the below you a window named close with comment by clicking on it you will Reject the pull request..

After rejecting the pull request it will show like this…

Graphical user interface, text, application, email

Description automatically generated

You can also reopen the pull request.

Enabling a depandabot in Github

Click on the repository and goto settings

Graphical user interface, text, application, email

Description automatically generated

Click on code security and analysis and activate dependabot it will give you all the alerts for you repo.

Stashing in Github:

Suppose you working on the branch and you created a new branch and start working on it and suddenly you want to move to the main branch, we cannot move because you didn’t commit the changes and you didn’t want to commit and move to the main branch use stash with the help of stash you can easily move from one branch to another branch without committing.

To stash use the command git stash

Text

Description automatically generated

I created a branch named dev and modified the code now I will try to change the branch..

Text

Description automatically generated

By using the stash I able to save the work.

Releasing you Package.

1.go to your repository

To the right of the list of files, clickreleases

Graphical user interface, text, application

Description automatically generated

Click on create a new release

And to create your own package click on the publish your first package.

Graphical user interface, text, application

Description automatically generated

You will get the next body like this:

Graphical user interface, application

Description automatically generated

If you want to release the package choose which one you want to use.

To release the project click on the release

You will get the page like these

Graphical user interface, text, application, Teams

Description automatically generated

Click on publish release your won package.

Released successfully

Text

Description automatically generated with low confidence

Setup a Projects Board for your project.

Go to your profile and then click on projects.

Graphical user interface, text, application

Description automatically generated

Click on new project you will get a project board like these

Graphical user interface, application

Description automatically generated

Add projectname and you can add repository from your project and add assignees the names of your group numbers you can also check the status also.. by assigning each task to a specific person.

Table

Description automatically generated with medium confidence

In these way you can create a project..