<https://guides.github.com/activities/hello-world/>  
Go through once you’re done with this.

**GIT**

> helps to keep different version of the same file.

> anyone can use it, but mostly used by programmers to MANAGE source code.

**GIT BASH**

> git --version

GIT configuring username and email id  
> git config --global user.name "Rahul Chawla"  
> git config --global user.email "rahulchawla2801@gmail.com"

> git config --list   
(for viewing details)

> git config user.name  
(to view one particular detail)

> clear   
(to clear screen)

HELP

> git help  
(for the list of different keywords)

>git help commit  
(will open details of it in the browser)

Directory  
  
>pwd  
(to know the current directory)  
  
> cd ..  
(to go back)

> cd folder/filname  
(to go front)  
  
 >ls  
(to view the list)

To navigate to C:\Users\rahul\Desktop\smsApp from C:\Users\rahul

>pwd  
>cd Desktop OR > cd Desktop/smsApp  
>ls  
>cd smsApp

NOW that our path is C:\Users\rahul\Desktop\smsApp which is our project, we can make it a git file or repository by  
>git init

working area --> staging area --> repository  
  
> git add .  
(to tell git that we added ALL the files to staging area)  
> git add filename.txt  
(to add particular file to staging area)

> git commit -m "First version of smsApp"  
(to save our project/repository till a particular point, thus whenever we want to come back to that point/version, we can rollback. Message in quotes is just used to identify which version is it. It takes files from the STAGING AREA and puts them in the REPOSITORY )  
  
  
> git log  
(to visit commit history)

> git log --author="Rahul"  
(to visit commit history by a particular author)  
  
> git status  
(gives you new files or files with modified data and haven't been committed/untracked)

**EXAMPLE**  
> file1.txt and file2.txt are newly created  
> git add file1.txt  
(file1.txt is in staging area )  
> git commit -m "version 1.01"  
(adds only file1.txt in repository)

**View difference between a file in working area and repository**   
1. file1.txt is commited in repository as "version1"  
2. file1.txt is modified, is in working stage.  
  
> git diff  
(will giv the difference)  
  
(If modified file1.txt is commited or put in staging area, it won't show a difference!)

**View difference between a file in staging area and repository**>git diff --staged

DELETE a file

> git rm file3.txt  
> git commit -m "version 1.02, deleted file3.txt"  
(removes file from repository as well as in the computer)  
(but if you want the file3.txt back, just rollback till version 1.02)

RENAME a file  
> git mv file1.txt newFirst.txt  
(renames the file, DO commit after that)  
  
  
MOVE a file to a folder  
>git mv file1.txt folder1/file1.txt  
(moves file1.txt to folder1. You can also change the name egs. folder1/newfile.txt)  
  
  
  
  
Working area to repository DIRECTLY  
> git commit -am "Version 1.0.22"  
(it commits all the files together. Thus if you dont wanna send ALL files to the repository, its better to go via staging area, i.e. put few files in staging area and only commit those)  
  
  
  
Undo few changes, i.e sending files from Repository to Working area  
> git checkout -- file1.txt  
  
Example-  
You made a change in file1.txt and file2.txt and you wanna undo changes made in file1.txt.  
> git checkout -- file1.txt  
> git add .  
> git commit -m "version 1.0.12"   
  
  
**> git checkout 01e7dba -- file1.txt**  
(undoing changes by sending files from OLDER repository to the working area. The id of the previous repository is 01e7dba......)

Unstage file i.e. Staging area --> Working area  
> git reset HEAD file1.txt  
  
  
  
  
  
  
**WORKING ON GITHUB**> Create new repository on github.com  
>Uncheck-->initialise the repository using README  
>Copy the URL which will help to link the github.com repository with your local repository in your system.  
  
TO PUSH an existing repository from local system to the github.com  
> git remote add origin <https://github.com/rahulchawla2801/Sports-Management-System.git>  
> git push -u origin master  
(origin is the name given by the user for the link)

**---Bash points to your system repository.  
---URL or origin points to your github.com repository**