

CHITKARA
UNIVERSITY



SUBMITTED BY:

NAME: Diwakar Papola

ROLL NO. : 2110990470

Group: G8

SUBMITTED TO:

Dr.Monit Kapoor

INSTALLATION OF GIT IN LINUX(UBUNTU)

```
diwakar@diwakar-HP-Laptop-15-da0xxx: ~  
diwakar@diwakar-HP-Laptop-15-da0xxx:~$ sudo apt-get install git -->THIS COMMAND IS FOR INSTALLTION OF GIT  
[sudo] password for diwakar:  
Sorry, try again.  
[sudo] password for diwakar:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
git is already the newest version (1:2.25.1-1ubuntu3.2).  
The following package was automatically installed and is no longer required:  
  libfwupdplugin1  
Use 'sudo apt autoremove' to remove it.  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
diwakar@diwakar-HP-Laptop-15-da0xxx:~$ git --version --TO CHECK THE GIT VERSION INSTALLED IN OS  
git version 2.25.1
```

- NOW WE HAD SUCESED TO INSTALL GIT IN OUR SYSTEM
- NOW WE NEED TO ADD OUR EMAIL AND CREATE A USERNAME FOR OUR GIT---

ADDING EMAIL AND USERNAME TO OUR GIT

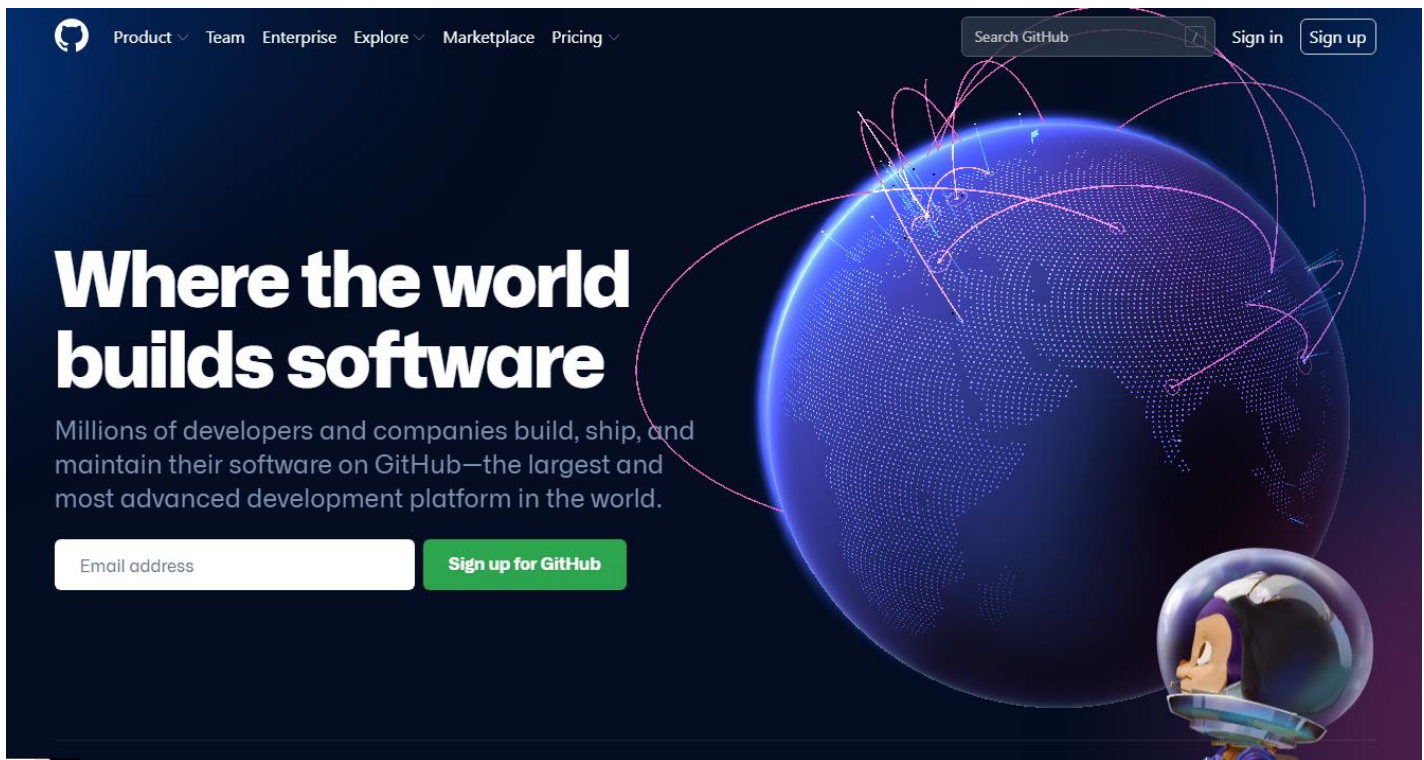
```
diwakar@diwakar-HP-Laptop-15-da0xxx: ~  
diwakar@diwakar-HP-Laptop-15-da0xxx:~$ $ git config --global user.email "eparis@atlassian.com" ---> TO SET THE EMAIL TO GIT  
$: command not found
```

WE CAN LIST OUR GIT USERNAME AND EMAIL USING FOLLOWING COMMANDS AND WE CAN ALSO EDIT THE DETAILS WHENEVER WE WANT USING THE FOLLOWING COMMANDS→

```
diwakar@diwakar-HP-Laptop-15-da0xxx:~$ git config --list --> TO GET THE LIST OF AUR GIT USERNAME AND EMAIL
user.name=diwakar4586
user.email=diwakar0470.be21@chitkara.edu.in
diwakar@diwakar-HP-Laptop-15-da0xxx:~$ git config --edit --> TO EDIT THE NAME OR EMAIL OF GIT
```

SETTING UP GITHUB ACCOUNT →

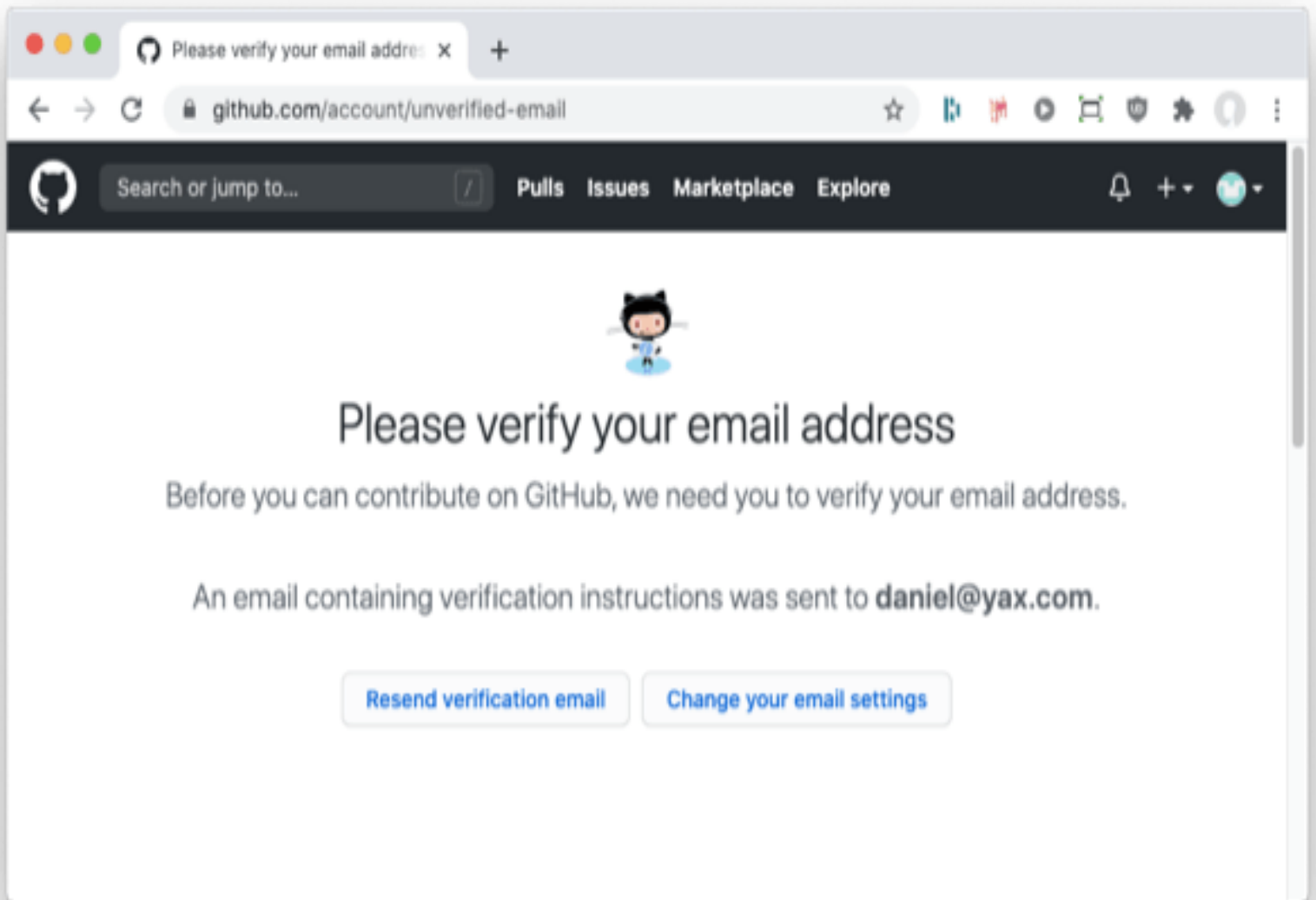
1. ACCOUNT –SETUP→ To Sign up for an account on Github.com ,got the <https://github.com/>



2. Verifying your email address→

To ensure you can use all the features in your Github plan,verify

Your email address after signing up for a new account.



STEP2→ After clicking **new repository** option, we will have to initialize some things like, **naming our project**, choosing the **visibility** etc. After performing these steps click **Create Repository** button.

Step1→ Creating a repository

The screenshot shows the GitHub 'Create a new repository' page. A red arrow points from the 'New repository' option in the top navigation bar to the page. Another red arrow points to the 'Repository name' field, which contains 'Resume', with a green annotation: 'This is going to be name of our project'. A third red arrow points to the 'Public' radio button, with a green annotation: 'Keep this as public selected'. A fourth red arrow points to the 'Description (optional)' text box, with a green annotation: 'We can add a project description if we want.'. A fifth red arrow points to the 'Initialize this repository with a README' checkbox, which is checked, with a green annotation: 'Tick the README option'. A sixth red arrow points to the 'Create repository' button at the bottom, with a green annotation: 'After performing above steps, Click this button'. The page also shows the owner 'Namanbhatia7' and the option to 'Add a license: None'.

Marketplace Explore

New repository
Import repository
New gist

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner: Namanbhatia7 / Repository name: Resume ✓

Great repository names are short and memorable. Need inspiration? How about [animated-memory](#)?

Description (optional)

☒ **Public** Keep this as public selected
Anyone can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☒ **Initialize this repository with a README** Tick the README option
This will let you immediately clone the repository to your computer.

Add .gitignore: None | Add a license: None ⓘ

Create repository After performing above steps, Click this button

BASIC COMMANDS TO ADD FILE AND COMMIT THEM

```
diwakar@diwakar-HP-Laptop-15-da0xxx:~$ ls -- TO LIST ALL THE FILES OF DIRECTORY
apport.save      cmpt100          countdown_project Desktop  'github token'
'c++ course coding ninjas' 'c++ new course' Cpp-Programms  Documents git_workspace
'c++ important stuff' 'coding ninjas course' cppsolutions  Downloads javascript

diwakar@diwakar-HP-Laptop-15-da0xxx:~$ cd Programs_01
diwakar@diwakar-HP-Laptop-15-da0xxx:~/Programs_01$ ls
2darrays.cpp  arraymodule.cpp  exercise.cpp  'lec7 Bitwiseoperators.cpp'  lecture1.cpp
array3.cpp    classs.cpp       functionexercise.cpp 'lec 8 array.cpp'          lecture2.cpp

diwakar@diwakar-HP-Laptop-15-da0xxx:~/Programs_01$ git init --TO INITIATE GIT IN THE DIRECTORY
Initialized empty Git repository in /home/diwakar/Programs_01/.git/

diwakar@diwakar-HP-Laptop-15-da0xxx:~/Programs_01$ git status --TO CHECK(OR TRACK) THE STATUS OF OUR FILES
On branch master --MASTER IS THE INBUILT BRANCH

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  2darrays.cpp
  array3.cpp
  arraymodule.cpp
  classs.cpp
  exercise.cpp
  functionexercise.cpp
  lec 8 array.cpp
  lec7 Bitwiseoperators.cpp
  lecture1.cpp

THESE FILES ARE IN RED COLOR BECAUSE THESE FILES NEED TO ADD AND GET COMMITTED
```

Figure 1.1

SO IN THE FIGURE 1.1 → We initialize a folder by adding git init

And then we added some files and as we can see those files having red color so it means they are not yet committed they just added to track them so we need to add commit them.

Let see how commit is done in fig 1.2 given below--

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

```
new file:   2darrays.cpp
new file:   array3.cpp
new file:   arraymodule.cpp
new file:   classs.cpp
new file:   exercise.cpp
new file:   functionexercise.cpp
new file:   lec 8 array.cpp
new file:   lec7 Bitwiseoperators.cpp
new file:   lecture1.cpp
new file:   lecture2.cpp
new file:   lecture3 pointer.cpp
new file:   lecture4 loops.cpp
new file:   lecture5 patterns.cpp
new file:   lecture6 character pattern.cpp
new file:   newexercises.cpp
```

```
diwakar@diwakar-HP-Laptop-15-da0xxx:~/Programs_01$ git commit -m "Initial commit"
```

```
[master (root-commit) 6806e6c] Initial commit
```

```
15 files changed, 2180 insertions(+)
```

```
create mode 100644 2darrays.cpp
create mode 100644 array3.cpp
create mode 100644 arraymodule.cpp
create mode 100644 classs.cpp
create mode 100644 exercise.cpp
create mode 100644 functionexercise.cpp
create mode 100644 lec 8 array.cpp
create mode 100644 lec7 Bitwiseoperators.cpp
create mode 100644 lecture1.cpp
create mode 100644 lecture2.cpp
create mode 100644 lecture3 pointer.cpp
create mode 100644 lecture4 loops.cpp
create mode 100644 lecture5 patterns.cpp
create mode 100644 lecture6 character pattern.cpp
create mode 100644 newexercises.cpp
```

↑
THIS COMMAND IS FOR COMMITTING OUR
ADDED FILE IN THE GIT

Figure 1.2 –

COMMITTING A FILE IN GIT I

As we already committed our files in the git .Now let see how can we push our git in github

Cloud sever.

```
create mode 100644 js_drumpadproject/com.wav
diwakar@diwakar-HP-Laptop-15-da0xxx:~/Content_01$ git branch -M firstbranch --THIS COMMAND IS FOR CREATE A NEW BRANCH
diwakar@diwakar-HP-Laptop-15-da0xxx:~/Content_01$ git remote add origin https://github.com/Group08-Chitkara-University/2110990470.git
diwakar@diwakar-HP-Laptop-15-da0xxx:~/Content_01$ git remote -v
origin https://github.com/Group08-Chitkara-University/2110990470.git (fetch)
origin https://github.com/Group08-Chitkara-University/2110990470.git (push)
diwakar@diwakar-HP-Laptop-15-da0xxx:~/Content_01$ git push -u origin firstbranch
Username for 'https://github.com': diwakar4586
Password for 'https://diwakar4586@github.com':
Enumerating objects: 35, done.
Counting objects: 100% (35/35), done.
Delta compression using up to 4 threads
Compressing objects: 100% (35/35), done.
Writing objects: 100% (35/35), 2.47 MiB | 780.00 KiB/s, done.
Total 35 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Group08-Chitkara-University/2110990470.git
 * [new branch] firstbranch -> firstbranch
Branch 'firstbranch' set up to track remote branch 'firstbranch' from 'origin'.
diwakar@diwakar-HP-Laptop-15-da0xxx:~/Content_01$ git status
On branch firstbranch
Your branch is up to date with 'origin/firstbranch'.

nothing to commit, working tree clean
diwakar@diwakar-HP-Laptop-15-da0xxx:~/Content_01$ git log
commit e9ceb08cfff3a50875b50d0b8bf9fb21f359bc2 (HEAD -> firstbranch, origin/firstbranch)
Author: diwakar4586 <diwakar0470.be21@chitkara.edu.in>
Date: Sun Apr 10 14:51:47 2022 +0530

    first commit
diwakar@diwakar-HP-Laptop-15-da0xxx:~/Content_01$ git branch -r ---> TO CHECK THE BRANCH STATUS
origin/firstbranch
diwakar@diwakar-HP-Laptop-15-da0xxx:~/Content_01$
```

↑
TO ADD THE PATH OF CLOUD WHERE THE GIT
WILL PUSHED LATER

↑
THIS COMMAND IS FOR PUSHING OUR FILE TO GITHUB CLOUD
SERVER

