**Lecture 1:** Git Intro

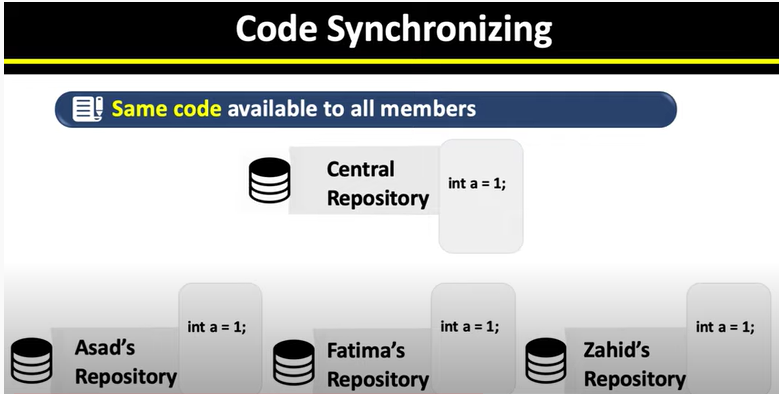
VCS: Version Control System is like keep tracking of all changes to code.

For example, if we write some code like

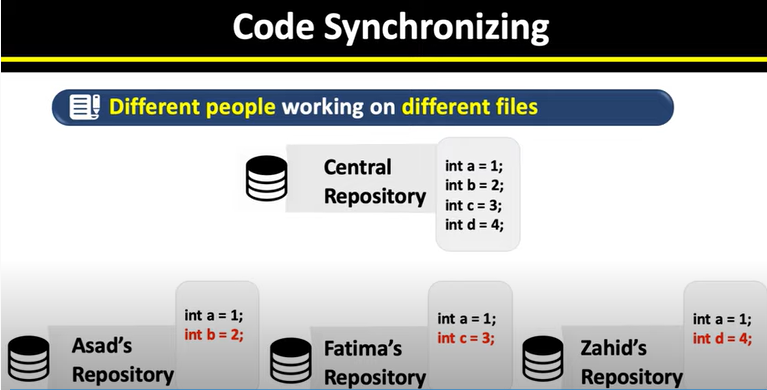
**File:1**

1. **Created file-1**
2. **Write some code:**
3. **Int x=12;**
4. **Int b=x+1;**
5. **After that, next time we require some change in the value of x let suppose we wang to make it 12 so we’ll do some changes like  
   (i) int x=~~12~~ 13;  
   (ii) int b=x+1**
6. **Now we want to keep track of change we just did in the code. Therefore, we require a *version control system.***
7. **Best way to keep track of all changes is to make an account on GitHub.**

**CODE SYNCHRONIZING BETWEEN DIFFERENT GROUPS:**

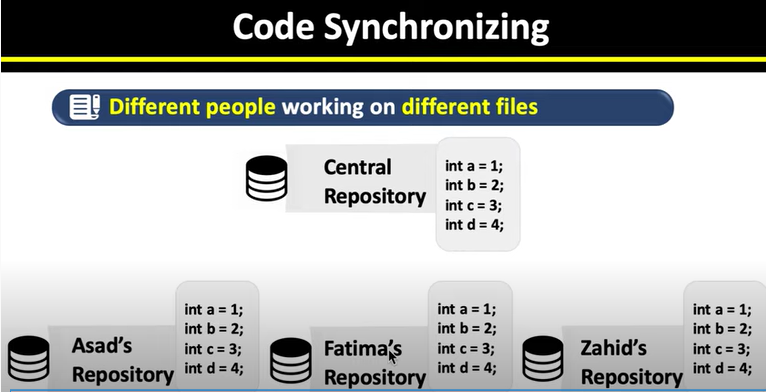


As you can see above there are three group members each have their own code.  
Also, we have a central repository.   
In VCS, we put all our code’s int this central repository so that everyone will have the same updated version of the same code.



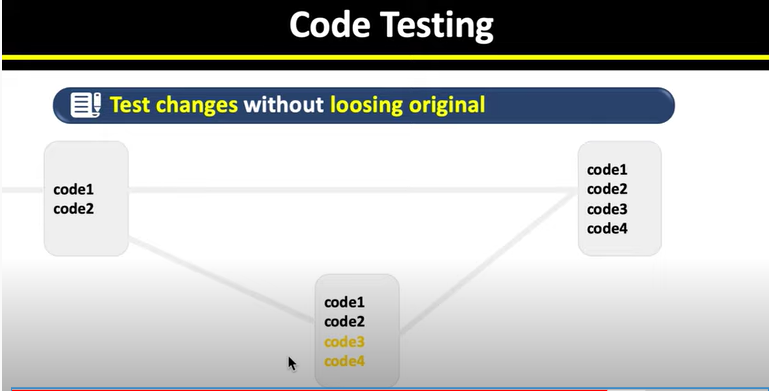
**As we can see, here we have all the updated code in out central repository.**

**Now everyone will have the same code/updated code in their systems.**



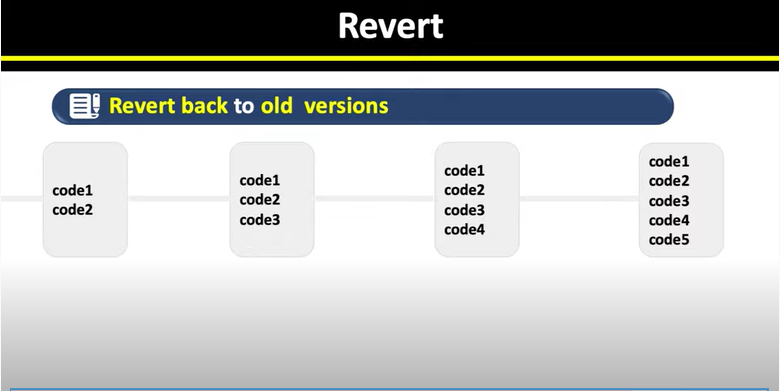
**Everyone now have the same code/updated code in their own systems.**

**TEST CHANGES WITHOUT LOSING ORIGINAL**



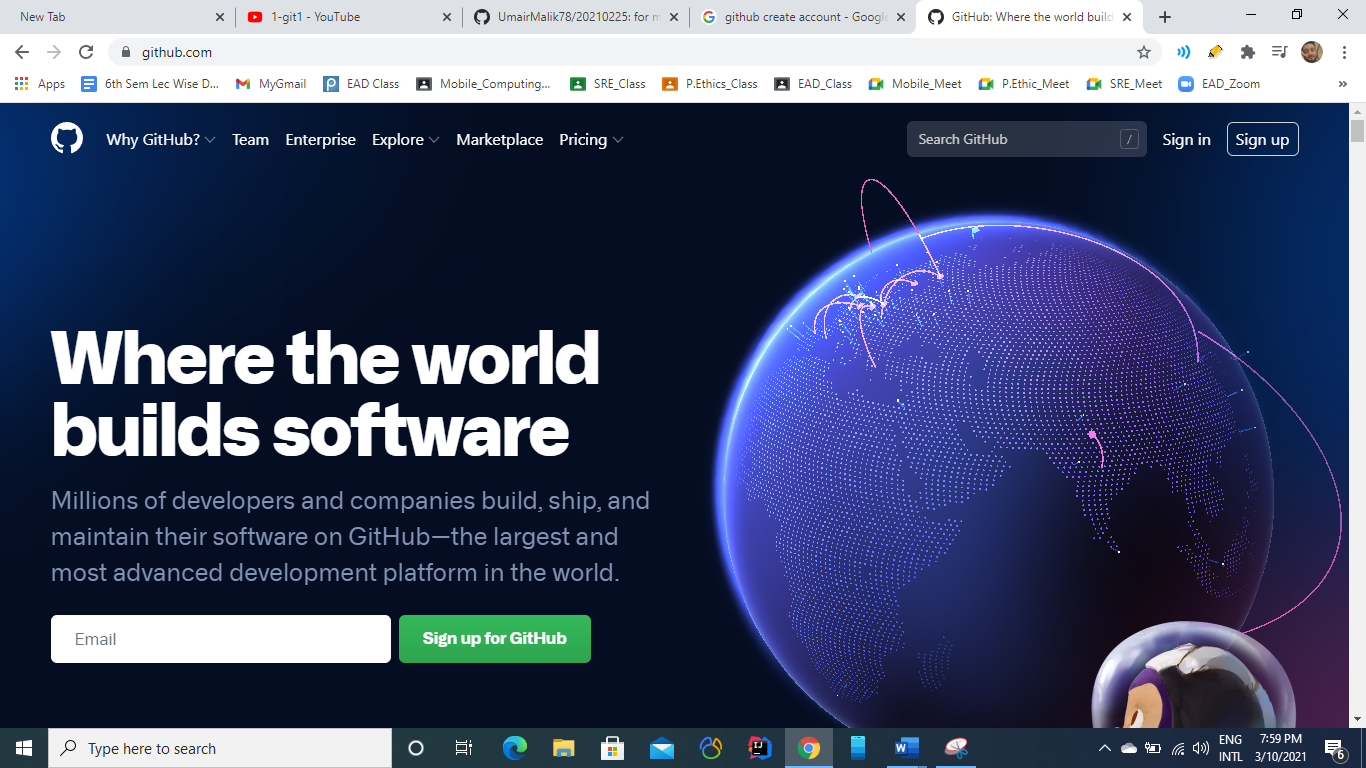
**Until we are not sure about the new added code, we can keep our previous code.  
After that, when we have tested our new code, we can add it to the original as shown in above figure.**

**REVERT BACK TO OLD VERSIONS FACILITY:**

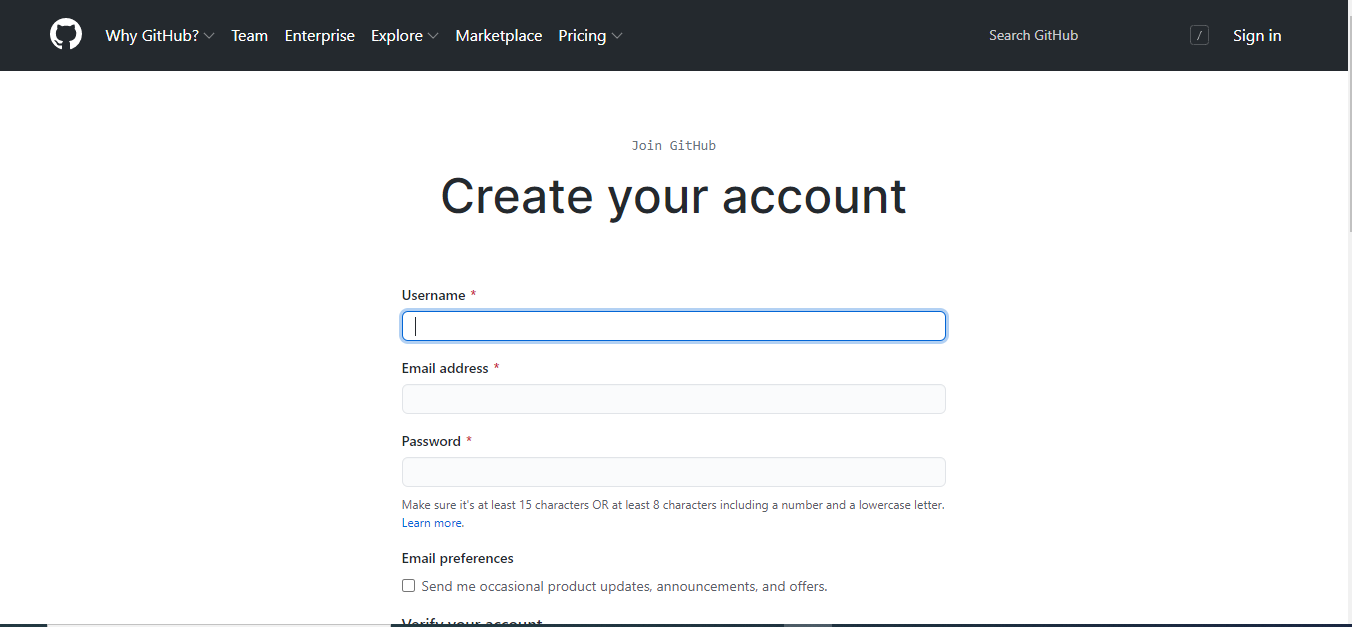


**We can also look back at our code or code history and can find revisions, upgradations, deletion etc easily.**

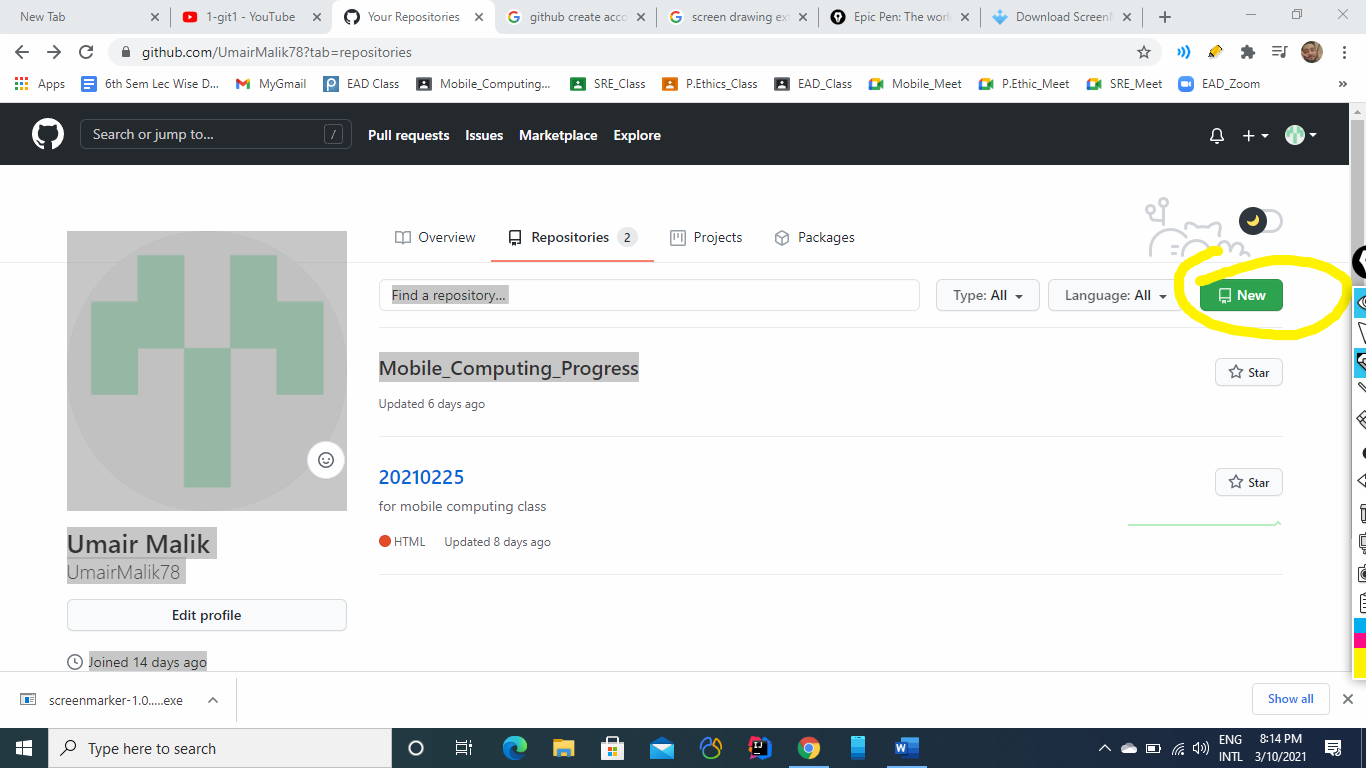
**CREATING ACCOUNT ON GitHub.com**



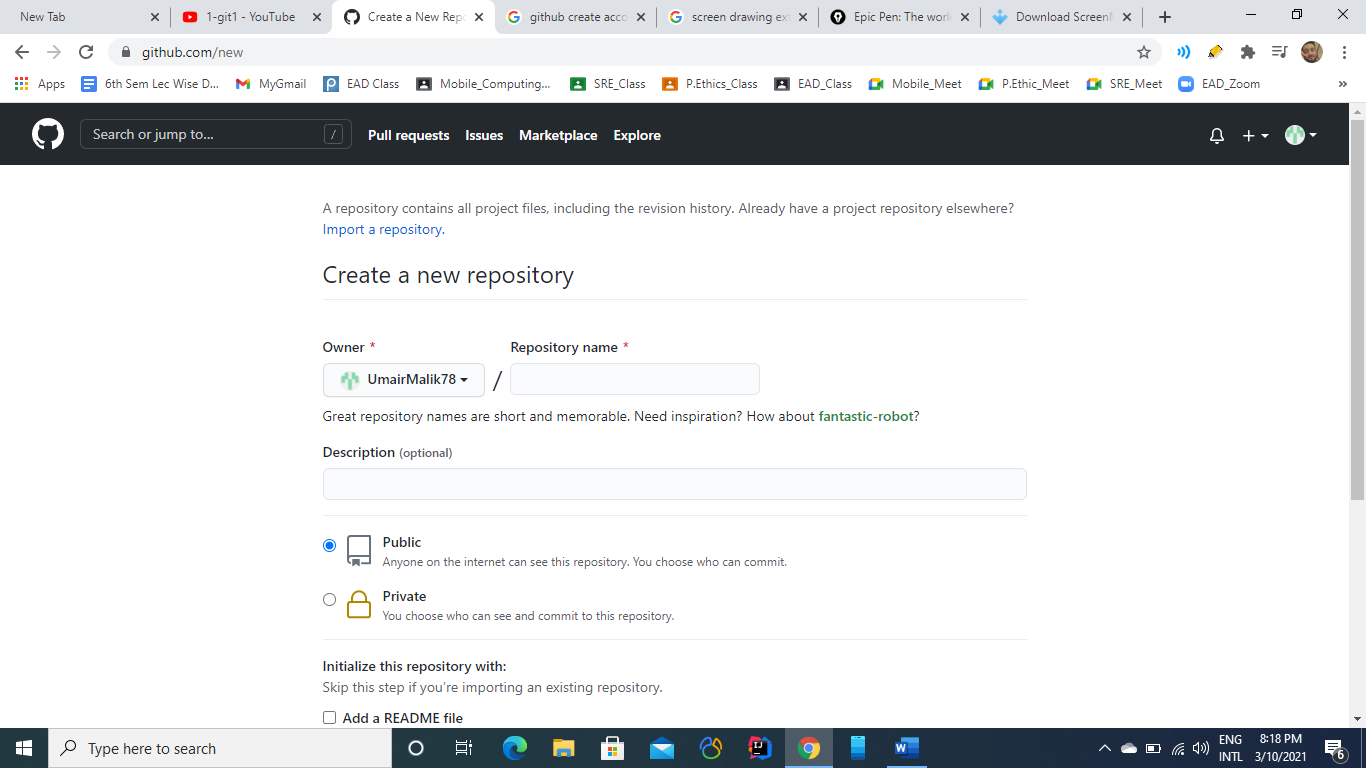
From there select ***sign up for GitHub.***

You’ll find a new page like this. 

Fill out all necessary fields with respective data and Select **Create Account** button.



From this screen creeate on **new button**  to create a new Repository.

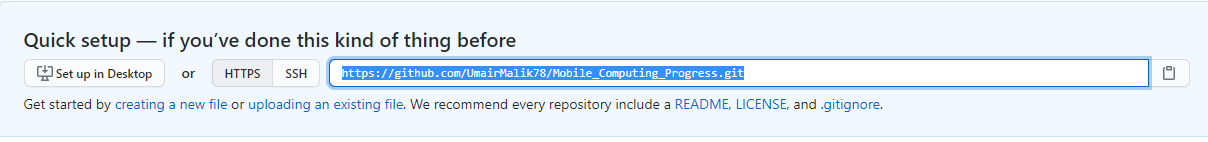


From this menu, Wrtie a new Repository name which weill be of couurse unique wuth reference to you id.

You can also make your repositry as ***Private*** or ***Public***:

**Private**: If you make private, then noone elese can make change to it

**Public:** In this case, everone will be able to see and make changes to your work.



This is the URL of your new Repository, which we’ll use ;ater to work with it.

**Lecture No 2**

**Git Clone:**

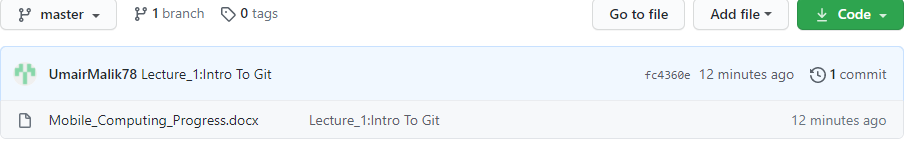
1. Git Clone means to cone the central repository into our ocal repositroy
2. For this we use command  
   ***git clone [url]***

Where URL is the url of our repository.

1. When we clone a repos, a folder/repos is created at our system locally.

**How to add files to central repository:**

1. **Place file into the local repository:**
2. Create File into the local repository we created with the help of ***git clone.***
3. Let suppose we create a file named as **Mobile\_Computing\_Progress.docs**
4. **git add *filename***(i) Add file in the local repository.  
   (ii) git add **Mobile\_Computing\_Progress.docs**
5. **git commit**  
   (i) It is used to make changes with message to track changes.  
   (ii) git commit -m “First Commit”
6. **git Push**  
   (i) It is used to finally add your work on the server.  
   (ii) git Push



After above steps, we’ll see our file located on the server of GitHub.com

Suppose we want to make changes to file.

1. We first add new changes onto the file
2. After that we’ll add file again using ***git add filename.***
3. Then we’ll make commit.
4. At the end we’ll run push command to make newly file available on the server.