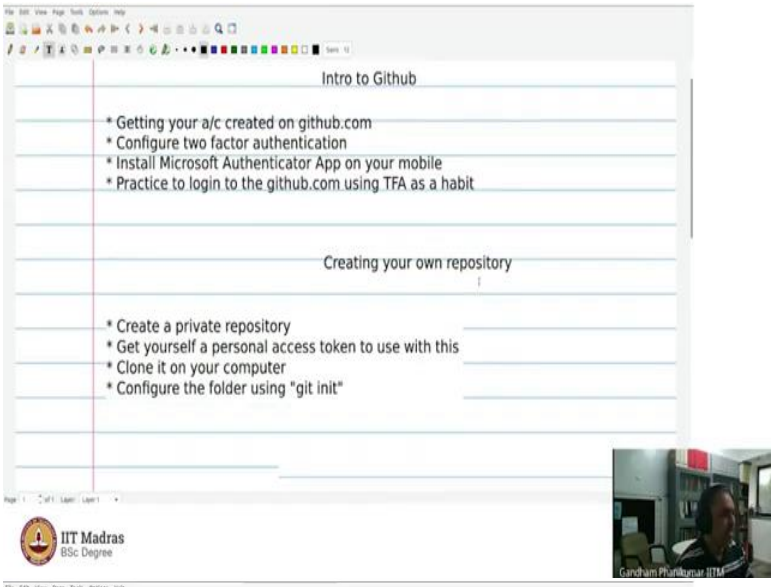
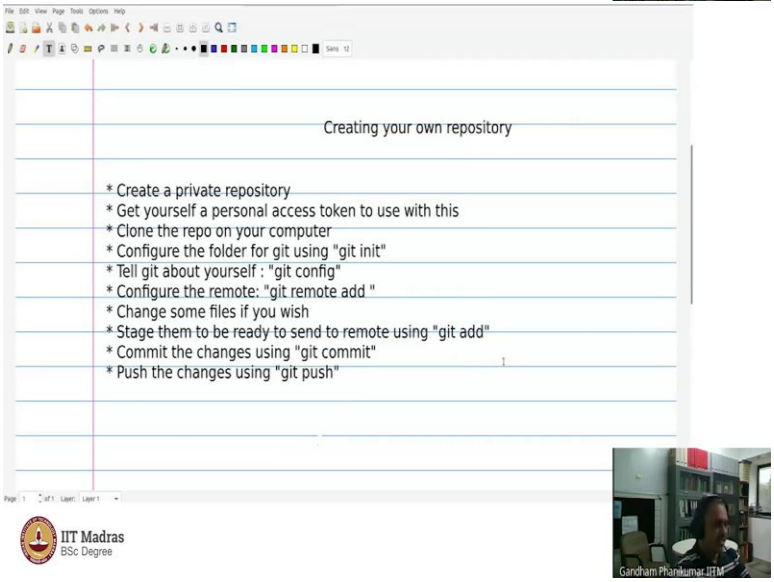
System Commands
Professor Gandham Phanikumar
Metallurgical and Materials Engineering
Indian Institute of Technology, Madras
Version Control - Part 02

(Refer Slide Time: 00:14)





So, let us get going. So, today's thing, I have just, again, restart with the GitHub introduction. And of course, I am not going to go through every detail, but I will tell you most important steps and then we will invite one of your classmates, I think (())(00:30) and we will ask him to come over.

And we will have two exercises done today as demo. And that should then clarify most of you walk up to. So, before we call (())(00:44), I would have the brief intro. Yes. So, brief intro of GitHub. So, what are we going to do in that. So, we have the steps as follows. The first thing we are going to do is getting your account on GitHub dot com.

Of course, as I told you, we do not need compulsorily to use Git concept, but it is a good idea. And I encourage you to use GitHub dot com. So, you should get your account created on GitHub dot com first, and then you should also configure two factor authentication on it. And it also means that maybe you can install Microsoft authenticator app on your mobile.

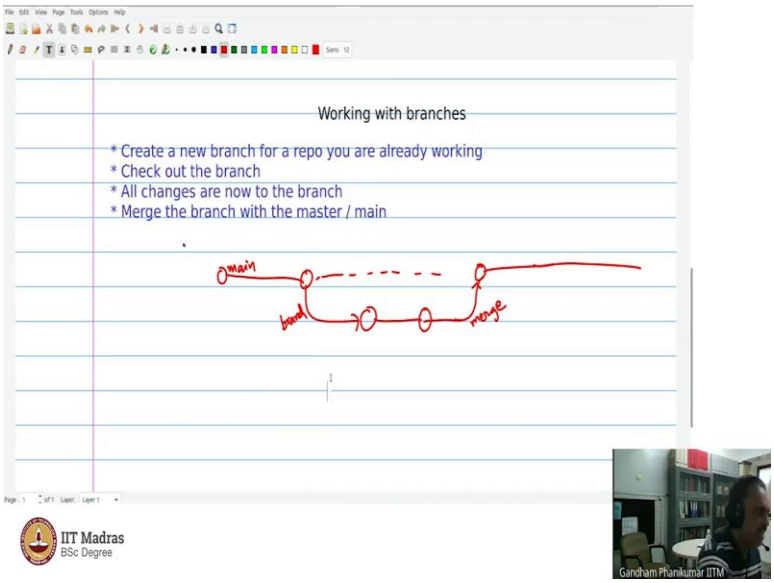
So, that will help you, it is a good thing actually to have. And then every time you open the app, there is a 6-digit number that comes which you enter in the site, then you log in. So, basically that you check so that, practice to login to the GitHub dot com using two factor authentication, as a habit. So, because after 11th of August, we cannot log in directly with the password alone.

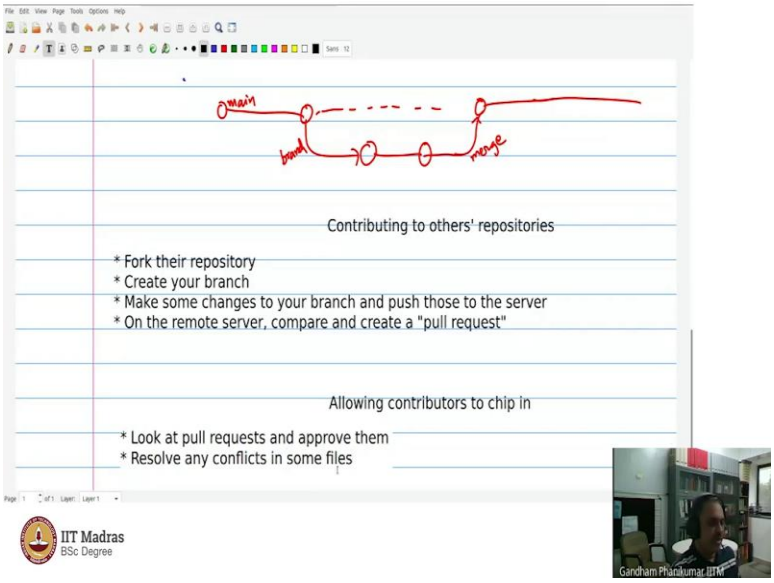
So, that is something that you need to do. That is the first thing. So, I would say here, I would call this as, I would say, intro to GitHub, that is the first steps, second set of steps are about, repositories. So, we will first have that, creating your own repository as a first activity that is important.

So, for that, what we do is, of course, create a private repository and then, get yourself a personal access token to use with this and then, clone it on your computer. And then configure by folder using git init. And of course, I will have this on the front and then we will go through the steps as we do so I am just telling you. So, we create a private repository, we get we have access token obtained and kept it safe so that we can use it by logging in et cetera.

We clone the repo so when I am saying clone, I would say clone it clone it means what? Clone the repo on your computer, configure the folder for git using git init, and I would say, Tell git about yourself. So, I would say that you would tell by telling git config and then configure the remote. So, you would write a git remote, add. And then you would, change some files if you wish, and then you would have that to be committed, so you need to say, stage them to be ready to send to remote using git add and then of course, commit the changes using git commit. And then the changes using git push, so this is the sequence of steps by which we have we can create own repository, make some changes and actually see those changes on the server.

(Refer Slide Time: 05:42)





Now, after that, of course, there are some more steps then the next step is working with branches, branches and that for that we again have the steps. So, create a new branch, for a repo you are already working kid then I would say check out, the branch all changes are now to the branch. And, at some time when you are done with the branch you can say most branch with the master or main. So, that the process is something that I can illustrate here using a sketch, I am done with this so.

So, we can say that you have got main and then some changes are done. And normally you would proceed in that manner, but you can actually have a branch and then have those changes

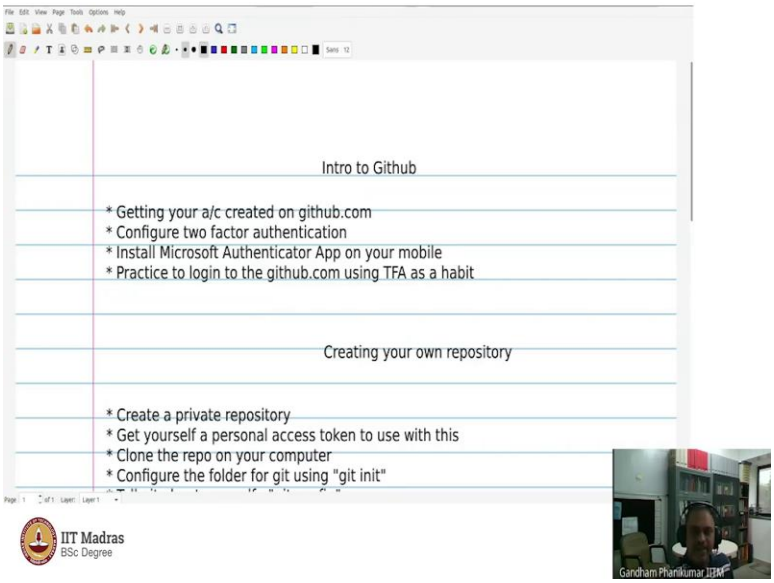
made and then work and then you can come back and merge. So, this process is merge process, this past is branch process and in between of course, others may also do some branching and merging we will talk about that how to resolve if there is a conflict.

But this is an activity that we do so, you can think of you can think of this as like, the main or master branch and then you have got some any user name and then you can work and then branch merge it and again you can repeat that activity by doing it again and again. So, the concept of a branch is something that we will try right away, we will work on that. So, once we have learned how to do that working with branches then I would say in terms of knowledge.

We are ready to… say working I was not working I would say contributing to others repositories. So, there of course we are going to contribute by these steps where we can say, fork their repository, create your branch, makes some changes to your branch and push those to the server. And then on the remote server compare and create a pull request.

So, this is something that we do and of course so this is one activity of course then other activity would be allowing contributors to chip in and that is where we say, look at pull requests and approve them and sometimes you have to do, resolve any conflicts in some files. So, these are the various, now I would say this by enlarge, we are done, so if you can work on these steps, all of these steps if how to do you are ready to go. So, let us save this fellow. Of course, I will share this thing across so we will practice.

(Refer Slide Time: 09:42)

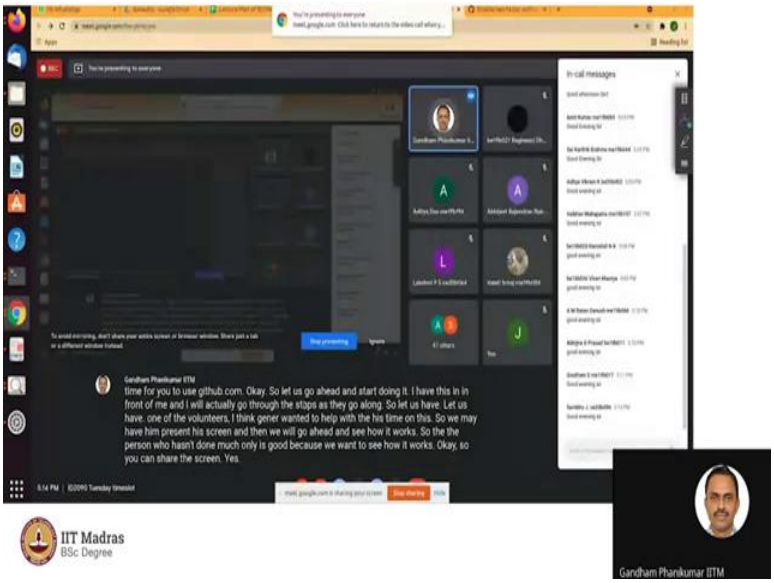


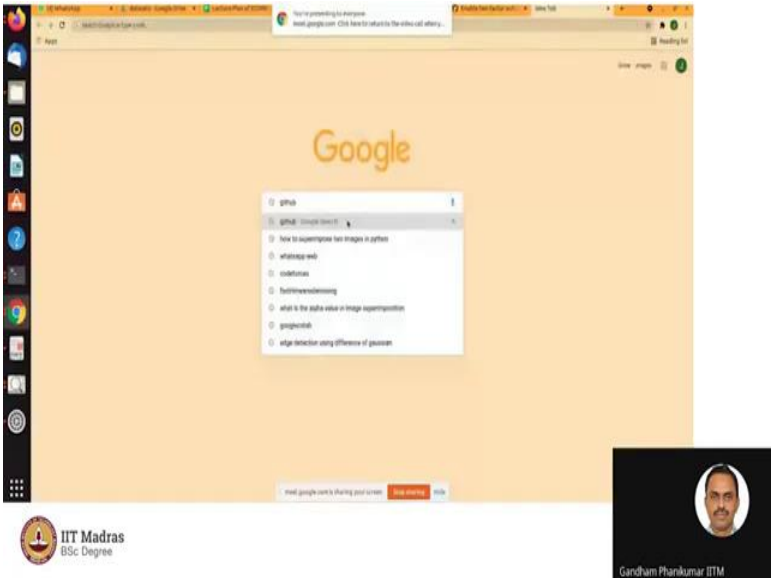
So, just before we have the volunteer I just want to tell you saying these are all steps that you may take few days to finish that is not a problem. And of course, our video is recorded so you can always go back and see what was that guy doing when he said he is going to do so and so activity you can always look it up and of course there is a set of, tutorials in GitHub dot com itself. Where you can look up and YouTube has plenty of resources.

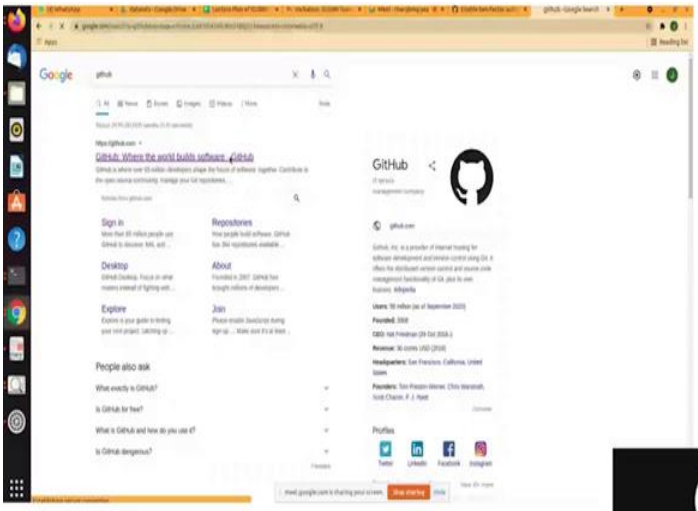
So, there is enough help but of course we are our own class also is a help for you ready me, but these are the steps I want you to know because this is sufficient for a long time for you to use GitHub dot com. So, let us go ahead and start doing it I have this in front of me and I will actually go through the steps as we go along.

So, let us have one of the volunteers I think Gandham Jaya wanted to help with his time on this so we may have him present his screen and then we will go ahead and see how it works. So, the person who has not done much only is good because we want to see how it works. So, you can share the screen Gandham Jaya.

(Refer Slide Time: 10:55)

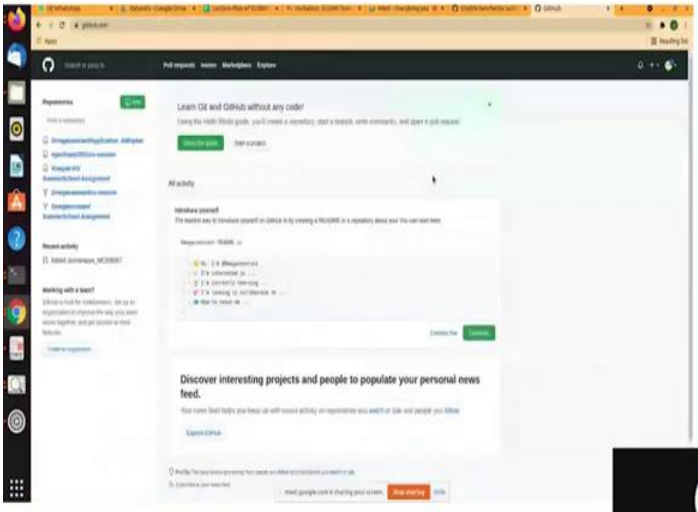






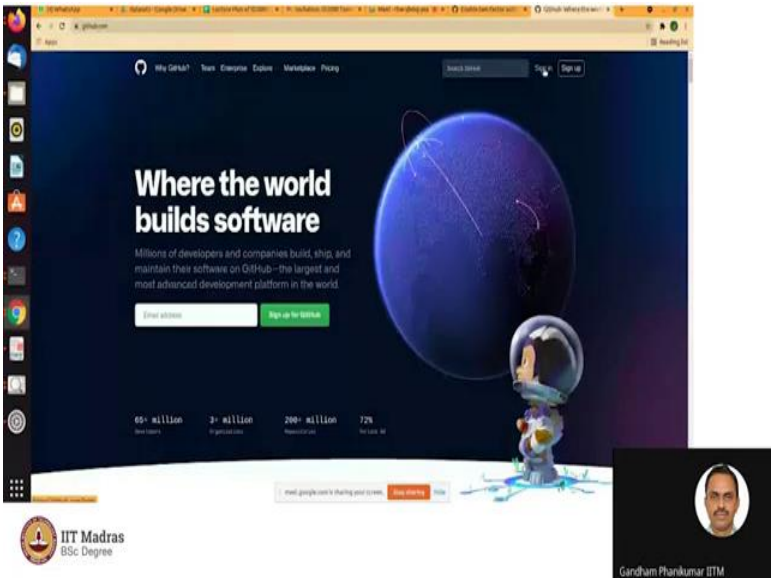
IIT Madras
BSc Degree

Gandham Phanikumar IITM



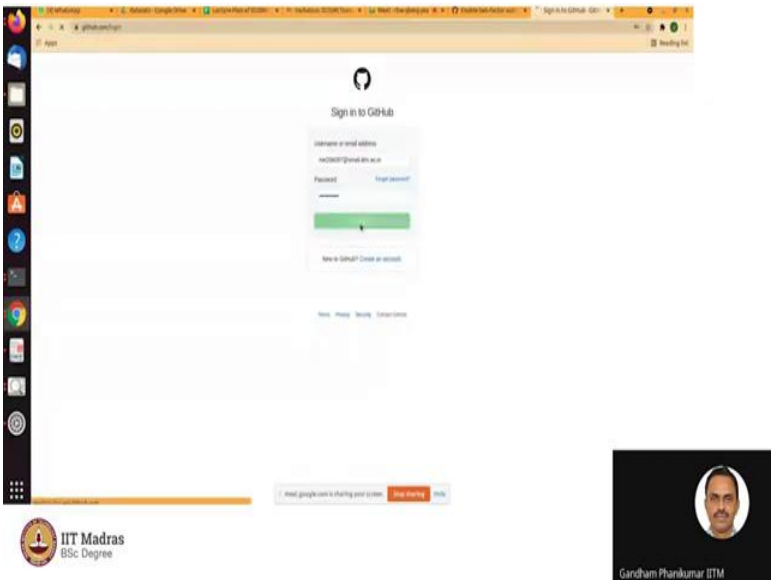
IIT Madras
BSc Degree

Gandham Phanikumar IITM



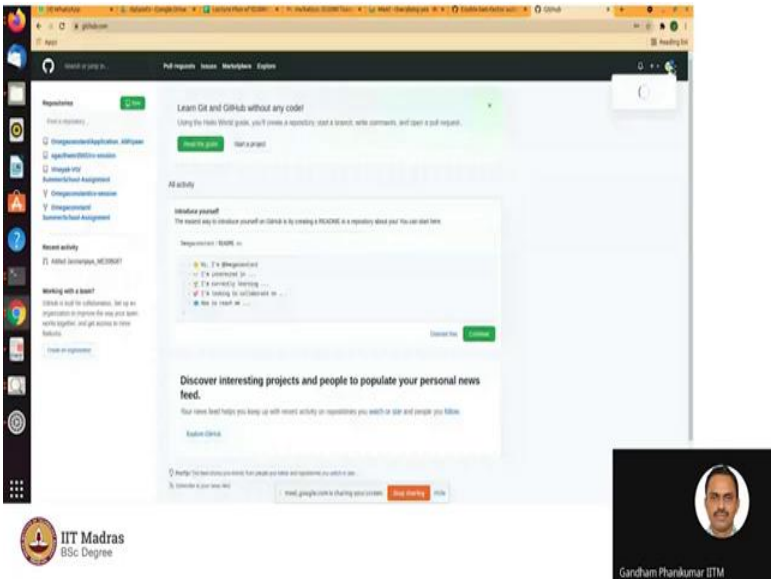
Yes, so for this session, do not worry so much about passwords and all that, I mean, we do not have to worry, so go ahead and open it GitHub dot com. So, log off because I want it to start from scratch alright, so now you can log in.

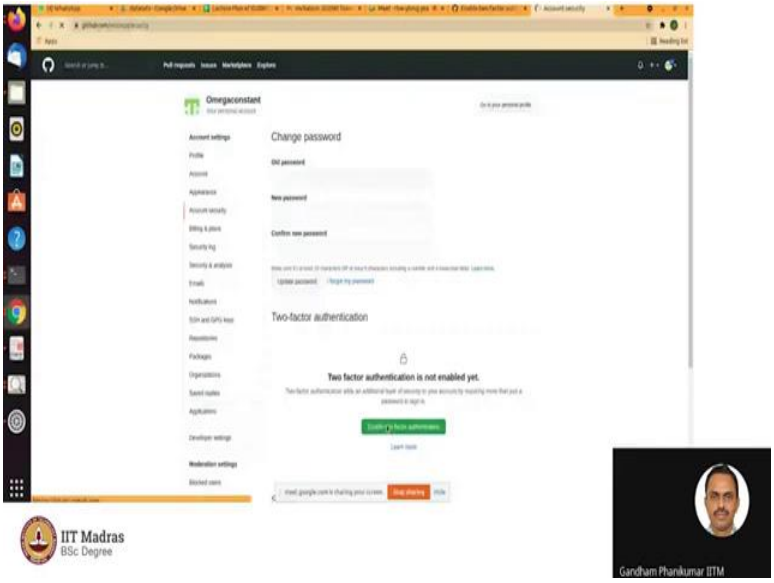
(Refer Slide Time: 11:42)

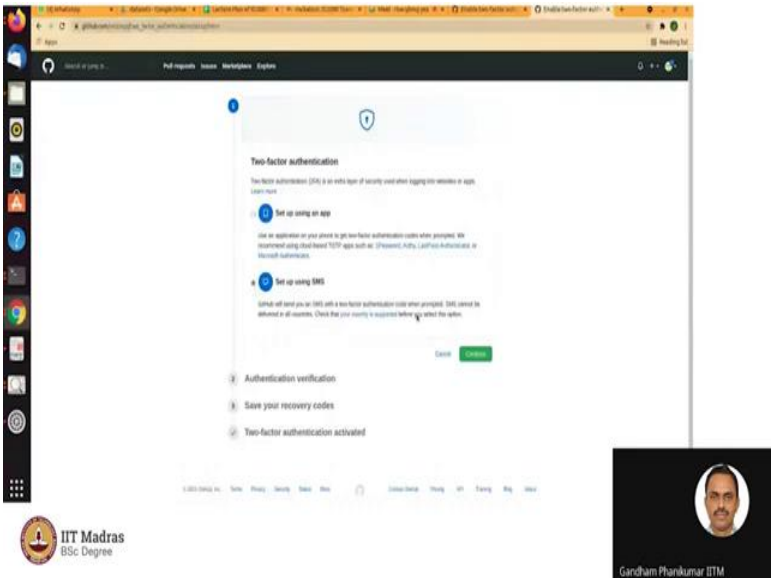


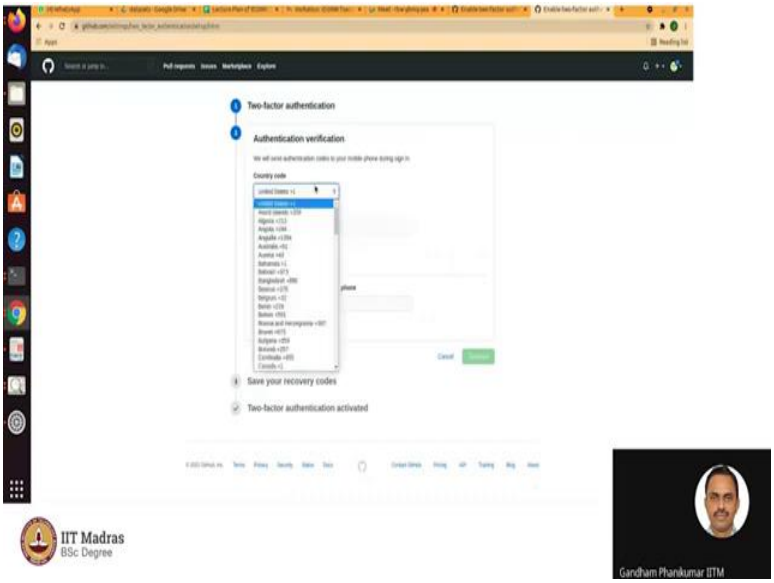
So, normally once you present that login screen, so you have put some id, email id is possible and also user name one of the two is sufficient and you can give a password.

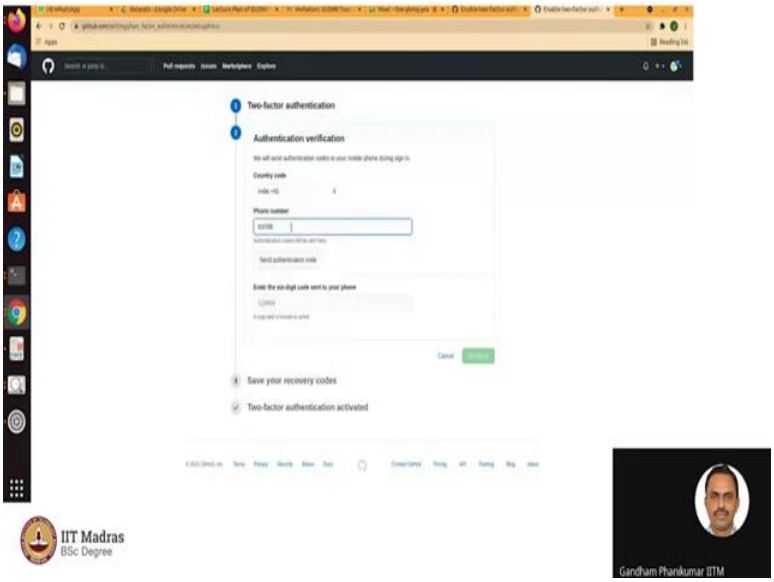
(Refer Slide Time: 12:04)

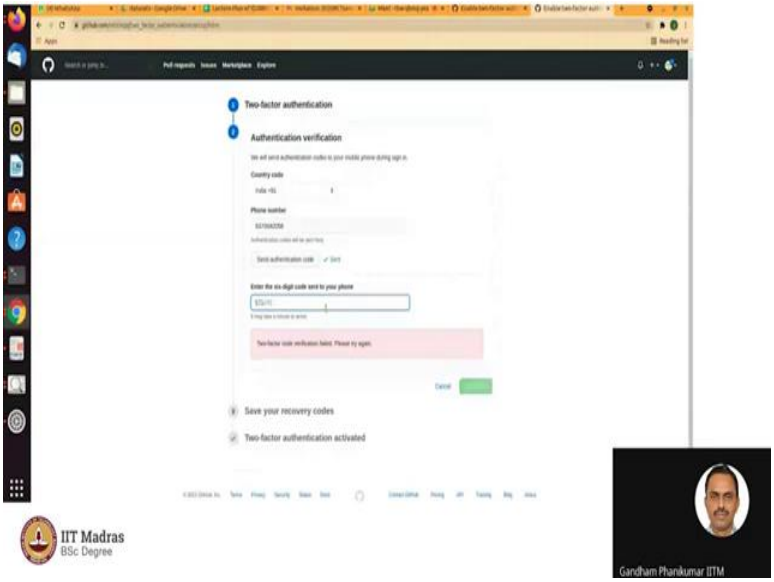


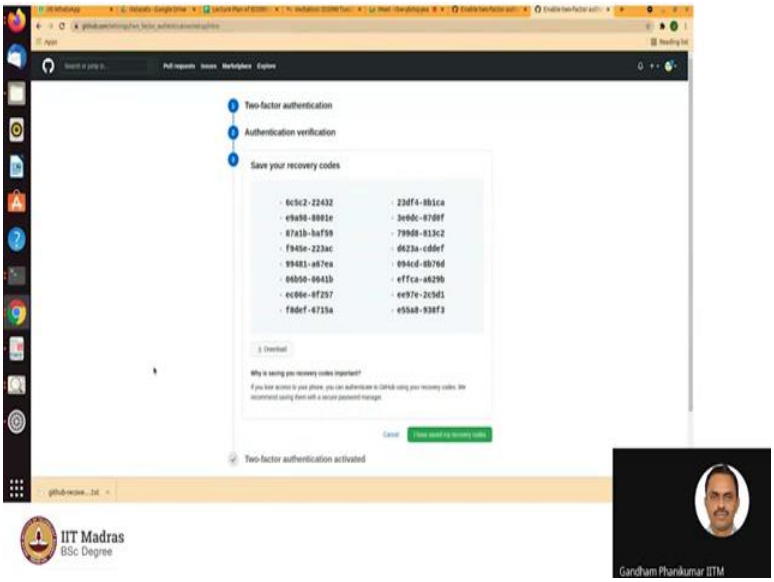



Two-factor authentication
Set up using an app
Set up using SMS
Authentication verification
Save your recovery codes
Two-factor authentication activated
IIT Madras
BSc Degree
Gandham Phanikumar IITM


Two-factor authentication
Authentication verification
Country code
Save your recovery codes
Two-factor authentication activated
IIT Madras
BSc Degree
Gandham Phanikumar IITM


Two-factor authentication
Authentication verification
Save your recovery codes
Two-factor authentication activated
IIT Madras
BSc Degree
Gandham Phanikumar IITM


Two-factor authentication
Authentication verification
Save your recovery codes
Two-factor authentication activated
IIT Madras
BSc Degree
Gandham Phanikumar IITM



So, it appears that you have not configured the two-factor authentication that is good because now we can show others how to do that. So, go to the top right corner and press that icon top right corner. You have got this icon? Yeah. So, come down and settings. And the left? Yeah, that is fine settings and the left-hand side account security is there one of the items I think the fourth one account security yeah.

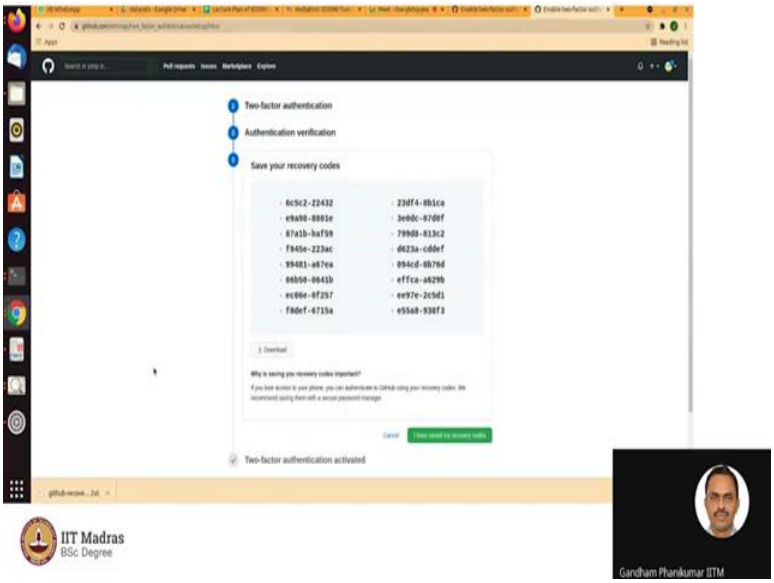
Now, there is a green button it says enable two factor authentication. Click on it, enable it, you can hide this Google meet thing. Stop Sharing hide that click on that hide there. There is a pop up meet dot google dot com is sharing your screen that pop up hide it? Yeah, good. Now, you have two ways of doing it.

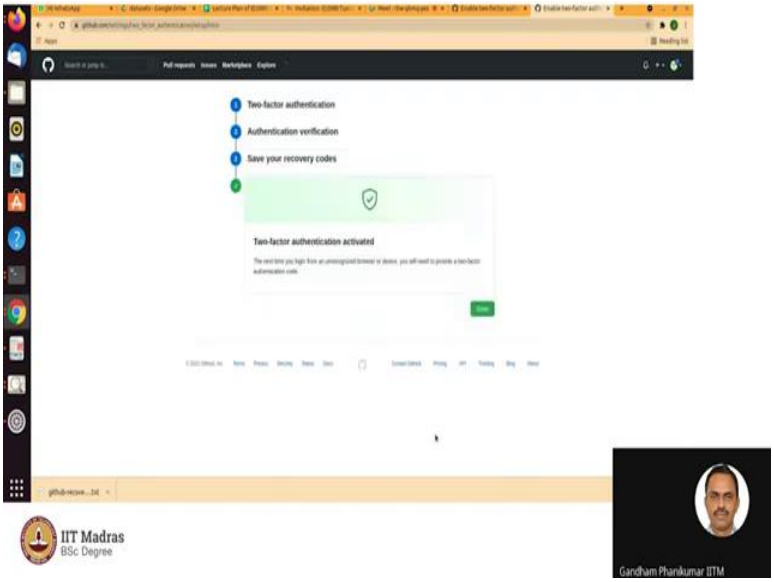
So, if you have your phone ready, you can do it or you can configure it with SMS that is fine. Just do one of them. No problem. So, continue and you have to give you a phone number and country code is first do not miss out. Go and first say is a country code to India. After that, you gives the phone number yes.

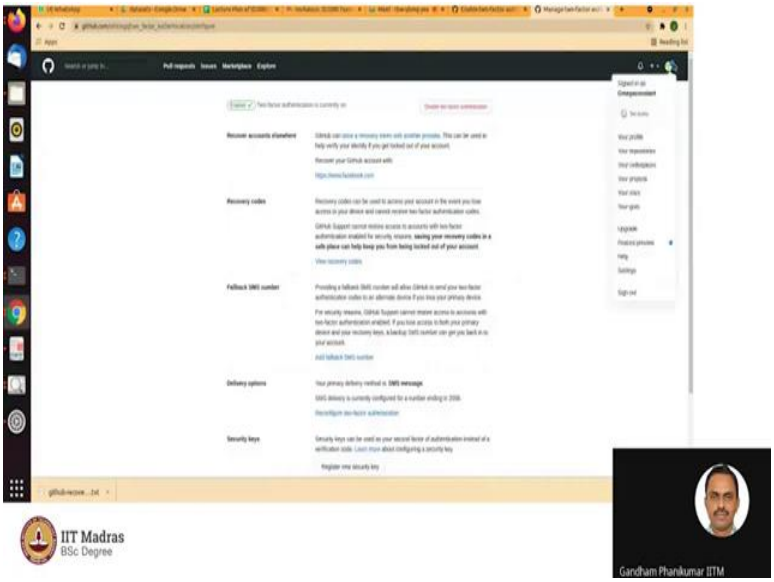
Now it is asking you to, first of all confirm the phone number whether you have it or not. So, ask it to send the authentication code, and it will send something in the SMS so you can type that in do not worry its sural one time use if others look at it does not matter. Of course, this if they click before you click that button. I do not think somebody is following you so closely behind your hills. So, enter that thing and confirm, so this might have come already or so this is one of

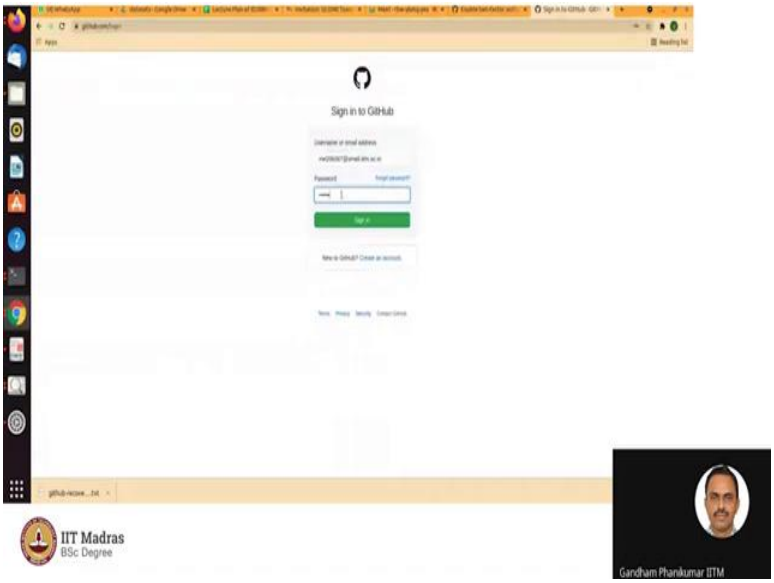
the reasons why I choose to use the app because the app there is no delay. Yeah, continue and see you might have not given a proper limit. Yes good.

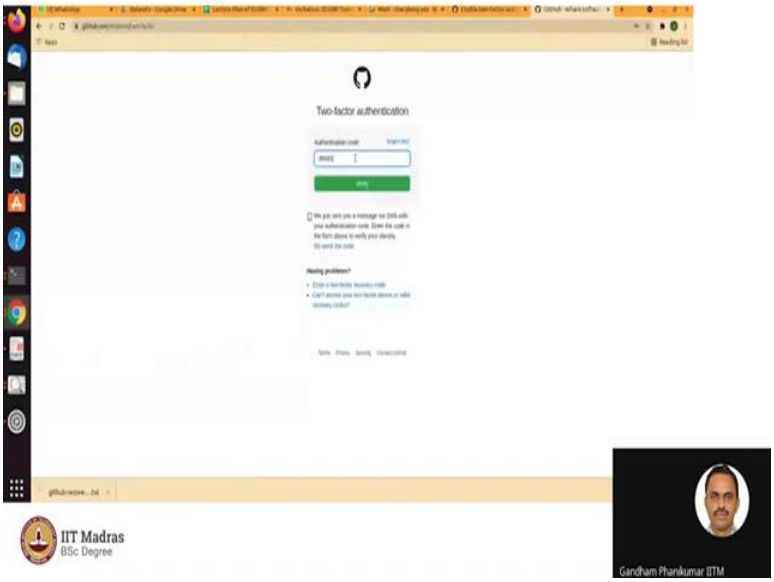
(Refer Slide Time: 14:15)






IIT Madras
BSc Degree
Gandham Phanikumar IITM


Sign in to GitHub
IIT Madras
BSc Degree
Gandham Phanikumar IITM



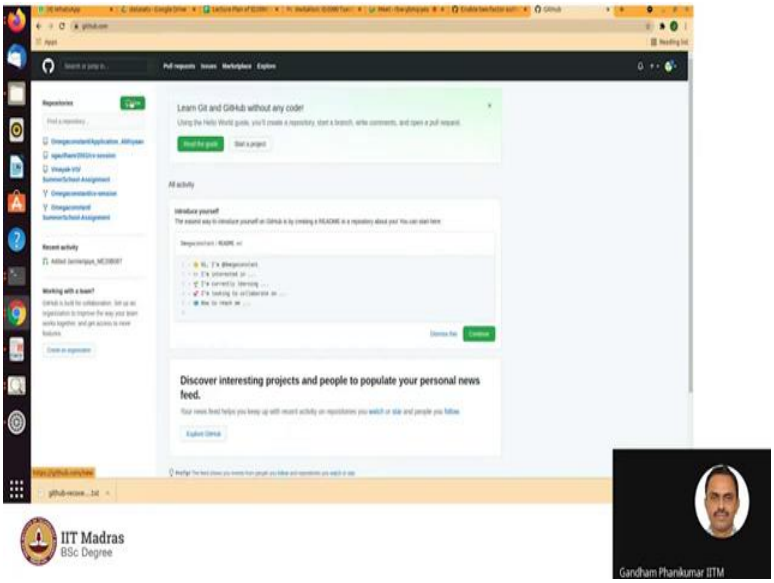
Now, these codes are something you should save so just a download just click on the download button. So, this is basically sometimes if you happen to lose your phone and therefore you are unable to get in then this course will help you for recovering so it is a good idea to keep them that is fine. Now, having saved it, you can say I have saved my recovery codes click on that button.

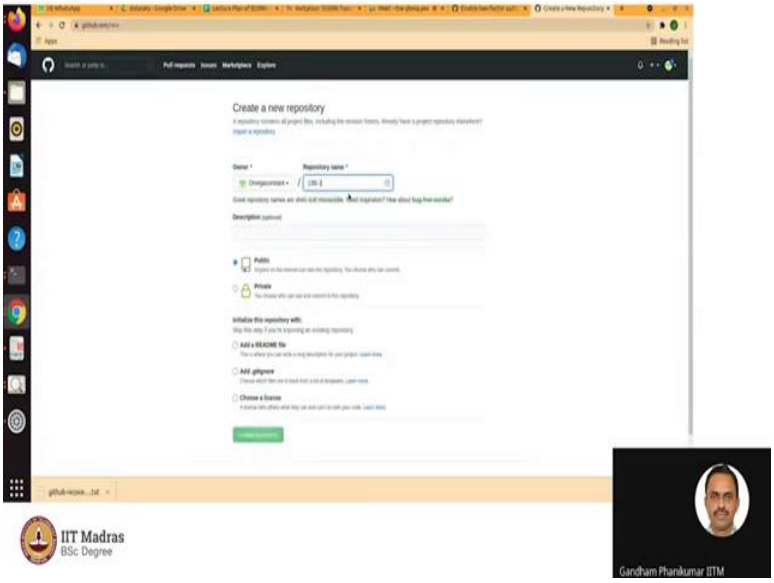
And that means that you are two factor authentications has been activated, done. So, there is some celebration, things on the background. That is happy sign. So, you can click on done and then you are yeah, now you are two factor authentication is setup. So, you can log out and try to login now.

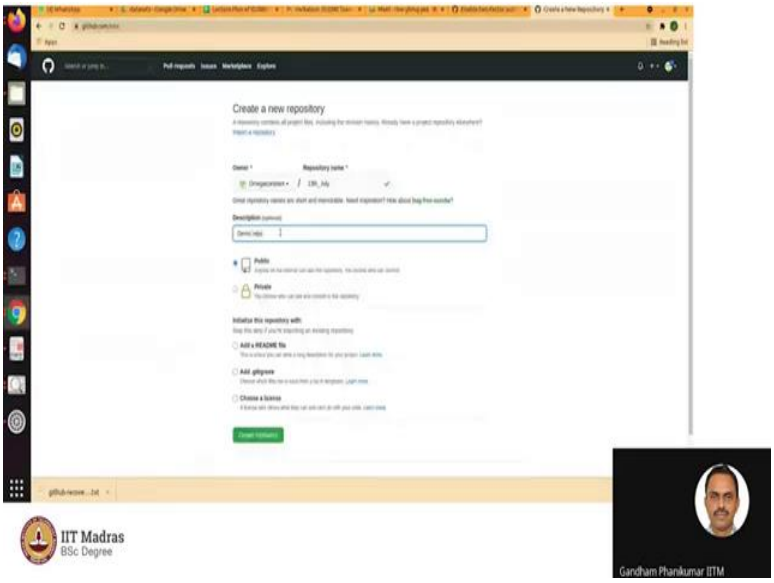
See what happens, so if you log in, of course, you get the same password. The only thing is that when you click on sign in, it will not go in, it will actually pop in one more in it last one more. And this code is what you are supposed to enter seeing your phone. So, they just sent an SMS you enter the SMS code OTP, and then once it verifies.

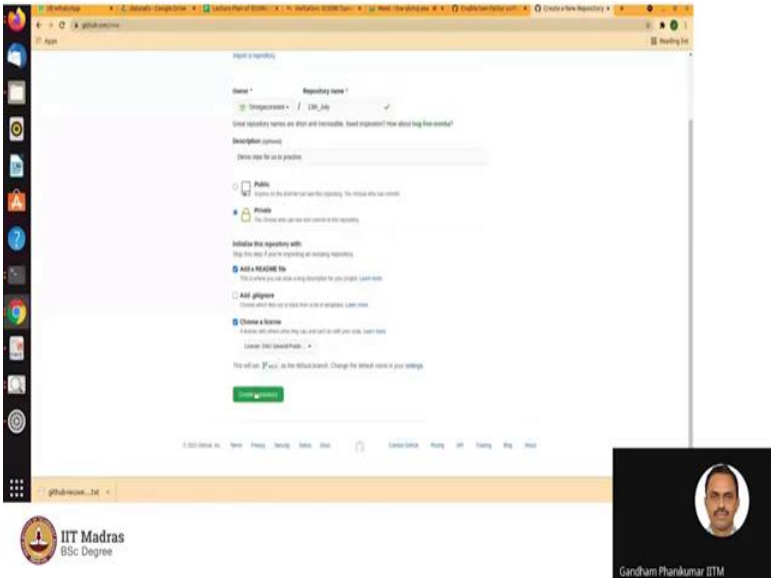
And you will be allowed to get in so, sometimes, so SMS delivery may take some time for things to work out. So, it is a good idea to use the Microsoft authenticator app that is (())(15:48) only. But you are in now so this is the first step we have done. So, the intro to GitHub, we are done.

(Refer Slide Time: 16:00)







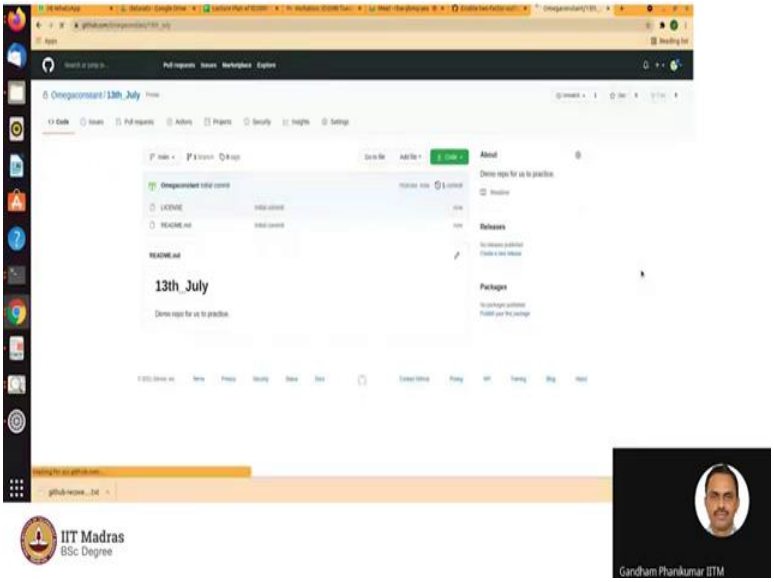


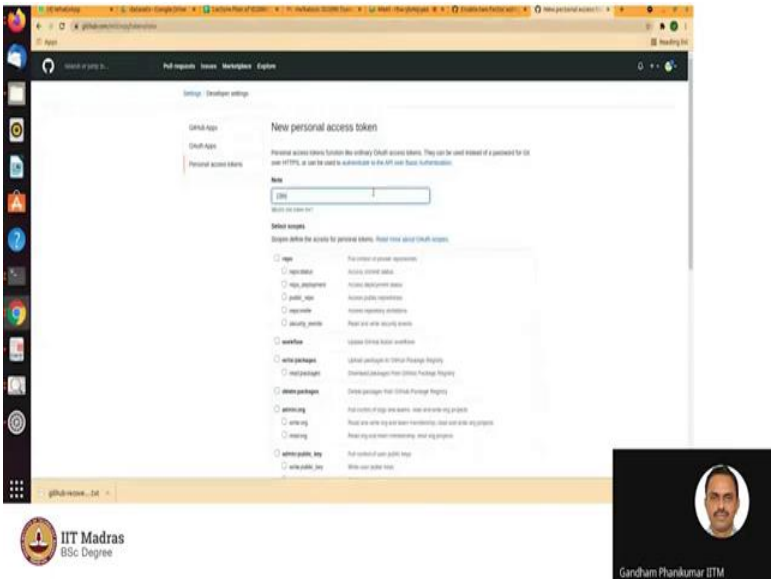
Now, the next step is to create your own repository. So, we will do that. So, what we do is click on that new button. Yes. And so, you can then give any name. So, you may give a name. Again, there are same named multiple reports do not do that. So, do not do that. You can just, give something else let us say 13th July or something 13th July.

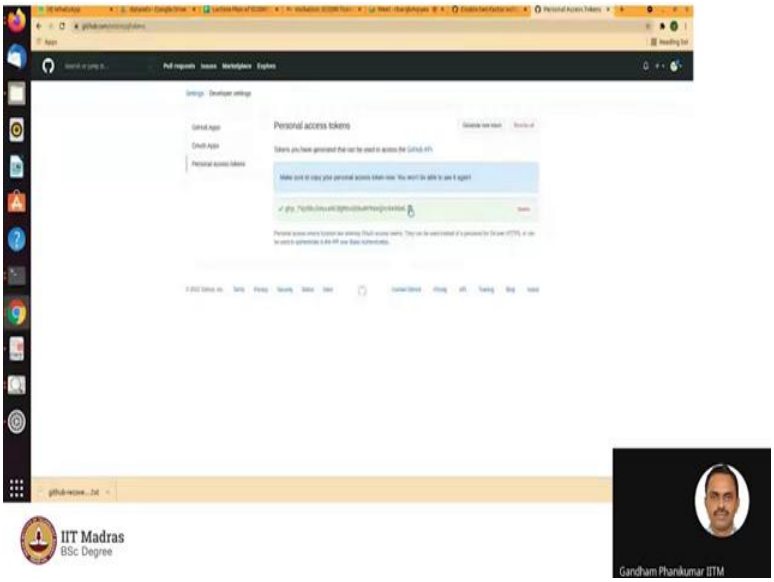
No, no, do not give any gaps. It may accept, but it is a pain for us to later on, do some URL, et cetera. So, you just give it such a way that there is no special character at all? 13th July, that is fine, description, you write some description right, right, right because I want you to do it properly.

So, you can say, a demo repo for us to practice and the very first repo, of course, it is do not have to make it public, it is, you can make it private. Also, you should learn to make it private, obviously, because you need to also tell, so this is fine, private is fine, add read me a file, and licensed, also, you can choose some, click on that button, and then it will show you some licenses that are there. So, you can click a license, you can choose something like, GNU Public License Version 3.O. That is the license, which is the line x and therefore, it is a good idea to do it, click create the repository.

(Refer Slide Time: 17:37)



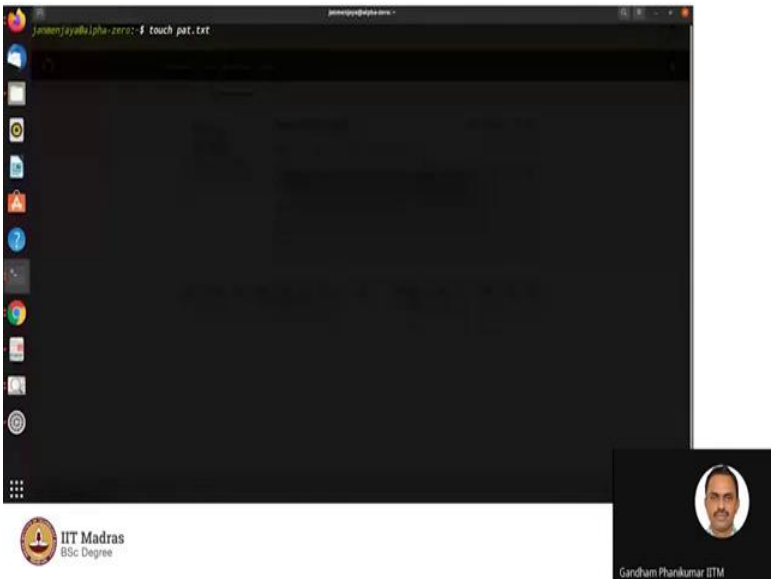


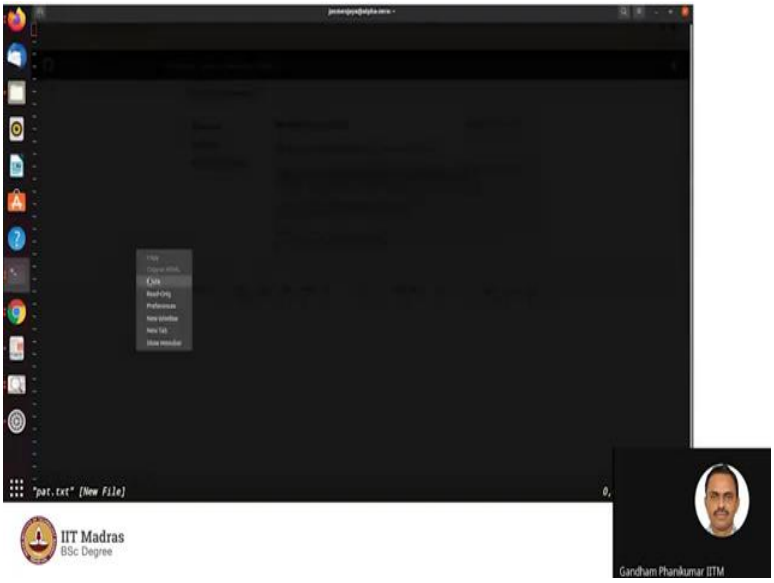


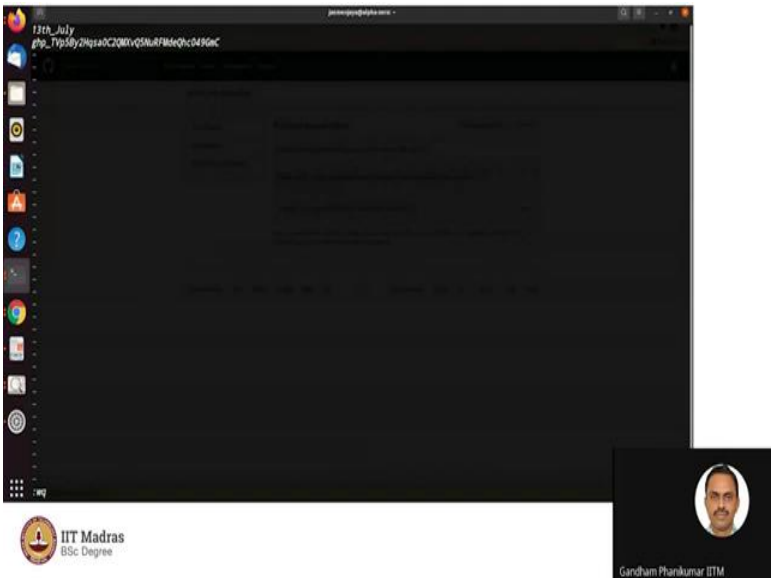
Now, this repo is ready, and now you need to clone it. Now, there is a difference from yesterday and today, yesterday, our colleague has created a public repo. So, to clone it, you did not require to authenticate, but for you, you need to authenticate even to clone it to your own computer. And therefore, you need a personal access token.

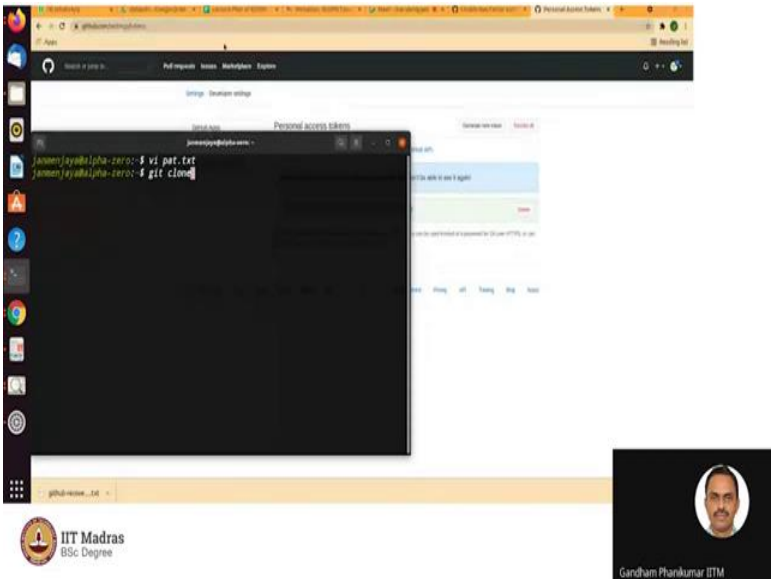
So, go ahead and pick it up first. So, click on that. Yeah. And come down to settings. And on the left-hand side developer settings and personal access token, yes. And you can give the token name same as the report so 13th July. So, you understand it is meant for that particular app. And click on that. Yeah, copy, copy it and save it to a file called PAT dot text so that when you are initially learning how to use it, it is a good idea to have that stored.

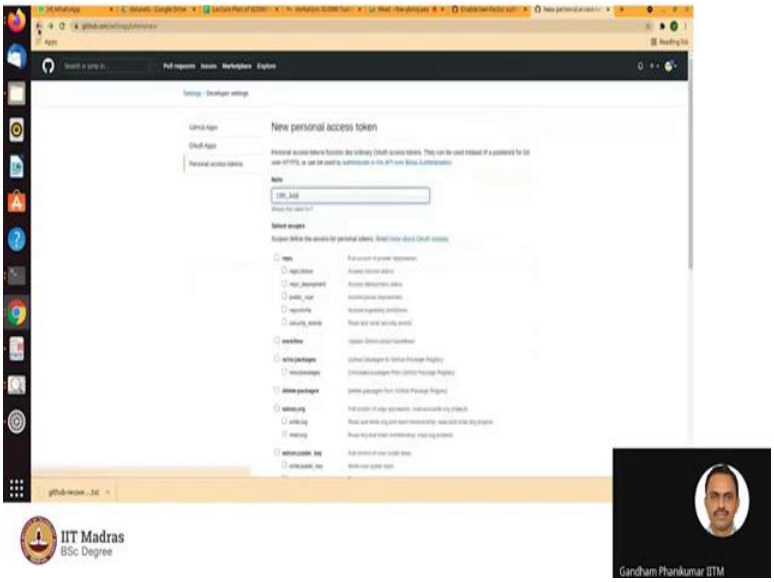
(Refer Slide Time: 18:53)

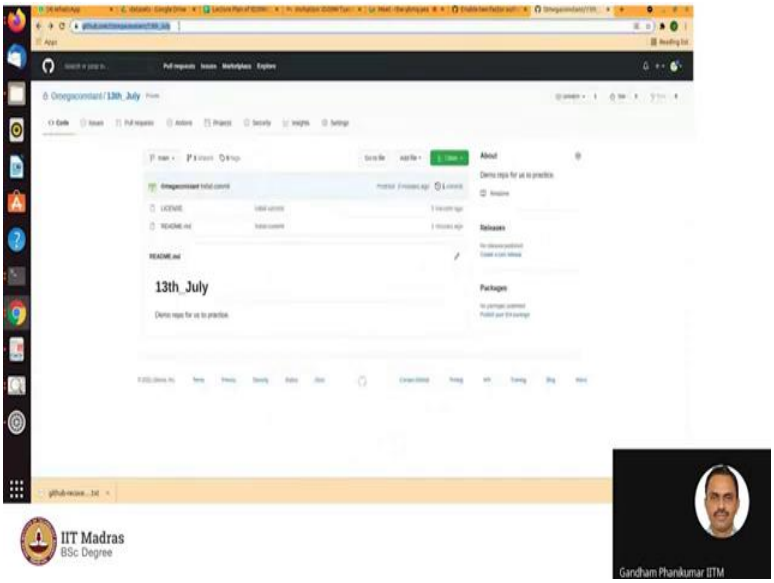


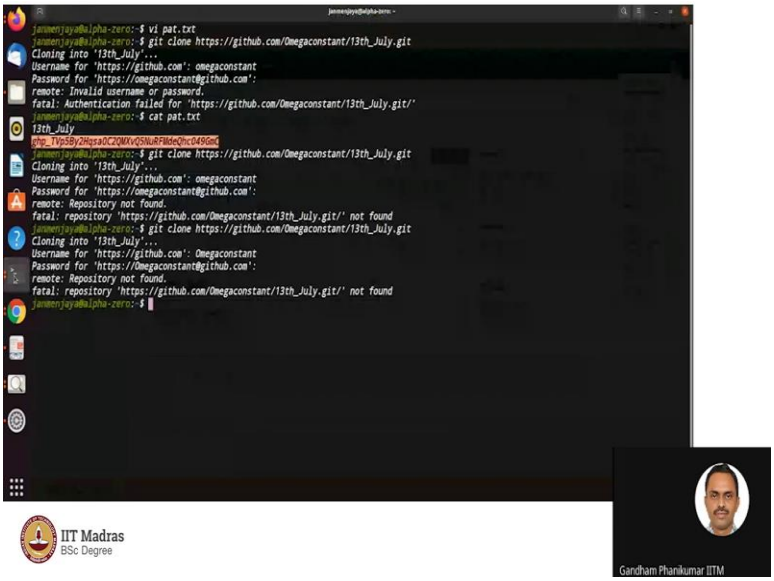



13th_July
ghp_TVp5By2HqsaOC2QMXvQ5NuRFMdeQhc049GmC
:wq
IIT Madras
BSc Degree
Gandham Phanikumar IITM


Personal access tokens
janmenjaya@alpha-zero:~$ vi pat.txt
janmenjaya@alpha-zero:~$ git clone
IIT Madras
BSc Degree
Gandham Phanikumar IITM


New personal access token
IIT Madras
BSc Degree
Gandham Phanikumar IITM


13th_July
IIT Madras
BSc Degree
Gandham Phanikumar IITM



So, open a text file and save that, click on that button it will copy and in a text file, you can save it, you have saved it, yeah. PAT dot text for example do not say touch because that is only going to create an empty file. So, you have to say vi so that you can actually enter this our nano or nano nano. Let us say that vi, that is fine. Press i after that you can paste it and then save it.

And you can actually add one more line above to tell what, what is it for, so that press capital O it will go to the line above create a new line above press escape press, escape capital O will press you edited it no, no, no, no, delete the whole line and paste the code again. Because you do not know what change that you have done so, yeah.

Now, escape press escape and press capital O, now you can read 13th July because that is something that is a reminder for you for which repo you have kept this and, and of course, you should also not forget to remove it later on. So, this is done save it. Now, what you do is earlier in the yesterday class I have clone and then it came to a directory, so therefore there is no issue.

So, in this case, we need to also, do that, and when you are trying to do git clone, it should clip and it should ask you authentication. So, let us go ahead and do that. So, git clone, and then you need clone and space and give the URL? And the URL would be the URL of that particular not the tokens go to the repo so click click on the top right. Yeah.

So, that one is ID, followed by dot git. So, paste it, dot git no no, no without space dot git. Yeah, now if you try to clone it would ask username password, which was not the case yesterday

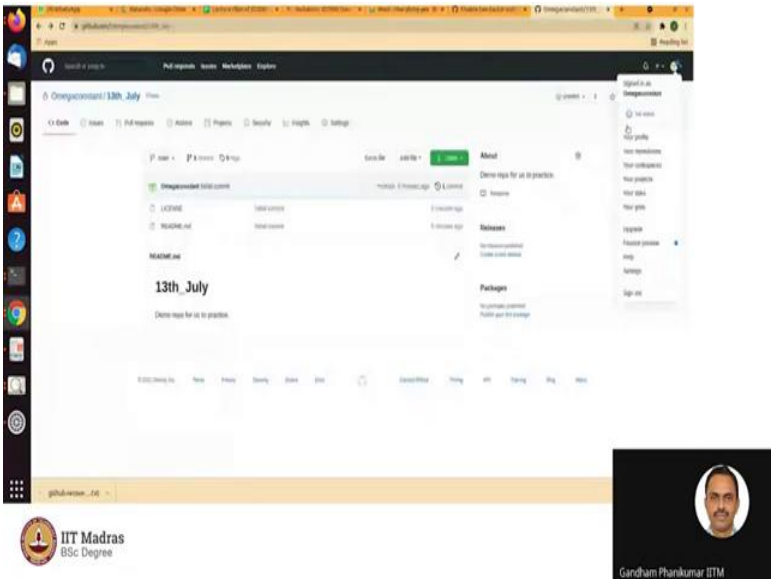
because that was public. Now of course it is a private. Now wait, wait, wait now you need the password to be pasted. So, which means that I mean you do not have it readily with you right? So, put a control C cancel the process, put a control C or just press enter it will crib that is not authenticated, that sort of problem.

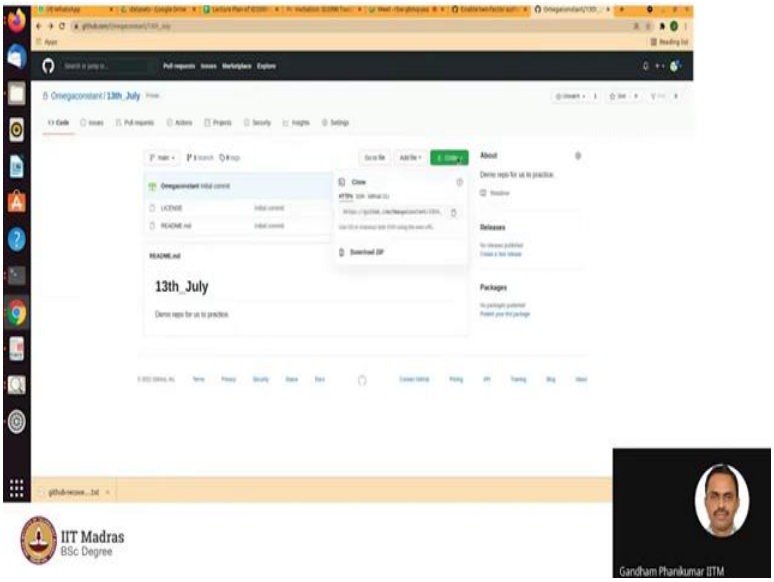
Just press enter, now cat pat dot text so that it is there on the screen for you to copy paste. I am just giving you some tips so that you can do it again. Now, you can yeah, and you now give the user name password now for the password you do not give the password of your account you give this code. Yes, or you to copy paste and the omega constant that pin PH dot git, you have to find out why you have not found it. So, you need to just go and.

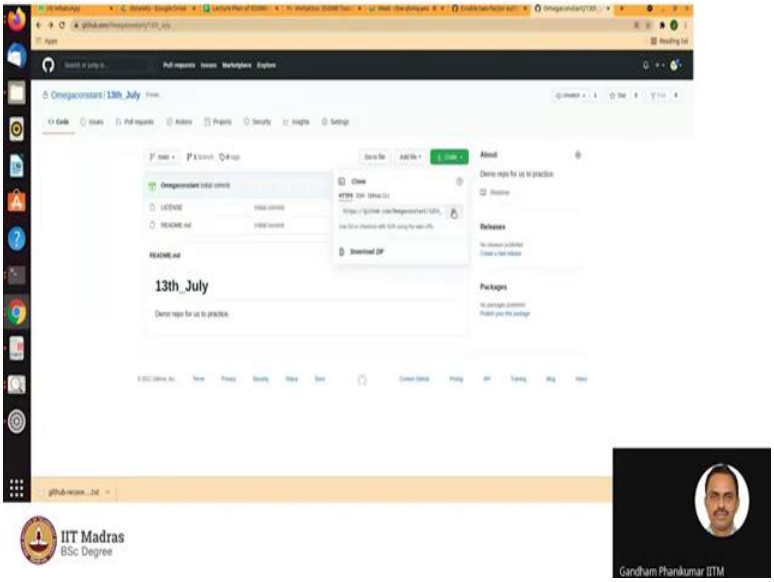
Student: Omegas capital in this username I guess.

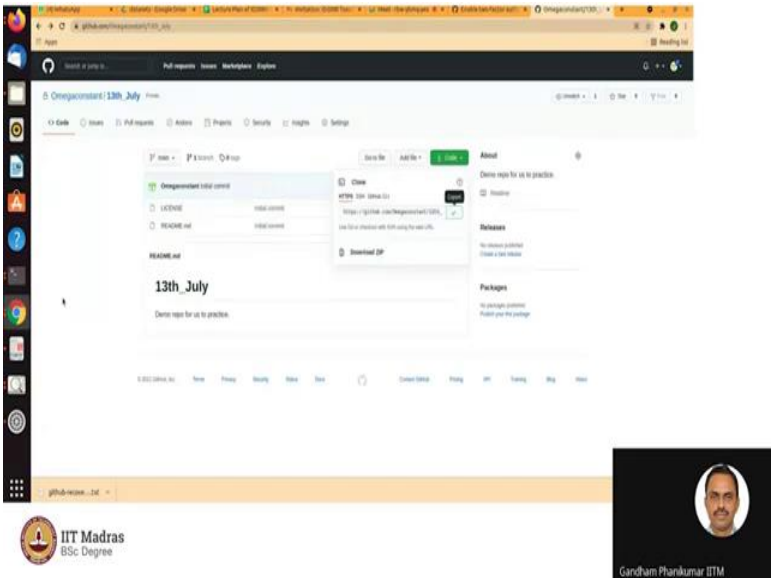
Professor Gandham Phanikumar: Yes, you forgot to put the first letter capital R supposed to be so I do not know which was the original name. So, you have to give me whatever is yeah, so I mean, if the URL has capital O it means that your user name is capital. So, check that out once go to the site and check what was your spelling? What was your user name and what was your.

(Refer Slide Time: 22:53)



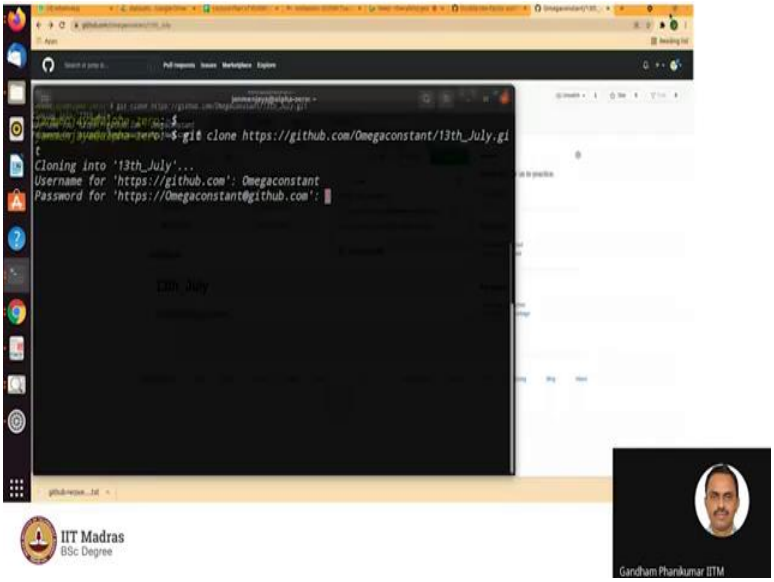

13th_July
Demo repo for us to practice.
IIT Madras
BSc Degree
Gandham Phanikumar IITM

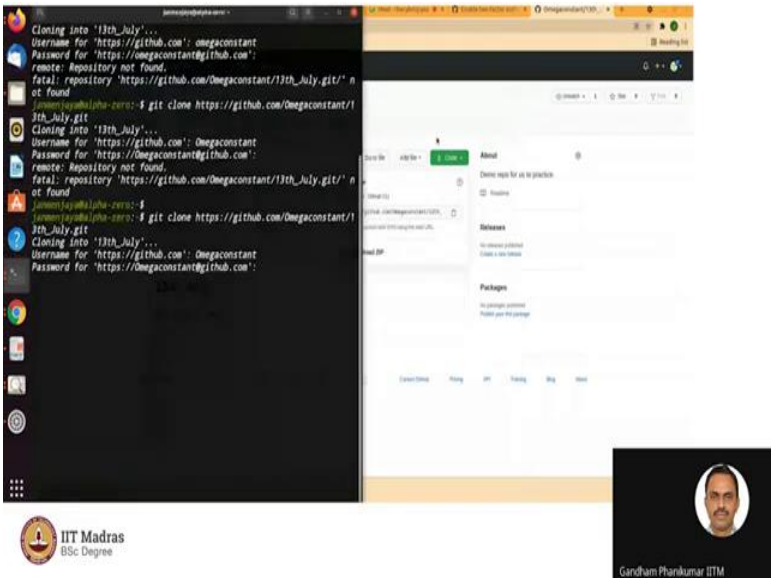

13th_July
Demo repo for us to practice.
IIT Madras
BSc Degree
Gandham Phanikumar IITM



So, user name has a capital O of course, and wait, wait, wait, what you do what you do wait, wait on the right-hand side in the site? There is a download there is a button the green button is there. Not that corner? Beside Add File, there is a green button code it says code no, no just come down yeah. So, there you click on that to drop down. And it has a URL that you copy yes that may help you because it may have something that you would have missed out.

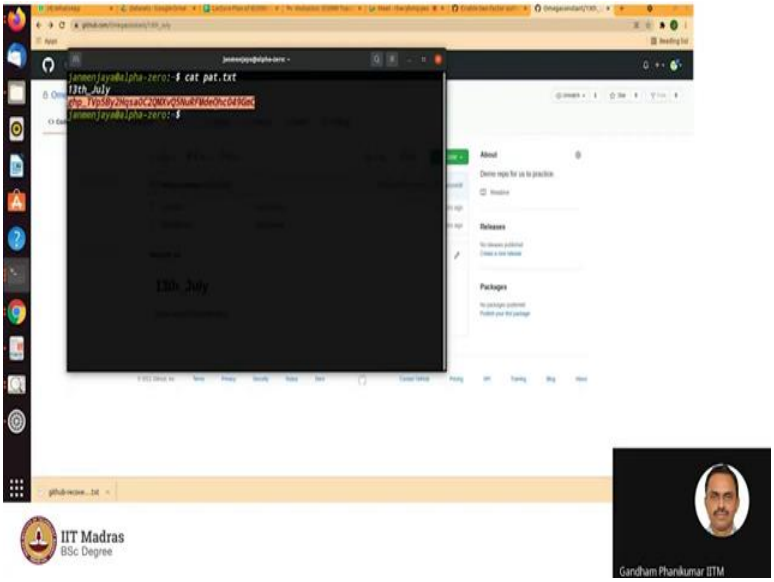
(Refer Slide Time: 23:25)

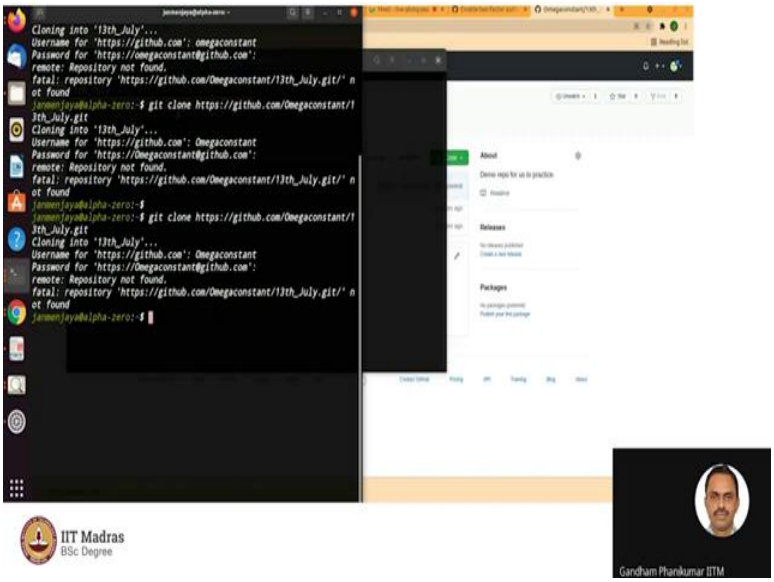




So, check that out now. Again, do not clear the screen you have lost the password now again, so you have to put cat, yeah, yeah fine. So, now you can try to do a git clone and see what happens.

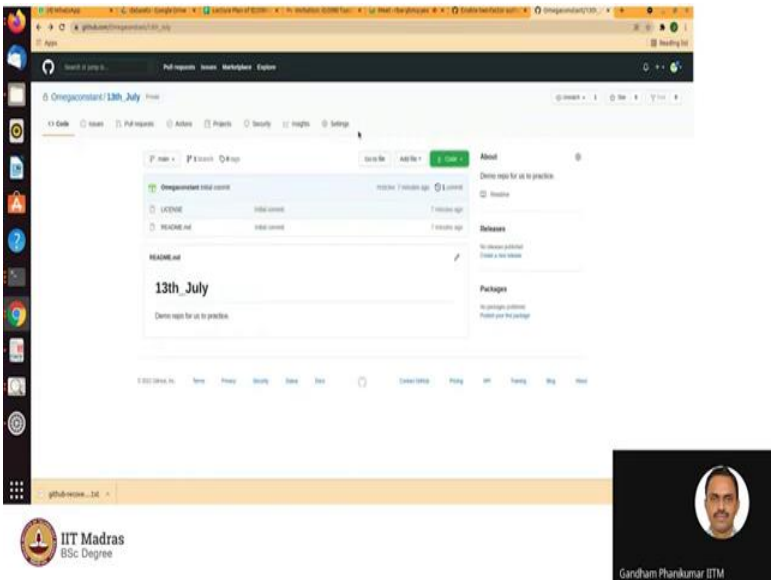
(Refer Slide Time: 24:12)

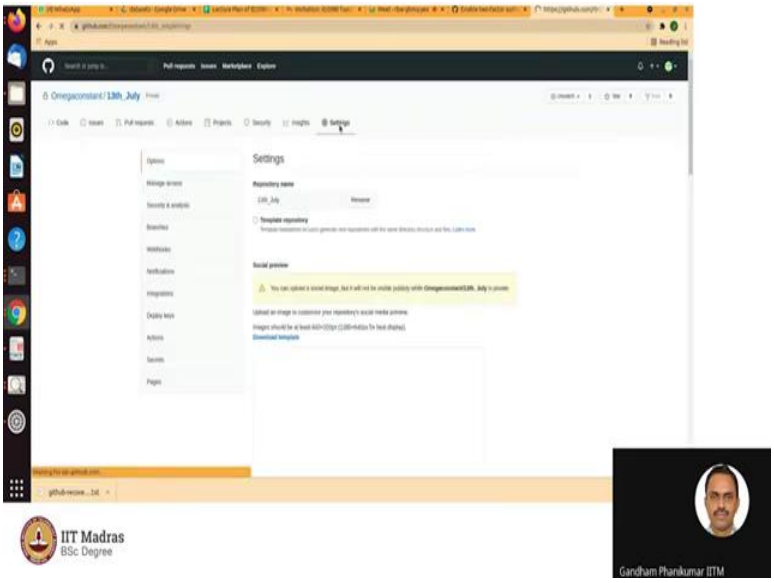


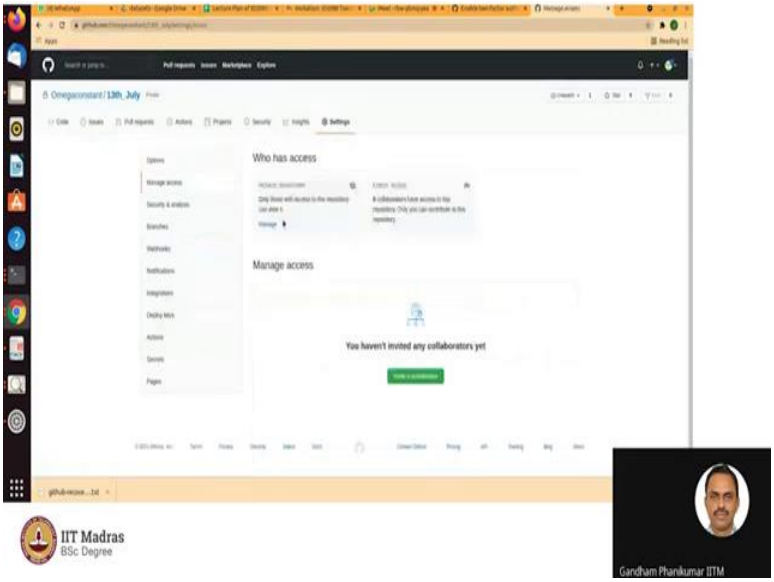


Just if you double click on that the string it will get copied usually, if it is a Linux machine, and you can use the middle button to paste typically, so that you do not need to spend too much time to do that no problem. So, I am surprised what was the problem, I will see what you can do but I do not want you to again spend too much time so go back and make it public so that at least you can get started. Because I know what was going wrong for you.

(Refer Slide Time: 25:03)

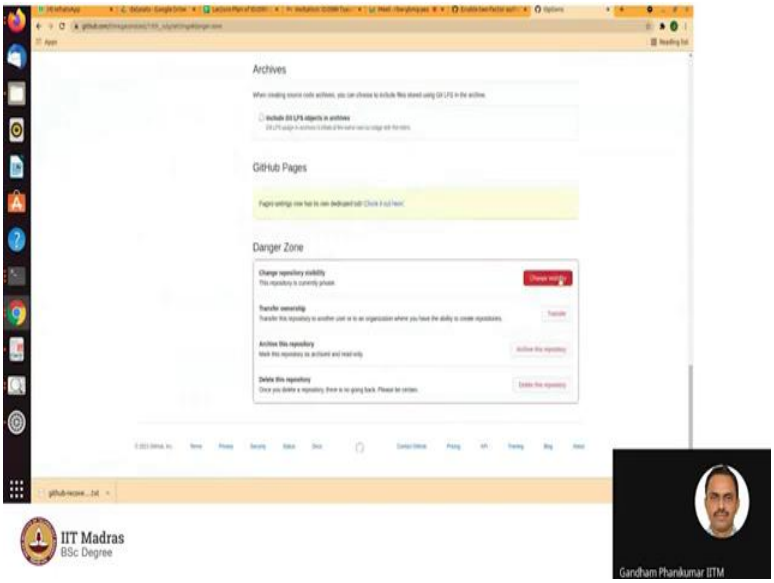


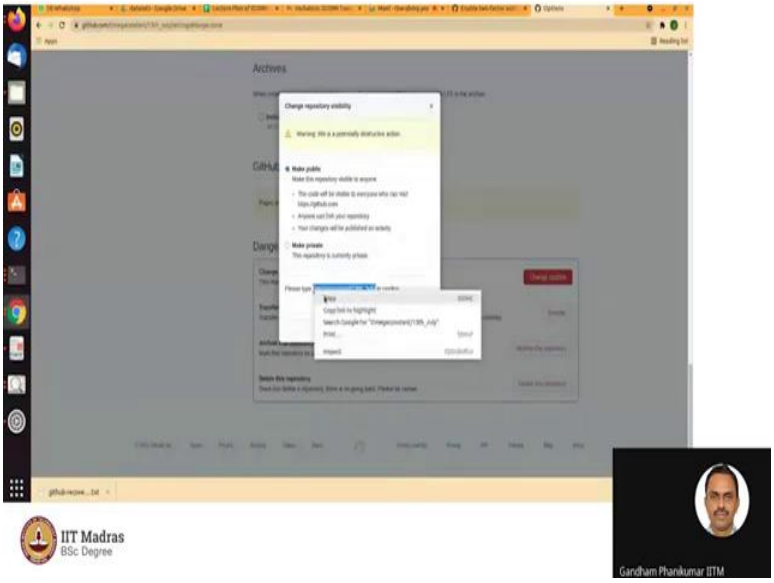


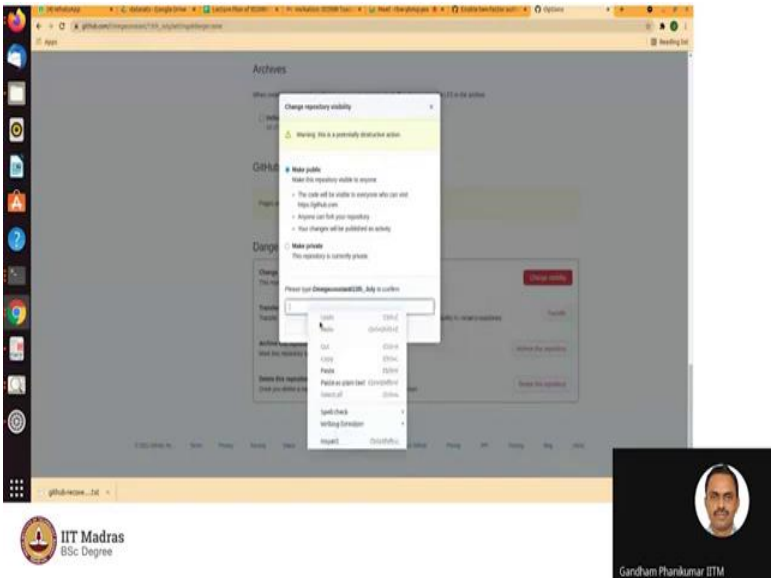


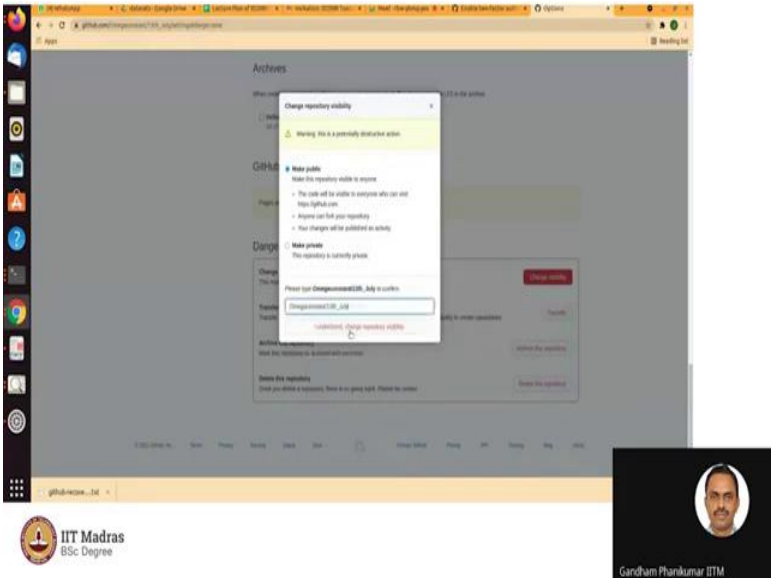
So, go to that repos click on your ID make it public click on it yeah, setting settings you can do settings also or you can do it in clicking on or your user name also. Manage Access, manage access. Yes, come down. Yeah, that the, there is a manage I can there is a link called manage.

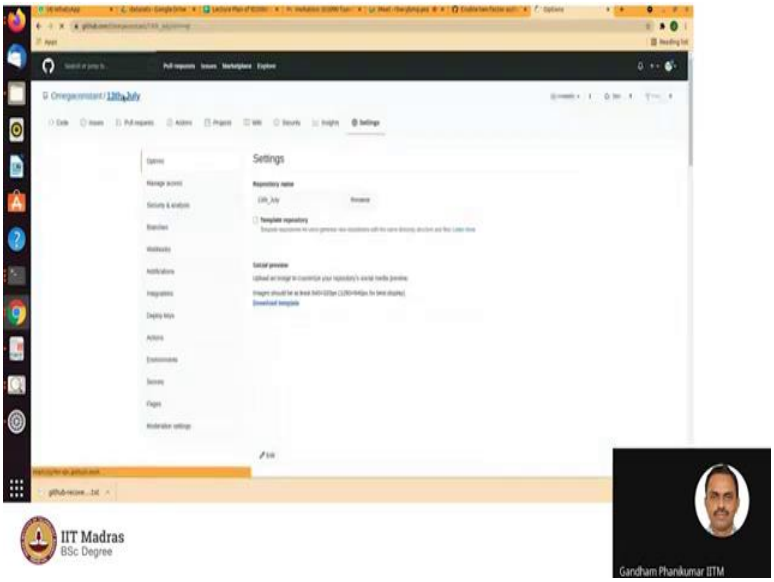
(Refer Slide Time: 25:39)

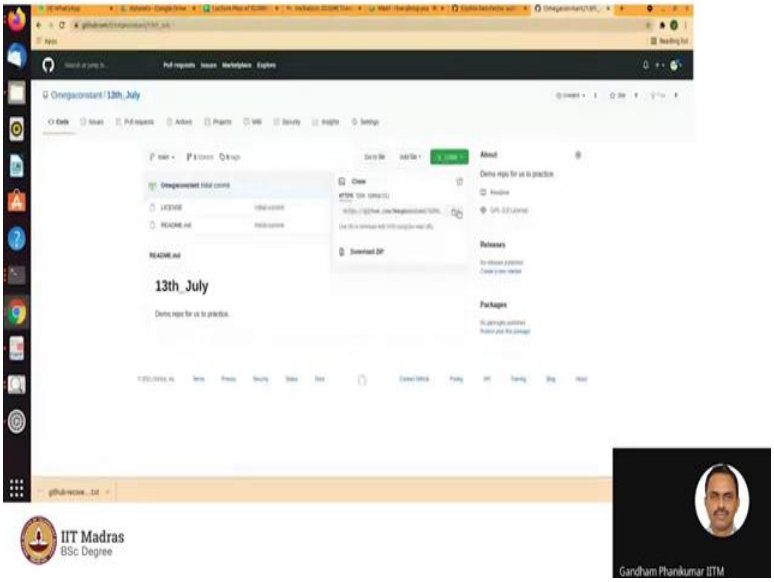


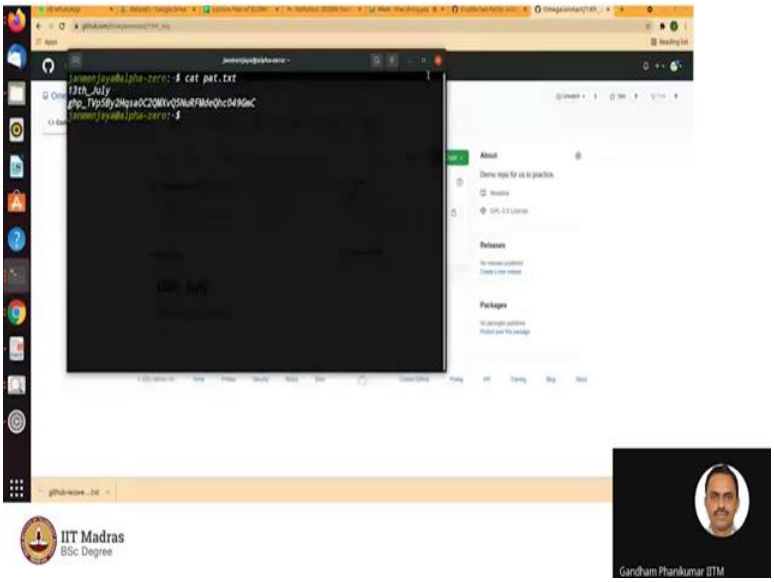



Archives
Change repository visibility
Make public
Make private
IIT Madras
BSc Degree
Gandham Phanikumar IITM


Archives
Change repository visibility
Make public
Make private
IIT Madras
BSc Degree
Gandham Phanikumar IITM

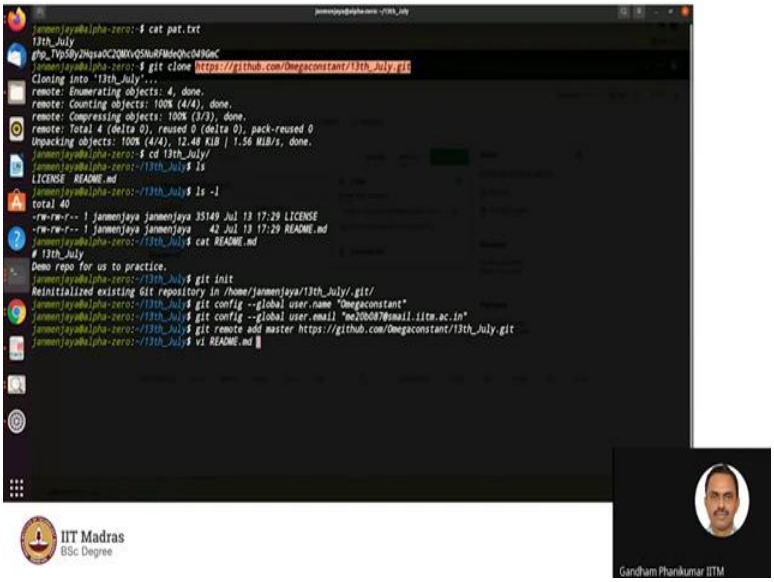

Settings
IIT Madras
BSc Degree
Gandham Phanikumar IITM


13th_July
IIT Madras
BSc Degree
Gandham Phanikumar IITM



So, there you can make it public. So, that at least you can get started. Now so, that was some way of confirming. Now, go back and then try the same thing now. And of course, it will clone without authentication so that at least you get started.

(Refer Slide Time: 26:10)

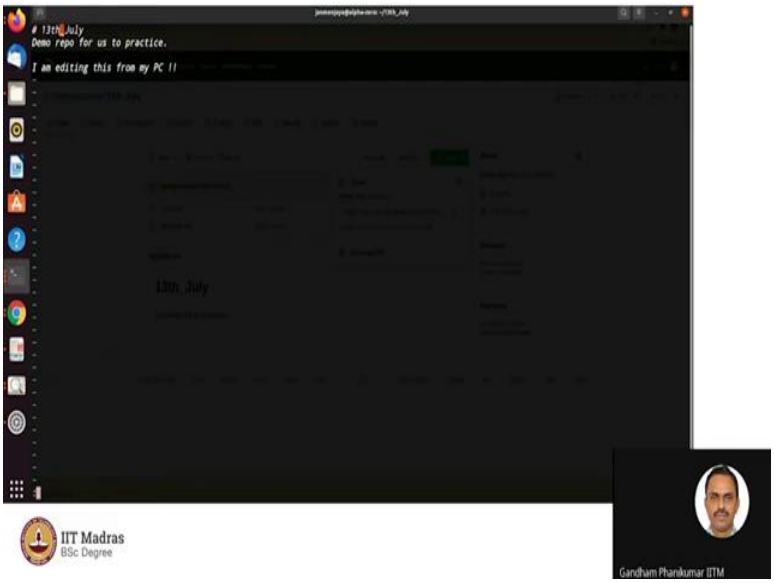


Alright, so it has come. So, you can now go to the directory yes, and now, put ls and check that there will be two files. So, there is one for ls minus l. There is the LICENSE file and the README file, both of them are there, the license file is same as the license for the GNU, which you do not have to worry, readme is what you have kept. And you can just put cat readme dot nd to check what has been entered normally it will contain the name of the repo itself, as the seed.

So, what you do now you need to initialize this, you can first say git init, git space init and therefore, you have got that initialized, so we know what is there in the content of dot gits. So, we do not need to worry now. And then you need to config also git config. And you have to tell some names. So, you can say git config minus minus global and user dot name and then in give a space, and then give in quotations your username.

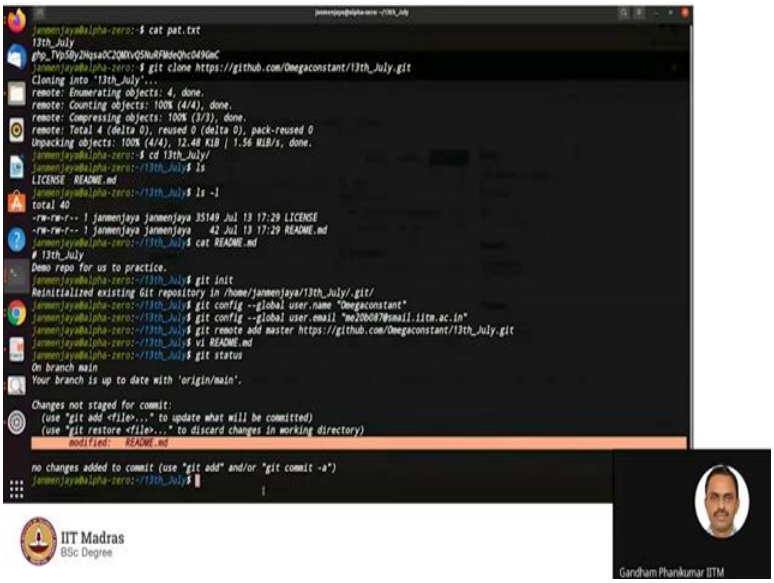
Same way, you also configured the email. Email, user dot email. Now, you have got configured so, we now need to tell what is the remote location it should be talking to so we have to do that git remote add, add, and then you have to give a nickname so we can say master maybe. And then give the URL. So, the URL is same as the dot git that URL, it is there. So, HTTPS up to git that part. So, now that git has been told, what you can do is you can now start doing some changes. So, go ahead and edit the readme dot file. So, that you have some changes being done that.

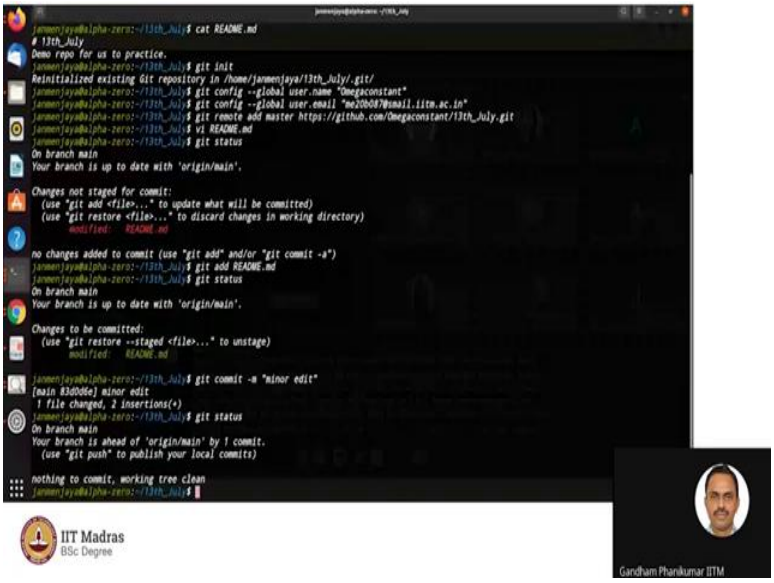
(Refer Slide Time: 29:04)



So, add a line below, which will tell you that you have done something from here. Then say, write something, write something. I am editing this from my PC, something like that. Because you are doing it only from your terminal, you are not doing it on the side. So, which means that you have done the changes from your computer yeah. Now, save it and come out.

(Refer Slide Time: 29:34)





Now, because you have already done git in it once it actually knows the initial status and it has now also seen that the file has changed. So, it will understand that there was something that changed. So, there is no, there is nothing wrong in running this git status command again, and again, how many times you run does not matter.

So, you can just run it every now and then just to check what is going on. So, git status so it understands that you just want to know what is happening. So, it will tell you whatever is happening. So, you can see that right now, you have got a file that has been modified, and it has

to be, going back to the server, so you need to do that, so what you do is first you app, you have to stage it for this pushing.

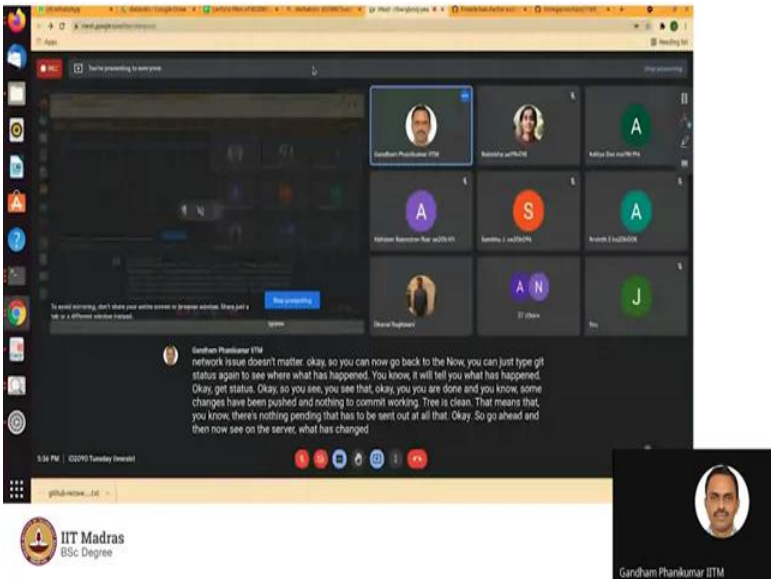
So, you say git add, git add and then capital R, do not put dot do not, do not because make it a habit to specifically add those files. So, capital R and put a tab, it will fill it. Yeah, enter. Now, git status again, to check have you added enough why I am telling you to say git add and then file name is because so sometimes, when you are working on lot of files, it is not as if you want to push everything immediately to the server, you want to push only some of them and remaining you want to push later.

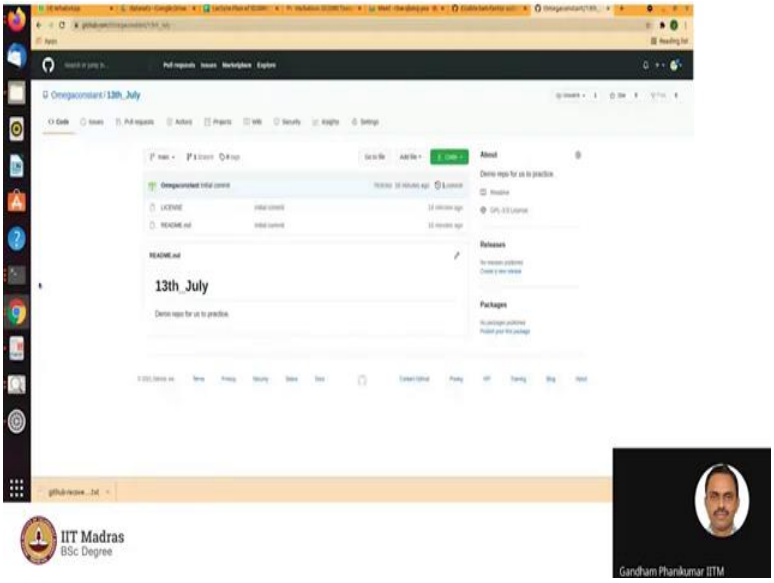
So, if you say git add dot it will push everything to the server, which you may not want to so that is why I was just telling you not to do that. Make it a habit to add files such as specifically. Now, you have only staged it for pushing, but you are not told, what is this going to be commented with? So, git commit minus m, git commit minus m, and then you should give the remark as to what you doing with that particular pushing.

So, you can say git cat commit minus m, git commit minus m. And then in quotes, you give some comment, saying that, minor edit something like that. Maybe Gandham Jaya is unable to hear me perhaps a network problem, you are done that good. Now, so I could not see some of the activity because I typed something in the text and it did not come either. So, I think there was a network issue does not matter.

So, you can now go back to the now you can just type git status again to see, it will tell you what has happened, git status. So, you see, you see that, you are done and, some changes have been pushed and nothing to commit working tree is clean that means that, there is nothing pending that has to be sent out at all that. So, go ahead and then now see on the server what has changed what is there on the server.

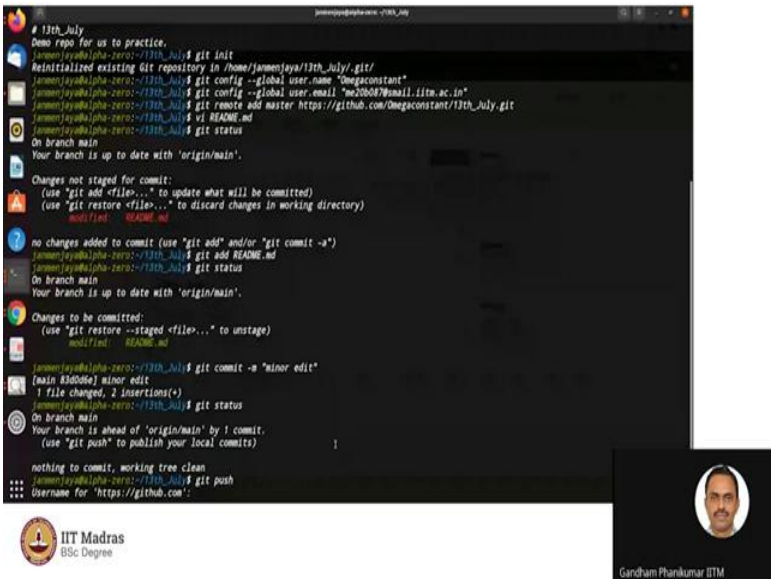
(Refer Slide Time: 32:55)

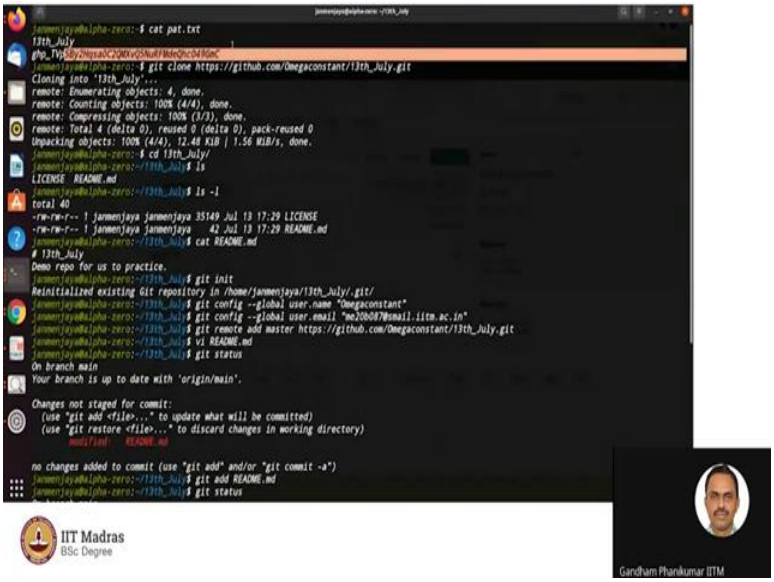




So, GitHub dot com website if you go and look at the page, you refresh the page and see what happens. So, what happened because you have not yet sent it is it pushed we have not done push yet. Go back to the terminal.

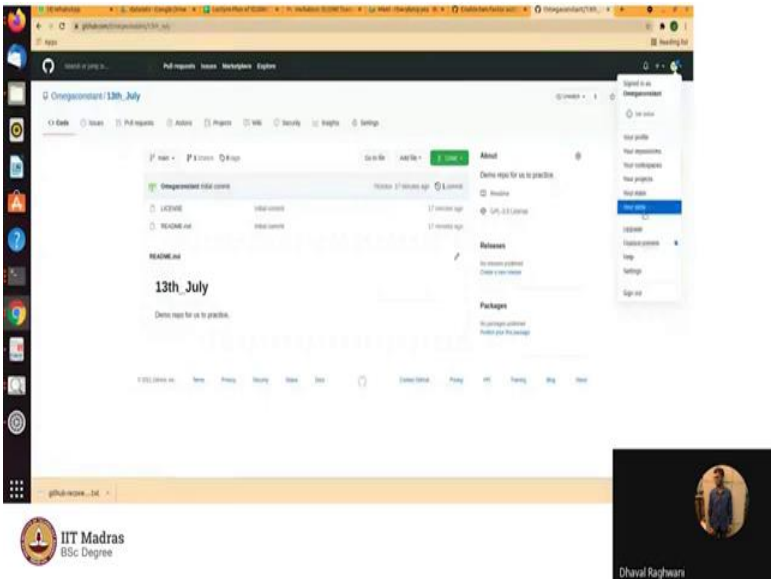
(Refer Slide Time: 33:38)

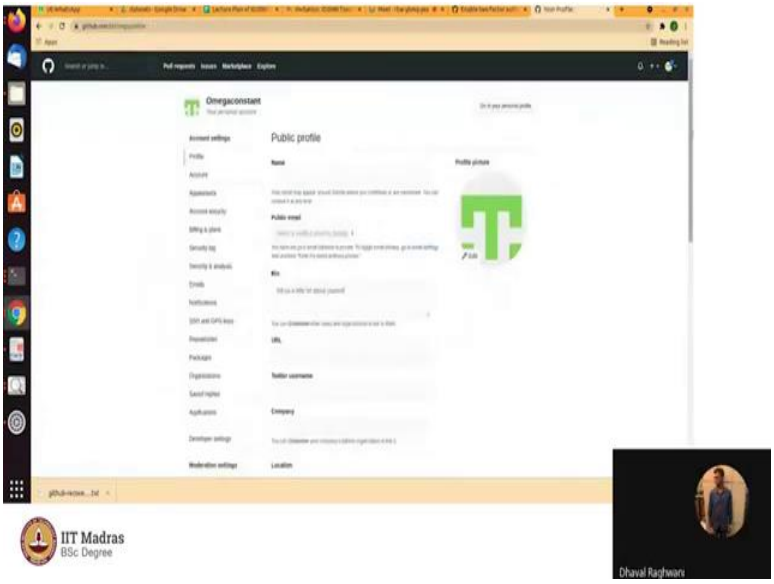


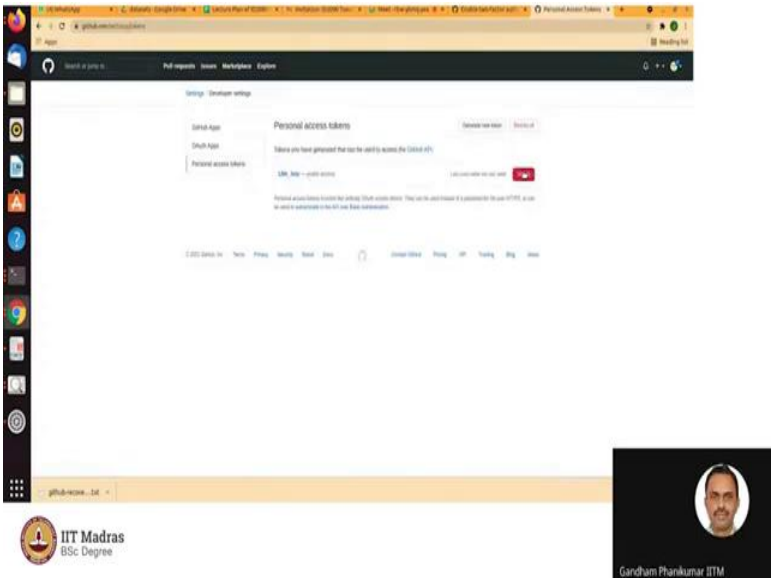


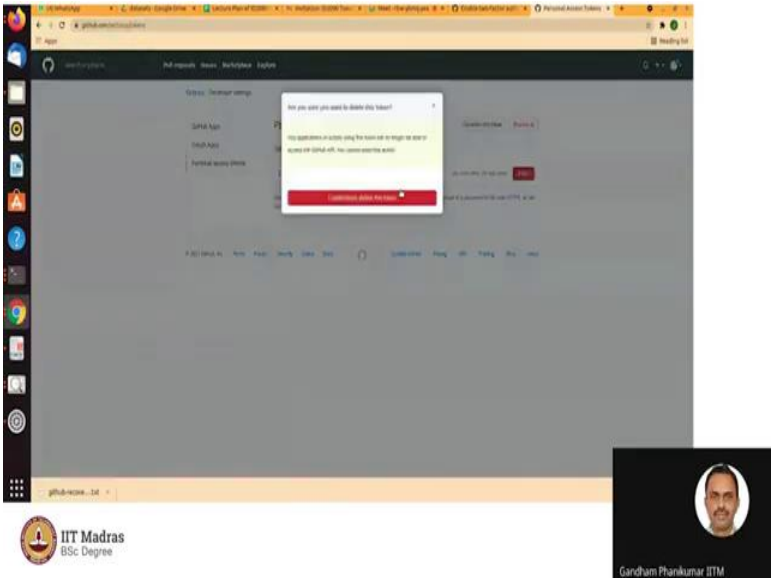
You have not done push it yet. Please do that push after git commit minus m. You have to do push git push. Now, while when you do git push it will ask for username and password so be ready to pick it up. So, if you want to yeah, if you want to copy paste it from somewhere have it ready. So, you have got a problem with your authentication. Your personal access token is actually not working basically. So, what you do is you just dump it and pick a new one.

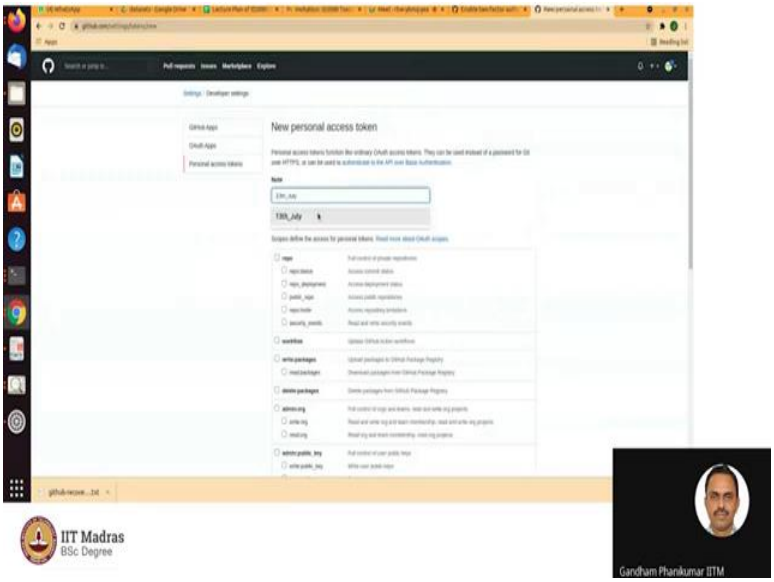
(Refer Slide Time: 34:36)

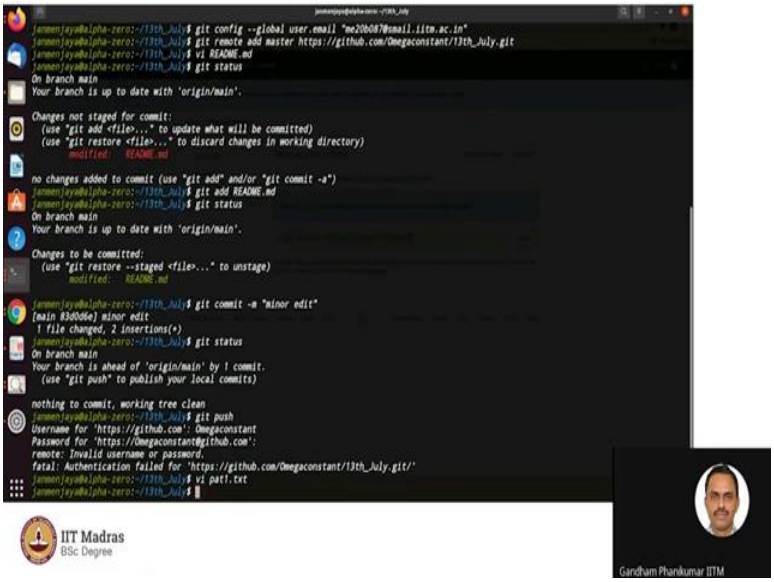





Personal access tokens
IIT Madras
BSc Degree
Gandham Phanikumar IITM


IIT Madras
BSc Degree
Gandham Phanikumar IITM

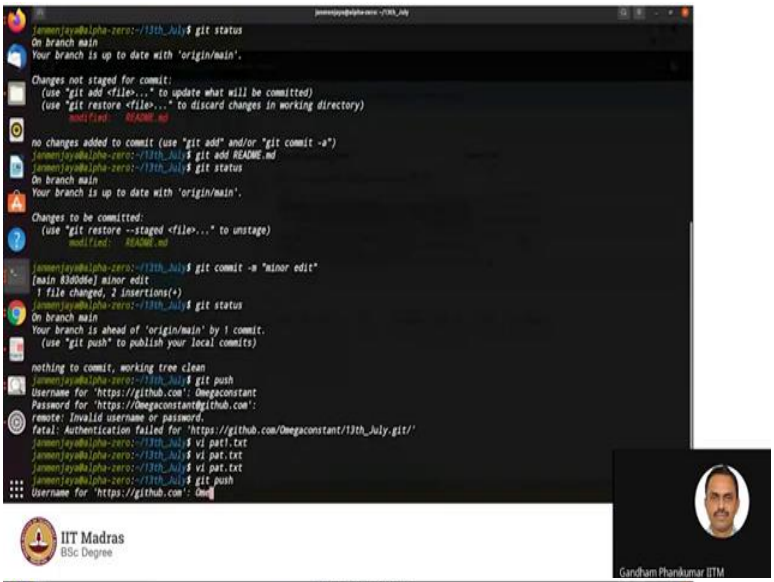


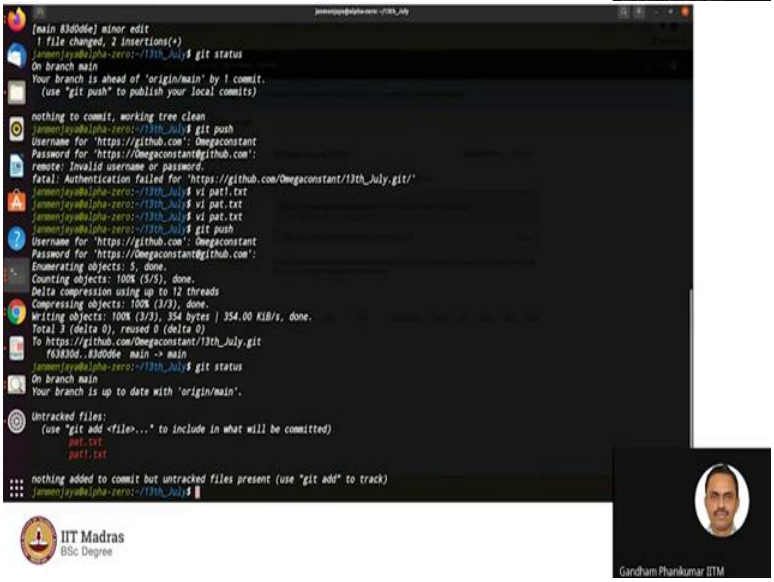


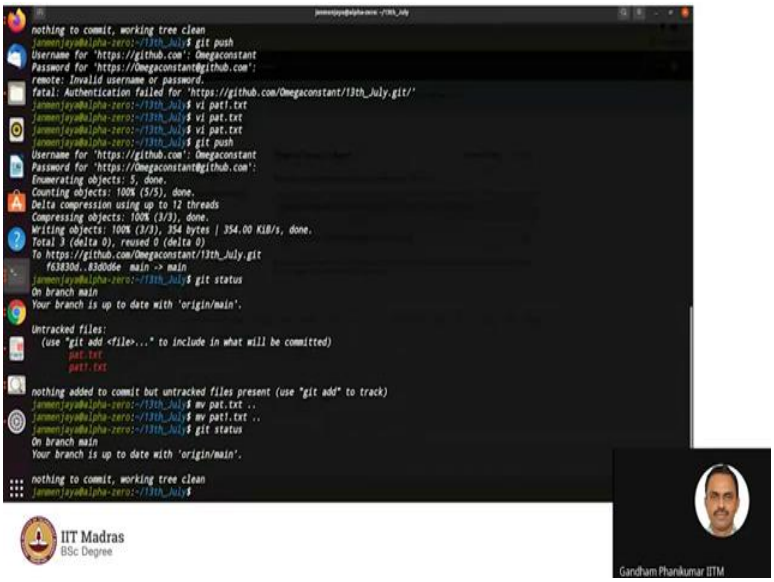
Student: So, when I am facing the same, the same error like what Gandham J is.

Professor Gandham Phanikumar: Yes, yes so you are PAT is not working whatever PAT you have had it dump it and get a new one. Delete it and create a new one copy that and maybe save it to someplace. No, no just over write it because do not keep the old ones just dump it to the old one just dump it just type no wq just q, just go to vi pat dot text and delete whatever was there, that is the new file no problem.

(Refer Slide Time: 36:08)





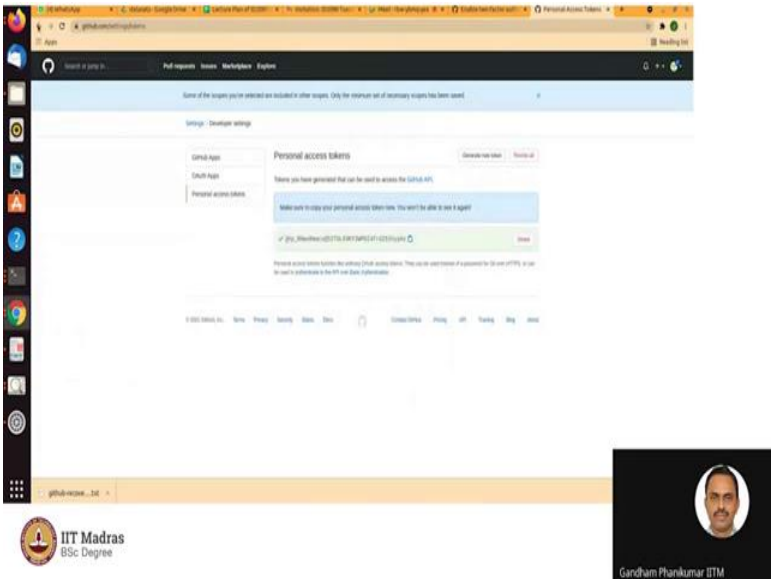


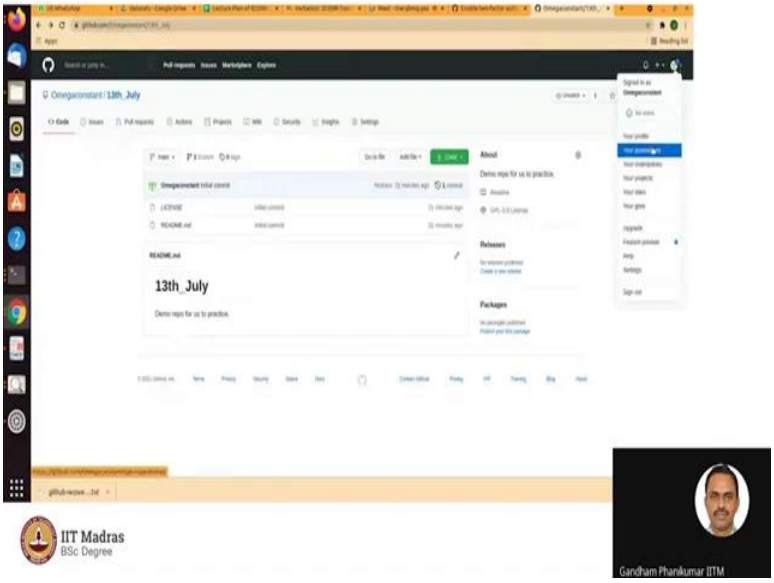
No, no paste it here know, have it accessible to you, and the press O escape and press O. Yeah so, what was it for some way by which you identify when back in the data that the (())(36:38) fine. So, you come out and then have that saved and so that you just simply copy paste it. So, you can try to push it again now so, it has authenticated and it has pushed, so, it was a problem of somewhere some copy pastes some edits something happened.

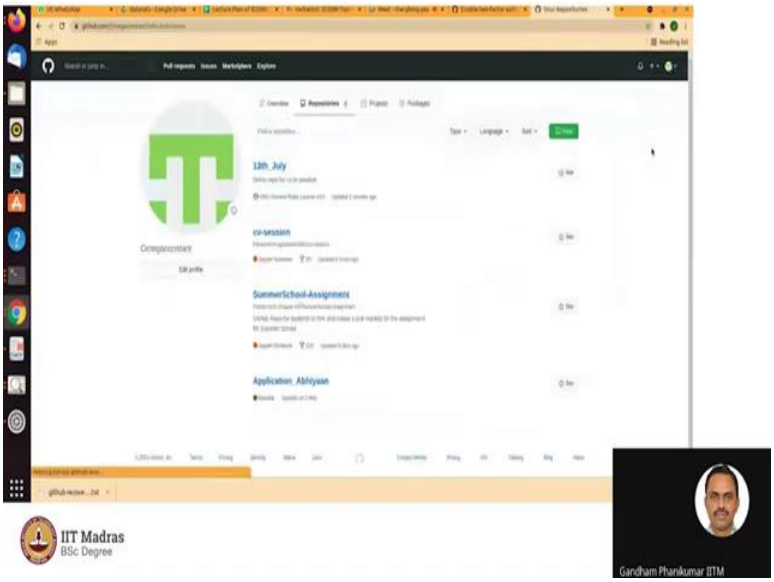
So, you have now successfully pushed it. So, the data has gone to the server now you put status git status. So, it will tell you that earlier it was saying that you are one step ahead of the master. Now, it will not tell that so it will tell that you have done of course, you have done a mistake of creating those two files here.

So, I normally keep it in some other location. So, please do one thing git rm pat dot wait, wait, wait, wait, do not do that. Move the files to one directory above mv pat dot text dot dot, and do that with the other cell also, so that you do not lose the files necessarily. And now, again type git status. So, there is nothing here nothing to work so nothing to worry now that is fine. Now, you can go back to the server and see what has happened. Go to the browser.

(Refer Slide Time: 38:15)

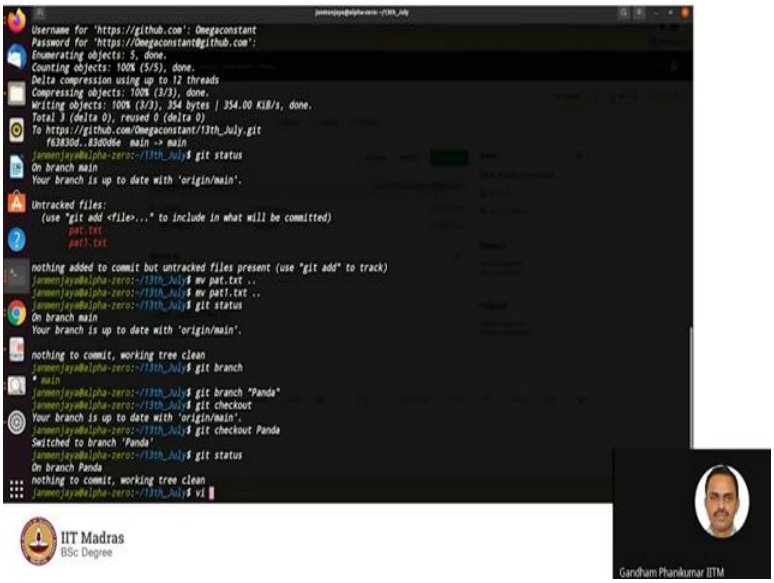






And open your repo do not do back, back back buttons generally, make it a habit to go and, click on your icon on the top right. And go to the repository. That is a good habit. Do not use back buttons generally, as far as possible, because it does have some forms. You can see that the line that you have added has come. So, now it is done. So, which means that you can edit the files and it can come to the server. Now, this is fine. This is not, a big progress but it is reasonable. Now, the next step is to actually do some branching and merging. So, we will do that go to the terminal.

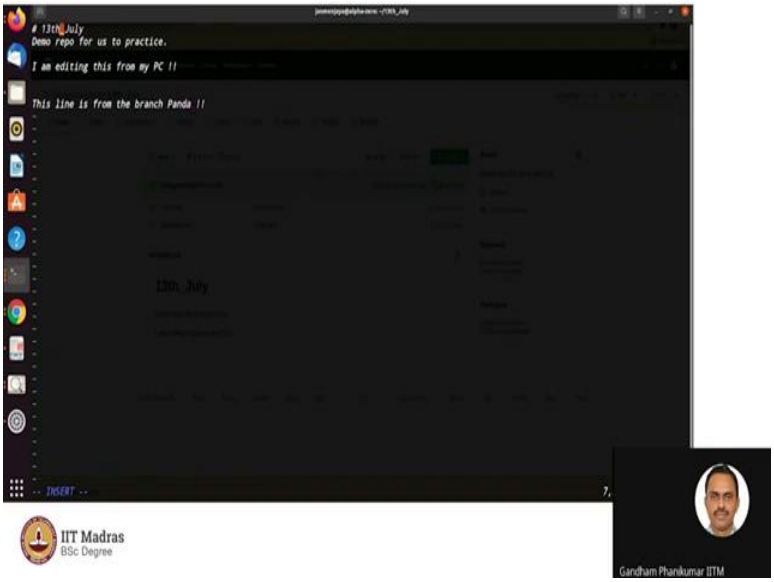
(Refer Slide Time: 39:09)



So, what you do is type a git branch and see where you are what branch you are. So, you are in the main branch. Now, what you do is, you can say git branch and then give a space and then you give a name for the branch, you can say, you can say some name, just the types of Panda some other names so that it is something very distinctly, something that you have created now.

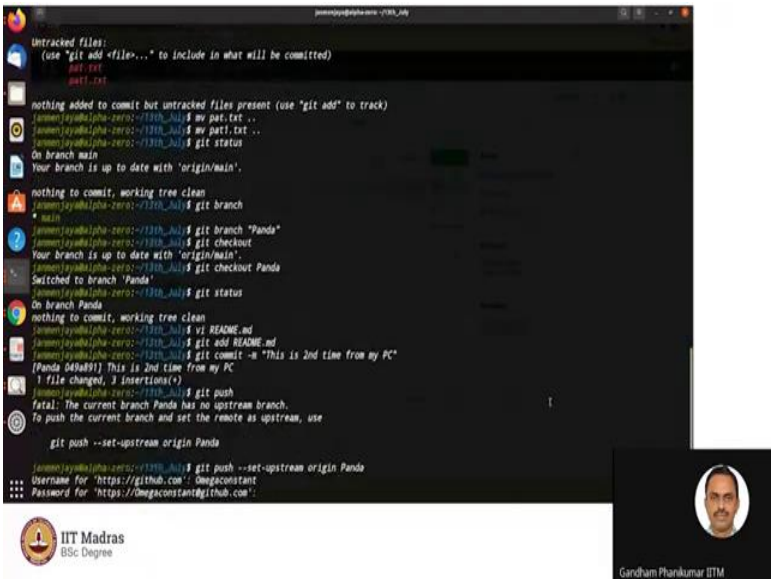
So, enter and now git checkout, and you can see what you can, you can make out what are what branches are available for you enter and then see what happens. So, you can now it is you are actually, linked with the main branch now so you can actually check out the other branch so you can say git checkout and then put the branch name now, Panda. Now, put again, git status, so it now knows that you are on the branch Panda, you are not on the main you are on the panda branch. Now what you do is that you go ahead and edit that file again, add one more line, vi README dot nd.

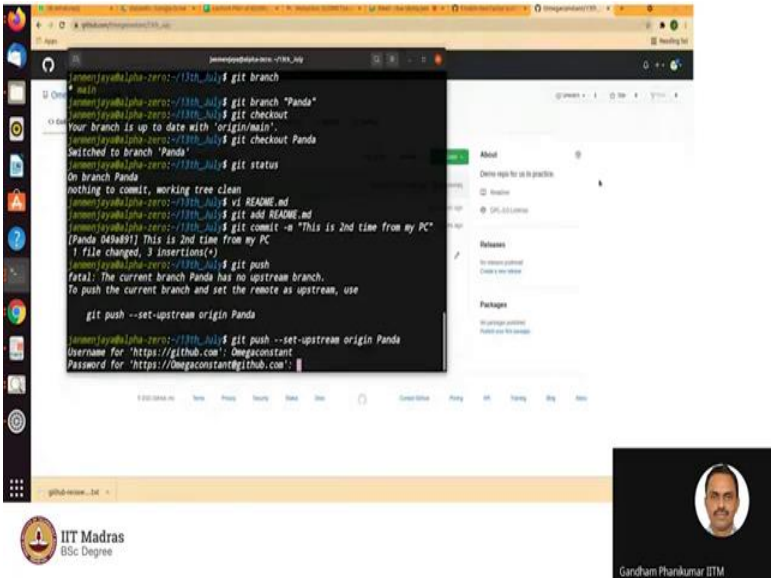
(Refer Slide Time: 40:55)



Now you write one line below you write saying this line is from the branch Panda. So, distinctly we understand that we have done some change from a different branch that is a silly way of keeping track, come out of it.

(Refer Slide Time: 41:17)



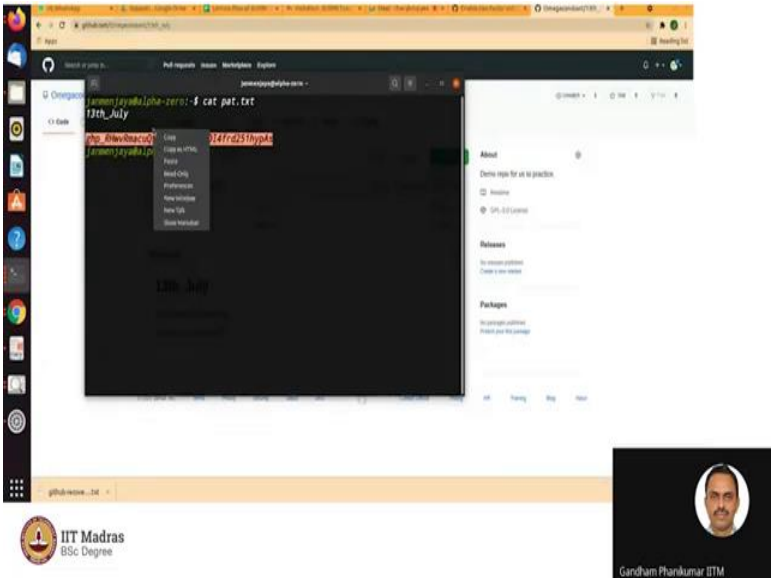


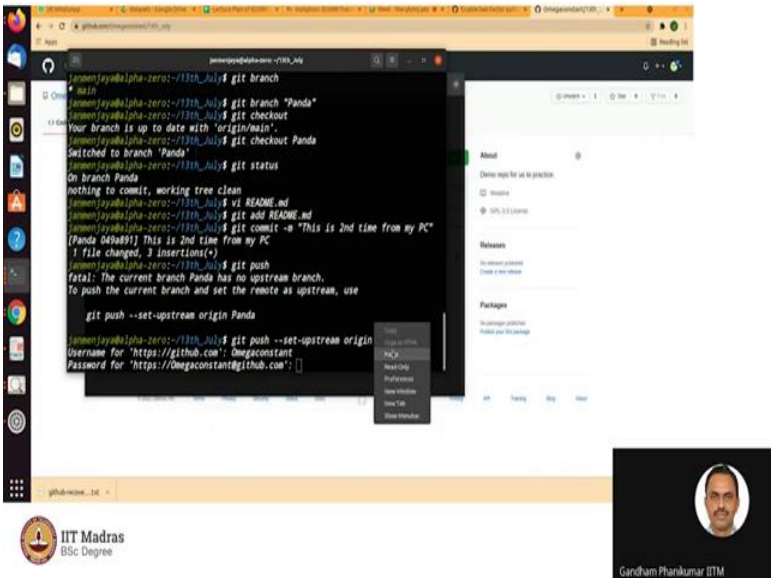
Now, you need to push it, so push this branch. So, we need to, so you can, you can see what would happen if, if you want to do that, so, of course, you have to do all those, git add and then, git commit minus m all of that you have to do. So, there is a talk, I need to attend at 6 o'clock and I need to join a bit mean, I had 5 minutes ahead.

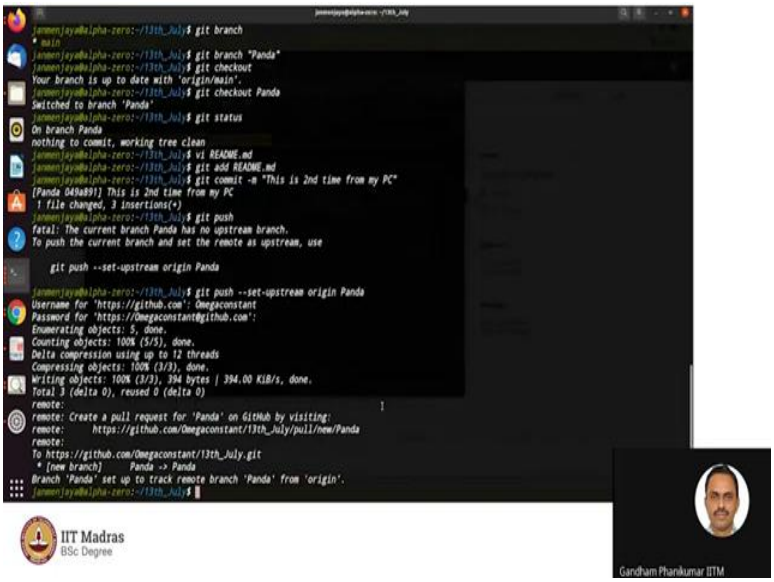
So, I will close shortly. And I will continue in the next session. So, do not worry, because there is a demonstration of software that is being happening. So, we will stop in a moment. In a few minutes and we will continue from where we left. And I request Gandham Jays also to sort of help me by being volunteers till we finish this.

So, yeah, do go ahead and push it and see what happens. So, it is pushing to what? So, there is no upstream branch called Panda. So, therefore, you have to set it and you can do that command to do that no problem, because there must be a branch on the remote location also for you to do that. So, it is requiring that to be created so you have got that password somewhere so you should always cat it to some files and then use it up. So, you can open a terminal no, no, you can open a terminal it is there.

(Refer Slide Time: 43:09)

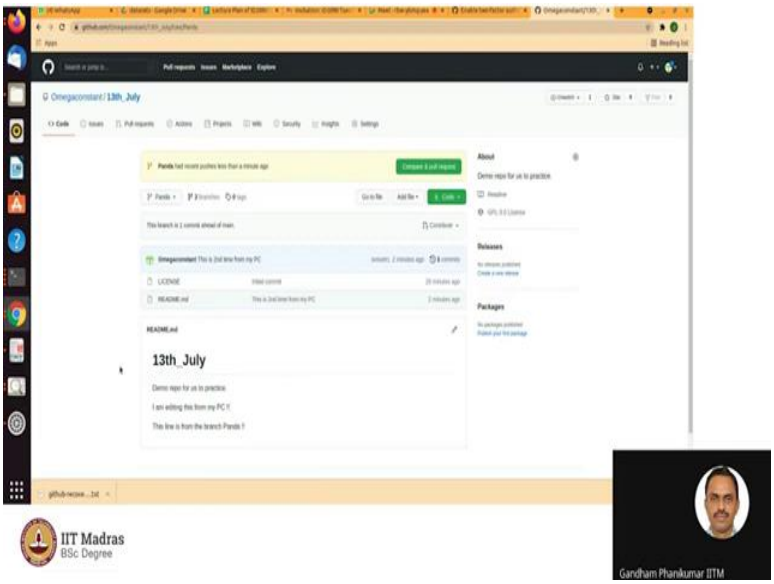






You have it in pat dot text somewhere yeah. So, that why I say you keep a new one, so that you have no confusion. Particularly when you are learning if you have too many things hanging around, you do not know what, is a problem, a problem is because of a wrong key or some wrong command. So, now we have got this boost. So, you can go back now.

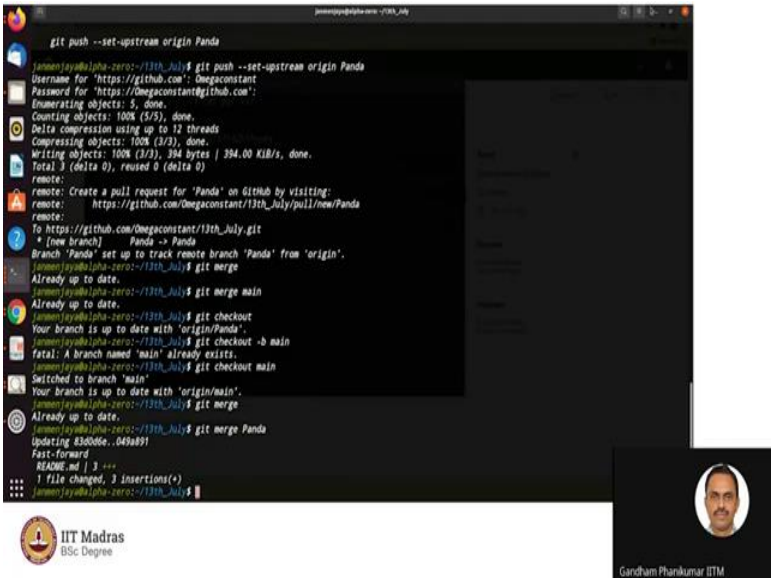
(Refer Slide Time: 43:41)



So, now you can see refresh this page, you can see that there are two branches? There are two branches. So, you can now see that the README dot md is coming is from the main branch. So, change the branch beside that, two branches, there is a button. So, change the branch to Panda, and then you see that from that branch, there is another version of it.

So, now you can see that there are two versions that have been kept. Now, one of the things that we need to learn to do is at some point when you are working with a branch, you want to say that, now I am done with this branch, so therefore I want to merge it. So, that is something that we can actually do it from the command prompt, and do that from the command prompt by typing git merge.

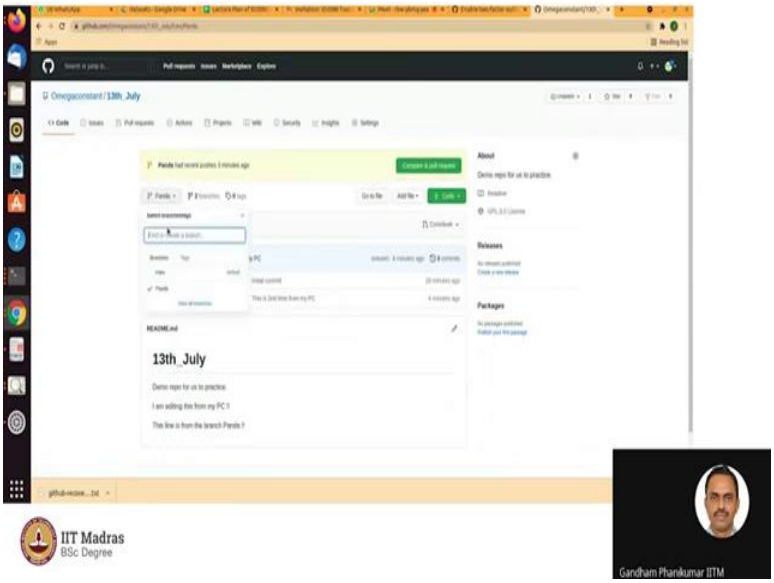
(Refer Slide Time: 44:35)

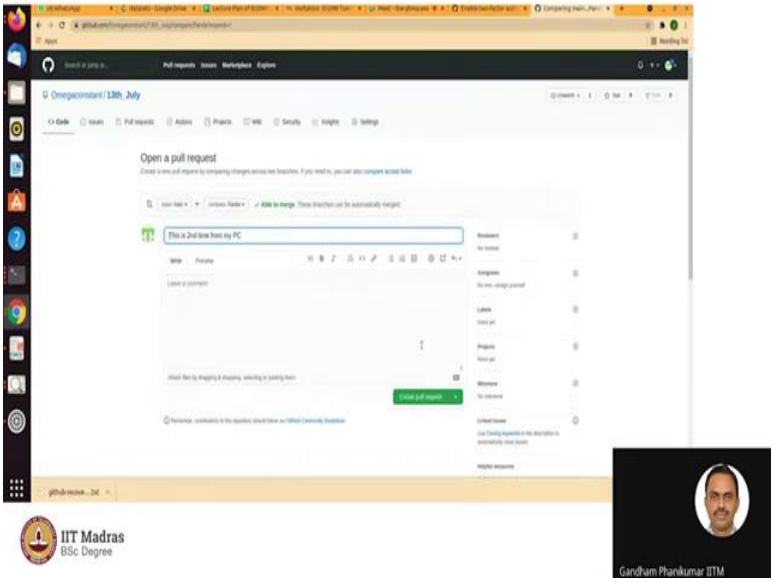


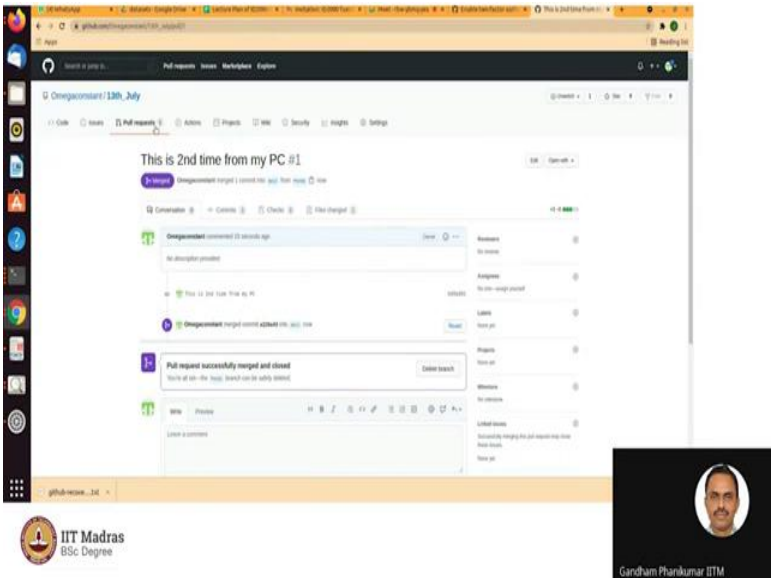
And see what happens what does it tell you, what help us tell you. So, you need to merge the panda branch with the main branch. So, you can tell git merge and tell the branch with which you would like to merge, git checkout and see where you are. So, checkout and be on main and run the command again to check where you are, do not create minus b means it will create that brand, do not create it solid is here, minus b is for creating it and all that.

So, you have now switched to the main branch, of course, now you need to merge and what you want to merge, you have to tell now you have to merge. So, now it is done, now you can see that you have merged. And this will reflect on the screen, and we can go to the browser and see what happens.

(Refer Slide Time: 46:24)

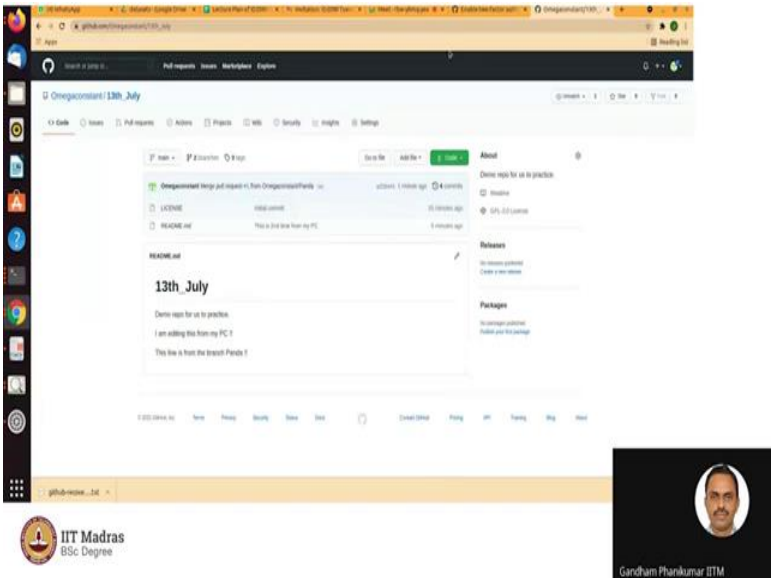






Refresh the screen. Now you will see that go to the main branch. Now you will see, refresh that, wait, wait, wait, no, you have done a push request. So, there is a button there on the top right. Compare and pull request. Yeah, click on that. So, it does check whether the files can be merged or not. And all that so create that, bottom there is a create pull request, come it will verify it will check and then go and say merge. Now, go ahead and merge. Now confirm it. Now, you can see that go to the screen 13th July click on it.

(Refer Slide Time: 47:28)

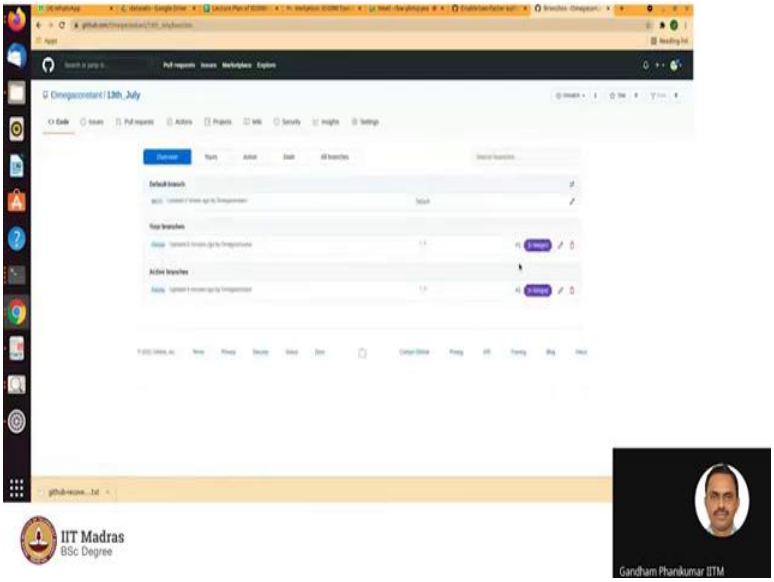


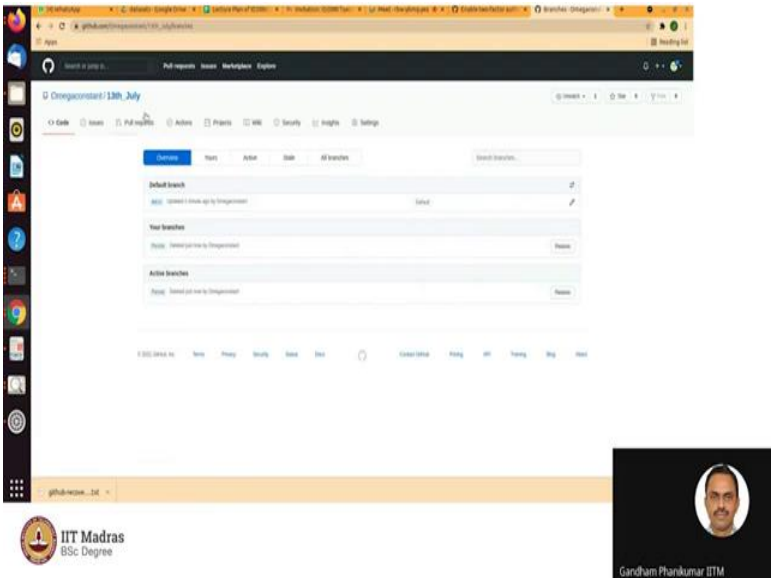
Now, you see that it has come there, on the main branch. So, what is happening is that you have merged the contents and on your local machine and also on the remote machine. And the remote

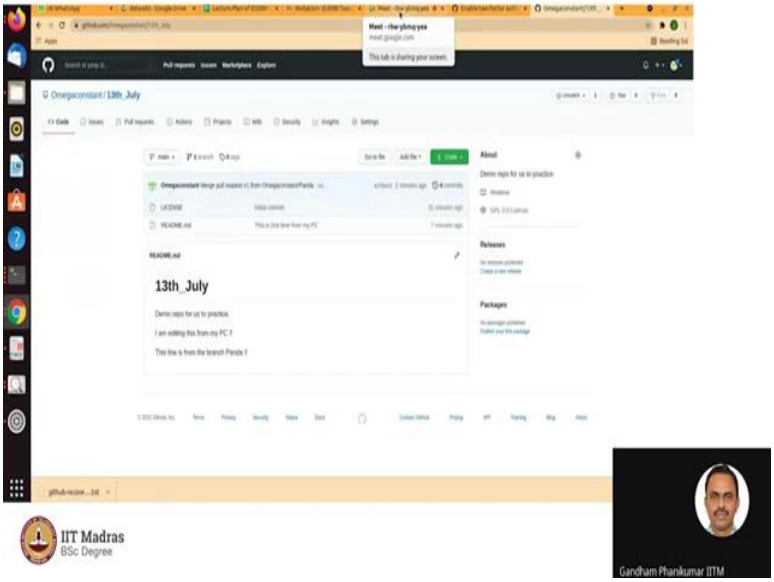
machine has certain workflow. That is it is actually checking the differences between the two versions, and whether there is any conflict, conflict in terms of when was the change done in terms of timestamp versus what was the content. So, if what you are going to merge is newer, then there is no problem. If both of them have changed, then there is a conflict. And that has to be resolved, we will look at it in the next class.

But for now, what has happened is that whenever you compare there was no problem of merging. So, it allowed you to go ahead and merge it. And that is usually the scenario when very few people are working because they have a way of working around in that. Now, what you can do is you can delete the other branch. So, you can click on the branches, two branches is there, click on the two branches.

(Refer Slide Time: 48:34)







Now you can actually delete that Panda, because you have merged it, so you do not have to have it. And now you can go to your repo and it says that there is only one branch and there is a main branch. So, this is what we do normally that is we have some activity; we create it on a branch because we do not want to disturb the original code.

And then after we are convinced that it is working fine, we will merge it. And then after that have dropped, the branch because we no longer need it. So, this is something that will continuously keep on doing there will be branches that will be created and merged, created and merged, created and merged. That is a part of the coding cycle. And that is what we have basically learned.