CS 240: GitHub Set Up Transcript

[00:00:00] All right. Now, let's, let's move our chess project up into GitHub.

Start visual description. The professor demonstrates how to move the chess project up into GitHub. End visual description

[00:00:06] So um as we said earlier, first thing you need to do with GitHub is create your own account.

[00:00:12] Once you've created an account, the next thing you'll probably do is create a repository.

Start visual description. The professor demonstrates creating a new repository on GitHub, naming it "chess demo," and adding a description. End visual description.

[00:00:18] So in this case, I'm just going to click the new button and I'm going to create a new repository.

[00:00:23] Of course, every repository needs a name.

[00:00:25] So in this case, I'm going to call it chess um demo.

[00:00:33] You would probably just name yours. Chess, I think I've already used that name.

[00:00:37] So I'm going to call it chess demo and it says yes, you can use that name. It looks good.

[00:00:41] You can put a description in there if you want. So, chess project for CS 240.

[00:00:51] Now, having given it a name and a description, the next thing you need to decide is whether you want your repository to be public or private public means anybody in the world can access it and see it.

[00:01:02] That doesn't mean they can modify a bit. It means they can see it, or they can read it.

Start visual description. The professor explains the difference between public and private repositories on GitHub and recommends making the repository public for the class. End visual description.

[00:01:06] So if you make a repository public, you're, you're essentially sharing it with the world.

[00:01:12] Now, other people won't be able to change it, like I said, but they will be able to read it.

[00:01:19] So if you don't want people to be able to see your code at all, then you need to make your repository private.

[00:01:25] So public or private.

[00:01:28] Now, in a basic get hub account, I think there's a, there's a limitation on how many private repositories you can actually make.

[00:01:36] Um, I, at least there used to be, you'd have to go check.

[00:01:40] But um, as a student, when you create your account, you can actually, um, tell GitHub that you're a student.

[00:01:50] And so if you're a student, you can get a, a better GitHub account that allows basically unlimited uh public reposit or private repositories.

[00:01:59] So, um, making this choice, you know, if, if you're really doing some work that you don't want other people to, to, to see or to steal, make it private in this class, we're going to ask you to make your repositories public.

[00:02:12] So just click public there, then there's some other options you can choose.

[00:02:17] Um, add to get, ignore.

[00:02:19] Yeah, I'm going to add to get ignore.

[00:02:22] Well, actually my source code already has a G ignore. So, I won't do that.

[00:02:25] You can add a read me to it if you want.

[00:02:27] Um, you can also pick a license for it.

[00:02:29] There's different open-source licenses that you can choose from.

[00:02:32] We'll just pick the MIT license and then just click, create repository.

[00:02:39] And so now I have a repository, it's empty.

[00:02:46] And so, um yeah, so now I've got my repository.

[00:02:51] Now, the next thing we want to do is we're going to need to create what's called a personal access token because you're going to need to access your repository remotely from your development machine. So, every time you do a push or a pull, essentially your, your git is going across the internet to access your, your GitHub repository.

[00:03:15] And so to keep that secure, um you have to essentially um log in when you run those GIT commands that, that actually connect with GitHub and you have your username and password, which you created when you set up your account.

Start visual description. The professor demonstrates creating a personal access token for secure authentication with GitHub. End visual description.

[00:03:30] That's one way you can authenticate with GitHub.

[00:03:34] But if you do that, um you're sending your password over the network over the internet probably more than you want to.

[00:03:40] So another way to do um authentication and, and GitHub is to use what's called a personal access token.

[00:03:48] So let's uh I'm going to click on my account here. So, I'm going to go up to my profile and down here, I'm going to select settings and there's lots of settings you can configure of course and get.

[00:04:02] Uh but down here at the very bottom, there's uh what's called developer settings. And so, your developer and so you can select developer settings and then um you can set up different kinds of security for your GitHub repository.

[00:04:18] In this case, we're going to recommend using a personal access token.

[00:04:25] And we're going to use classic tokens.

[00:04:26] They have some new ones called Fine Grain Tokens, but we're going to stick with the ones that are a little more tried and true.

[00:04:33] And we're going to use uh classic tokens.

[00:04:37] And so basically what a token is, it's just a random string that you can use in place of your password.

[00:04:42] So once you've created a personal access token, you can use that instead of your password.

[00:04:49] So you could authenticate with, with GitHub, using your username and your access token instead of your password if you wanted to.

[00:04:56] And that just, you know, results in you um passing your password across the network less frequently, which is more secure.

[00:05:06] And so to create a personal access token, all you have to do is click generate new token and I'm going to create a classic token and it's going to make me um authenticate to do that. So, I'm going to, you do my two-factor authentication.

[00:05:24] And so now once I've uh authenticated, then I can create a new axis token.

[00:05:33] You can give it a note just to describe to yourself what this axis token is used for.

[00:05:40] And you can say how long it will last, when will it expire? And you can select all kinds of uh different permissions basically that the token has.

[00:05:52] And so you can decide what you want it to be able to do, I'm going to select repo because I'm going to use this token to access my repositories and then having configured my token, I'm going to go ahead and, and generate it and here's my token.

[00:06:09] So I want to copy that token and I want to put it somewhere safe.

[00:06:14] I need to probably put it in a file and save it somewhere.

[00:06:17] Um, because that's the token that I'm going to need to use.

[00:06:20] Um When I access GitHub from my local development computer.

[00:06:26] Now, once you've created your token, then you can configure your, get on your development machine to use that access token.

[00:06:37] There's a lot of different options on how you actually um configure your access token.

[00:06:42] And so um I'll just refer you to um websites that talk about credential storage with GIT.

[00:06:52] So here's a good, good website you can go to.

[00:06:55] It's on the website git dash scm.com.

[00:06:59] And it has an article here on credential storage.

[00:07:01] That's basically different options you have for storing your access token on your development machine so that other people can't steal it.

[00:07:10] Um And once you've installed that, that token on your development machine, then g will automatically use that token when you access GitHub.

[00:07:20] OK.

[00:07:21] Having created our personal access token, the next thing I'll probably want to do in my repository is, is, uh I'm going to go back to my repositories, find the one I just created.

[00:07:46] Next thing I want to do is I'm going to need to grab the URL for my repository.

[00:07:50] I'm going to need this URL to access the repository from my development machine.

Start visual description. The professor shows how to grab the URL for the GitHub repository and clone it to the local development machine. End visual description.

[00:07:55] And so if you click on this button named code, you can see it brings up this um http URL, that's, that's the URL of my repository.

[00:08:08] So having copied that URL. Now, what I can do is I can go to my local development machine and I'm going to go back to my CS 240 folder.

[00:08:24] So this is the folder where I'm keeping all my CS 240 projects.

[00:08:28] If I look inside there, I have this chess project already created.

[00:08:32] This was just created locally. This is not the GitHub project.

[00:08:35] This is the one I just created locally.

[00:08:37] So what I'm going to do here is I'm going to clone my GitHub repository.

[00:08:43] So I'm going to type in GIT clone and paste in the URL and we can see here that I now have a, a local copy of the chess demo repository. So that is the one that came from GitHub.

[00:09:04] So if I change into that directory, now, I can see that um I do have a copy of that repository. Now it's mostly empty.

[00:09:16] But you can see, I do have a doc git folder.

[00:09:19] So I do have a complete copy of that repository on my local development machine.

[00:09:23] And like I said earlier, you can do commits to your local copy of the repository all you want and then eventually you're going to want to run a get, get push and push whatever local changes you've made up to get.

[00:09:39] And you also occasionally want to do, get pull as well.

Start visual description. The professor explains how to commit changes to the local copy of the repository and push them to GitHub. End visual description.

[00:09:42] If I do a get pull right now, it won't do much because I already had the up to date code. But if other people had made changes to the repository since I pulled last, then it would bring those down at that point.

[00:09:56] Ok. So that's, that's basically how you get set up with, with GitHub.