SAP Data Audit Master Class

Python Meetings

How to use GitHub to automatically obtain code for python projects in the SAP Data Audit Masterclass python meetings

Update: How to create your own repositories in GitHub and send your projects to them

1/ Create an account in GitHub

It is necessary to create an account so that you can be added to the 300FMasterclass group on Github.

Incase you don't have account, please click here to create an account: https://github.com/

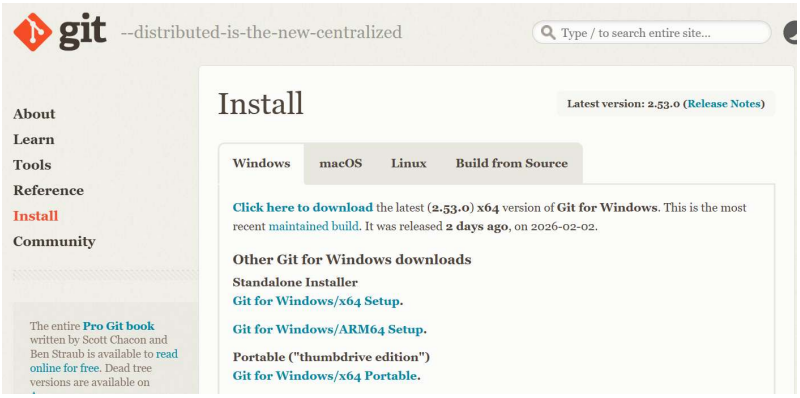
Send your account name to us and we will add you to the 300Framework private GitHub group.

Note: You will need to do this step before being able to download the repositories for the masterclass to your machine.

2/ Download and install the git application

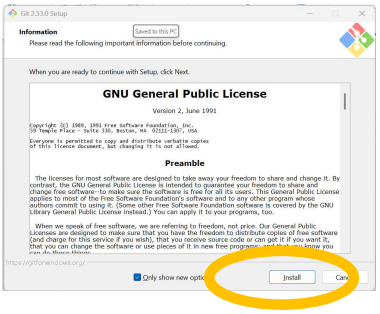
If you would like to be able to clone the python projects for each python meeting, it is necessary to install git on the machine where you use Visual Studio Code.

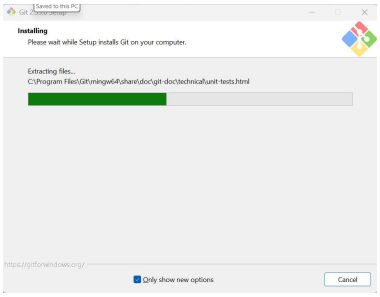
Download the git application from this link and install it on your machine: https://git-scm.com/download/win

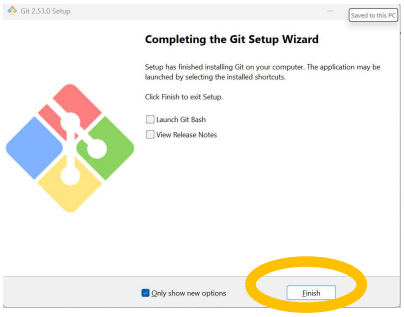


The following installer will be downloaded to your machine: Double-click on the .exe file to launch it.

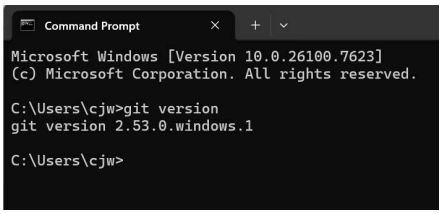
Click Install, choose Next for all questions, then at the end, ***un-tick*** both boxes and click Finish.







To test that you have correctly installed the git application, type CMD in the Windows Search bar, and then in the black box, type: git version



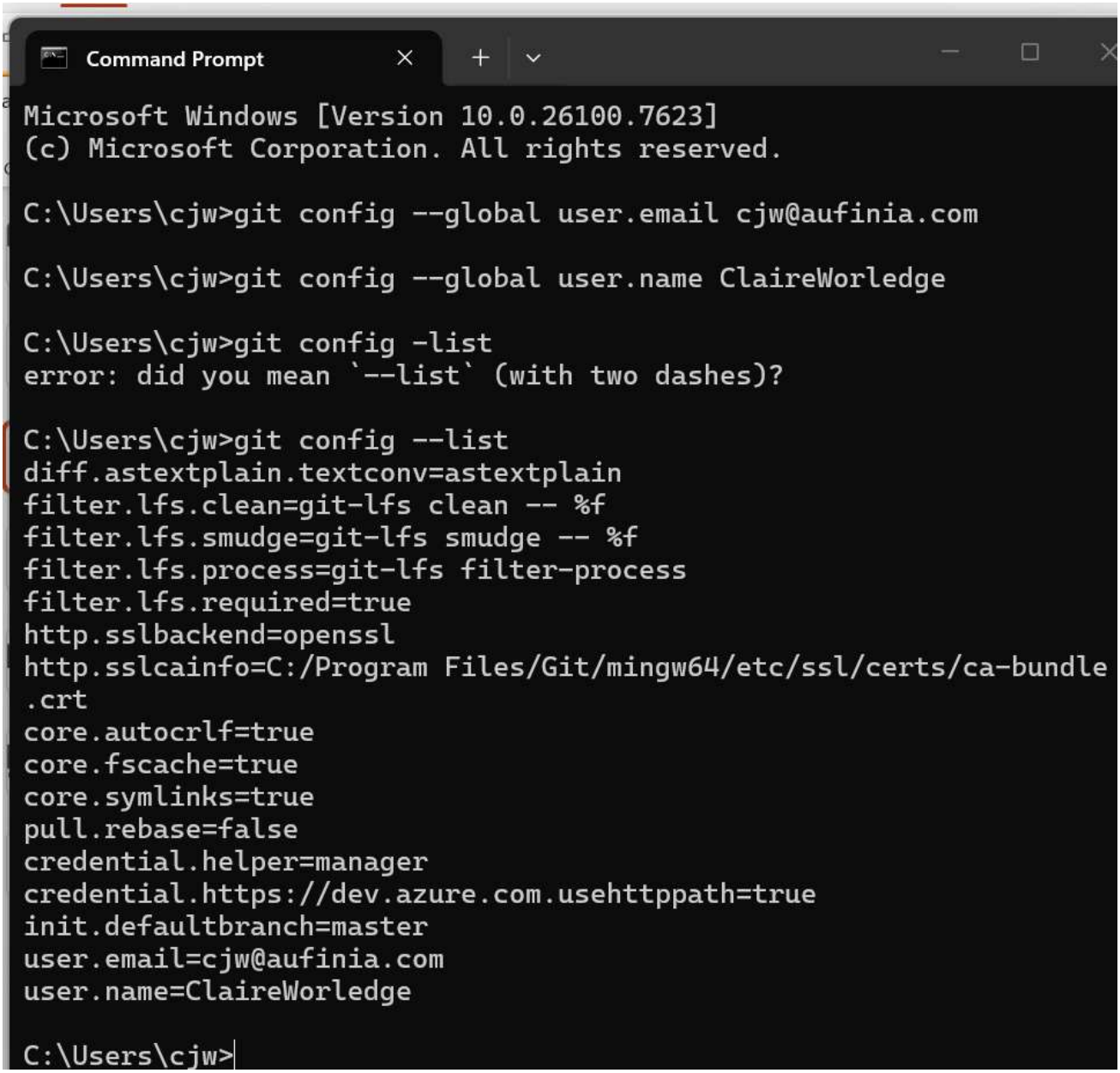
If the git application is correctly installed, you will see the version number.

3/ Configure git application

Next, we need to configure our git application, so that it can login to the GitHub account that you created in step 1.

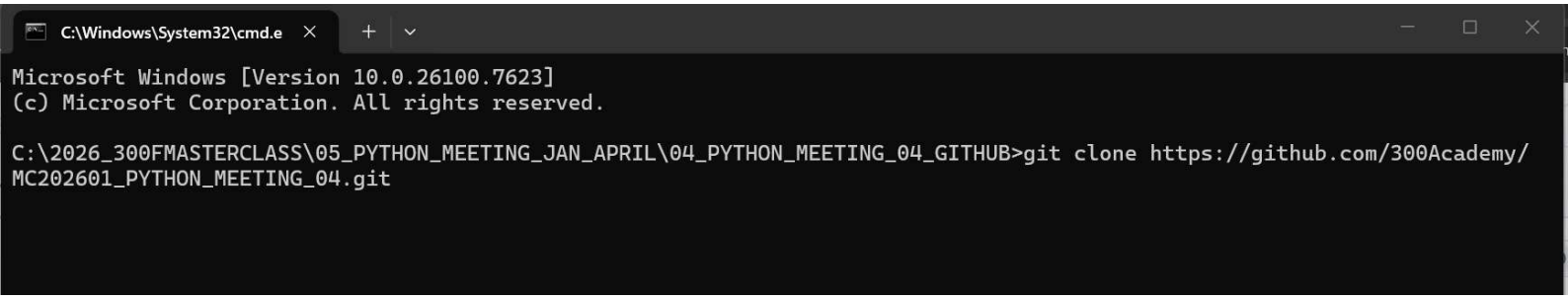
- Type CMD in the search bar of Windows
- Then, in the black box type the following, replacing <your email> and <your name> with the email and username for your GitHub account:
    - git config --global user.email <your email>
    - git config --global user.name <your name>

- To test that your email and username were configured correctly, type
    - git config --list

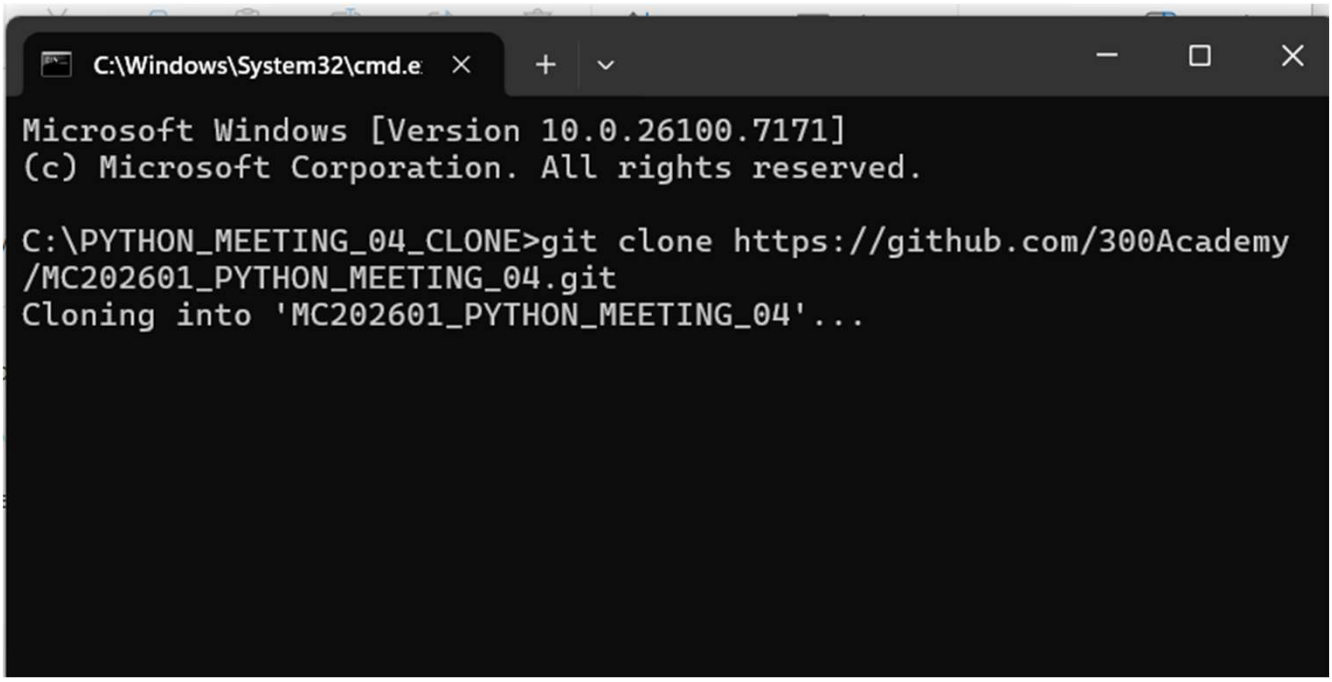


4/ Clone the project from GitHub (1/3)

- Go to the Windows folder, where you would like to put the python meeting project.
- In the Windows search bar of that folder, type CMD
- In the black box that then appears, type the following: (*On one line and replacing for the link of the repository that you would like to clone. Each week, we will give you a new repository link*)
    - git clone https://github.com/300Academy/MC202601_PYTHON_MEETING_04.git



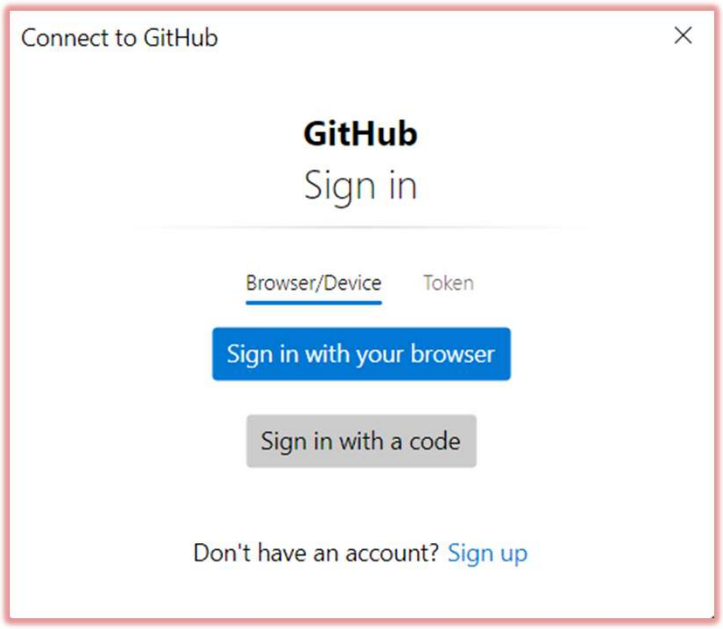
- You will see this message:



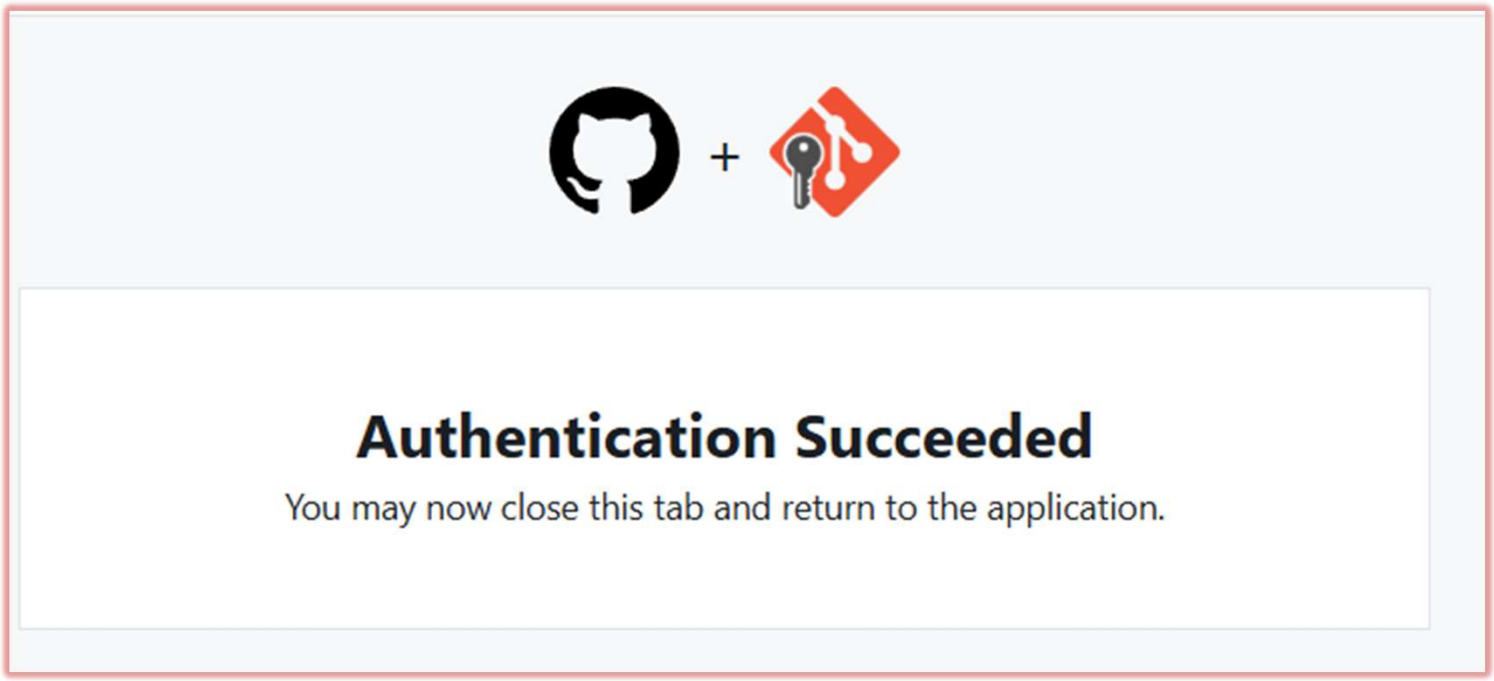
- At the same time, you will see a pop-up, as below. Click on the pop-up icon to sign in to GitHub: *(Note, if you are stuck at this point, it might be because you are signed in to a different GitHub account on your machine – see the trouble shooting section*).

4/ Clone the project from GitHub (2/3)

- When you click on the little icon from the previous page, you will see this Window. Here you can sign in to GitHub.

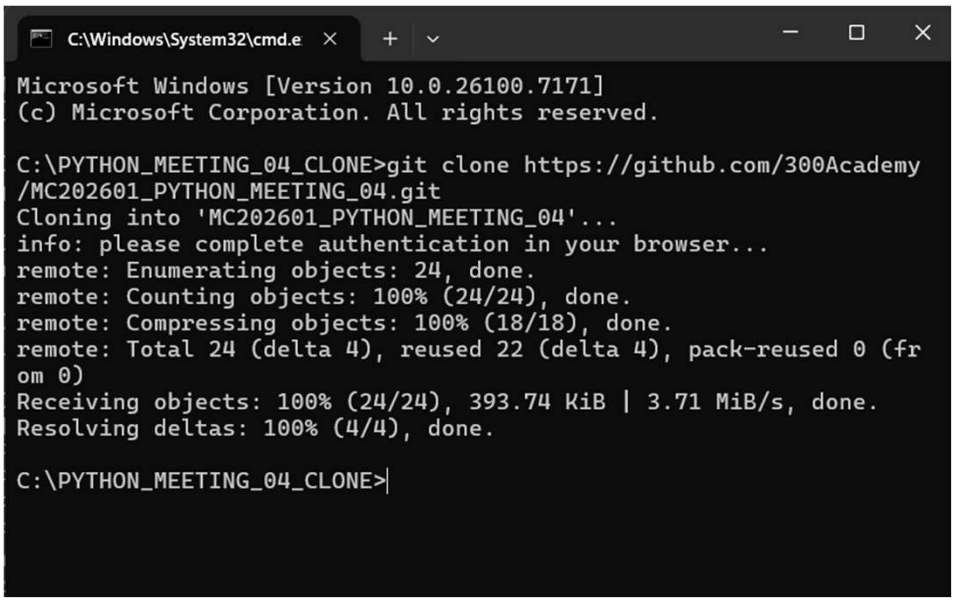


- If the sign in is successful, then you will see this window:

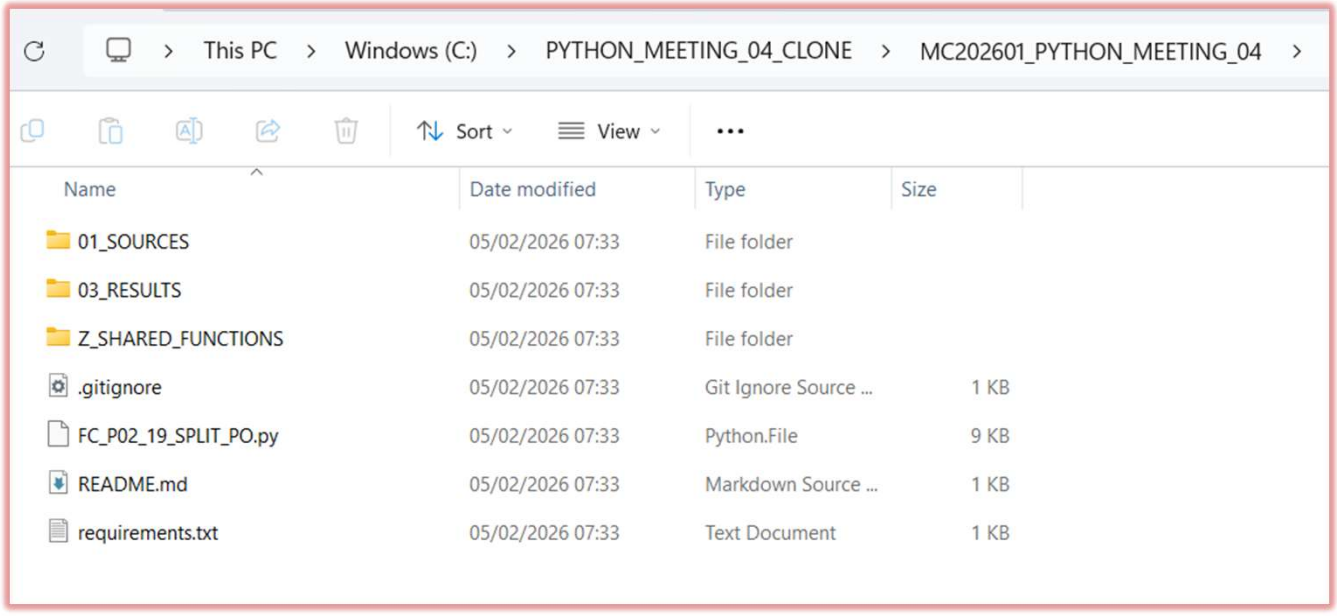


4/ Clone the project from GitHub (3/3)

- Once you have signed-in successfully to GitHub, the folder will be downloaded to the location that you chose before:



- You should then be able to see the folder structure and content in Windows:



5/ Run the code that you downloaded

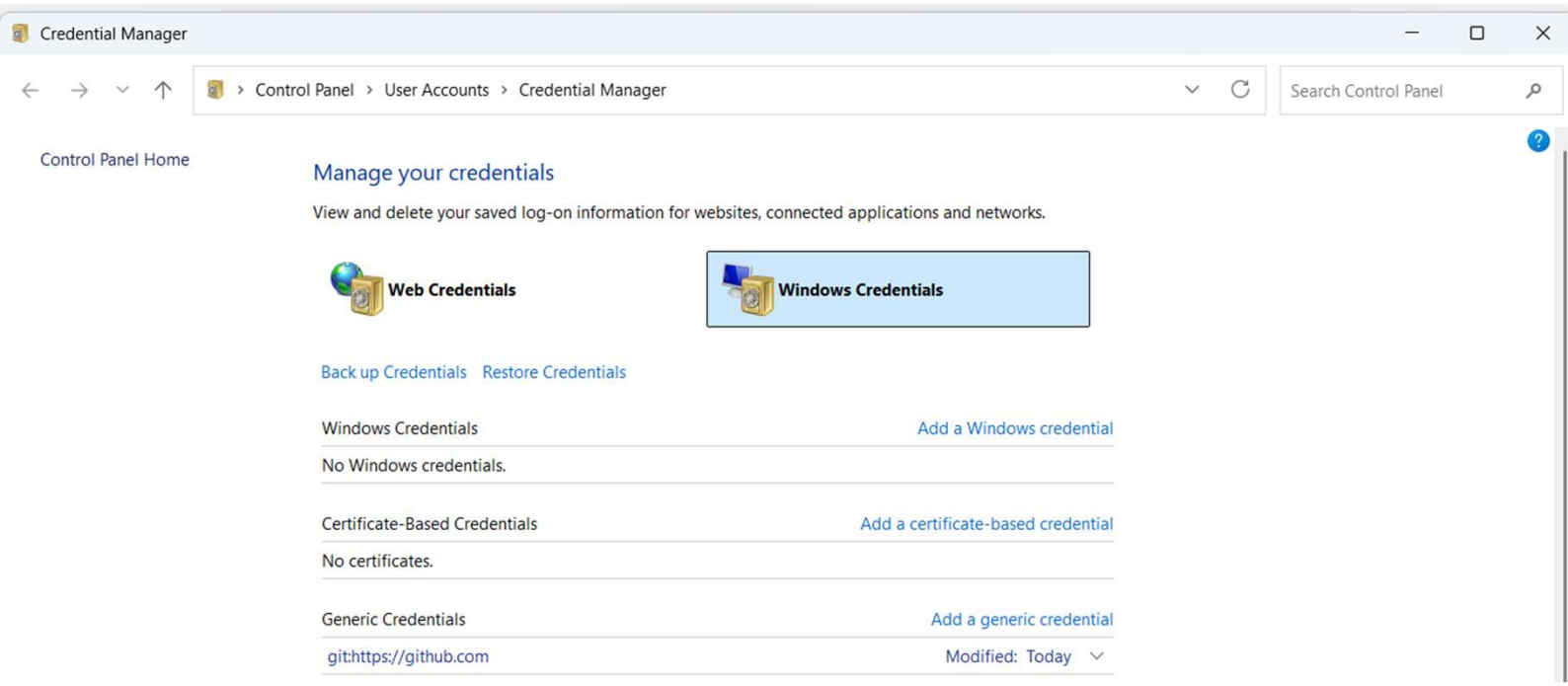
- When the project is downloaded from GitHub:
    - Go to the 02_PROGRAMMES folder in Windows
    - Type **CMD** in the Windows search bar
    - Type **code .** in the black box
    - Create venv: **python -m venv venv**
    - Active venv: **.\venv\Scripts\activate**
    - Install libraries: **pip install -r requirements.txt**
    - Update the variables

```
# 2/ Variables
ZV_ST_SOURCES_FOLDER = 'D:/03_DEMO/300ACADEMY_PYTHONMEETING_2026/01_SOURCES/' # fill in the folder where you put the A_AM_EXCLUSION_WORDS and your supplier data
ZV_04_C_PHASE_FOLDER = 'D:/03_DEMO/300ACADEMY_PYTHONMEETING_2026/03_RESULTS/' # fill in the folder where you want to put the results
ZV_ST_NAME_EKPO_FILE = '20251025_180621_1_EKPO.csv' # fill in the name of your EKPO file
ZV_ST_NAME_EKKO_FILE = '20240510_152615_1_EKKO.csv' # fill in the name of your EKKO file
ZV_ST_NAME_JSON_FILE = 'A_APPROVAL_LIMITS.json' # fill in the name of your JSON file
ZV_NU_WINDOWDAYS = 7
```

- Run the program

6/ Trouble-shooting cloning from gitHub

- If you are not able to clone the GitHub library, it could be that you already have a GitHub account configured for your machine. In this case, you may need to remove that GitHub account (but check with your IT department first).
- To remove the GitHub account, you can type Credential Manager.
- You will then see the Credential Manager box pop-up:



- You could try removing the github account from Web credentials and starting again the process from step 1.

The following steps are optional

- You do not need to do the following steps for the SAP Data Audit Master Class.
- The following steps are optional, in case you want to create your own GitHub repository.

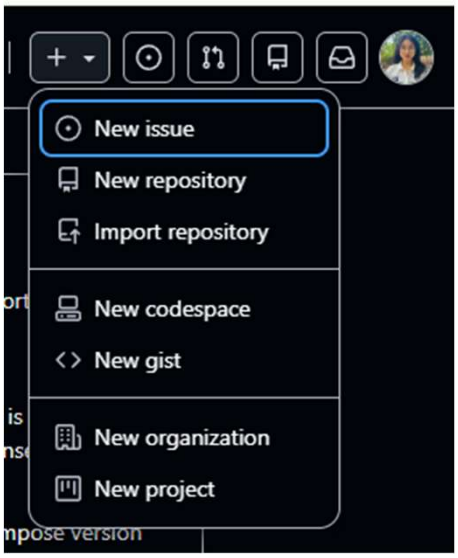
- Note: never store any keys or passwords in code that is uploaded to GitHub. If you store keys, such as ChatGPT keys or the like, then it is highly likely that your code will be scrapped (even if it is private) and your keys compromised. This situation can cause you unexpected bills for the use of services, such as ChatGPT, Azure or AWS, etc.

- For this reason, it is important that only authorized team members be allowed to upload code to your GitHub.

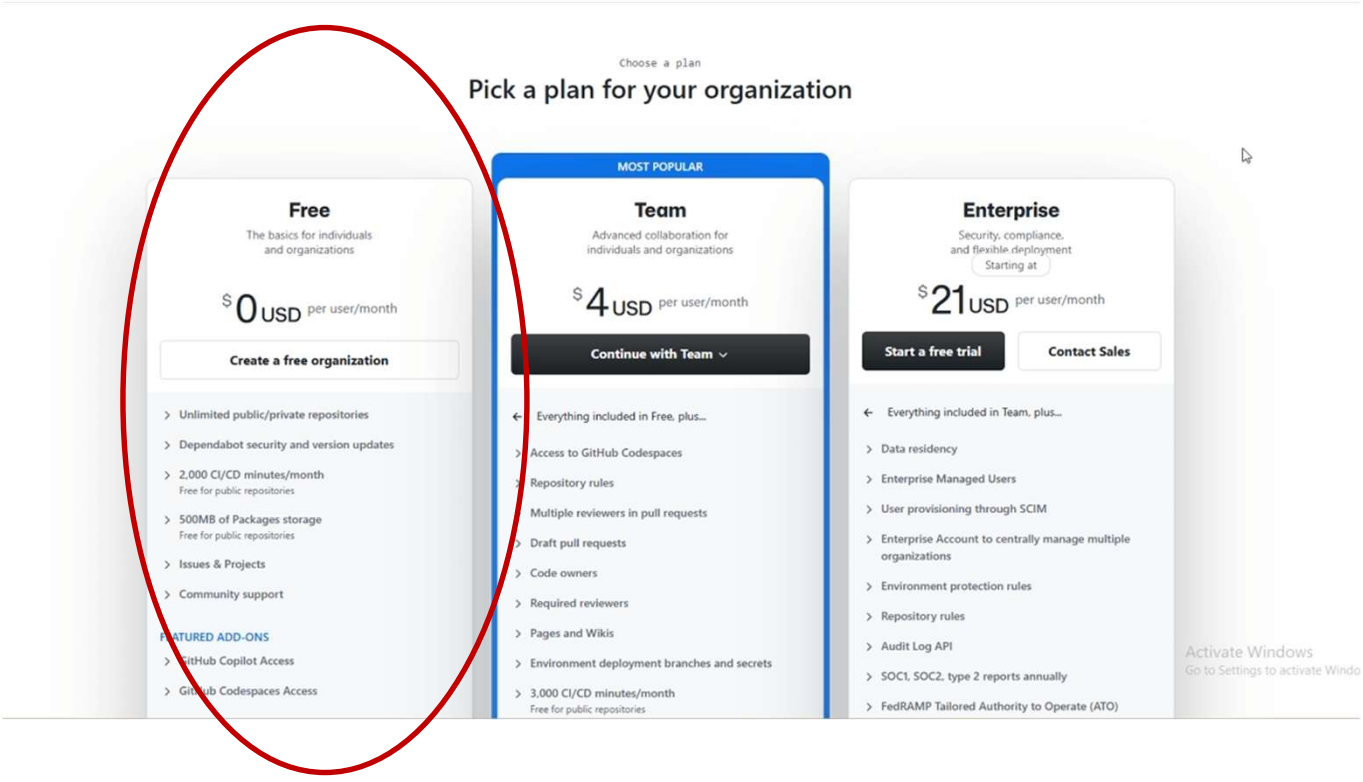
7/ Create your own organization in gitHub

- If you want to be able to store your projects in your own GitHub, or share them with your colleagues, then you could create your own organization

- Click https://github.com/ to login
- Click on button '+' → choose New organization.

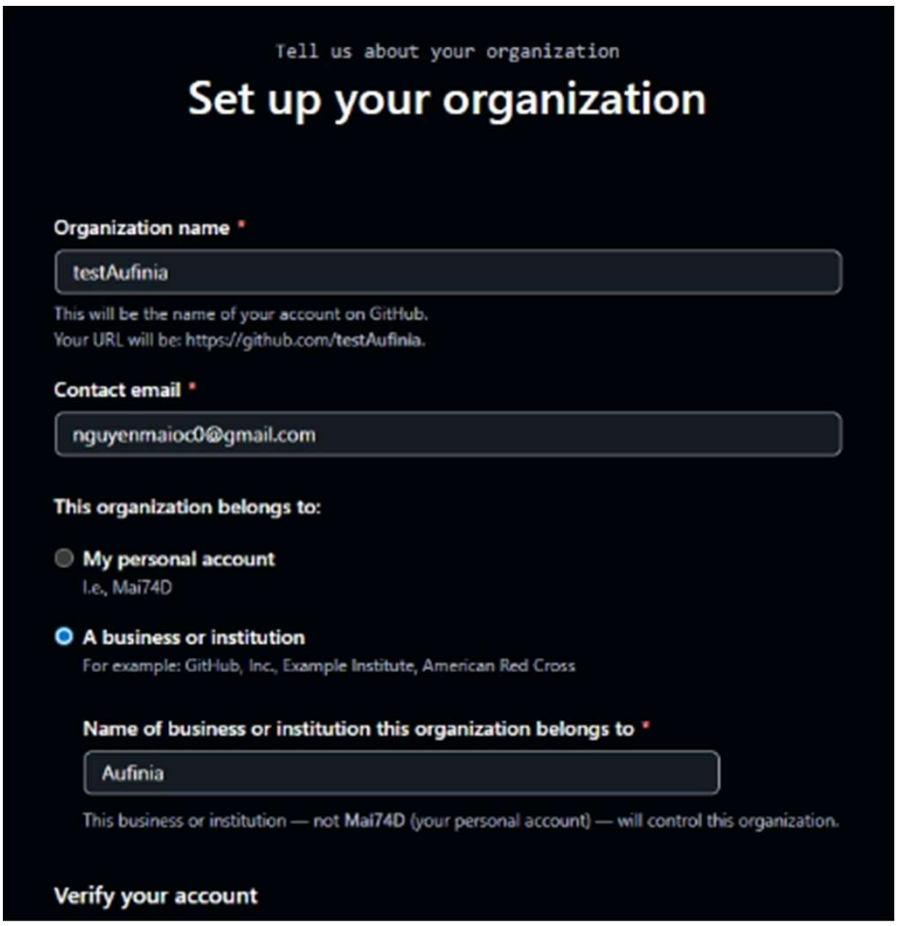


- Choose a plan for your organization: (you can probably use the free plan for small projects):

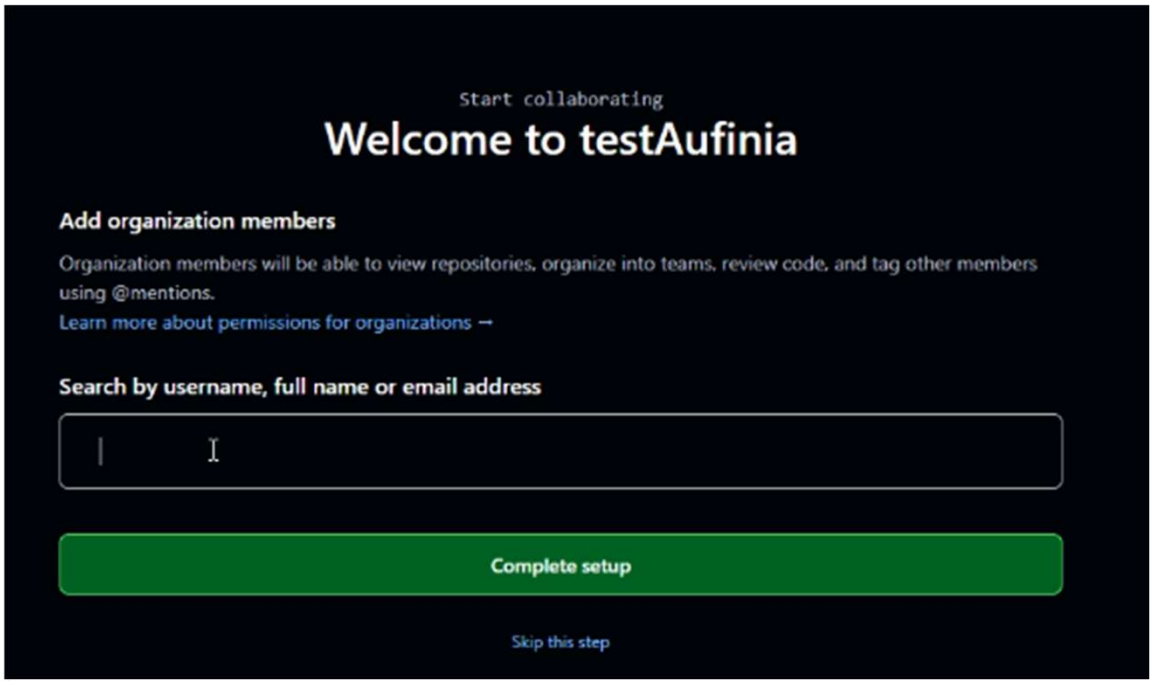


8/ Add name and team members to your organization

- Add information about your organization

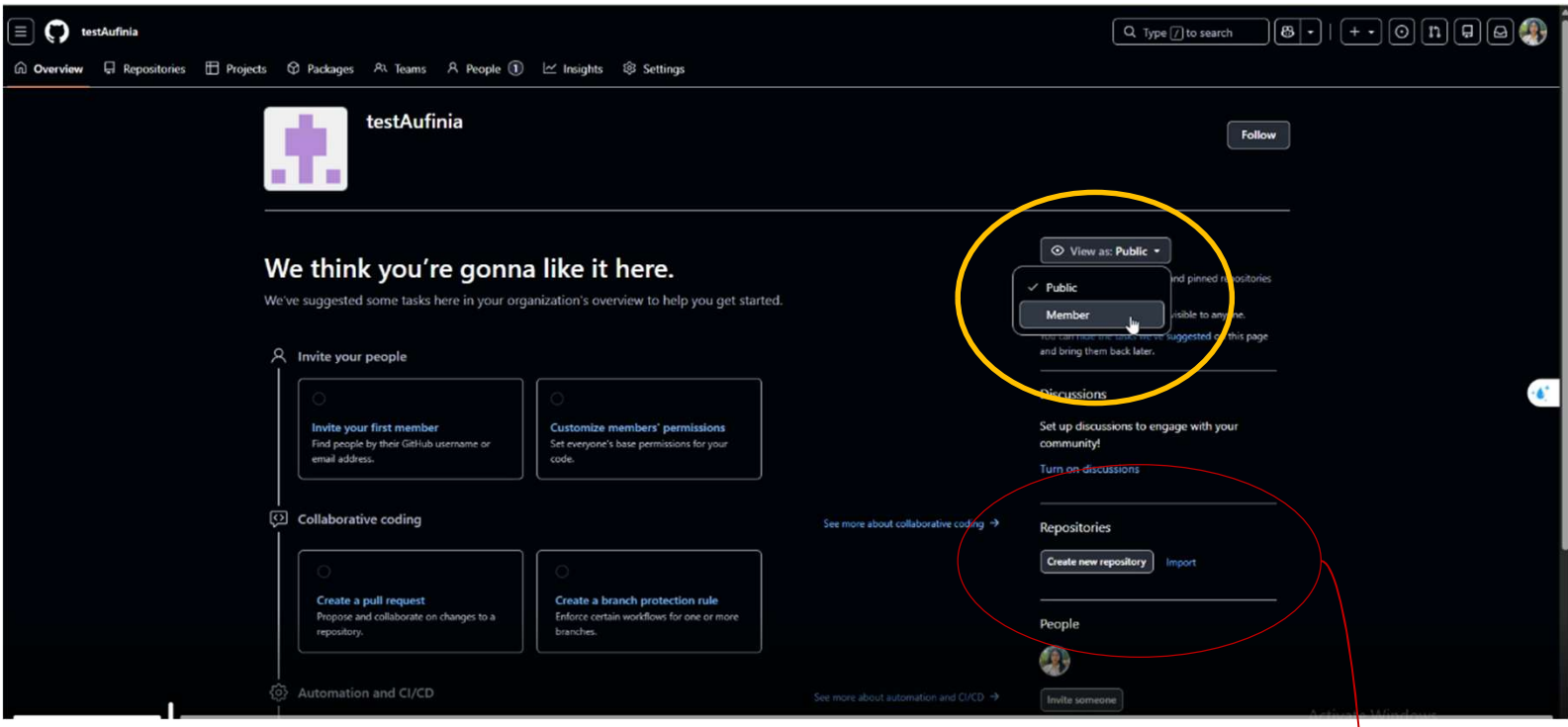


- Add members (optional)



9/ Create a repository in your organization (1/3)

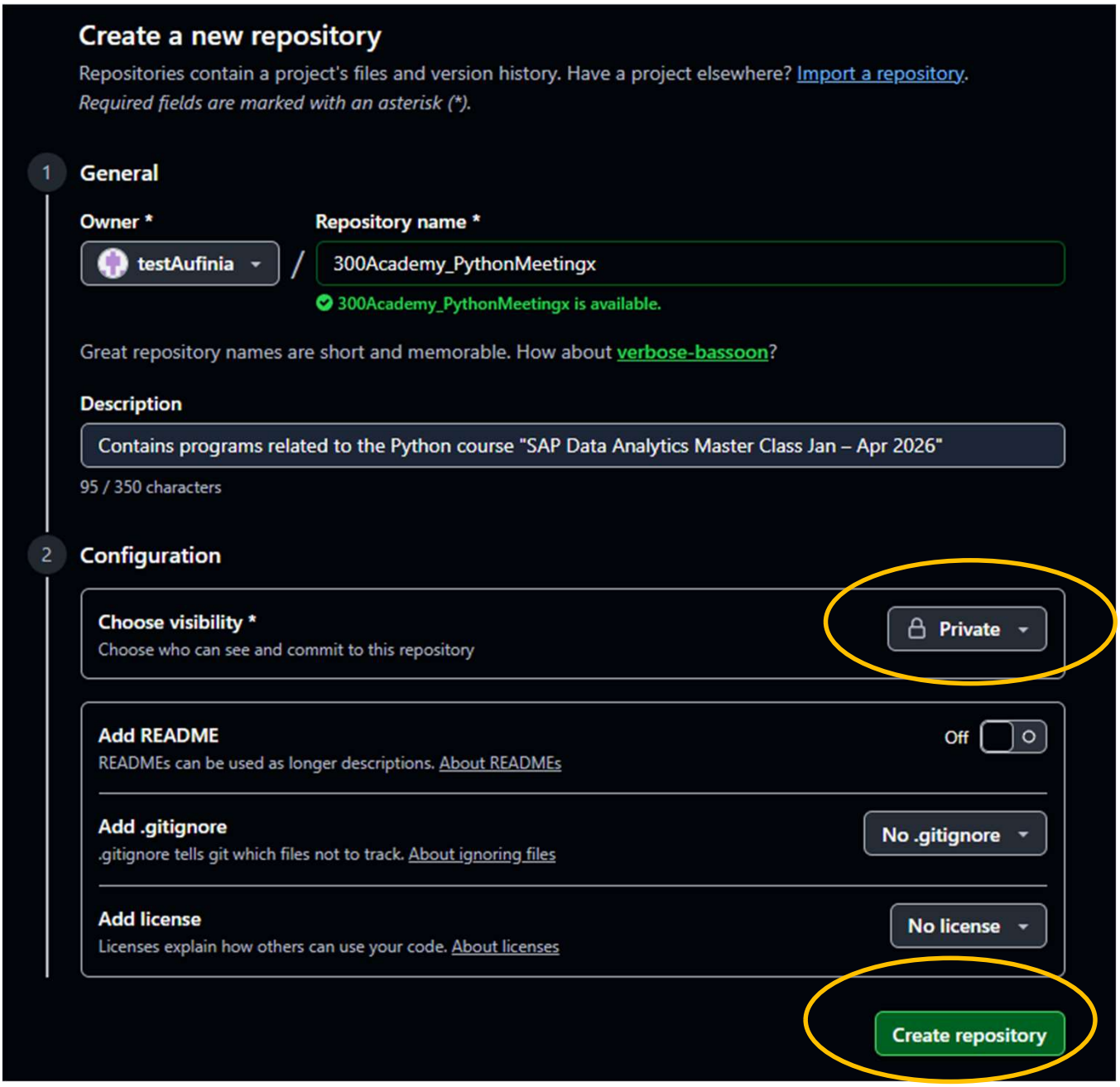
- You can view your organization as the public will see it or as members will see it. Note: this setting is only for the organization page, it does not impact the security settings of the repositories.



- To add a project (for example a python meeting project), click on **Create new repository**

9/ Create a repository in your organization (2/3)

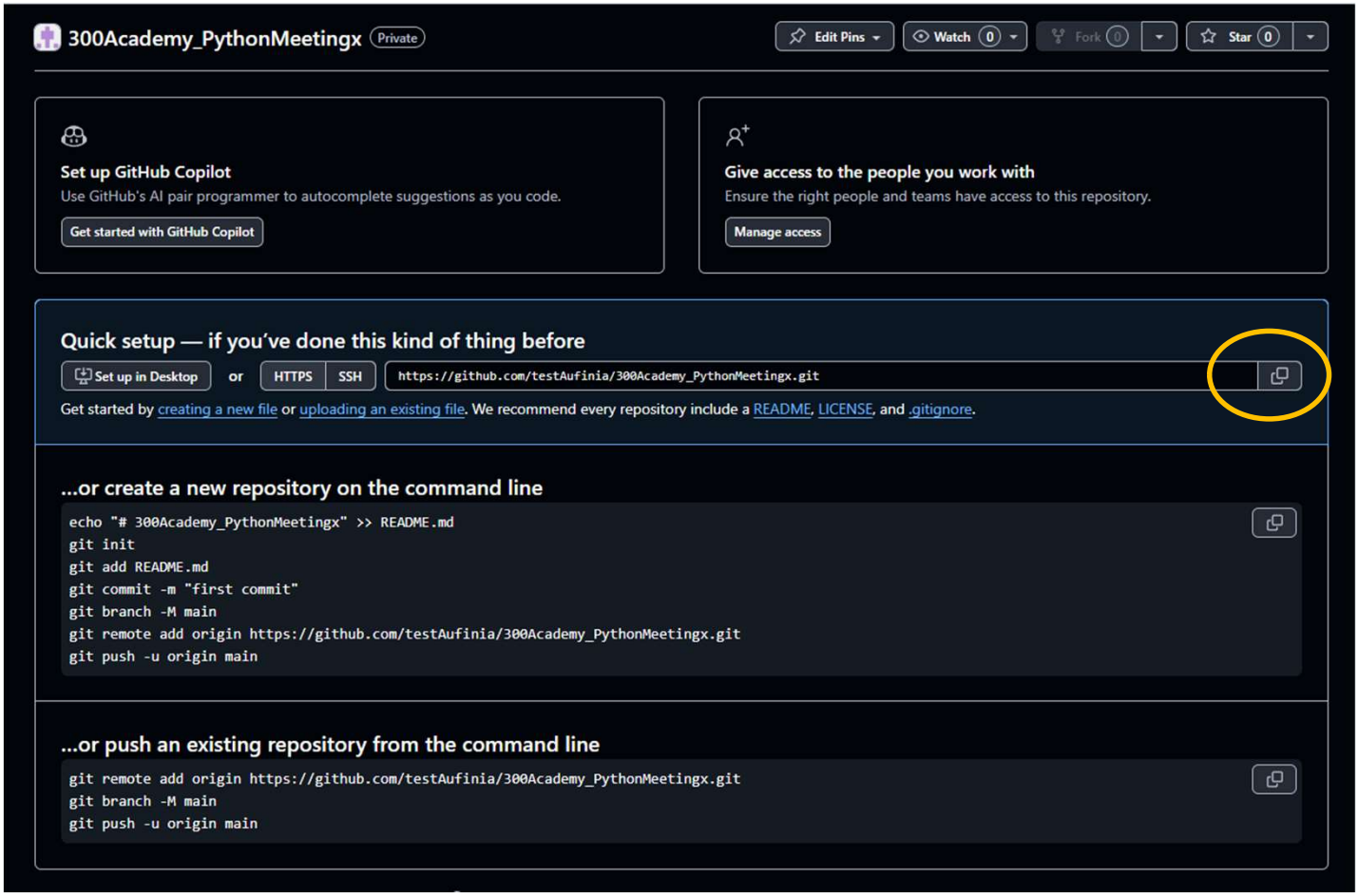
- Fill in the repository name and set to private if you only want your repository members to see this repository.



9/ Create a repository in your organization (3/3)

- Once you have finished creating the repository you will see the below screen.

- Click to copy the link for the next step.



10/ Create a token

- Before being able to push to the GitHub, you need to generate a token..
- Login to your GitHub account and then got to: https://github.com/settings/tokens
- Click generate new token
- Choose Classic
- Give your token a name in Note
- Choose the token expiration (for example 90 days)
- Tick the scope: repo
- Click Generate token
- **Copy the token and remember it** – when you do a push following the creation of the token, you will see a pop-up box, in which you can add a token.

Erase expired token – so that you are requested new one at next push

Action	Type the following in t
git config --get credential.helper	Find out if your token is manager or manager-core. If so continue with the following steps to erase the token
@" protocol=https host=github.com "@ | git credential-manager erase	

11/ Clone your local project to your gitHub repository

- Assuming that you have installed the git application on your machine (as mentioned in steps 2 and 3, we can now use git to copy your project to your new gitHub repository.
- Assuming that you already have Visual Studio Code on your machine, go to the root Windows folder of the project that you would like to push to GitHub.
- In the Windows address bar, click and then type CMD to open a command dialogue box at that location. Then type code . In order to open Visual Studio Code at the same location.
- In Visual Studio Code, open the terminal.
- In the terminal of Visual Studio Code, type the following commands to clone your project to your new repository. Note: we mention <RepositoryLink>. Where you see <RepositoryLink>, replace this text with the link of your repository that you obtained in step 9:

Commands to type in Visual Studio Code terminal:

Action	Type the following in t
git init	Creates a hidden git folder inside your folder. If you want to see this hidden folder, go to the folder location in Windows and select View-> Show -> Hidden items.
@" *.pptx *.pdf *.log .env __pycache__/ venv/ *.pyc .vscode/ .idea/ dist/ build/ *.sav "@ | Set-Content .gitignore	Tells git that these folders will not be copied to GitHub. For example, typically any keys will be stored in .env files. These files you would not want to put in git. Furthermore, the libraries that you downloaded to venv are heavy and not worth copying. __pycache__ is the cache of python and is heavy and not worth copying either. We might not want to include the PPTX in the GitHub or even the PDF. Use code .gitignore to check if you correctly created the ,gitignore file with the correct contents.
git add .	This stages all content (except for the content to be ignored). When using git, content is staged, before it is committed (copied to GitHub). Instead of the . You can also stage individual files. git restore --staged . To un-stage everything.
git commit -m "Initial commit“	To add a message to the commit.
git remote add origin <RepositoryLink>	To tell git where the staged project should go
git branch -M main	To create a main branch in the repository. -M to force, even if branch already exists, or has a different name Ensure that the - is a minus dash (hyphen)
git push -u origin main	To finally send the staged project (origin) to your gitHub repository (main). The -u here adds tracking. This means that next time you can just write git push (instead of git push origin -u main). Ensure the -is a minus (hyphen) not a dash. A window will pop up to request authentication. Choose Token and enter the token that you created in step 10. If this is the first time with your new token, choose token in pop-up and add token

Any questions or feedback:

contact@aufinia.com