1. **Video 85: Module Introduction**

* Collaboration and Contribution
* Module content:

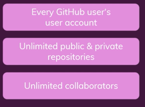


1. **Video 86: The 4 GitHub Usecases**

* Single User uses GitHub for:
  + Cloud storage
  + Portfolio Page
  + To add other people on your repository to collaborate with you
  + To contribute to other open source projects

1. **Video 87: Understanding GitHub Account Types**

* **Types of Accounts:**
  + **Personal User Account:**
    - Free Plan is more suitable for this account :For starters, or people using it on a small to medium scale.



* + **Organisational User Account:**
    - Free/Team (Paid Plan)/Enterprise (paid Plan) plans are suitable for this account depending on the size of the organization and the number of features required.



* + **Enterprise User Account:**
    - Enterprise Plan (Paid Plan) is more suitable for this account.



1. **Video 88: Changing the Repository Type from Public to Private**

* In your GitHub account go to **Settings**, click on **Organisations. I here you can:**
  + Create a new organization
  + Transform your account from Personal to Private/Organisational and vice versa:
    - You can check/access Repositories
      * Once in the specific repository, click on **Settings**
      * Scroll down to **Danger Zone,** under this you can **Change repository visibility**
        + Click on **Change visibility**
        + Select the **Make Private** option
        + Confirm the repository
        + Then click on **I understand…..**
        + Once done, page will not be available to others!
    - Accessing repository using URL code without having access to **Settings tab**:
      * Copy the URL of your GitHub account
      * Open a new browser
      * Paste the URL of your GitHub account to access your page

1. **Video 89: Pushing Commits to a Public Repository**

* Create a folder in VSC
* Clone the repository using **git clone [repository code/URL]**
* **cd [folder name]:** Navigate to selected folder
* Add a file, and commit the file **: git add . , git commit -m “commit message”**
* **Git push:** to push your changes to GitHub
* To get access to a repository**:** Go to **Setting,** click on **Developer Setting** to get a token

1. **Video 90: How GitHub Manages Account Security**

* **Security and Access**



1. **Video 91: Understanding & Adding a Collaborator to a Private User Account**

* **Option one:**
  + To give access to a repository**:** Go to **Setting,**
  + Click on **Manage Access,**
  + Click on **Invite a collaborator** and add an email address or account name,
  + The collaborator will receive and email and can accept the invite,
  + Now the collaborator will have push access to the repository.
* **Option two:**
  + To get access to a repository**:** Go to **Setting,** click on **Developer Setting** to get a token
  + Click on **Personal access tokens**
  + Click on **Generate new token**
  + And select rights by ticking specific ones you would like to give
  + Click on **Generate token**
  + Copy the token
  + Got to VSC and insert the code on the pop-up screen to sign in
  + Now your pushes will be successful

1. **Video 92: Collaborating in Private Repositories**

* Go to **settings**
* Scroll down to **Danger zone**
* Change visibility from **public to private**
* Go through option 1 or 2 in video 91

1. **Video 93: Comparing Owner & Collaborator Rights**

* GitHub docs: **GitHub / User accounts / User account settings / Permission user repositories**
  + Read through section: **Owner access for a repository owned by a user account**
  + Read through section: **Collaborator access for a repository owned by a user account**

1. **Video 94: Limiting Interactions**

* Go to your **profile**
* Click on **Settings**
* Click on **Moderation settings: In here you can**
  + **Block User**
  + Set **Interaction limits**
    - Set **temporary interaction limits**
    - **NB: account limits always override repository limits**
    - **NB: Limits cannot be applied to 1 user (not specific)**

1. **Video 95: Introducing Organizations**

* GitHub Security and Access



1. **Video 96: Creating an Organization**

* Go to profile
* Go to **Settings**
* Click on **Organisations**
* You can also see other organization accounts that you have created before
* Click on **New organization**
* Select a plan: Free plan
* Give the organization a name
* Enter a contact email
* Choose between: Personal or business
* Click next
* Add organization names
* Then complete setup
* Click submit
* To access the organization, click on the GitHub logo and select the organization account

1. **Video 97: Exploring Member Repository Permissions**

* Under **organization account**, click on **New** to create a Private repository
* **To add members:**
  + **Go to settings**
  + **Click on Manage Access:**
    - **Base Role:** Click on manage to manage access roles: You can manage the following:
      * Base permission, Repository Creation, Repository Forking, Page Creation, Admin Repository Permissions, Member Team Permissions

1. **Video 98: Adding Outside Collaborators**

* Under **organization account**, Click on the repository
* Click on **Settings**
* Click **Manage Access**
* Click on Invite teams or people: You can invite a member or an organisation
* Choose a role: Read/write etc.
* And add the member
* To see added members Go to settings and they will be under the **Direct Access** tab
* To clone the repository copy the organisational URL, In VSC use the git command: **git clone [org. URL] .** (space and full stop at the end)
* Ensure the access writes allow the member to commit and push changes (Roles: Write)
* Stage the commit: **git add .**
* Commit the change: **git commit -m “commit message”**
* **git push**: To push /update remote repository

1. **Video 99: Adding Organization Members**

* Under the Organisational account, go to your repository
* Click on **People**
* Click on **Invite member**
* Search for a member and click on **invite**
* Select role: **Member/Owner**
* Then click on send invitation
* To which access rights of a **Member**, click on members and manage access (here you will have the option to edit)

1. **Video 100: Failing to Manage Access for Individual Repositories**

* Under your repository, click on People
* Then click on member privileges
* Under Base permissions select the read/write permission
* The problem with this option is that it changes permission to all the repositories that a member has access to. This option cannot change access rights for individual repositories. Problem can be solved when creating teams.

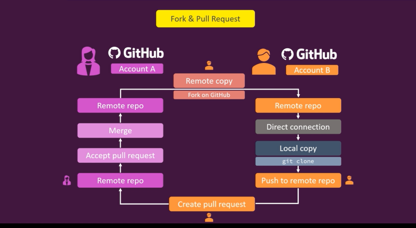
1. **Video 101: Introducing Teams**

* Allocating different access rights for 1 member with access to multiple repositories:
  + In your Organisational account, click on **Teams**
  + This will allow for flexible repository access
  + Click on **new team**
  + Create a team name
  + Add description
  + Define Team visibility
  + Click on **Create Team**
  + Click on **Repository** and allocate it to the team
  + On the right you can change permission level for this repository (Read/Write)
  + Click on Members and add a member
  + Assign a repository to the desired team (read only or write only access team)

1. **Video 102: Managing Team Repository Access Efficiently**

* In your Organisational account, click on People
* Under the specific member click on the setting icon on the right
* Click on manage access
* Click edit
* And set as desired: Read/write/no permission
* Be aware of individual and global settings of access right as the often clash.

1. **Video 103: Understanding Forks & Pull Requests**



1. **Video 104: Forking a Repository**

* In your personal user account, go to any Public repository
* Click on Fork
* The forked repository will now be part of your list of repositories
* Next, you can clone the repository and make changes
* However, your changes will not appear in the original repository in GitHub
* Owners of repositories can see when their repositories have been forked

1. **Video 105: Pull Requests in Practice**

* Click on pull request
* Select the **base repository** and the corresponding branch, and select the **head** repository and the branch to **compare** with
* Add a comment
* Click on **Create** Pull Request
* **The owners actions:**
  + Click on Pull Request
  + Open the pull request
  + Review the changes
  + Leave a comment
  + Click on Merge Pull Request / Close Pull Request
  + **NB:** Closed pull requests can be re-opened by the owner
  + If merge was confirmed, the changes made by the collaborator from the forked repository will now appear in the owners original repository.
  + **NB:** Merged commits can be reverted.

1. **Video 106: Opening & Closing Issues**

* Go into a repository
* Click on Issue tab
* Click on New Issue
* Add a title
* Add a comment (what is wrong and how it could be fixed)
* Assign a team member
* Add a label
* Click on submit new issue
* Owners Actions:
  + In the repository, click on Issues
  + Review and add a comment
  + Click on close with comment

1. **Video 107: Working with GitHub Projects**

* In a repository, click on Projects tab
* Click on New Project
* Add a Project Board Name
* Add a description
* Select a template
* Click on Create Project
* Add cards under TO DO, In Progress, and Done boards
* Add an issue by clicking Issue tab
* Link the issue to the created project
* Created issue card can be dragged to any boards
* Manage automation under cards/boards: Move issues here when………., places issues automatically as per settings selected.

1. **Video 108: Creating a README File in a Repository**

* In a repository at the bottom, click on add Readme
* Communicates important info on project, what its all about, manage contribution through core rules
* Md. stands for markdown, lightweight markup language that you can use to add formatting elements to plaintext text documents.
* Add info . using md.
* Commit the file

1. **Video 109: Presenting Yourself as Developer on GitHub**

* Updating your landing page to look attractive.

1. **Video 110: About GitHub Stars**

* Stars are likes and indicate the popularity of a project
* To add a star, open a repository and click on Star tab

1. **Video 111: Module Summary**

