**GIT HUB DOCUMENTATION**

**Start:**

**Github** is a web-based platform used for version control. Git simplifies the process of working with other people and makes it easy to collaborate on projects. Team members can **work** on files and easily merge their changes in with the master branch of the project.

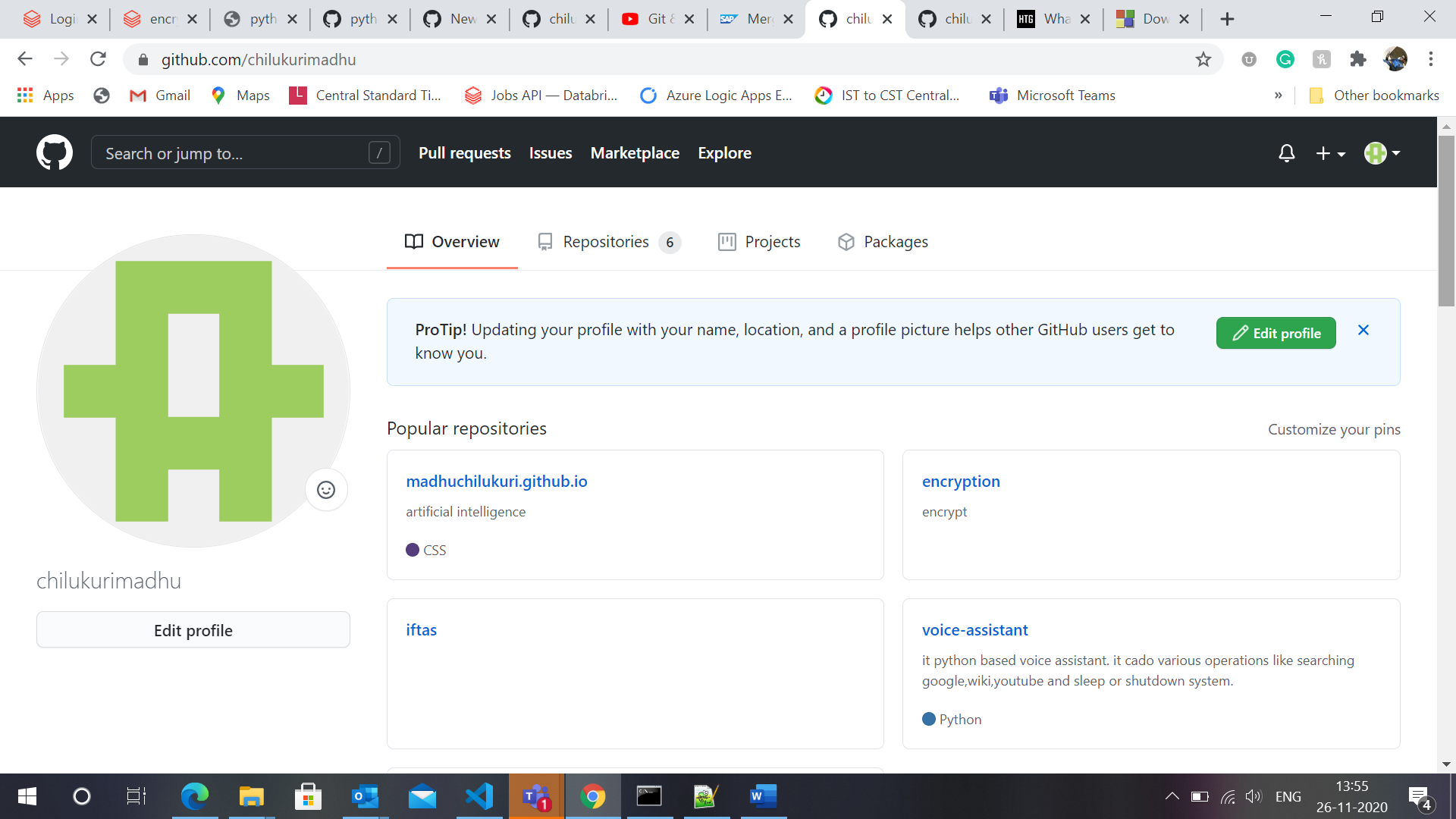
To understand GitHub, you must first have an understanding of Git. Git is an open-source version control system that was started by Linus Torvalds—the same person who created Linux. Git is similar to other version control systems— [Subversion](https://www.howtogeek.com/66731/version-tracking-with-subversion-svn-for-beginners/), CVS, and Mercurial to name a few.

So, Git is a version control system, but what does that mean? When developers create something (an app, for example), they make constant changes to the code, releasing new versions up to and after the first official (non-beta) release.

Version control systems keep these revisions straight, storing the modifications in a central repository. This allows developers to easily collaborate, as they can download a new version of the software, make changes, and upload the newest revision. Every developer can see these new changes, download them, and contribute.

Similarly, people who have nothing to do with the development of a project can still download the files and use them. Most Linux users should be familiar with this process, as using Git, Subversion, or some other similar method is pretty common for downloading needed files—especially in preparation for compiling a program from source code.

1.create new repository for ur project.



2.create file .

3.write some data.

4.write some comment suitable for it then click commit new file.

Then new file will be created in master branch.(main project branch.)

IMP:

>while saving save it as master.

If there are multiple people working then create each branch for each person,there they will upload their code there.

at the end if code is finalized they all will push the code to master branch.

5.if u don’t want to disturb the previous version then create new version if it is finalized then only merge to master . while saving save it as branch.

Note:

U can can track previous version of application or code go to commits there u can check ur previous versions.

6.if some person want to dedicate some code to u .

This is in another person git account.

# Search username/project name ex: [chilukurimadhu](https://github.com/chilukurimadhu)/[learning](https://github.com/chilukurimadhu/learning)

# There create file in that person repo then it says fork created .

# Then click propose new file then click on create pull request.

# Then click it same again.

# Then pull request goes to actual person (owner of repo),he will click on pull requests there it will show requested file.

# 

# Then click on file ->click merge pull request.->confirm merge ->merged succefully.

# Note:if another person when he sent pull request to branch it will go to branch from our side it will appear in branch .then again we have to merge it to.

# 

# 

# 

# GIT CMD:

# 1.in your local machine go to folder right click open git bach.

# 2.execute->git init ->will get .git folder

# Now u need clone aster into local

# -> git clone url

# 3.cmd->git remote add origin “git https url”

# 4.git pull origin master(it will pull the code to local machine)

# 5.if u add any file in the folder u should make commit of it, for that first check ->git status

# Then it will show what is status of then.

# When ur adding file or data means u should add it to index like below

# Ex: git add filename.txt

# Then cmd->git commit -m “write some comments here for commit”

# 6.if u want to add multiple files cmd->git add -A

# 7.multple files in single commit cmd->git commit -a -m “commit comment”

# Check cmd->git status now

# 

# 

# 8.now create branch in local repo.

# ->Git branch branchname (here clone of master create into branch)

# ->now move to branch ->git checkout branchname

# Now ur in branch folder

# ->add some files it doesnot reflect master

# When u add files in branch add them to index and commit it.(git add filename.txt,git commit -a -m “comment”)

# ->now go to master ->git checkout master

# If u do -> ls ->it will not show branch files ,here u have to merge that with master like below ->git merge branchname

# ->again go to branch and modify file commit them there and angain come back to master and merge them again.

# 

# 