

What is version control and GIT?

How does it help team working?

Anto Antony George
Shehnaz Katteth Salim

Content Management
MA Web Design and Content Planning

What is Version Control?

Version control allows you to keep track of your work and helps you to easily explore the changes you have made on your files.

Eg: **Our Coding files, Design files**

Our conventional version control be like..

Design_script_25thFeb

Design_script_25thFeb.version_1

Design_script_28thFeb.version_1.2

Design_script_30thFeb.version_1.2.1

Design_script_6thMar.version_1.2.newdesign

Design_script_25thMar.version_1.2.newdesign_final



How version control system helps..

- Generate backups
- Test and experiment
- Keep history and track changes
- Collaborate and contribute (**Team Working**)



git

Founder: **Linux Torvalds**

Initial release: **2005**

What is GIT?

Git is the World's popular **version control system**.

Git is software for tracking changes in any set of files, usually used for **coordinating work** among programmers **collaboratively** developing source code during software development.

Advantages of Git

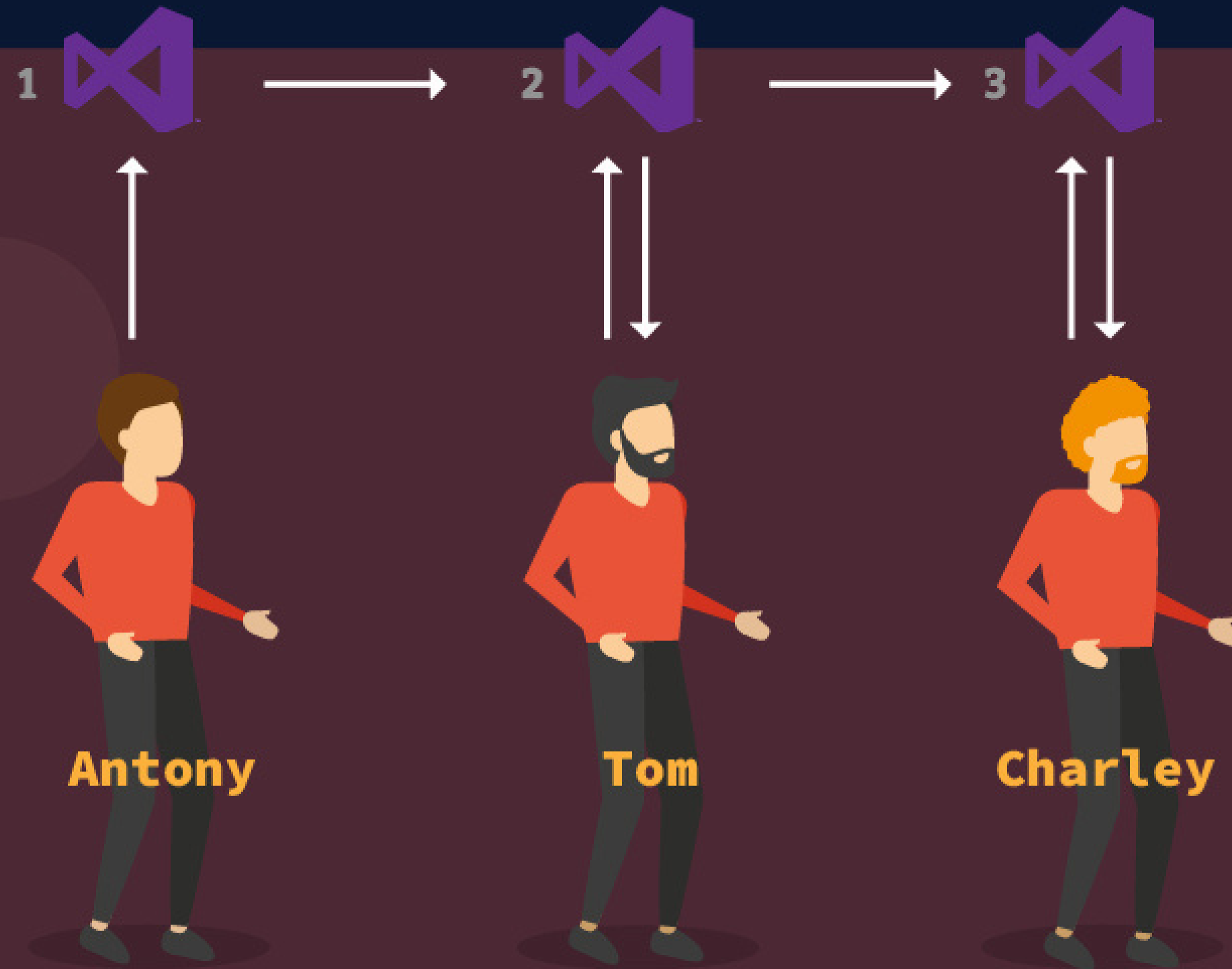
- Open Source
- **Team Working**
- Compatibility
- Reliability
- Speed
- Code Backup
- Branching and Merging

How git is helping for team working.



History
Collaboration
Feature Branches

Example Of Git As A team Work



Example Of Git As A team Work



Writing a Book



Email the copy



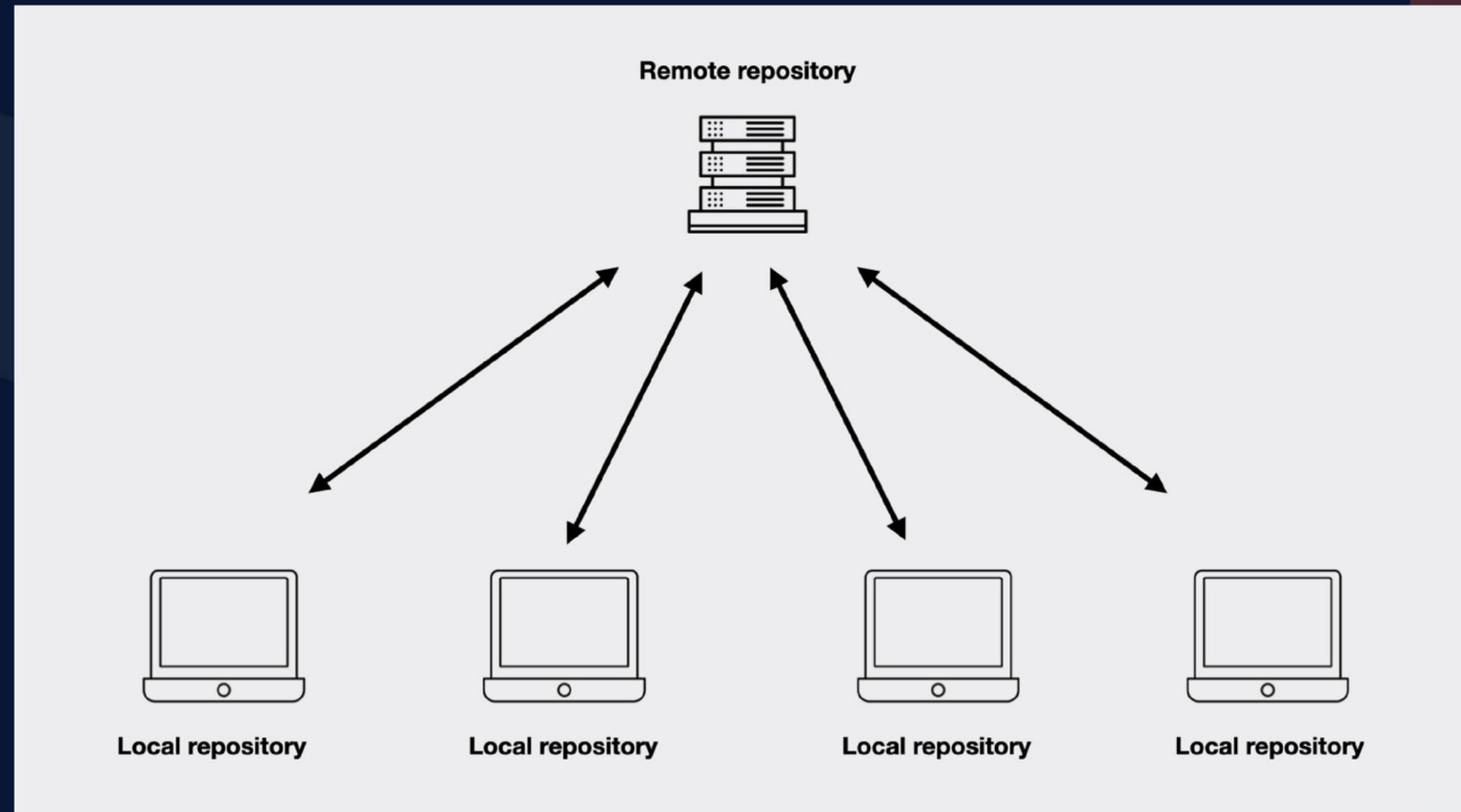
**Stuck, waiting for
friends updates**

Repositories – Local & Remote.

Local repository: It is just a file location residing in your system.

Remote repository: lies somewhere outside your system, on a remote machine. Important when you are working as a **Team**. This is the place where everyone will be sharing their code.

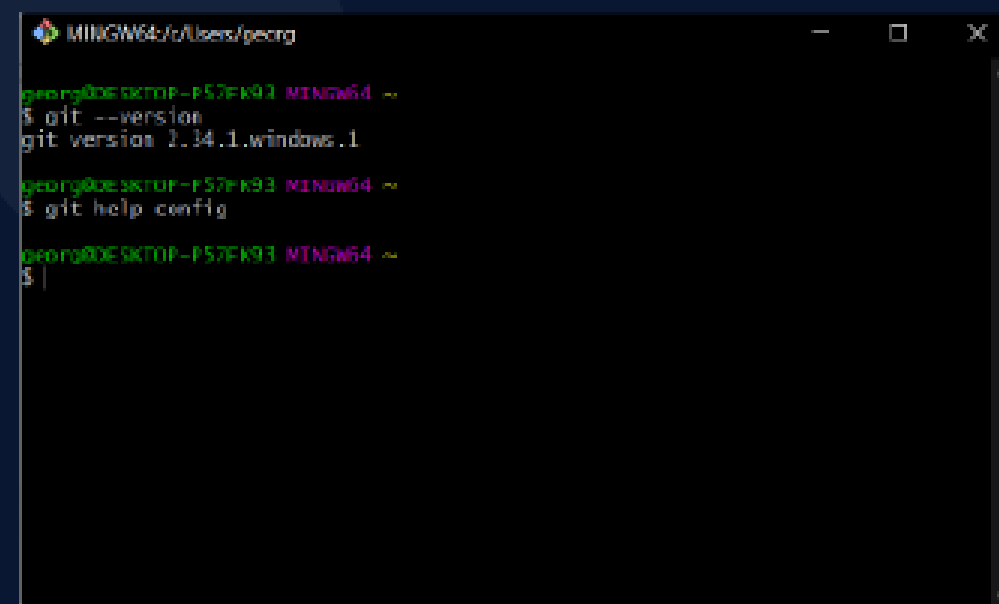
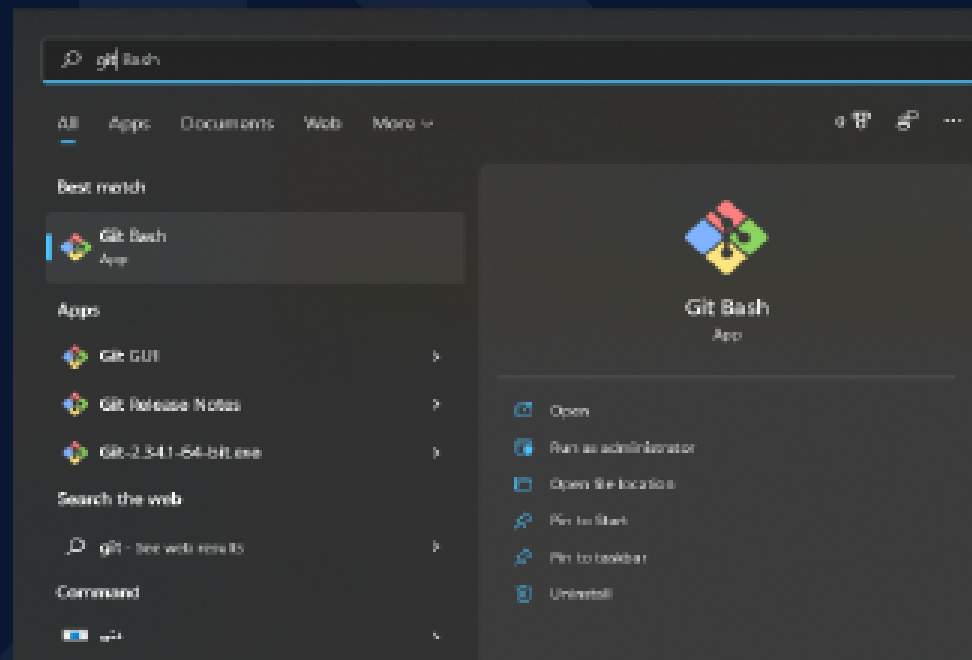
How it works..



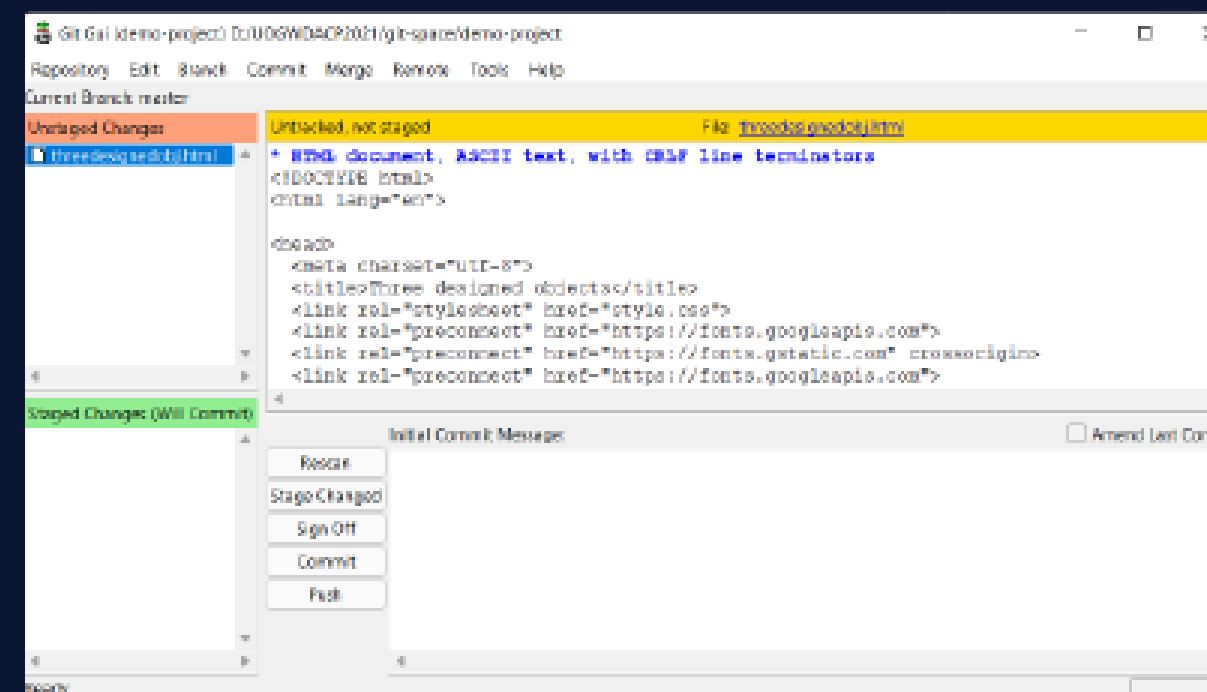
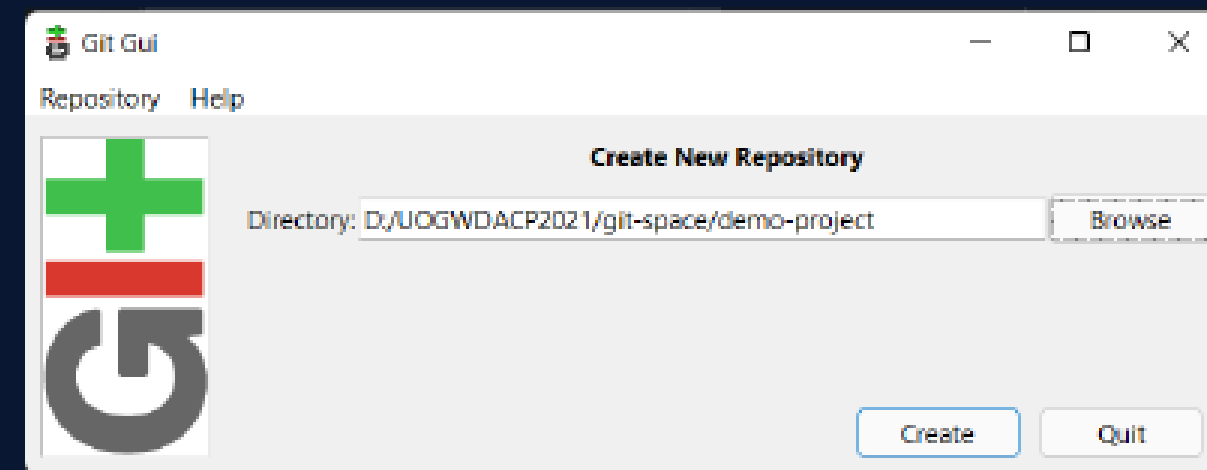
How it works..

Install **git** from <https://git-scm.com/>

Command line Interface



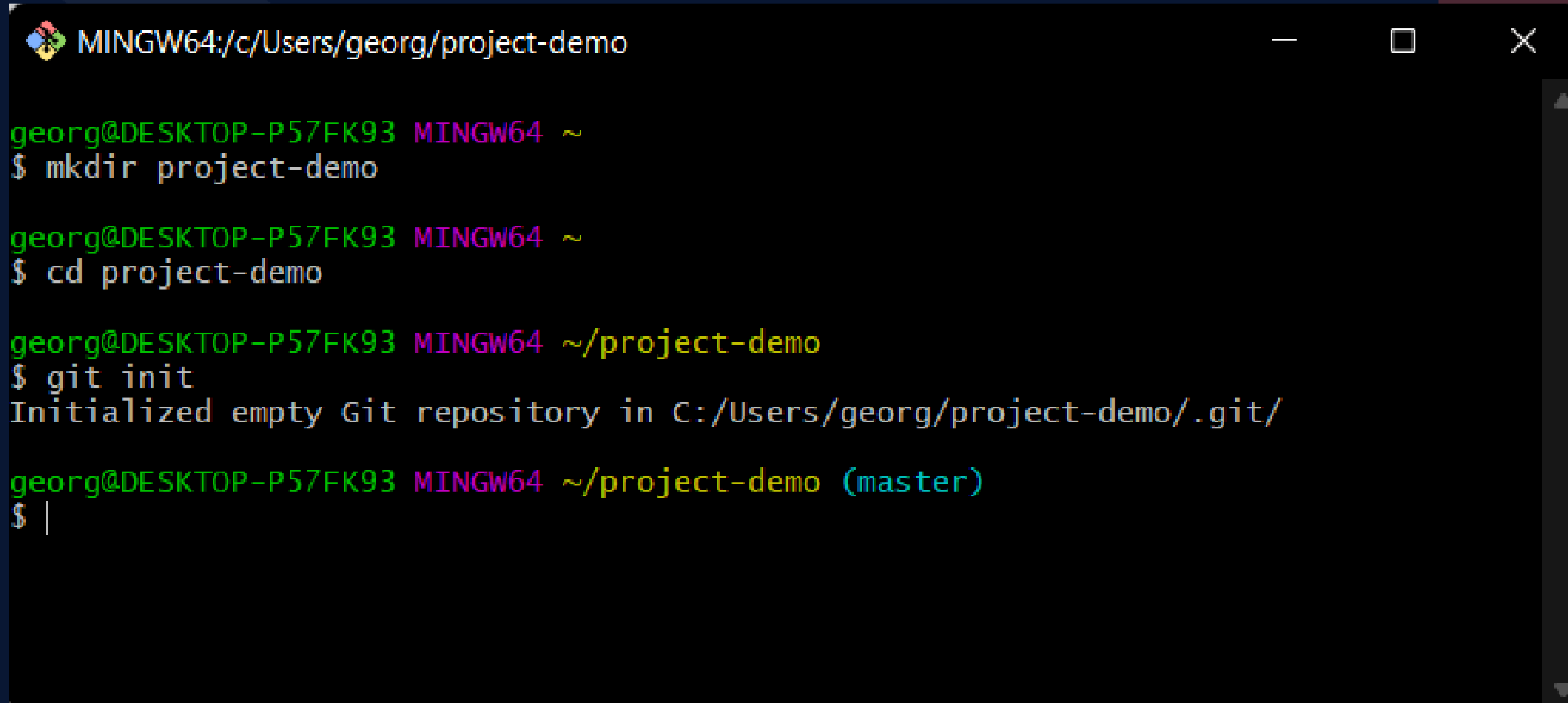
Graphical user Interface



Free Graphical User Interface for team work Eg:

- GitHub Desktop (Win/Mac) – Commonly Used
- SourceTree (Win/Mac)
- Git Extensions (Win/Mac)
- Fugitive (Win/Mac)
- Magit (Win/Mac)
- git g (Win/Mac)

Create a local directory and Initialize the directory:



```
MINGW64:/c/Users/georg/project-demo

georg@DESKTOP-P57FK93 MINGW64 ~
$ mkdir project-demo

georg@DESKTOP-P57FK93 MINGW64 ~
$ cd project-demo

georg@DESKTOP-P57FK93 MINGW64 ~/project-demo
$ git init
Initialized empty Git repository in C:/Users/georg/project-demo/.git/

georg@DESKTOP-P57FK93 MINGW64 ~/project-demo (master)
$ |
```


Add any project related file to the **project folder** which we have created.

```
MINGW64:/c/Users/georg/project-demo

georg@DESKTOP-P57FK93 MINGW64 ~/project-demo (master)
$ git add index.html

georg@DESKTOP-P57FK93 MINGW64 ~/project-demo (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   index.html

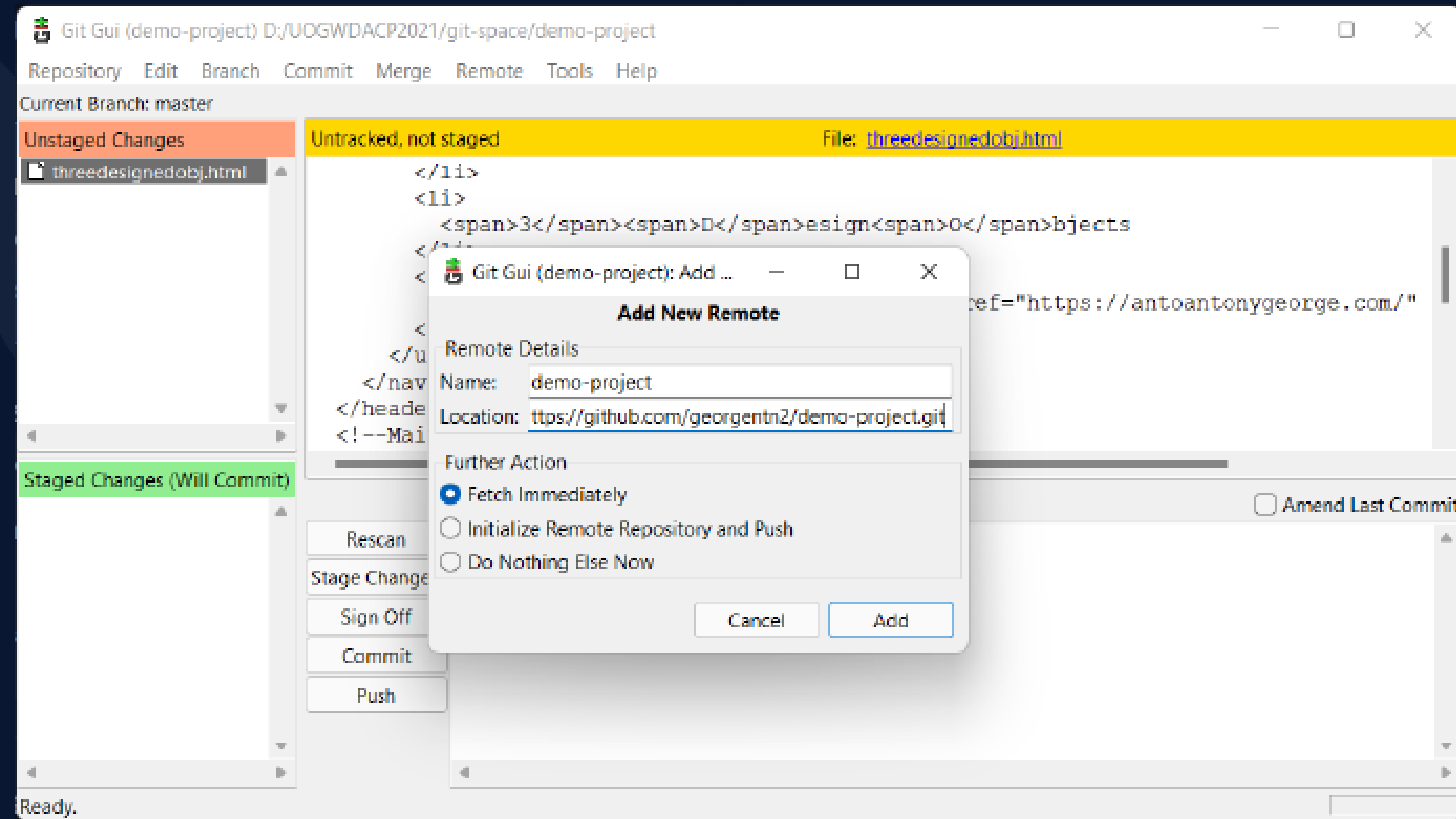
georg@DESKTOP-P57FK93 MINGW64 ~/project-demo (master)
$ git commit -m "Committing the demo index.html file"
[master (root-commit) 554fae6] Committing the demo index.html file
1 file changed, 165 insertions(+)
create mode 100644 index.html

georg@DESKTOP-P57FK93 MINGW64 ~/project-demo (master)
$
```

Checking the **status**

Committing the file
with a note to
local repository

We will explain it with the help of **GUI**



Now we need GitHub

What is GitHub?

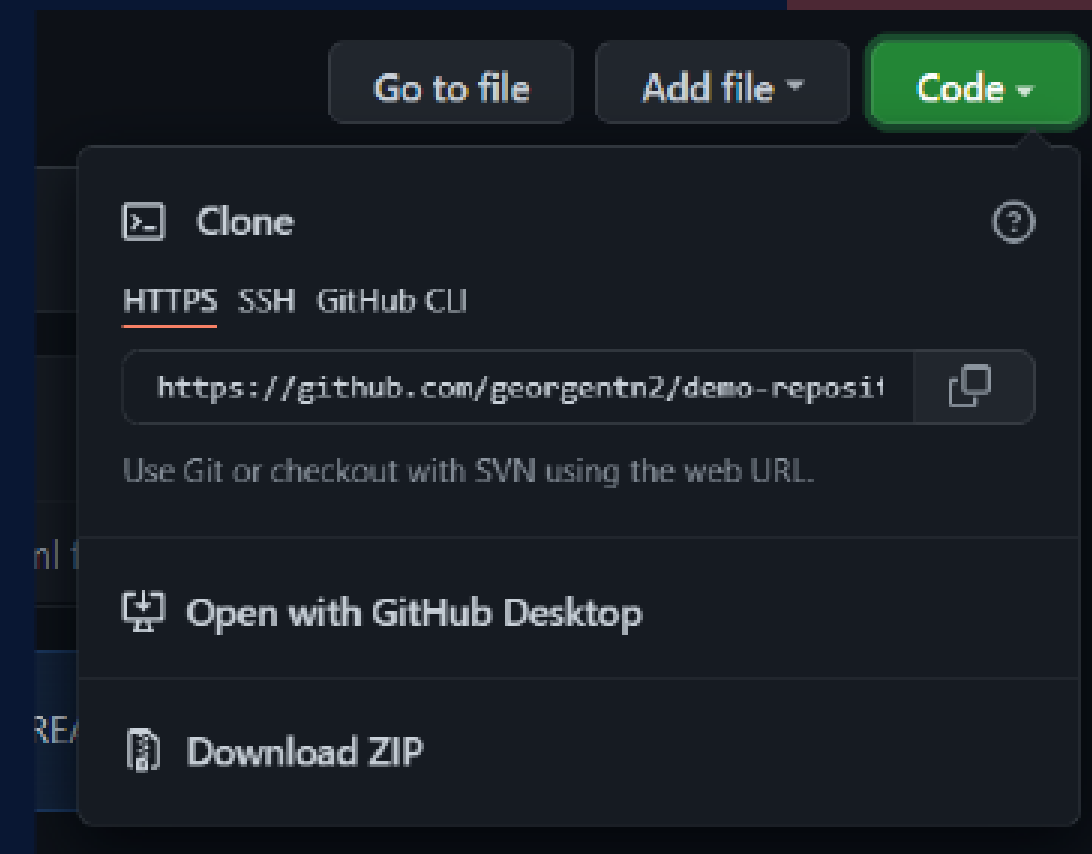


GitHub is a GUI platform that allows to host your **Git projects on a remote server** somewhere (or in other words, in the cloud).

Millions of **developers** and **companies** build, ship, and maintain their software on GitHub—the largest and most advanced development platform in the world.

go to <https://github.com/> to create a new account

create a **new repository** in
github



Copy the newly created **repository url**

He or she could

Create branches— Creating a new copy of master code as branch and work on it

Pull— retrieving latest changes from repo

Push— Updating the new changes made

merge— Merging multiple branches together or to master branch

Now one of the team member wants
our code in his local machine

he/she can use **git clone** <repository link>
command on CLI in his PC where **git** is installed

```
Select C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.493]
(c) Microsoft Corporation. All rights reserved.

D:\UOGMDACP2021\file>git clone https://github.com/georgentn2/demo-project.git
Cloning into 'demo-project'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

D:\UOGMDACP2021\file>
```

thank you.