



# INTRODUCTION TO GIT & GITHUB

READY TO LEARN ?

GIT





01

# What will we be learning today?

- What is a Version Control
- Basic CLI Commands
- Introduction to Git
- Setting up VS Code
- Introduction to GitHub
- Final Quiz



# What is a Version Control System?



# Let's Visualize



You are a designer



poster.png



final-poster.png



pakka-final-poster.png



maa-kasam-  
final-poster.png



# What is Version Control System?

**Version Control System manages the entire process of version and keeps the track of the changes made.**

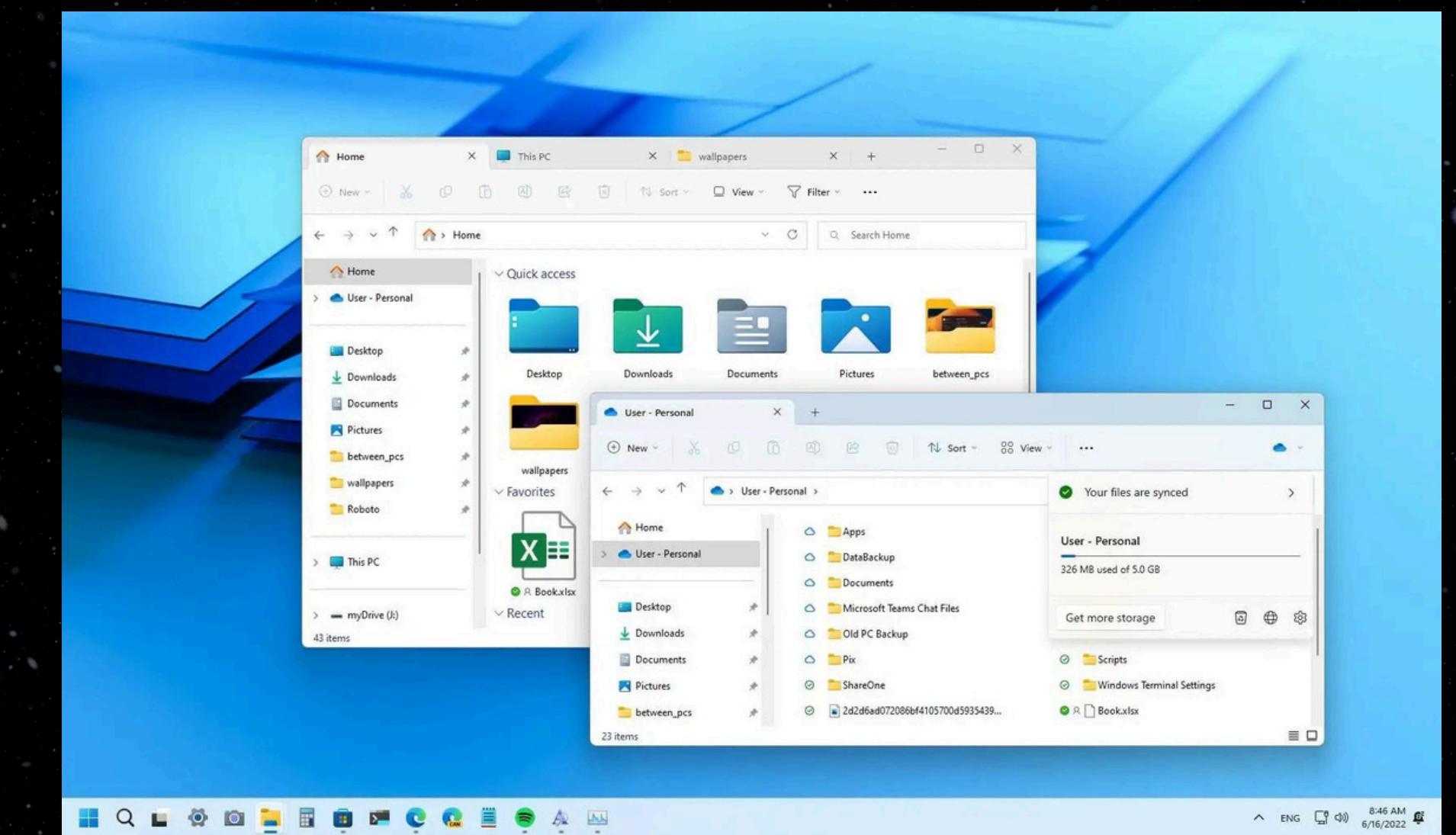
We need VCS to:

- track the changes made,
- revert back to any versions when needed,
- compare the changes made between any two versions,
- to know information like who, when, why made the changes,
- to get all these benefits with minimal effort.



Before we learn Git  
Let's learn about CLI

```
user@tecmint: ~
File Edit View Search Terminal Help
[user ] [user ]
/home/user [home/user]
Name Size Date Name Size Date
.. 4096 27 Jan 13:50 ..
.cache 4096 11 Feb 16:00 .cache 4096 11 Feb 16:00
.config 4096 11 Feb 16:36 .config 4096 11 Feb 16:36
.gnupg 4096 10 Feb 13:49 .gnupg 4096 10 Feb 13:49
.local 4096 27 Jan 13:55 .local 4096 27 Jan 13:55
.mozilla 4096 11 Feb 15:57 .mozilla 4096 11 Feb 15:57
.ssh 4096 10 Feb 13:49 .ssh 4096 10 Feb 13:49
.vim 4096 11 Feb 16:27 .vim 4096 11 Feb 16:27
/Desktop 4096 27 Jan 13:55 /Desktop 4096 27 Jan 13:55
/Documents 4096 27 Jan 13:55 /Documents 4096 27 Jan 13:55
/Downloads 4096 27 Jan 13:55 /Downloads 4096 27 Jan 13:55
/Music 4096 27 Jan 13:55 /Music 4096 27 Jan 13:55
/Pictures 4096 27 Jan 13:55 /Pictures 4096 27 Jan 13:55
/Public 4096 27 Jan 13:55 /Public 4096 27 Jan 13:55
/Templates 4096 27 Jan 13:55 /Templates 4096 27 Jan 13:55
/Videos 4096 27 Jan 13:55 /Videos 4096 27 Jan 13:55
.bash_history 57 10 Feb 17:39 .bash_history 57 10 Feb 17:39
.bash_logout 220 27 Jan 13:50 .bash_logout 220 27 Jan 13:50
.bashrc 3771 27 Jan 13:50v .bashrc 3771 27 Jan 13:50v
F1=Help
```



# Command line Interface

# Graphical User Interface



# Basic CLI Commands

```
$ pwd
```

Prints the full name (the full path) of current/working directory

```
$ cd Desktop
```

Changes the working directory to Desktop.  
Use this to navigate around CLI

```
$ ls
```

List directory contents.

```
$ mkdir name
```

Create a new directory called *name*.

Let's Try it!



# What is Git?

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.



## Let's Install Git

<https://git-scm.com/downloads>



## Let's Install VSC

<https://code.visualstudio.com/>



```
$ git --version
```

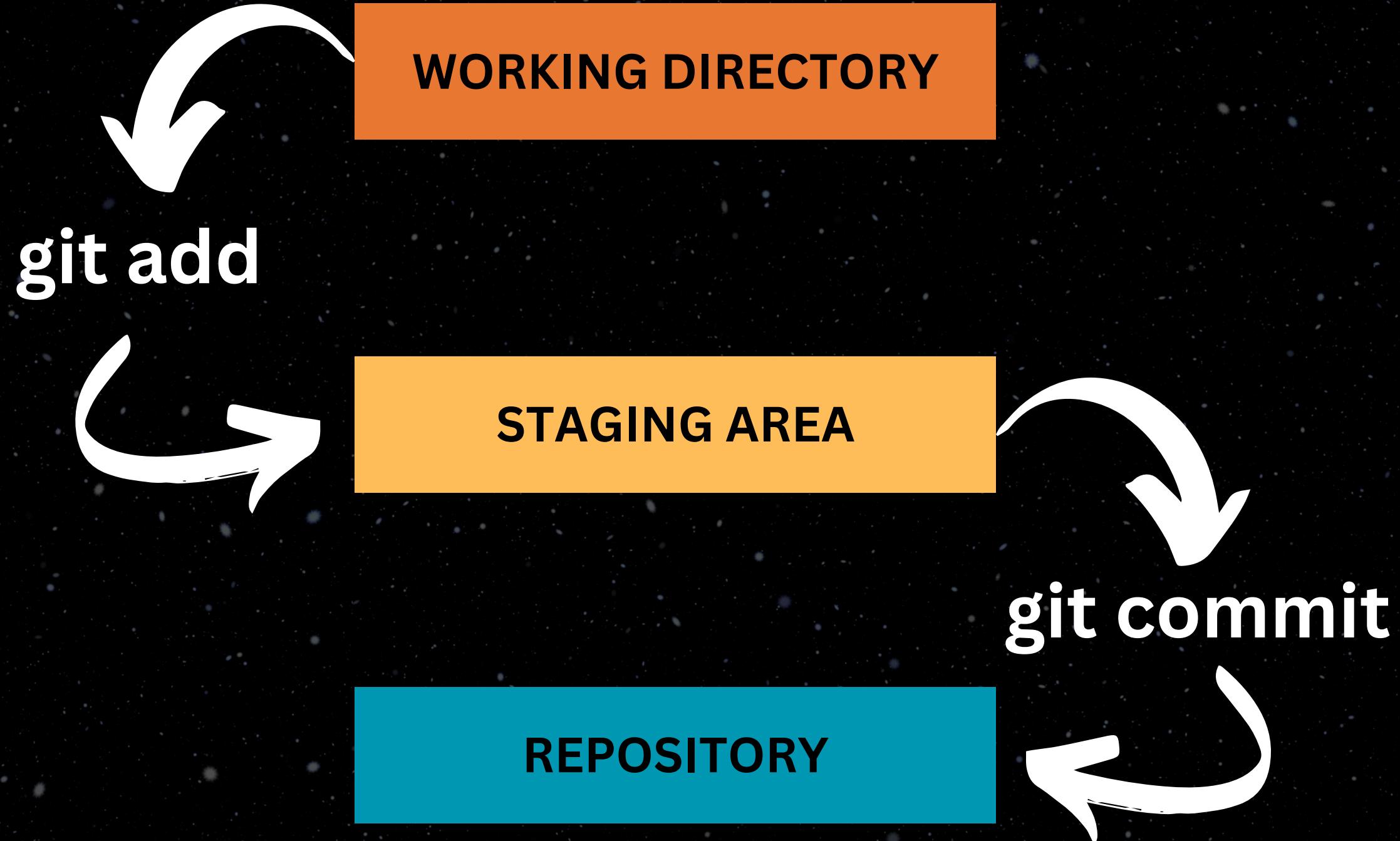
```
$ git config --global user.name "[firstname lastname]"
```

```
$ git config --global user.email "[valid-email]"
```

```
$ git init
```



# How Git works



# Git Commands



```
$ git status
```

```
$ git add [file]
```

```
$ git reset [file]
```

# Git Commands

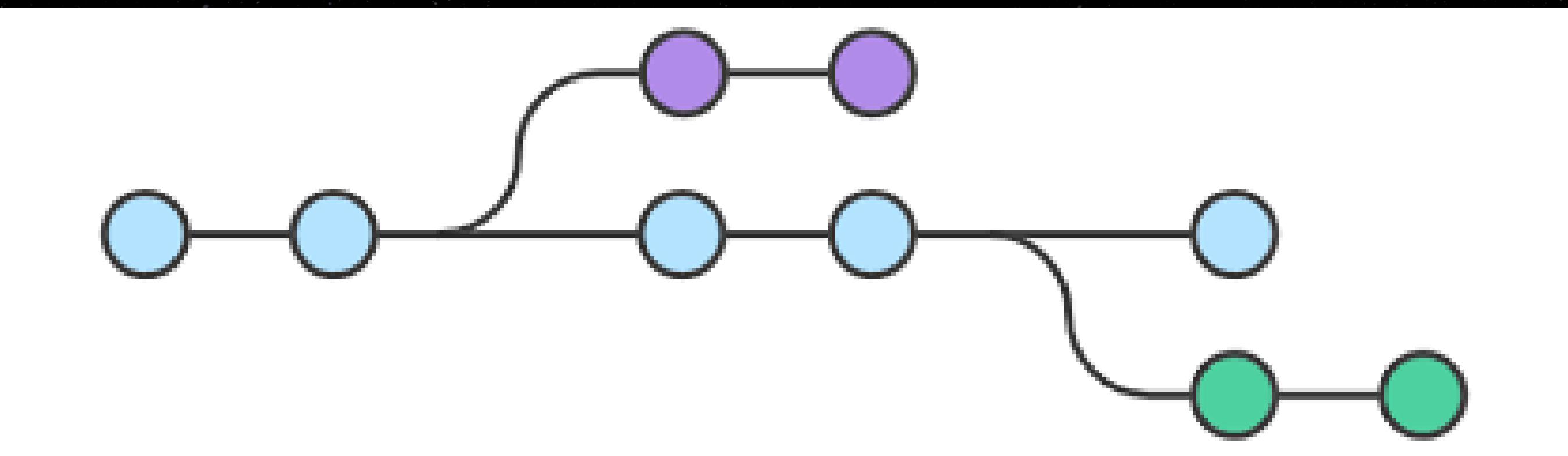


```
$ git commit -m "Some valuable message"
```

current branch.

```
$ git log
```

## Branching



# Git Commands



```
$ git branch
```

```
$ git branch [name]
```

```
$ git checkout [name]
```

```
$ git merge [branch]
```



# Git ignore



```
1 # .gitignore file is used to ignore some files in a repo
2 # such files always remain untracked
3 # the use of such file is done as follows
4
5 # any line starting with # is a comment
6 # ignoring file named password
7 password
8 # ignoring all files ending with .exe
9 *.exe
10 # but not ignoring final.exe
11 !final.exe
12 # ignoring all files inside build folder
13 build/
14 # ignoring file TODO not inside any subdirectory
15 /TODO
```

# What is GitHub



- Till now, we have all the changes made saved in our local system only.
- To host our repository online, we need a server or hosting platform.
- GitHub is a repository hosting platform that uses Git in its core.

Let's make an account!

# GitHub Student Developer Pack

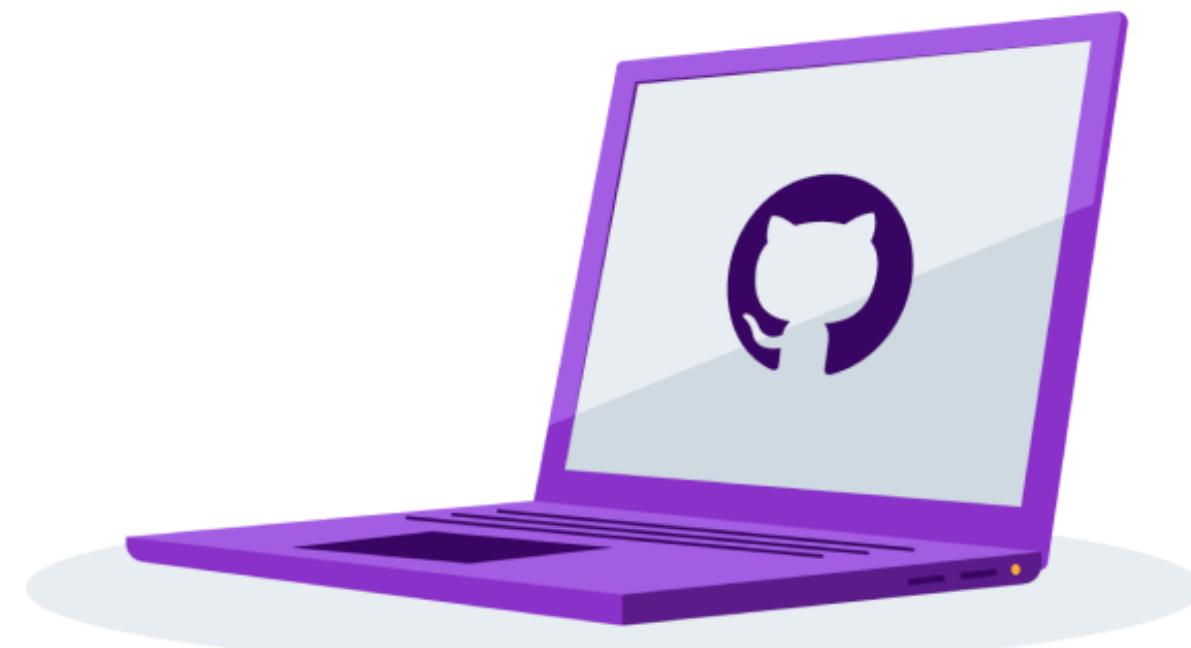
Learn to ship software like a pro. There's no substitute for hands-on experience. As a student, real world tools can be cost-prohibitive. That's why we created the GitHub Student Developer Pack with some of our partners and friends.

[Sign up for Student Developer Pack](#)

Love the pack? Spread the word



# Goodies



## Experiences



### Aspiring Creatives

Working on a creative project? Develop your design and collaboration skills to get your clever intentions off the ground. Unleash your originality and start to



### Primer: Codespaces

Wondering how to get started with GitHub Codespaces? This Primer makes it easy by giving you templates, videos and step-by-step instructions to help you get started.

# Btech-FY



# Let's Connect Git and GitHub



# Some More Git Commands

Add a remote repo link

```
$ git remote add origin [link]
```

Push local repo changes to remote repo

```
$ git push
```

Fetch and merge changes from remote repo

```
$ git pull
```

Retrieve an hosted repository from a URL

```
$ git clone [link]
```



# Some Terminologies

- **Fork:** A fork is a copy of a repository that allows you to freely experiment with changes without affecting the original project.
- **Pull requests:** A PR is request sent to the repo to merge new commits into the repo.
- **Issues:** Issues is what it sounds. It can be a new feature, bug, documentation problem and many more.



# Workflow to Contribute

01

Find a issue or Suggest a  
feature

02

Ask the maintainer to  
assign the issue

03

Fork and Clone the  
Repo

04

Branch and make  
changes

05

Commit your changes

06

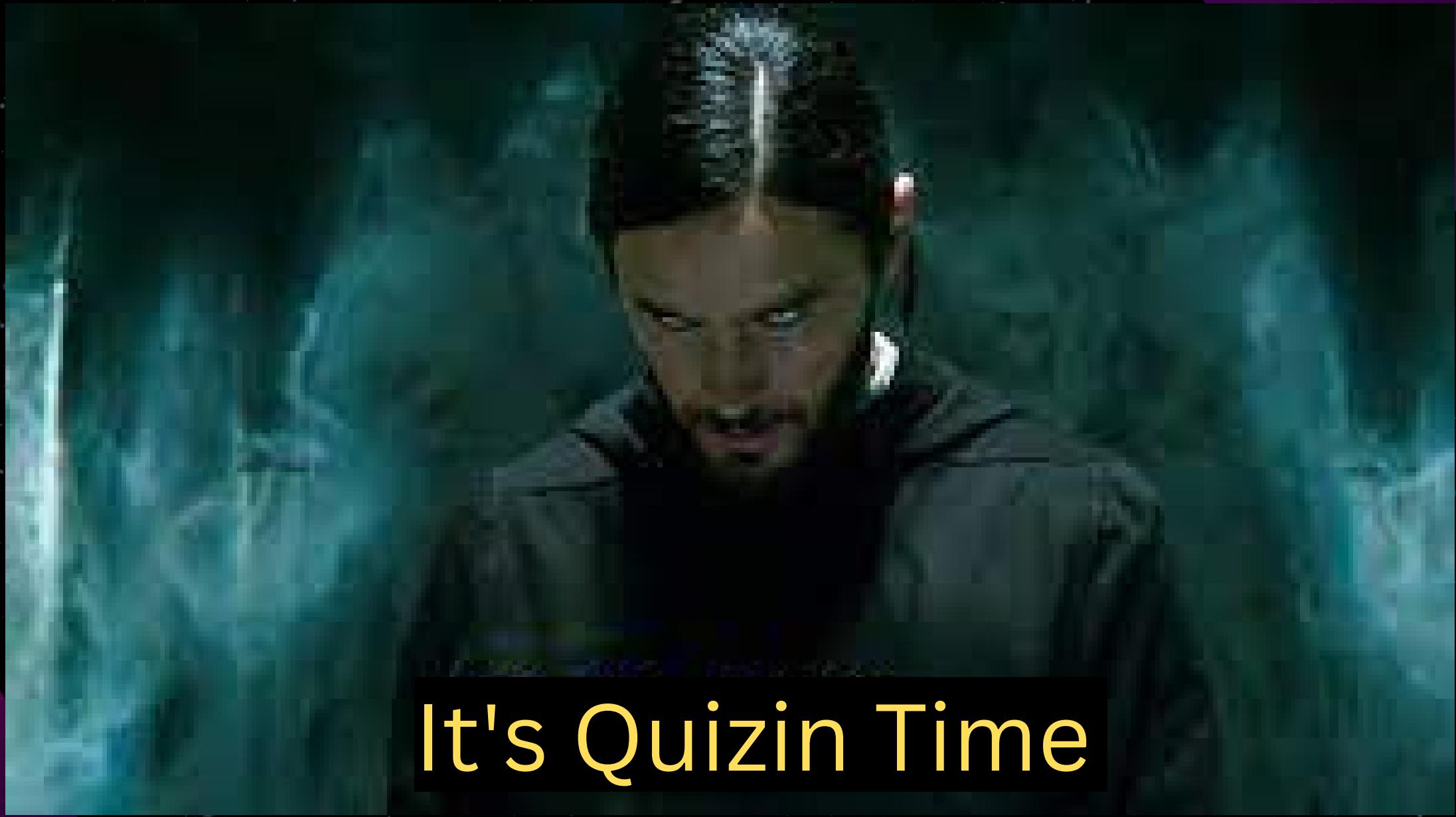
Create a Pull Request

07

Listen to Review and  
do changes if needed

08

Hurray! 🎉 The PR got  
accepted



It's Quizin Time



# Feedback Form





# Thank You!