



Git

A distributed version control system





Agenda

- Introduction to Version Control Systems
- Git VS Github
- Code Management
- Collaboration
- Conclusion

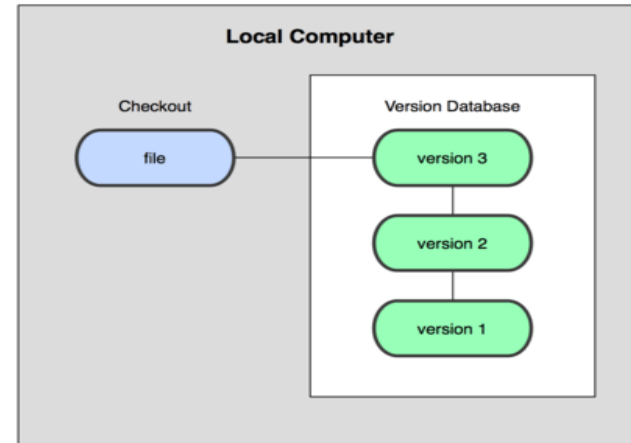


Version control systems

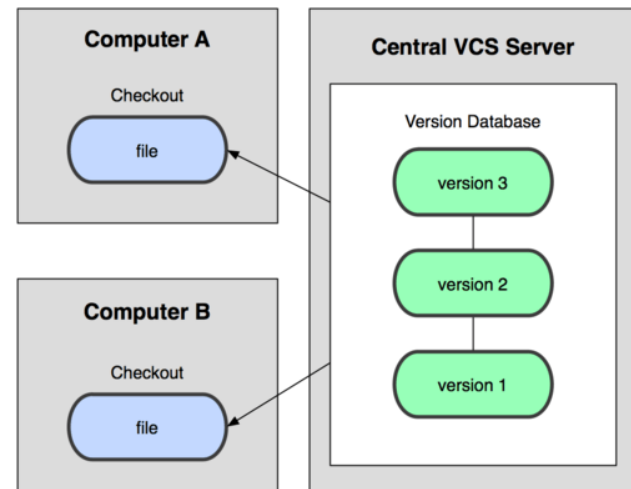
- **Version control** (or **revision control**, or **source control**)
Is tracking and managing of changes to software code.
- It keeps track of every modification in the code and store it in a database
 - If a mistake is made, developers can turn back the clock and compare earlier versions of the code.
- version control systems help software teams work faster and smarter.
 - It helps to minimize the disruption to all team members
 - It saves the source code from catastrophe and damaging

Types Of VCS

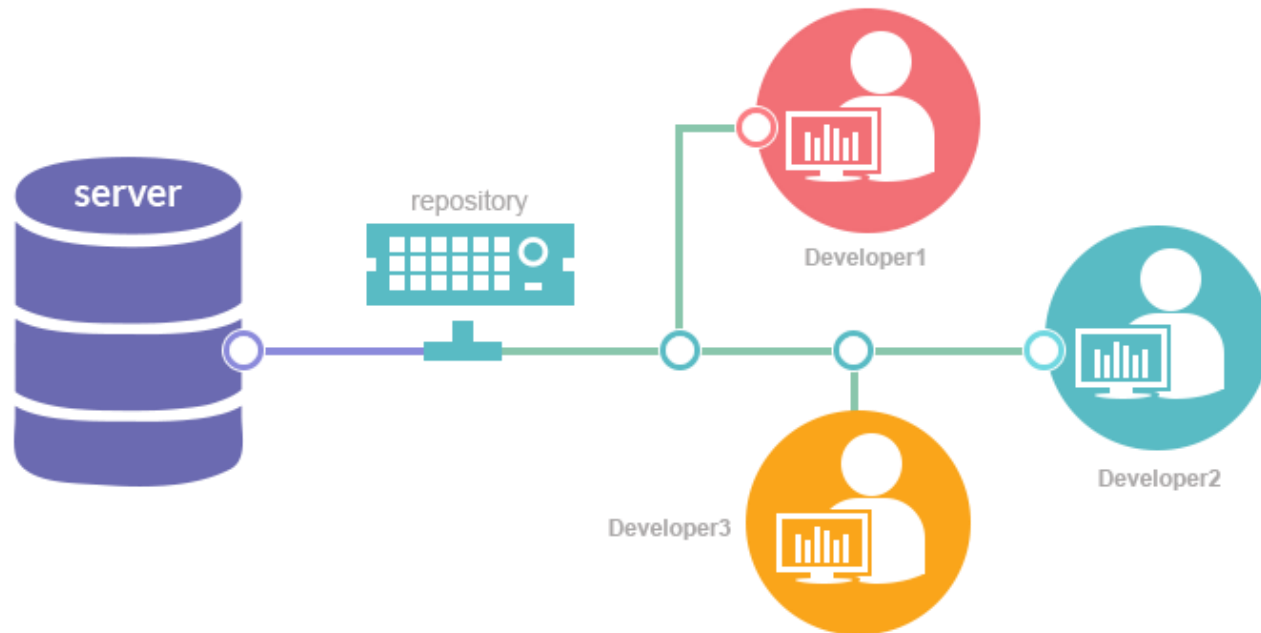
- Local version control system



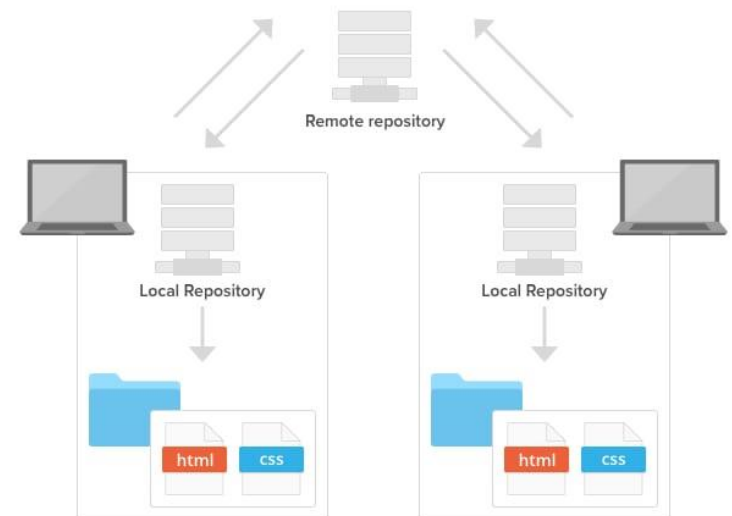
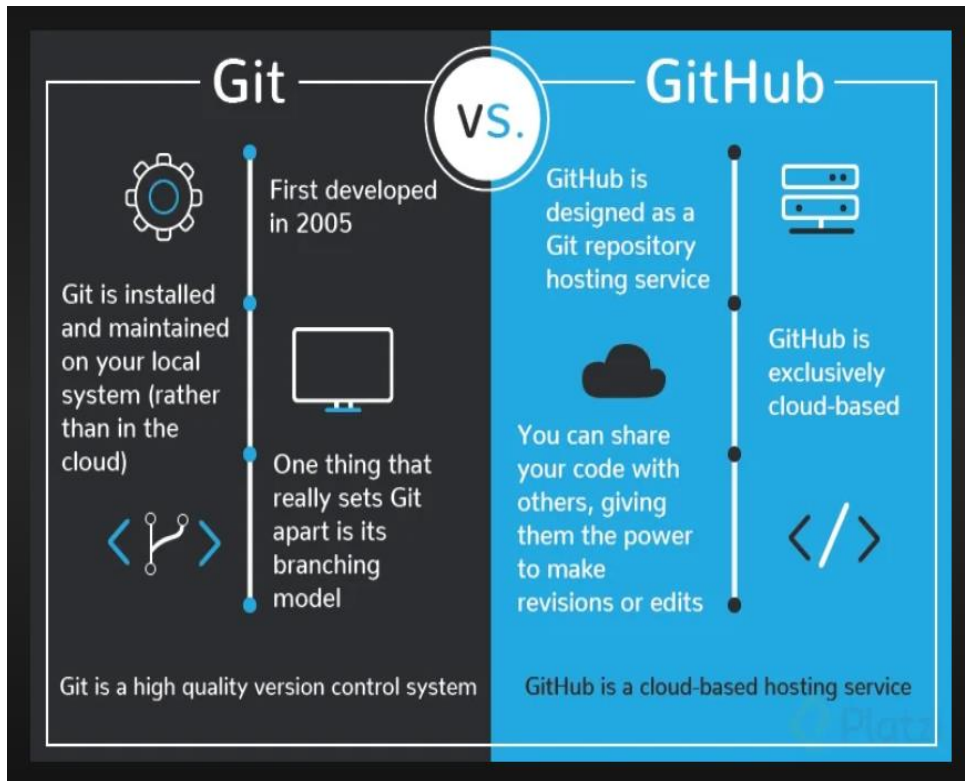
- Centralized version control system



- Distributed version control system

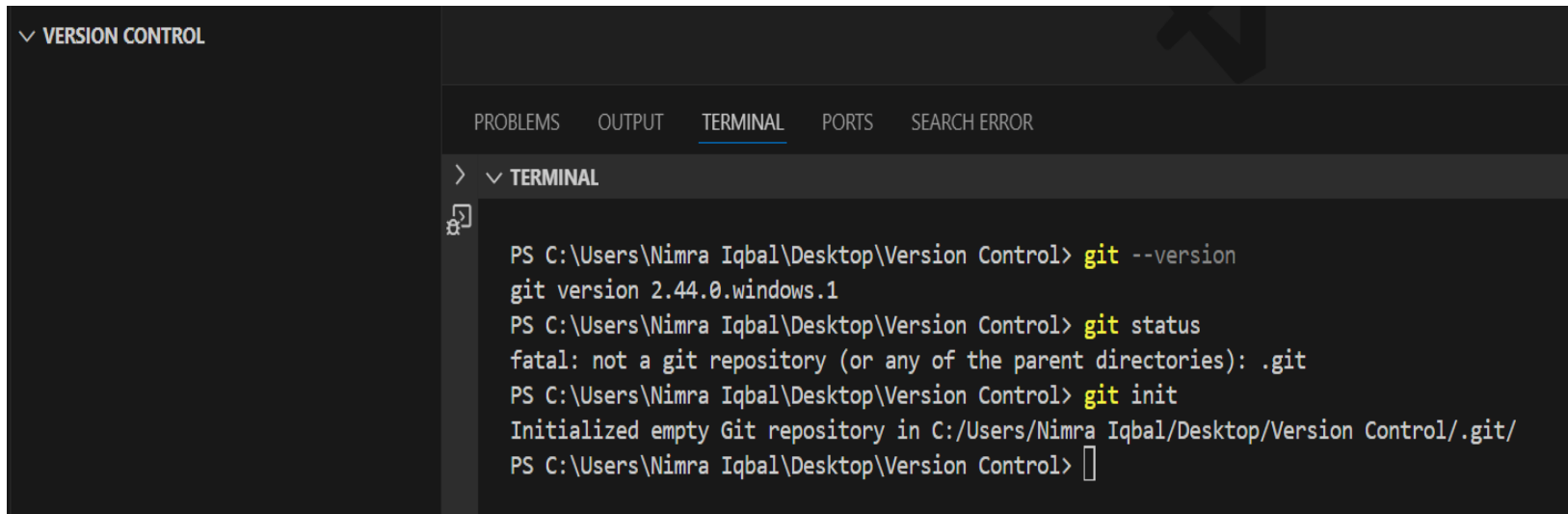


Git VS Github



Code Management

- ❑ **git --version**: check whether git is installed or not.
- ❑ <https://git-scm.com/downloads>: Link to install git on your machine
- ❑ **git status**: check the state of your Git working directory.
- ❑ **git init**: create an empty Git repository or reinitialize an existing one.



```

✓ VERSION CONTROL

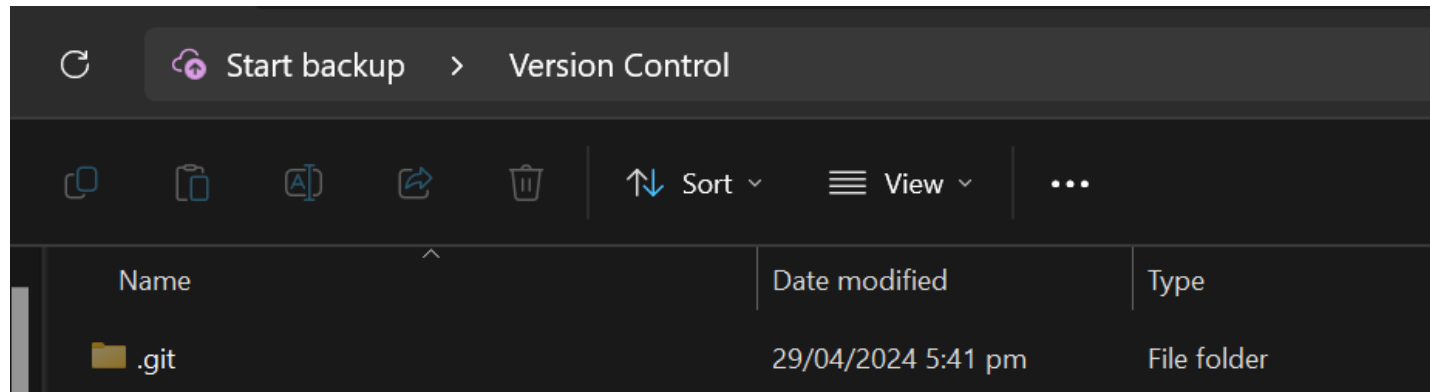
PROBLEMS  OUTPUT  TERMINAL  PORTS  SEARCH ERROR

> ✓ TERMINAL
❏
PS C:\Users\Nimra Iqbal\Desktop\Version Control> git --version
git version 2.44.0.windows.1
PS C:\Users\Nimra Iqbal\Desktop\Version Control> git status
fatal: not a git repository (or any of the parent directories): .git
PS C:\Users\Nimra Iqbal\Desktop\Version Control> git init
Initialized empty Git repository in C:/Users/Nimra Iqbal/Desktop/Version Control/.git/
PS C:\Users\Nimra Iqbal\Desktop\Version Control> 
```



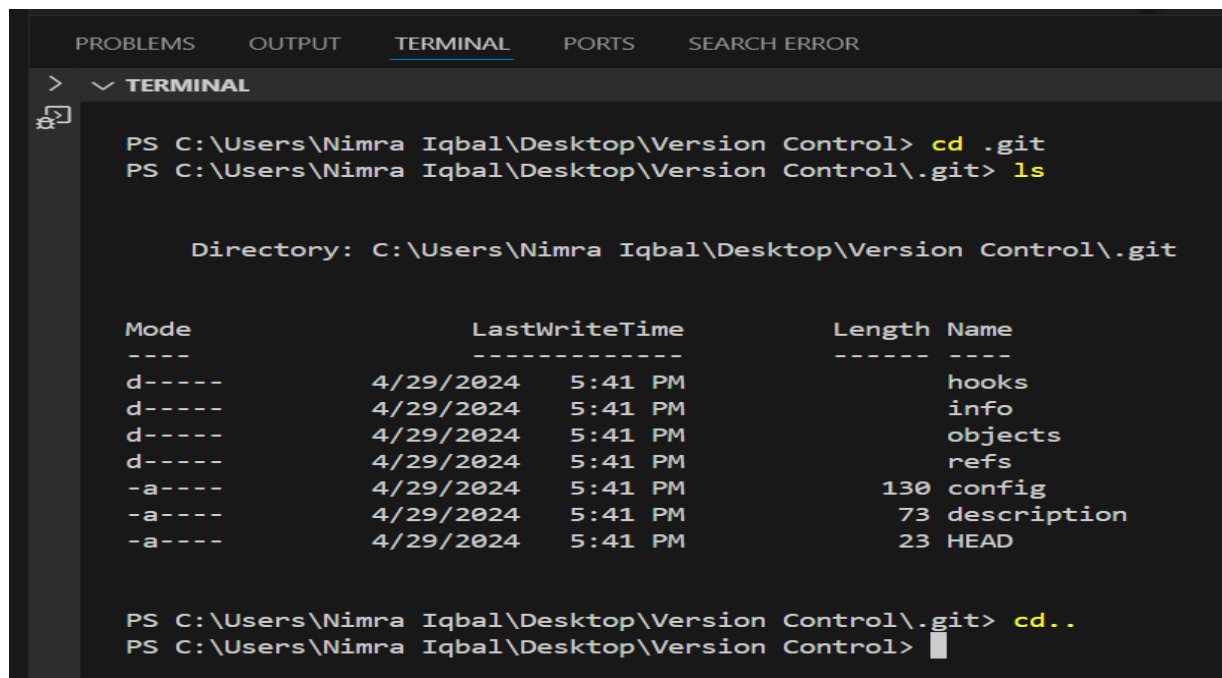
Code Management

.git repository created inside the Version Control repository.



Code Management

- ❑ **cd .git**: move inside .git folder
- ❑ **ls**: list of all files and directories inside .git folder.
- ❑ **cd..**: back to Version Control directory.



```
PROBLEMS  OUTPUT  TERMINAL  PORTS  SEARCH ERROR
>  ▾  TERMINAL
PS C:\Users\Nimra Iqbal\Desktop\Version Control> cd .git
PS C:\Users\Nimra Iqbal\Desktop\Version Control\.git> ls

Directory: C:\Users\Nimra Iqbal\Desktop\Version Control\.git

Mode                LastWriteTime         Length Name
----                -
d-----          4/29/2024   5:41 PM             hooks
d-----          4/29/2024   5:41 PM             info
d-----          4/29/2024   5:41 PM            objects
d-----          4/29/2024   5:41 PM             refs
-a----          4/29/2024   5:41 PM           130 config
-a----          4/29/2024   5:41 PM           73 description
-a----          4/29/2024   5:41 PM           23 HEAD

PS C:\Users\Nimra Iqbal\Desktop\Version Control\.git> cd..
PS C:\Users\Nimra Iqbal\Desktop\Version Control>
```

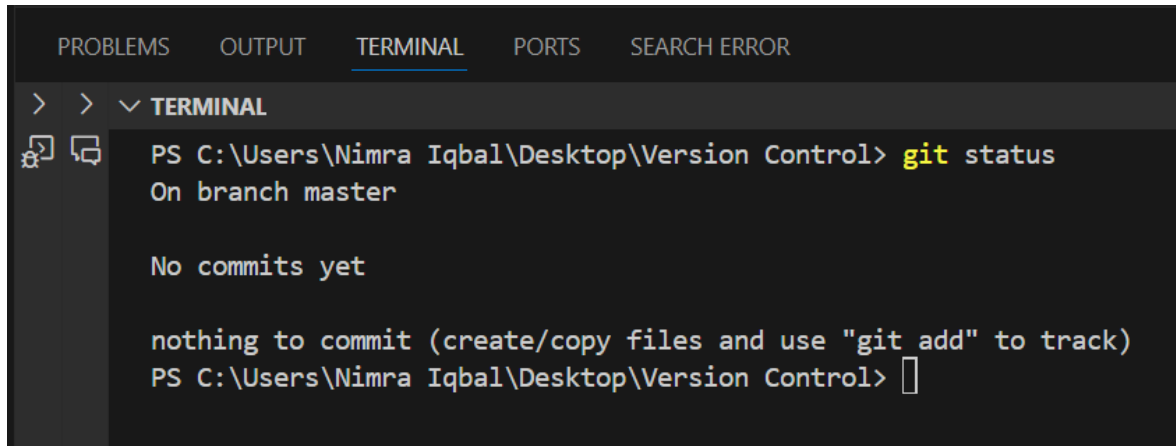


Code Management

- ❑ **git status**: check the state of your Git working directory.

On branch master

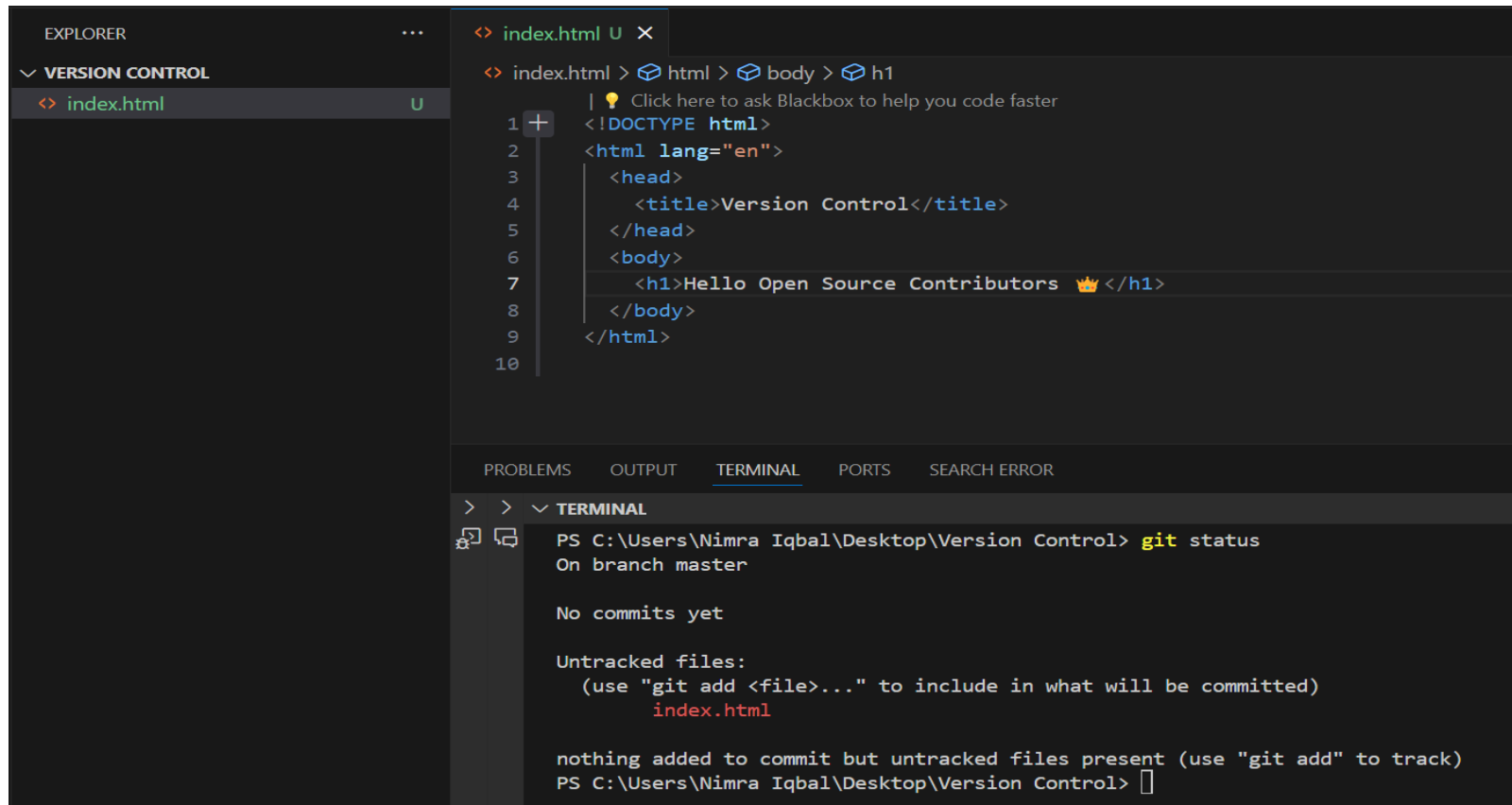
No commits yet



```
PROBLEMS  OUTPUT  TERMINAL  PORTS  SEARCH ERROR  
> > ∨ TERMINAL  
PS C:\Users\Nimra Iqbal\Desktop\Version Control> git status  
On branch master  
  
No commits yet  
  
nothing to commit (create/copy files and use "git add" to track)  
PS C:\Users\Nimra Iqbal\Desktop\Version Control> 
```

Code Management

An **index.html** file is created, but it is untracked.



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar shows a project named 'VERSION CONTROL' with a file 'index.html' marked with a green 'U' for untracked. The main editor displays the content of 'index.html', which is a basic HTML document with a title 'Version Control' and a heading 'Hello Open Source Contributors'. Below the editor, the TERMINAL panel shows the output of the command 'git status'.

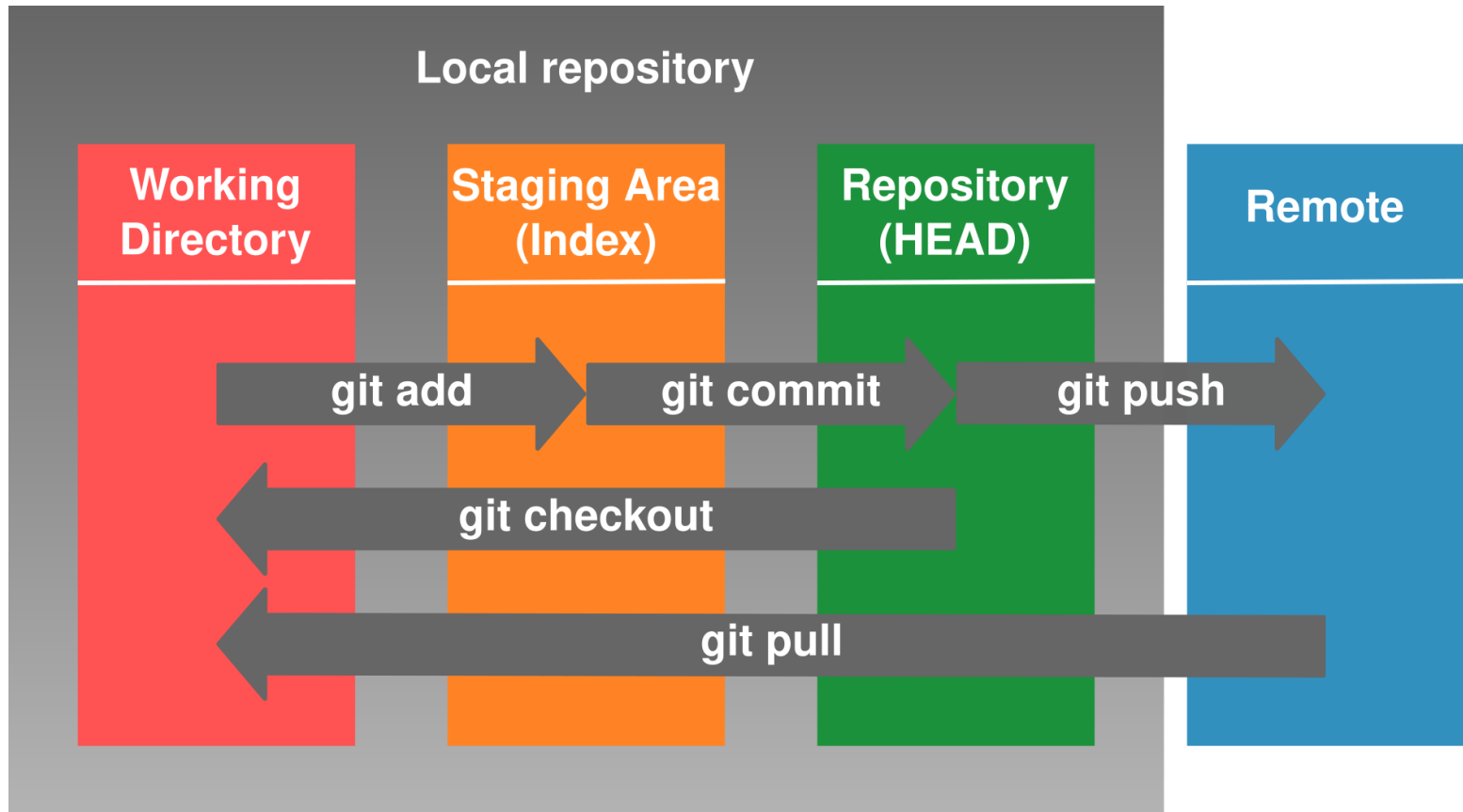
```
PS C:\Users\Nimra Iqbal\Desktop\Version Control> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\Nimra Iqbal\Desktop\Version Control>
```

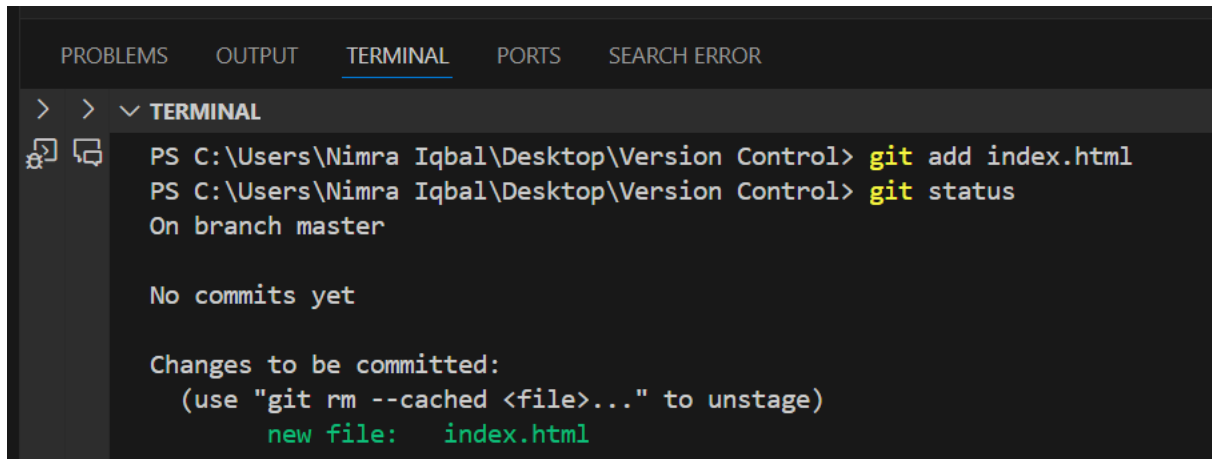
Git Different Zones





Code Management

- ❑ **git add index.html**: a copy of the file index.html is added to the staging area.
- ❑ **git rm --cached index.html**: to remove index.html from staging area.



```
PROBLEMS  OUTPUT  TERMINAL  PORTS  SEARCH ERROR

> > ▼ TERMINAL
❏ ❏ PS C:\Users\Nimra Iqbal\Desktop\Version Control> git add index.html
PS C:\Users\Nimra Iqbal\Desktop\Version Control> git status
On branch master

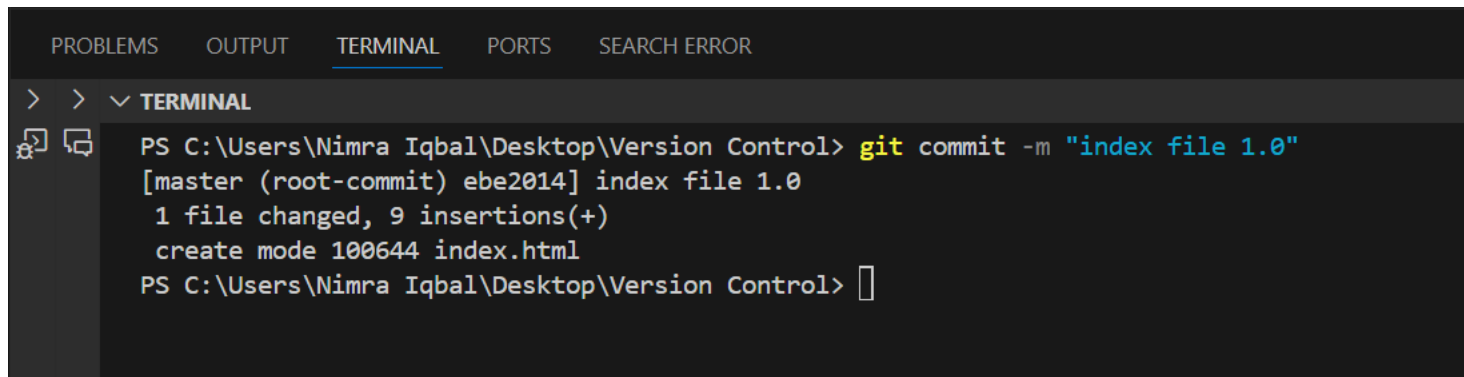
No commits yet

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



Code Management

- ❑ **git commit -m “index file 1.0”**: snapshot of the repository along with the message “index file 1.0”.



```
PROBLEMS  OUTPUT  TERMINAL  PORTS  SEARCH ERROR  
> > ▼ TERMINAL  
PS C:\Users\Nimra Iqbal\Desktop\Version Control> git commit -m "index file 1.0"  
[master (root-commit) ebe2014] index file 1.0  
1 file changed, 9 insertions(+)  
create mode 100644 index.html  
PS C:\Users\Nimra Iqbal\Desktop\Version Control> 
```



Code Management

- ☐ You must have an account on github
- ☐ Create a new repository.
- ☐ Here I have created a repository with the name “Version Control”.

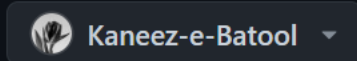
Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Required fields are marked with an asterisk ().*

Owner *



Repository name *

/ Version Control

✔ Your new repository will be created as Version-Control.

The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

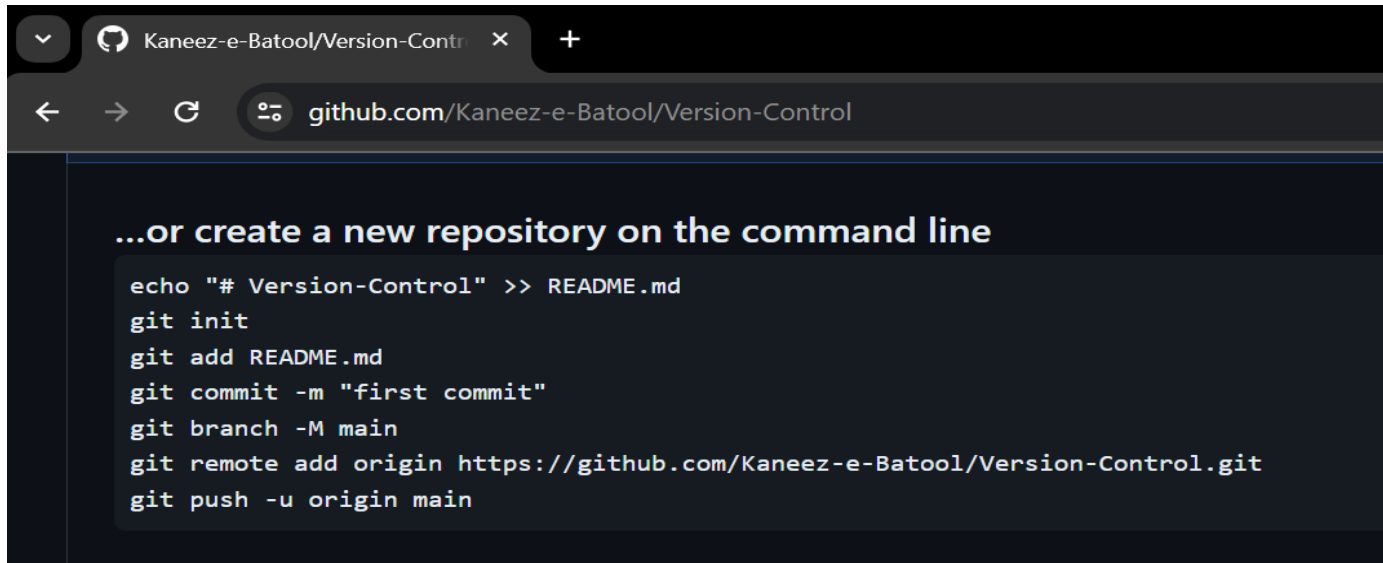
Great repository names are short and memorable. Need inspiration? How about

super-duper-computing-machine ?



Code Management

- ❑ These are the commands provided by github in order to link our git repository with the github repository.
- ❑ For now, we will run the last two commands on our terminal.



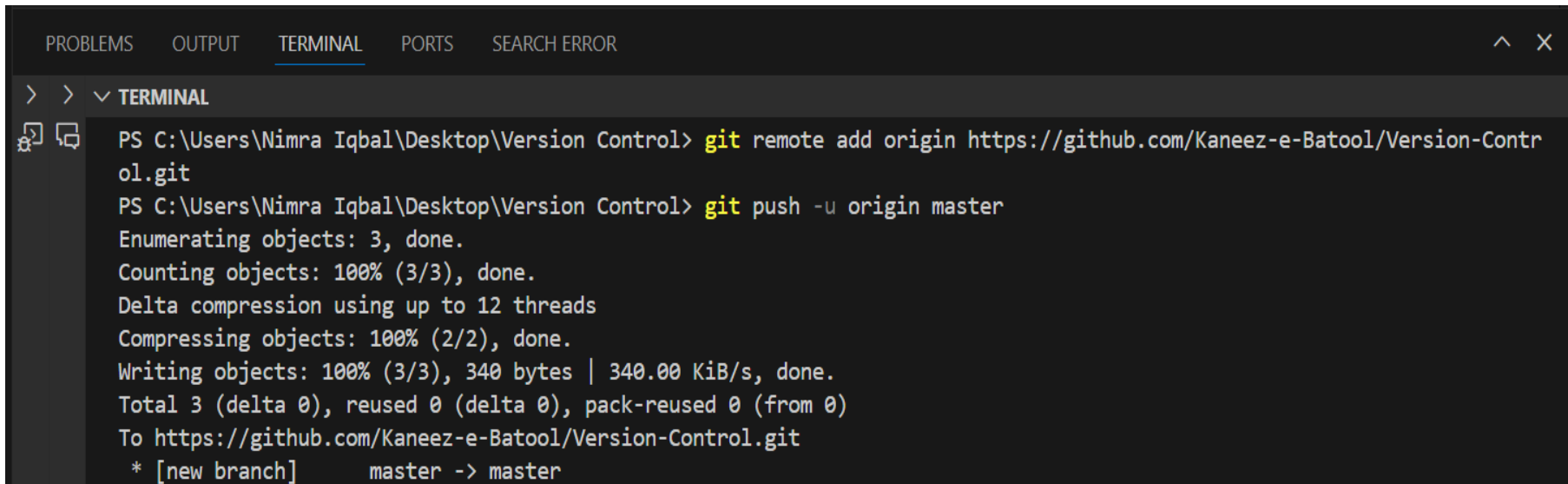
```
...or create a new repository on the command line

echo "# Version-Control" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Kaneez-e-Batool/Version-Control.git
git push -u origin main
```




Code Management

- ❑ The connection between our git repository and github repository has been established.



```
PROBLEMS  OUTPUT  TERMINAL  PORTS  SEARCH ERROR
> > v TERMINAL
PS C:\Users\Nimra Iqbal\Desktop\Version Control> git remote add origin https://github.com/Kaneez-e-Batool/Version-Contr
ol.git
PS C:\Users\Nimra Iqbal\Desktop\Version Control> git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 340 bytes | 340.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Kaneez-e-Batool/Version-Control.git
 * [new branch]      master -> master
```

Code Management

- ❑ Files have been transferred on our github “Version Repository”.

The screenshot shows the GitHub interface for a repository named 'Version-Control' by user 'Kaneez-e-Batool'. The repository is public. The main navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The repository header shows the name 'Version-Control', a 'Public' badge, and buttons for 'Pin' and 'Unwatch' (with a count of 1). Below the header, there's a section for branches and tags, showing 'master' as the selected branch, '1 Branch', and '0 Tags'. A search bar 'Go to file' is present. To the right are buttons for 'Add file' and 'Code' (with a dropdown arrow). The commit history section shows a single commit by 'ebe2014' titled 'index file 1.0', made '8 minutes ago', with '1 Commits'. Below this, a file named 'index.html' is listed, also titled 'index file 1.0' and dated '8 minutes ago'.

☰ Kaneez-e-Batool / Version-Control 🔍 Type / to search

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Version-Control Public Pin Unwatch 1

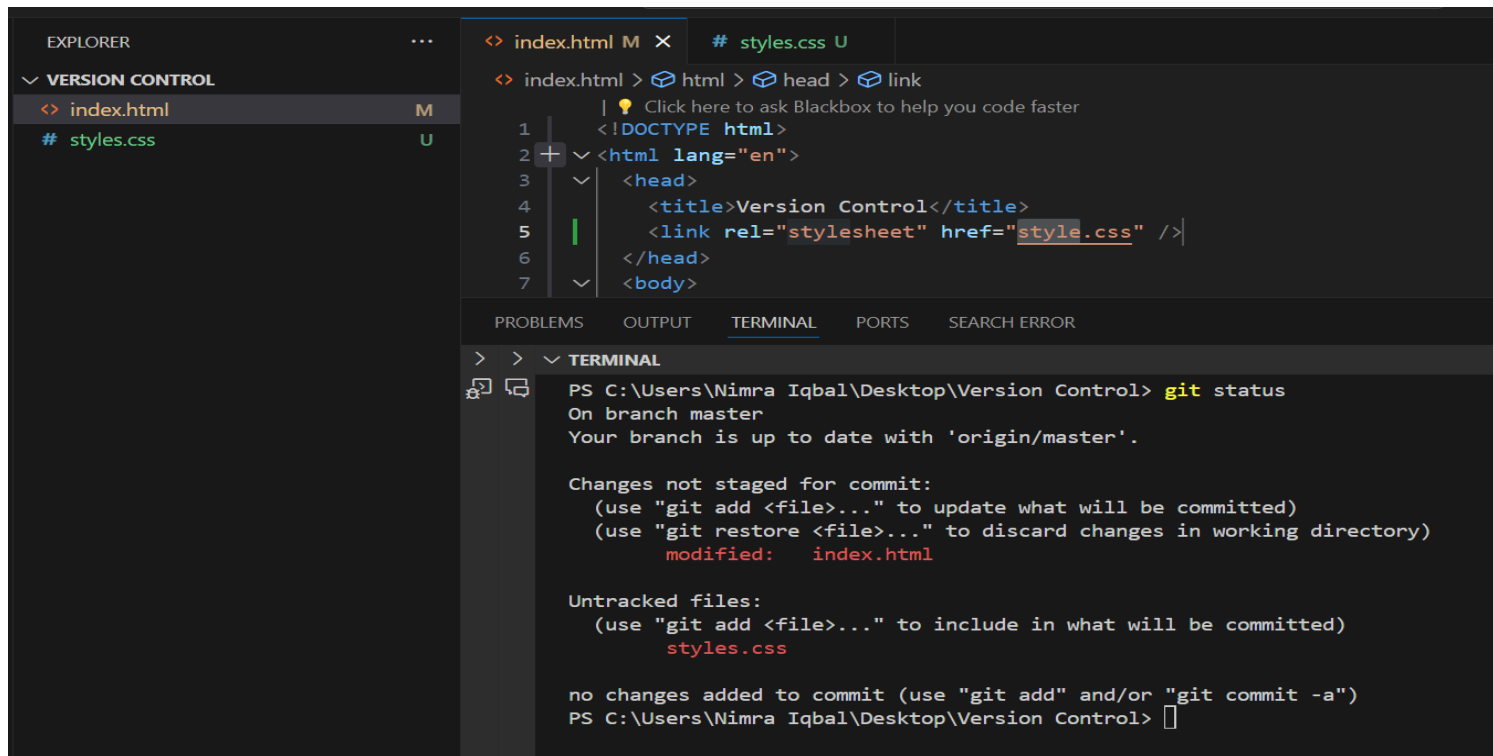
master 1 Branch 0 Tags 🔍 Go to file t Add file <> Code ▾

Kaneez-e-Batool index file 1.0 ebe2014 · 8 minutes ago ⌚ 1 Commits

index.html index file 1.0 8 minutes ago

Code Management

- ❑ After transferring files from git to github, changes have been made on our local repository.
- ❑ A new file styles.css is created and index.html is modified/
- ❑ M says that the file is modified, while U says the file is untracked.



The screenshot displays the Visual Studio Code interface. On the left, the Explorer sidebar shows the 'VERSION CONTROL' section with two files: 'index.html' marked as 'M' (modified) and 'styles.css' marked as 'U' (untracked). The main editor area shows the 'index.html' file with the following content:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Version Control</title>
<link rel="stylesheet" href="style.css" />
</head>
<body>
```

Below the editor, the TERMINAL panel shows the output of the 'git status' command:

```
PS C:\Users\Nimra Iqbal\Desktop\Version Control> git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

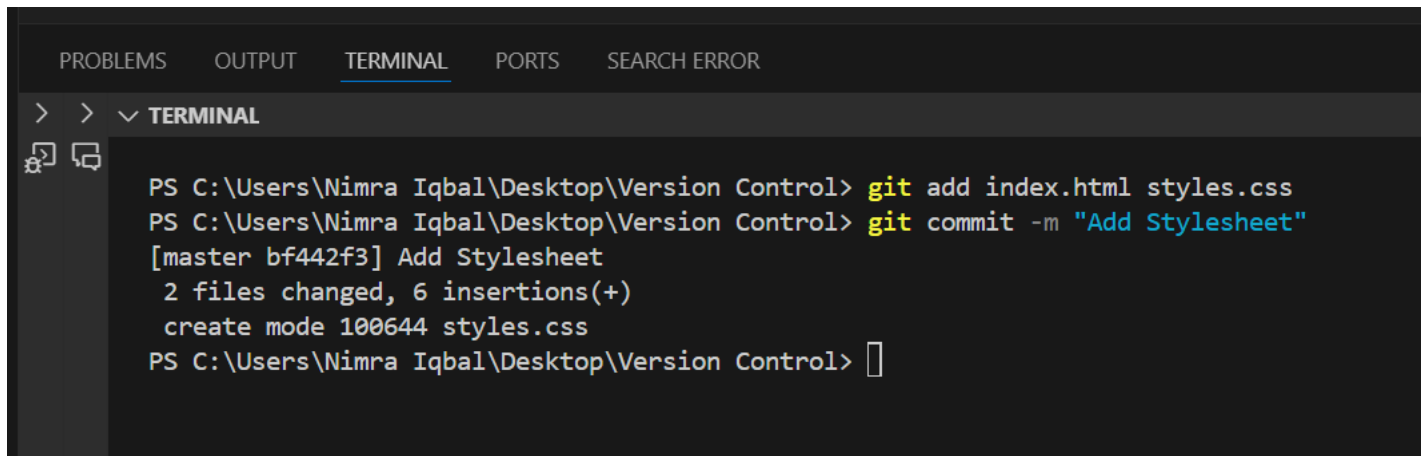
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        styles.css

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\Nimra Iqbal\Desktop\Version Control>
```



Code Management

- ❑ **git add index.html styles.css** : Add index.html and styles.css to the staging area
- ❑ **git commit -m “Add stylesheet”**: snapshot of the repository along with the message “Add stylesheet”.

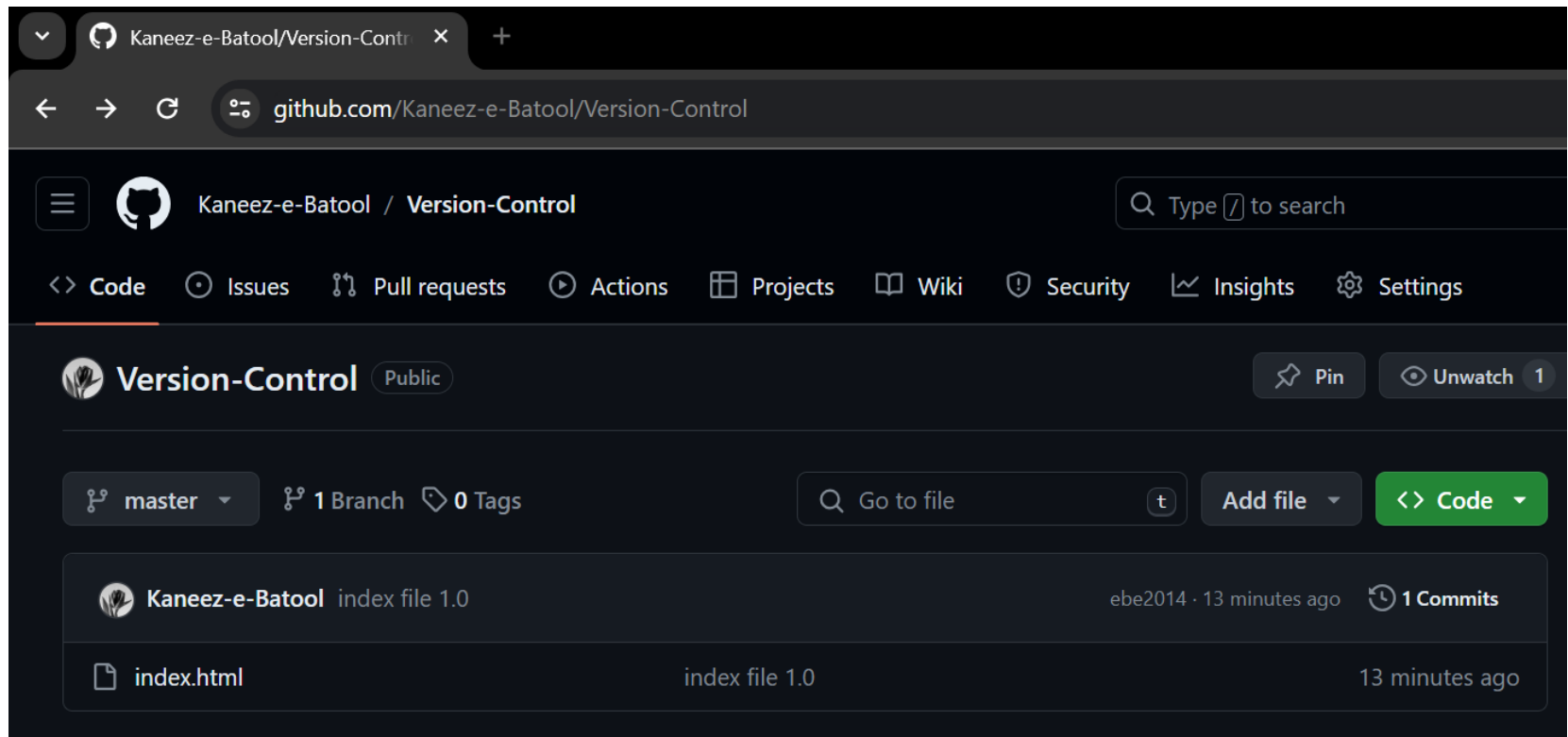


```
PROBLEMS  OUTPUT  TERMINAL  PORTS  SEARCH ERROR

> > ∨ TERMINAL
❏ ❏
PS C:\Users\Nimra Iqbal\Desktop\Version Control> git add index.html styles.css
PS C:\Users\Nimra Iqbal\Desktop\Version Control> git commit -m "Add Stylesheet"
[master bf442f3] Add Stylesheet
 2 files changed, 6 insertions(+)
 create mode 100644 styles.css
PS C:\Users\Nimra Iqbal\Desktop\Version Control> 
```

Code Management

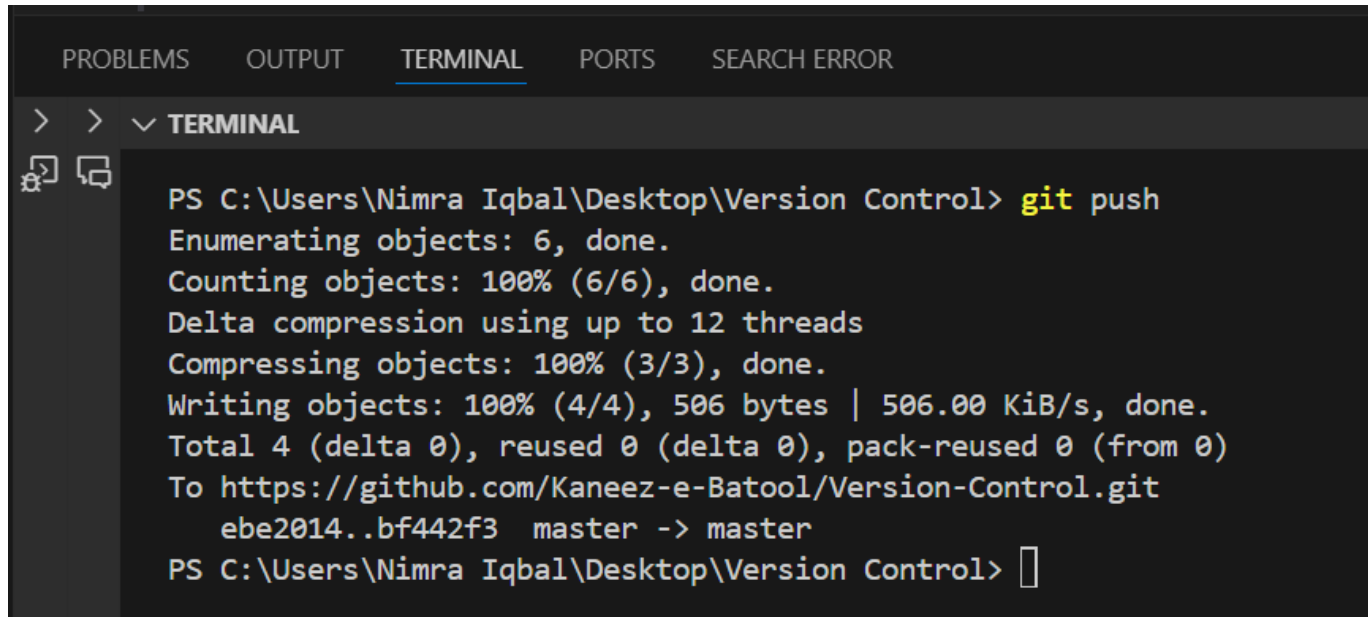
- ❑ Until now, the file styles.css is not added to our Github repository Version Control, because we have not push it from Git repository.





Code Management

- ❑ All updations are saved from commit to the github repository Version Control.



```
PROBLEMS  OUTPUT  TERMINAL  PORTS  SEARCH ERROR

> > ∨ TERMINAL
❏ ❏
PS C:\Users\Nimra Iqbal\Desktop\Version Control> git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 506 bytes | 506.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Kaneez-e-Batool/Version-Control.git
    ebe2014..bf442f3  master -> master
PS C:\Users\Nimra Iqbal\Desktop\Version Control> 
```

Code Management

- ❑ Now we can see our sysles.css and modified index.html on github.

The screenshot shows a GitHub repository page for 'Kaneez-e-Batool / Version-Control'. The page is in dark mode. At the top, there's a navigation bar with the GitHub logo, the repository name, and a search bar. Below this is a secondary navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The 'Code' tab is selected. The repository name 'Version-Control' is displayed with a 'Public' badge. To the right are 'Pin' and 'Unwatch' buttons. Below the repository name, there's a section for branches and tags, showing 'master' as the current branch, '1 Branch', and '0 Tags'. A 'Go to file' search bar is present. To the right are 'Add file' and 'Code' buttons. The main content area shows a commit by 'Kaneez-e-Batool' titled 'Add Stylesheet' with commit hash 'bf442f3' and '2 Commits' made '1 minute ago'. Below the commit, there's a list of files: 'index.html' and 'styles.css', both with the message 'Add Stylesheet' and a timestamp of '1 minute ago'.

github.com/Kaneez-e-Batool/Version-Control

Kaneez-e-Batool / Version-Control

Type to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Version-Control Public

Pin Unwatch 1

master 1 Branch 0 Tags

Go to file Add file Code

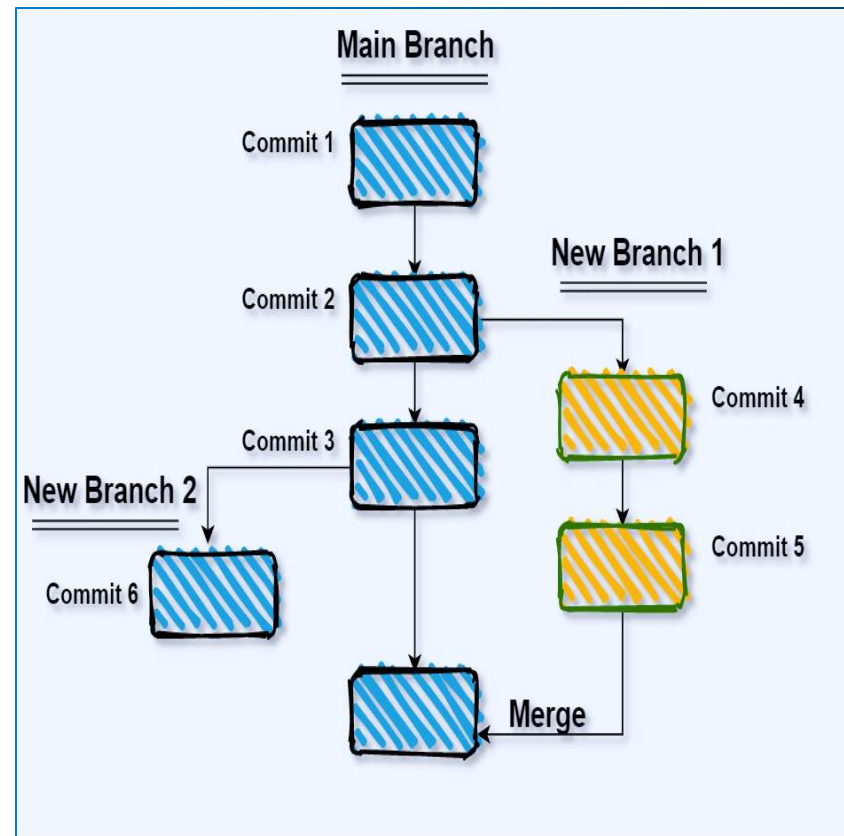
Kaneez-e-Batool Add Stylesheet bf442f3 · 1 minute ago 2 Commits

index.html	Add Stylesheet	1 minute ago
styles.css	Add Stylesheet	1 minute ago

Branching and merging

Git uses branching heavily to switch between multiple tasks.

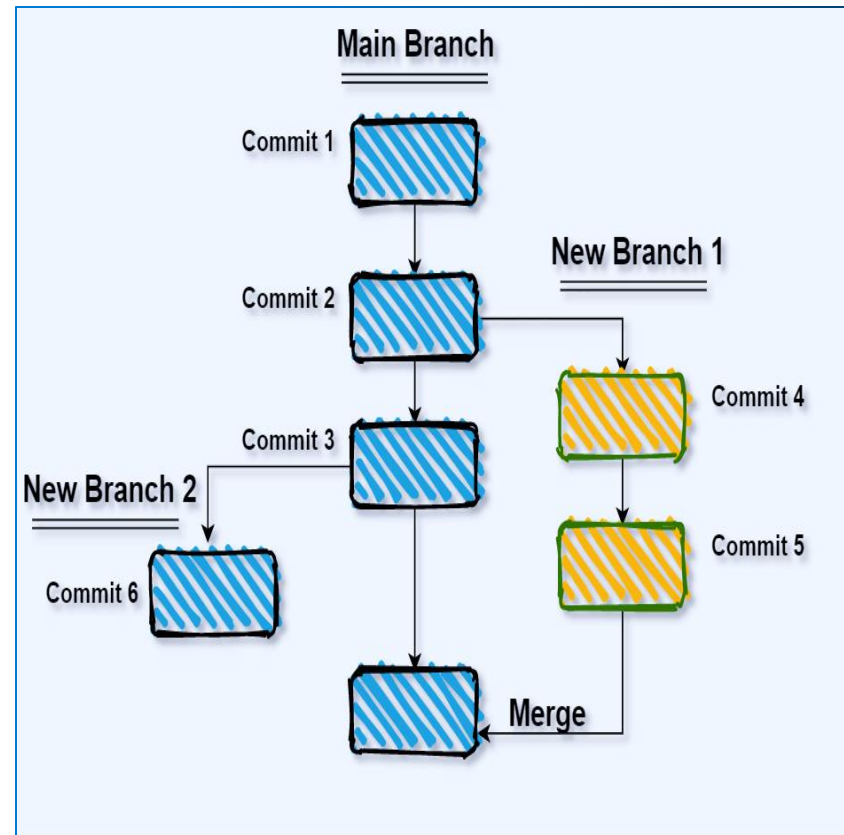
- To create a new local branch:
– **git branch name**
- To list all local branches:
(* = **current branch**) – **git branch**



Branching and merging

Git uses branching heavily to switch between multiple tasks.

- To switch to a given local branch:
 - **git checkout branchname**
- To merge changes from a branch into the local master:
 - **git checkout master** –
 - git merge branchname**





Enhancing Collaboration in Distributed Development

- Distributed development teams face challenges in coordinating efforts and sharing code effectively.
- Version control systems like Git provide a centralized place for developers to collaborate on code from different locations.
- Features such as branching and merging enable parallel development without conflicts.
- Real-time updates and notifications keep team members informed about changes made by others.



Enhancing Collaboration in Distributed Development

- Allows for flexible and remote work arrangements
- Enables cross-functional collaboration
- Reduces conflicts and errors with version control and branching
- Increases productivity and efficiency
- Enhances code quality and maintainability.



Benefits of Version Control

- **Traceability:** Detailed history of code changes aids project oversight.
- **Commit Logging:** Essential commit details aid comprehension of code progression.
- **Code Reviews:** Structured reviews ensure quality and teamwork.
- **Automated Testing and CI/CD:** Integration streamlines validation and deployment, boosting code reliability.



Benefits of Version Control

- **Versioning:** Easy management of different code versions
- **Security:** Access control and authentication ensure only authorized access
- **Backup and Recovery:** Regular backups and easy recovery in case of data loss
- **Code Reuse:** Existing code can be easily reused in other projects