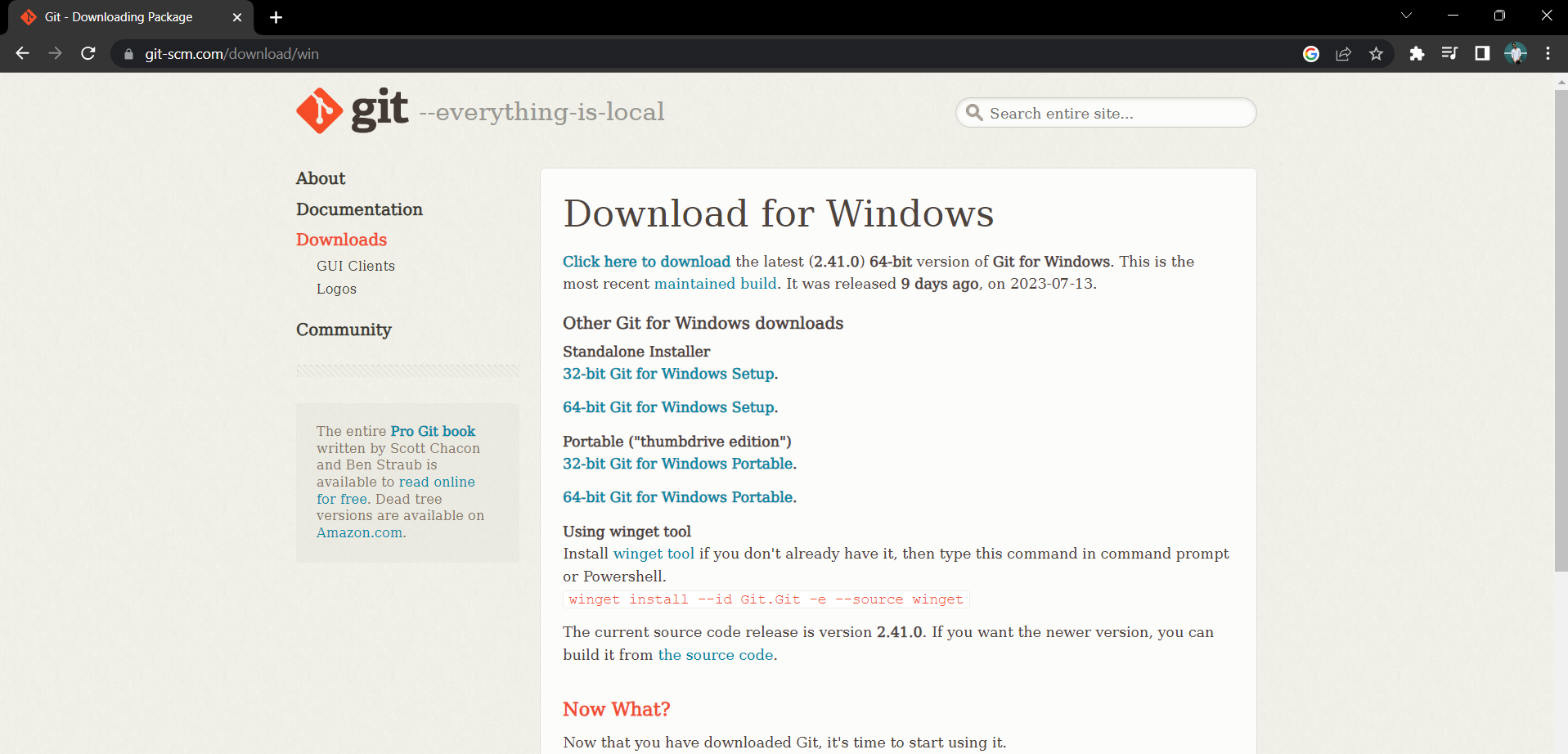
**GIT and GIThub Documents**

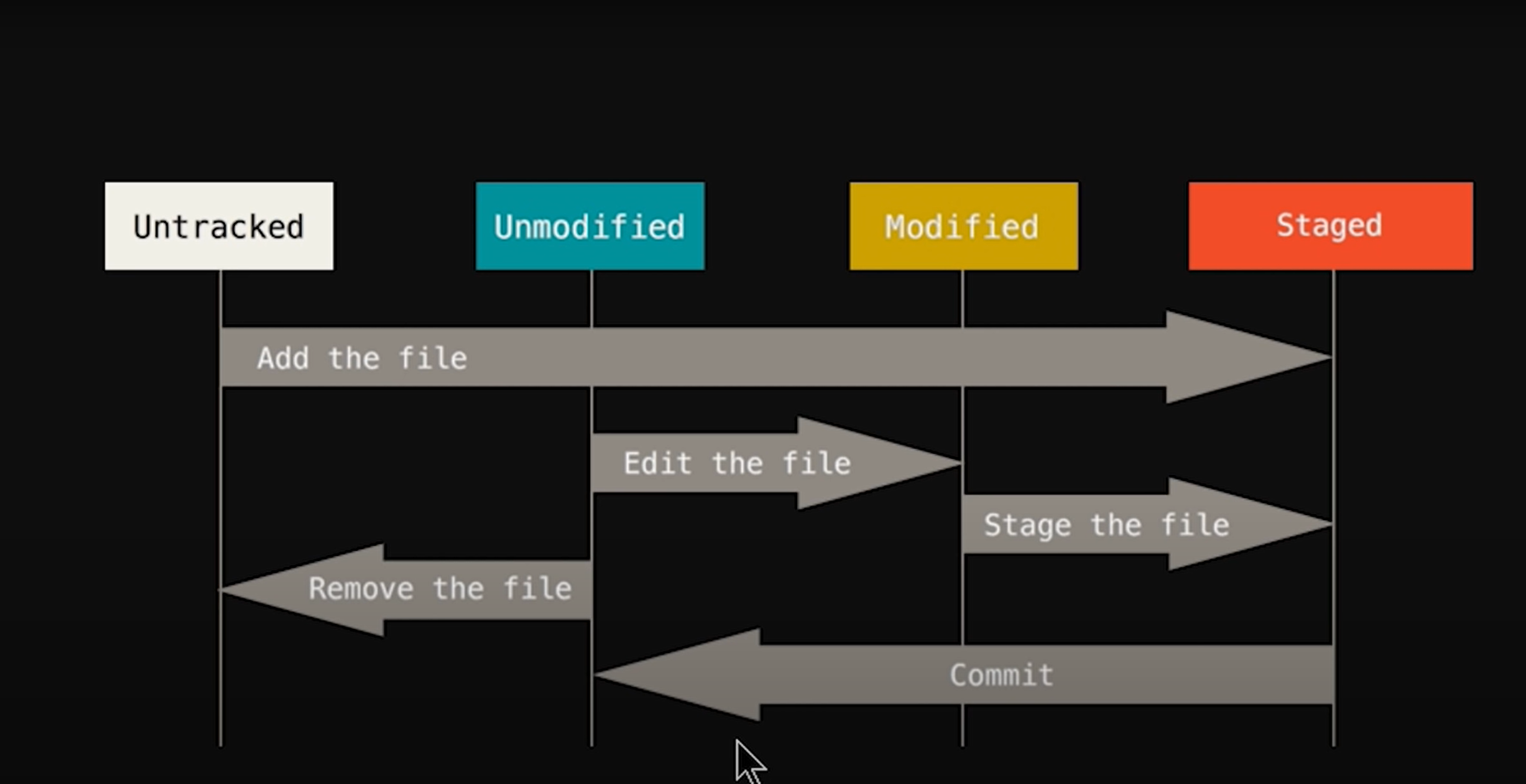
|  |  |
| --- | --- |
| SI | Topic |
| 1 | 1.1 Installation:  1.2 GIT Staging Concepts |
| 2 | 2.1 To set up or config username and email  2.2 Open visual code and then Initialized empty Git repository  2.3 Ls –lart shows the hidden folder and git folder contains the file we store and receive  2.4 It shows me my status that file is untracked  2.5 Creating 3 empty files and check status  2.6 Staged all files by git add –A  2.7 Commit shortcut:  2.8 Checkout the contact.html  2.9 Checkout –f for all the file reform \*\*  2.10 Git log tells us about author date and commit details  2.11 Git diff Compares working area VS Staging area  2.12 Git diff –staged Compares last commit VS Staging area  2.13 Skipping the stage area and direct commit  2.14 Normal operating system CLI for get file list  2.15 creating the waste.html and then untracked it from commit  2.16 creating the waste.html and then delete it permanently  2.17 Status –s (summerise status) Not so important |
| 3 | 3.1 Gitignore  3.2 Branch ADDED,VIEW,DELETE  3.3 Switched branch  3.4 Create branch and switched in 1 line  3.5 Git merge to insert the modified code of branch in main branch |
| 4 | 4.1 Creating a private repository  4.2 URL to host repository ,To connect local repository to remote repository  4.3 Set remote as origin (name) for short name of URL  4.4 Check git remote and remote url  4.5 go to setting and SSH and GPG key for give access my github account of my pc: We have to deploy it |
| 5 | 5.1 Git clone |

**1.1 Installaltion:**

**WE have to install and set up git from**



**1.2 GIT Staging Concepts**

****

**2.1 To set up or config username and email :**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz

$ git config --global user.name Shammo34

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz

$ git config --global user.email kazishammo2020@gmail.com

**To check:**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz

$ git config --global user.email

kazishammo2020@gmail.com

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz

$ git config --global user.name

Shammo34

**2.2 Open visual code and then Initialized empty Git repository**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz

$ code .

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz

$ git init

Initialized empty Git repository in C:/Users/Lenovo/Desktop/git and github atoz/.git/

**2.3 Ls –lart shows the hidden folder and git folder contains the file we store and recieve**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ ls -lart

total 16

drwxr-xr-x 1 Lenovo 197121 0 Jul 22 21:50 ../

drwxr-xr-x 1 Lenovo 197121 0 Jul 22 22:10 ./

drwxr-xr-x 1 Lenovo 197121 0 Jul 22 22:10 .git/

**2.4 It shows me my status that file is untracted**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

No commits yet

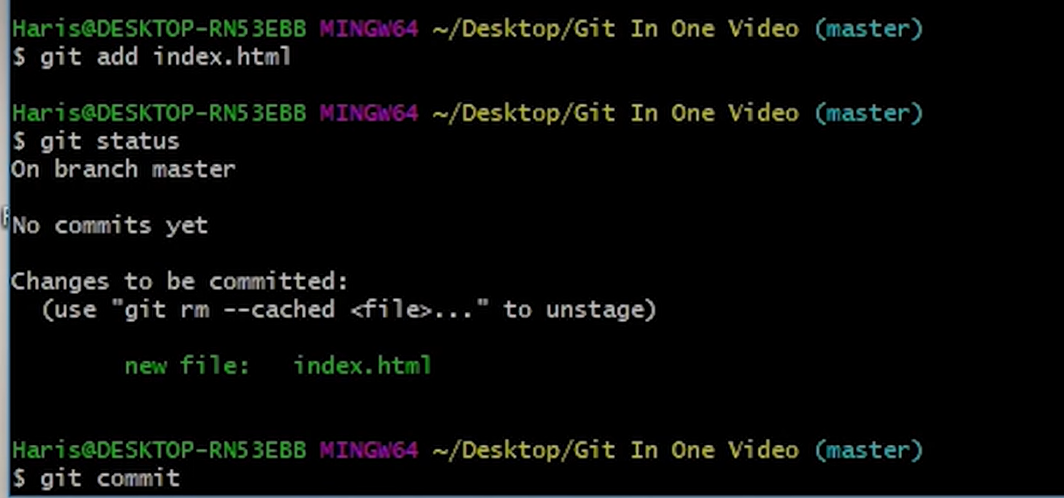
Untracked files:

(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)

**We have to add file to be staged(from untracked) and then commit it to save it is called Initial commit:**

****

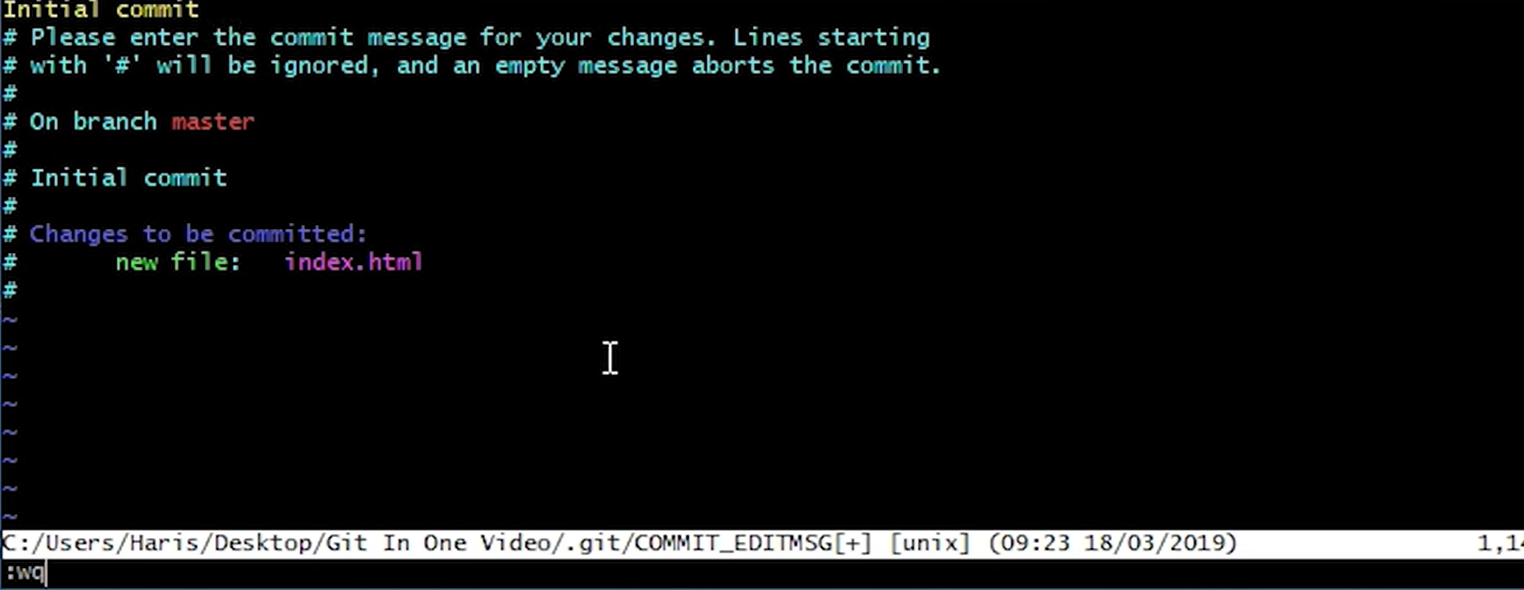
**After commit a new editor will be appeared :**

**We have to name the commit message for this**

**1,press i for write**

**2.esc after writing done**

**3. :wq to end and go back to the terminal**



**We have to check regularly so git status is important \*\***

**2.5 Creating 3 empty files and check status**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ touch about.html

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ touch contact.html

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ touch monuments.html

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

Untracked files:

(use "git add <file>..." to include in what will be committed)

about.html

contact.html

monuments.html

nothing added to commit but untracked files present (use "git add" to track)

**2.6 Staged all files by git add –A**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git add -A

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: about.html

new file: contact.html

new file: monuments.html

**After modifing the code in vscode of contact.html if we check the git status:**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: about.html

new file: contact.html

new file: monuments.html

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: contact.html

**Modified to staged form of contact.html(we can use –A too)**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git add contact.html

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: about.html

new file: contact.html

new file: monuments.html

**2.7 Commit shortcut:**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git commit -m "Added more htmls"

[main 364ca49] Added more htmls

3 files changed, 12 insertions(+)

create mode 100644 about.html

create mode 100644 contact.html

create mode 100644 monuments.html

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$

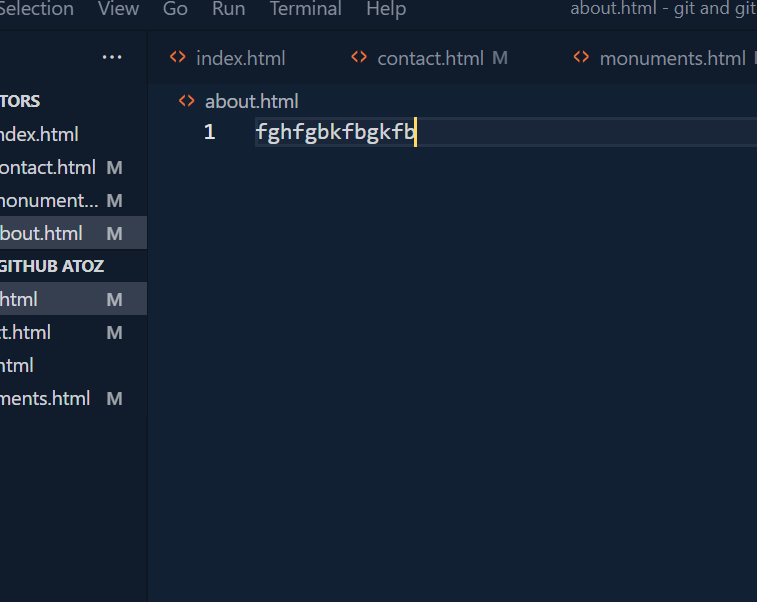
Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

nothing to commit, working tree clean

**\*\*\*\*Very Very important\*\*\*\***

****

**If accidently modify the code and shut down the pc it wll be great threat so we have to follow these steps**

**2.8 Checkout the contact.html**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

nothing to commit, working tree clean

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

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: contact.html

no changes added to commit (use "git add" and/or "git commit -a")

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git checkout contact.html

Updated 1 path from the index

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

nothing to commit, working tree clean

**2.9 Checkout –f for all the file reform \*\***

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

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: about.html

modified: contact.html

modified: monuments.html

no changes added to commit (use "git add" and/or "git commit -a")

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git checkout -f

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

nothing to commit, working tree clean

**2.10 Git log tells us about author date and commit details**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git log

commit 364ca4990807ed9a0f7e4f775a3ecfd35fcb7c99 (HEAD -> main)

Author: Shammo34 <kazishammo2020@gmail.com>

Date: Sun Jul 23 02:32:16 2023 +0600

Added more htmls

commit 67eb5d540b0d50b933cfa3271446d59080a27fa3

Author: Shammo34 <kazishammo2020@gmail.com>

Date: Sat Jul 22 22:32:49 2023 +0600

Initial commit

**To clear the terminal use $ clear\*\*\***

**\*\*\*\*\*Suppose we have 100 commits and we want 5 of them we will use git log -p -5 and to exit we will press q \*\*\*\*\*\*\*It tells more details**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git log -p -1

commit 364ca4990807ed9a0f7e4f775a3ecfd35fcb7c99 (HEAD -> main)

Author: Shammo34 <kazishammo2020@gmail.com>

Date: Sun Jul 23 02:32:16 2023 +0600

Added more htmls

**diff --git a/about.html b/about.html**

**new file mode 100644**

**index 0000000..e69de29**

**diff --git a/contact.html b/contact.html**

**new file mode 100644**

**index 0000000..56efbdb**

**--- /dev/null**

**+++ b/contact.html**

@@ -0,0 +1,12 @@

+<!DOCTYPE html>

+<html lang="en">

+<head>

+ <meta charset="UTF-8">

+ <meta http-equiv="X-UA-Compatible" content="IE=edge">

+ <meta name="viewport" content="width=device-width, initial-scale=1.0">

+ <title>Document</title>

+</head>

+<body>

2.11 Git diff Compares working area VS Staging area

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git diff

**diff --git a/index.html b/index.html**

**index d78f2c4..47f2941 100644**

**--- a/index.html**

**+++ b/index.html**

@@ -9,5 +9,5 @@

<link rel="stylesheet" href="style.css" />

<script src="main.js"></script>

</head>

- <body></body>

+ <body shammo></body>

</html>

But if we make the working area into staging area then there is no diff :

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git add -A

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git diff

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$

2.12 Git diff –staged Compares last commit VS Staging area

After that we reform into the last commit directory

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git diff --staged

**diff --git a/index.html b/index.html**

**index d78f2c4..47f2941 100644**

**--- a/index.html**

**+++ b/index.html**

@@ -9,5 +9,5 @@

<link rel="stylesheet" href="style.css" />

<script src="main.js"></script>

</head>

- <body></body>

+ <body shammo></body>

</html>

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git checkout -f

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

nothing to commit, working tree clean

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$

2.13 Skipping the stage area and direct commit

(BUT we should not use this cz staging area is safe and secured\*\*\*\*\*\*)

|  |  |
| --- | --- |
| commit -m | Commits the changes that are currently staged in the index. |
| commit -a -m | Commits all of the changes that are tracked by Git, including changes that are not staged in the index. |
| -m option | Tells the commit command to take a message as input. |
| -a option | Tells the commit command to automatically stage all of the changes that are not staged. |

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

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

no changes added to commit (use "git add" and/or "git commit -a")

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git commit -a -m "skipped staging area and fixed"

[main 467e9a3] skipped staging area and fixed

1 file changed, 3 insertions(+), 1 deletion(-)

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git log

commit 467e9a38721c963dc4d69a8ffb5db1be7123f7ed (HEAD -> main)

Author: Shammo34 <kazishammo2020@gmail.com>

Date: Sun Jul 23 03:15:59 2023 +0600

skipped staging area and fixed

commit 364ca4990807ed9a0f7e4f775a3ecfd35fcb7c99

Author: Shammo34 <kazishammo2020@gmail.com>

Date: Sun Jul 23 02:32:16 2023 +0600

Added more htmls

commit 67eb5d540b0d50b933cfa3271446d59080a27fa3

Author: Shammo34 <kazishammo2020@gmail.com>

Date: Sat Jul 22 22:32:49 2023 +0600

Initial commit

**2.14 Normal operating system CLI for get file list**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ ls

about.html contact.html index.html monuments.html

**2.15 creating the waste.html and then untracked it from commit**

**(rm - -cached waste.html)**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ touch waste.html

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ ls

about.html contact.html index.html monuments.html waste.html

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git add -A

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: waste.html

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git commit -a -m "adding waste"

[main 64ab4cd] adding waste

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 waste.html

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

nothing to commit, working tree clean

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git rm --cached waste.html

rm 'waste.html'

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

deleted: waste.html

Untracked files:

(use "git add <file>..." to include in what will be committed)

waste.html

**2.16 creating the waste.html and then delete it permanently**

**(rm waste.html)**

**(\*\*\*\*\*To give command u have to pick the code into staged area otherwise command will not work if untracked\*\*\*)**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git add -A

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

nothing to commit, working tree clean

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git rm waste.html

rm 'waste.html'

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ ls

about.html contact.html index.html monuments.html

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

deleted: waste.html

**2.17 Status –s (summerise status) Not so important**

**1.About.html is being changed in working tree**

**2. About.html is being changed in staged area**

**3. About.html is being changed in working tree and staged area**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status -s

M about.html

M monuments.html

D waste.html

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git add about.html

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status -s

M about.html

M monuments.html

D waste.html

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status -s

MM about.html

M monuments.html

D waste.html

**3.1 Gitignore:**

**Suppose we don’t want to commit locked files,auto generated images in our projects so we use it**

**Otherwise push pull will be slow\*\*\*\***

**1st of all we have to create .gitignore and mylogs.log in terminal to test the ignore process**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ touch .gitignore

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

Untracked files:

(use "git add <file>..." to include in what will be committed)

.gitignore

nothing added to commit but untracked files present (use "git add" to track)

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ touch mylogs.log

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

Untracked files:

(use "git add <file>..." to include in what will be committed)

.gitignore

nothing added to commit but untracked files present (use "git add" to track)

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git add -A

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

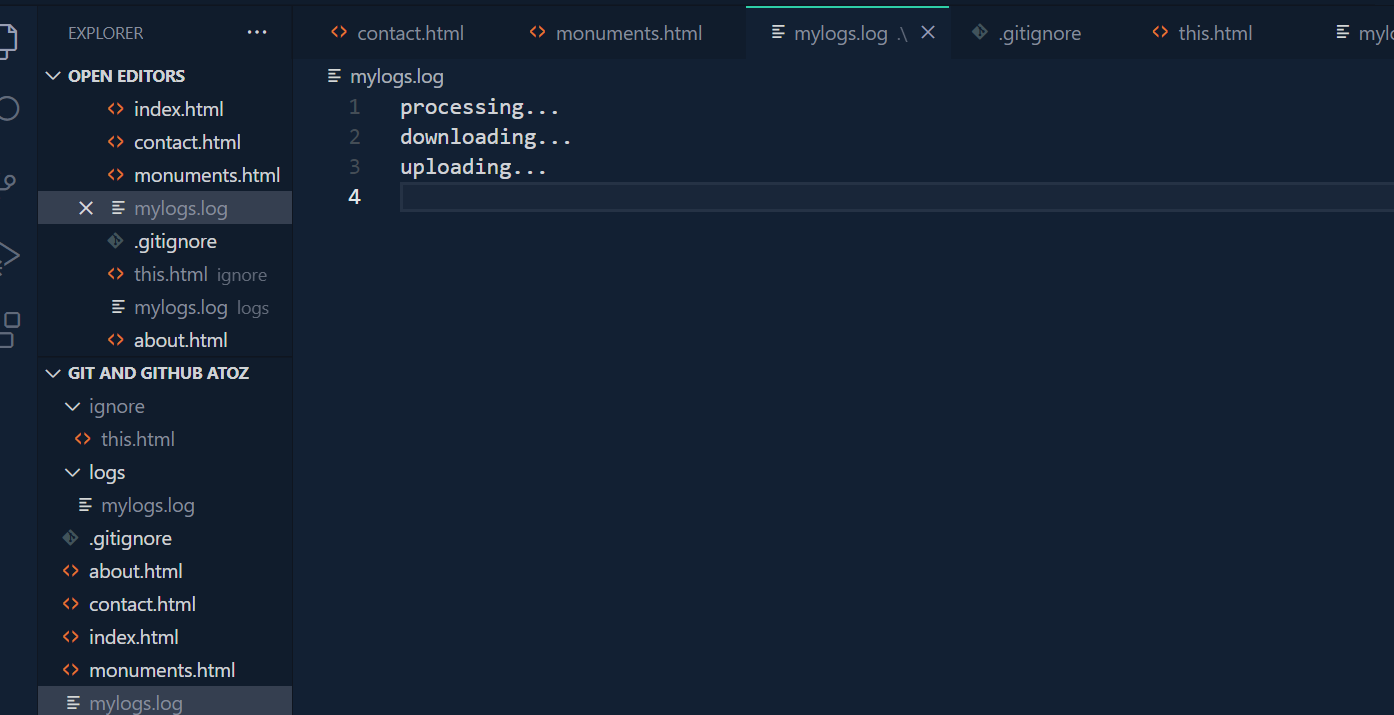
On branch main

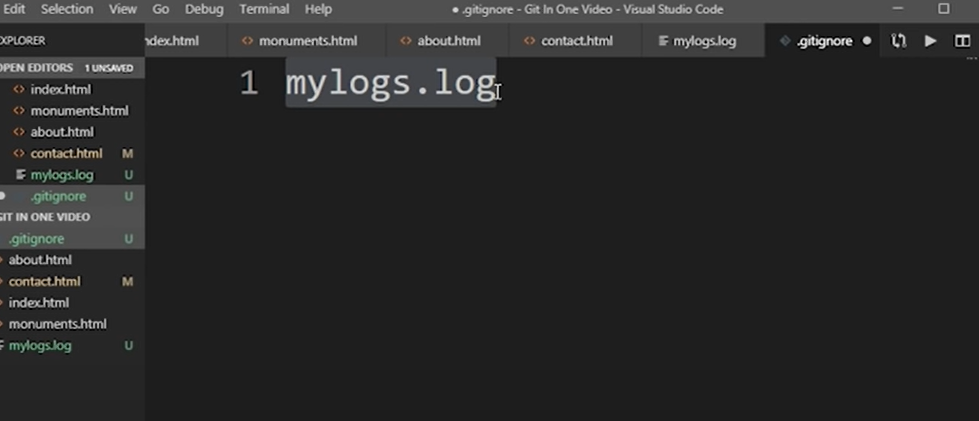
Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: .gitignore

**The mylogs.log file is ignored**

****

****

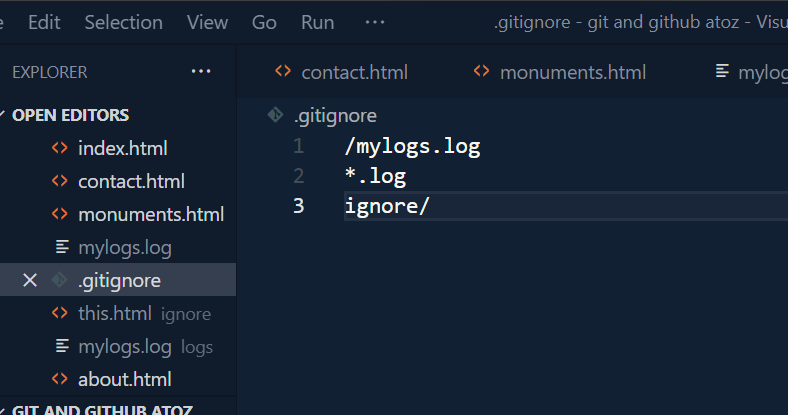
**1.To avoid the ignore command**

**Use” /” before**

**2.\*.log ignore all type of log files**

**3.ignore folder or directory**

**Use “/” after**

****

**WE have untracked the mylogs.log then again focus on new work**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: .gitignore

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: .gitignore

Untracked files:

(use "git add <file>..." to include in what will be committed)

logs/

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git add -A

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: .gitignore

new file: logs/mylogs.log

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git rm --cached logs/mylogs.log

rm 'logs/mylogs.log'

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: .gitignore

Untracked files:

(use "git add <file>..." to include in what will be committed)

logs/

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git add -A

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: .gitignore

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: .gitignore

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git add -A

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: .gitignore

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git commit -a -m "ignore directory"

[main 742b1fb] ignore directory

1 file changed, 3 insertions(+)

create mode 100644 .gitignore

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git log -p -2

commit 742b1fb98d37179cc87d02a43b7458942d67e39f (HEAD -> main)

Author: Shammo34 <kazishammo2020@gmail.com>

Date: Mon Jul 24 01:04:15 2023 +0600

ignore directory

**diff --git a/.gitignore b/.gitignore**

**new file mode 100644**

**index 0000000..444fcc3**

**--- /dev/null**

**+++ b/.gitignore**

@@ -0,0 +1,3 @@

+/mylogs.log

+\*.log

+ignore/

\ No newline at end of file

commit 62df3c4b3f1e4a20c4e1b8561d86fb2aab590ca6

Author: Shammo34 <kazishammo2020@gmail.com>

Date: Sun Jul 23 03:45:47 2023 +0600

dlt waste.html

**diff --git a/waste.html b/waste.html**

**deleted file mode 100644**

**index e69de29..0000000**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ touch .gitignore

**Braanches:**

**If I work for a big company and the website getting down for me for modyfing code…I should avoid this with the use of branch to copy the code and then work on it\*\***

**Ex: on branch main 🡪 means by default main branch**

**3.2 Branch ADDED,VIEW,DELETE**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git branch freature1

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git branch

freature1

\* main

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git checkout feature1

error: pathspec 'feature1' did not match any file(s) known to git

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git rm branch freature1

fatal: pathspec 'branch' did not match any files

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git branch

freature1

\* main

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git branch -d freature1

Deleted branch freature1 (was 742b1fb).

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git branch

\* main

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$

**3.3 Switched branch**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git branch feature1

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git branch

feature1

\* main

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git checkout feature1

Switched to branch 'feature1'

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (feature1)

$ git status

On branch feature1

nothing to commit, working tree clean

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (feature1)

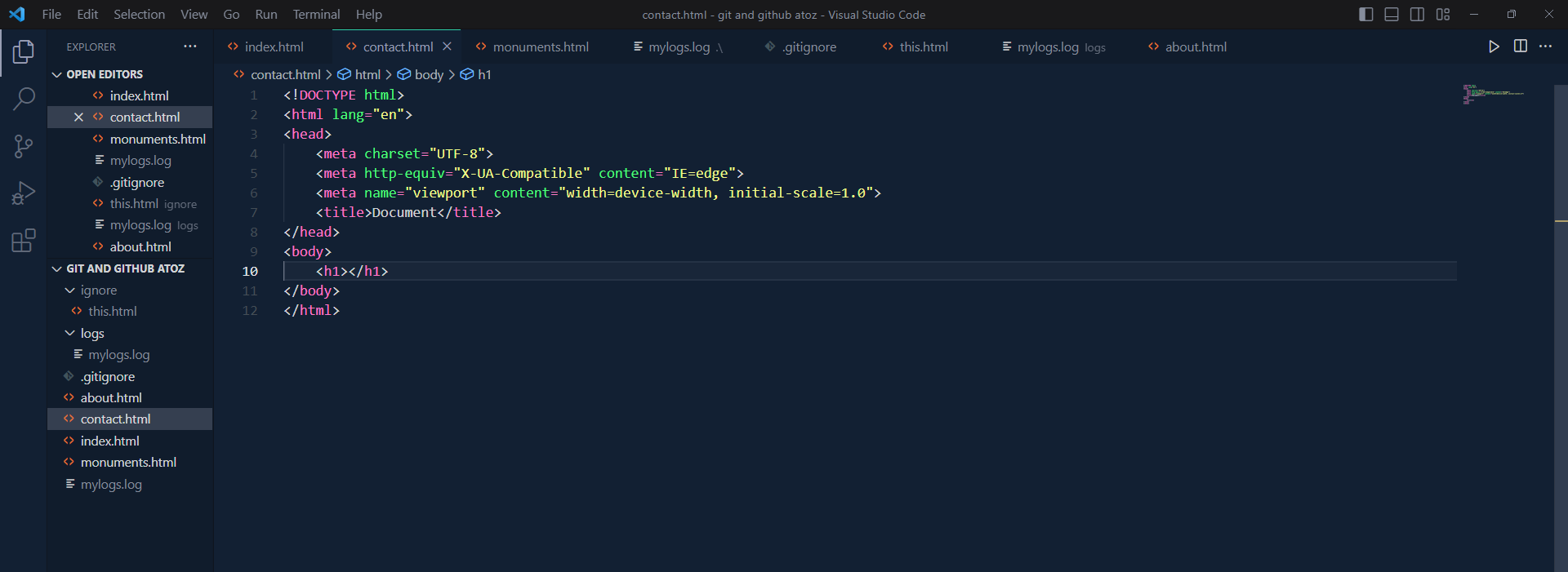
$ git branch

\* feature1

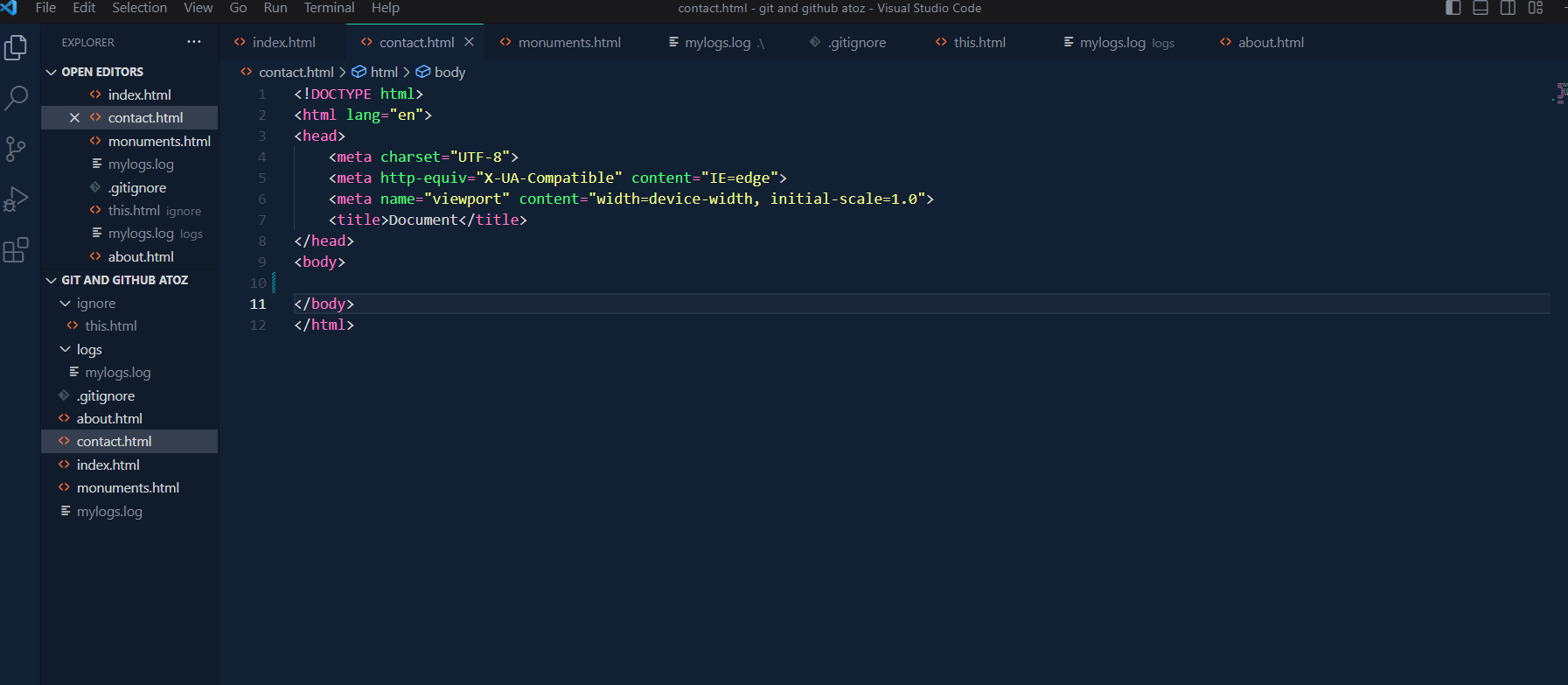
main

**After switching change the code in other branch and then back to main**

**Feature1 Branch:**

****

**Main:**

****

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git branch feature1

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git branch

feature1

\* main

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git checkout feature1

Switched to branch 'feature1'

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (feature1)

$ git status

On branch feature1

nothing to commit, working tree clean

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (feature1)

$ git branch

\* feature1

main

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (feature1)

$ git status

On branch feature1

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: contact.html

no changes added to commit (use "git add" and/or "git commit -a")

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (feature1)

$ git add -A

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (feature1)

$ git commit -m "Added h1 tag"

[feature1 389504c] Added h1 tag

1 file changed, 1 insertion(+), 1 deletion(-)

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (feature1)

$ git status

On branch feature1

nothing to commit, working tree clean

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (feature1)

$ git checkout main

Switched to branch 'main'

**3.4 Create branch and switched in 1 line:**

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git checkout -b feature2

Switched to a new branch 'feature2'

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (feature2)

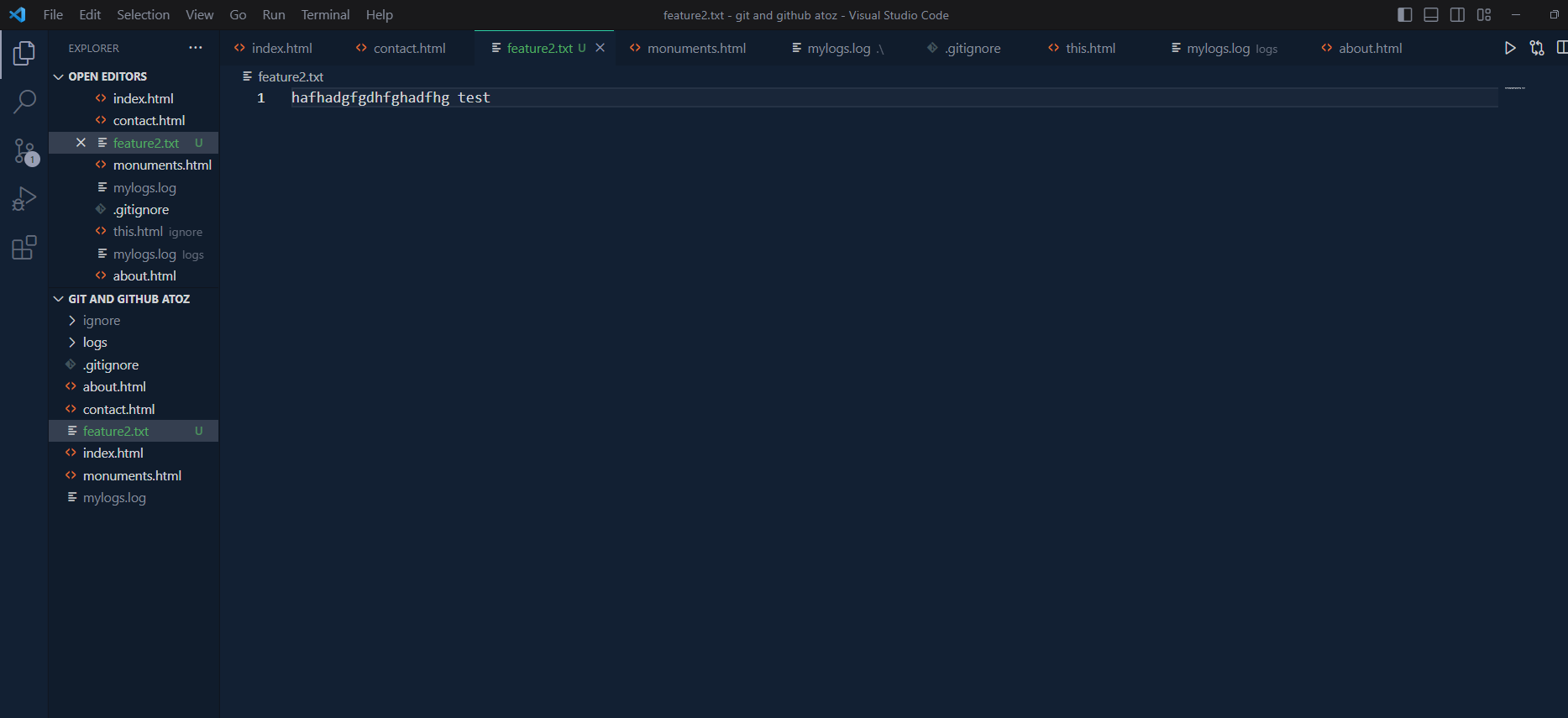
$ git branch

feature1

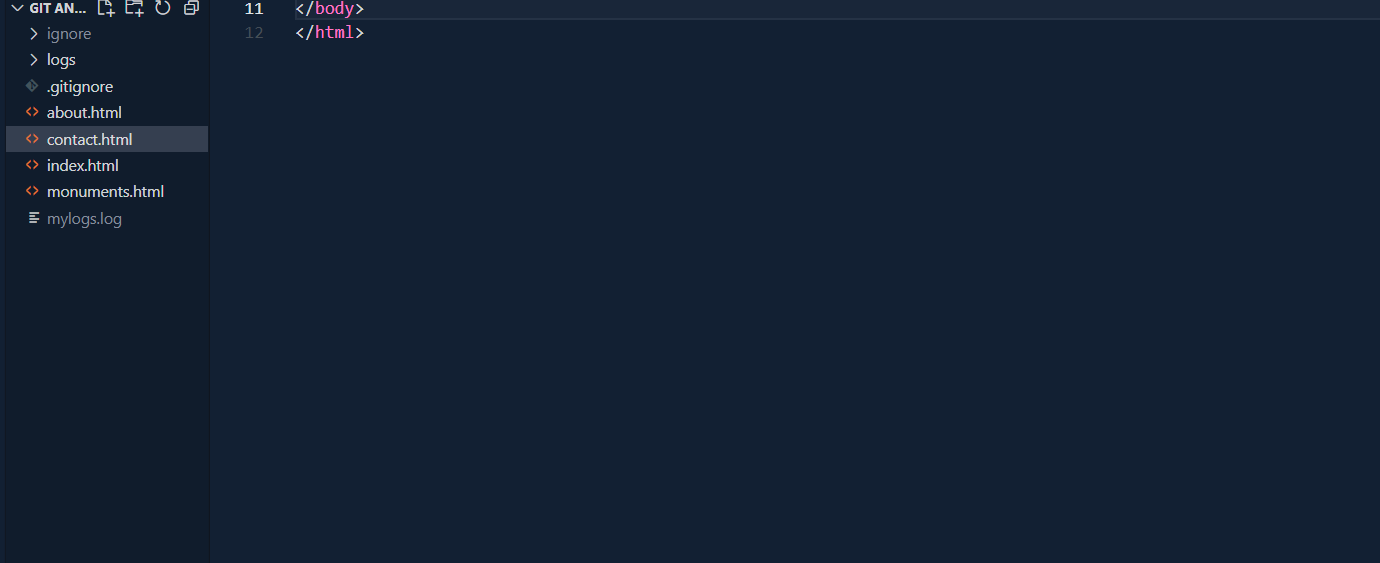
\* feature2

main

Feature2:

****

Main:no text file in this branch



Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (feature2)

$ git commit -a -m "Added text"

On branch feature2

Untracked files:

(use "git add <file>..." to include in what will be committed)

feature2.txt

nothing added to commit but untracked files present (use "git add" to track)

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (feature2)

$ git add -A

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (feature2)

$ git commit -m "Added text"

[feature2 ef2e46e] Added text

1 file changed, 1 insertion(+)

create mode 100644 feature2.txt

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (feature2)

$ git log -p -2

commit ef2e46ebb75d9a10cd509fa6f8e4984826ca1e27 (HEAD -> feature2)

Author: Shammo34 <kazishammo2020@gmail.com>

Date: Mon Jul 24 20:23:39 2023 +0600

Added text

**diff --git a/feature2.txt b/feature2.txt**

**new file mode 100644**

**index 0000000..f032185**

**--- /dev/null**

**+++ b/feature2.txt**

@@ -0,0 +1 @@

+hafhadgfgdhfghadfhg test

\ No newline at end of file

commit 742b1fb98d37179cc87d02a43b7458942d67e39f (main)

Author: Shammo34 <kazishammo2020@gmail.com>

Date: Mon Jul 24 01:04:15 2023 +0600

ignore directory

**diff --git a/.gitignore b/.gitignore**

**new file mode 100644**

**index 0000000..444fcc3**

**--- /dev/null**

**+++ b/.gitignore**

@@ -0,0 +1,3 @@

+/mylogs.log

+\*.log

+ignore/

\ No newline at end of file

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (feature2)

$ git checkout main

Switched to branch 'main'

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git branch

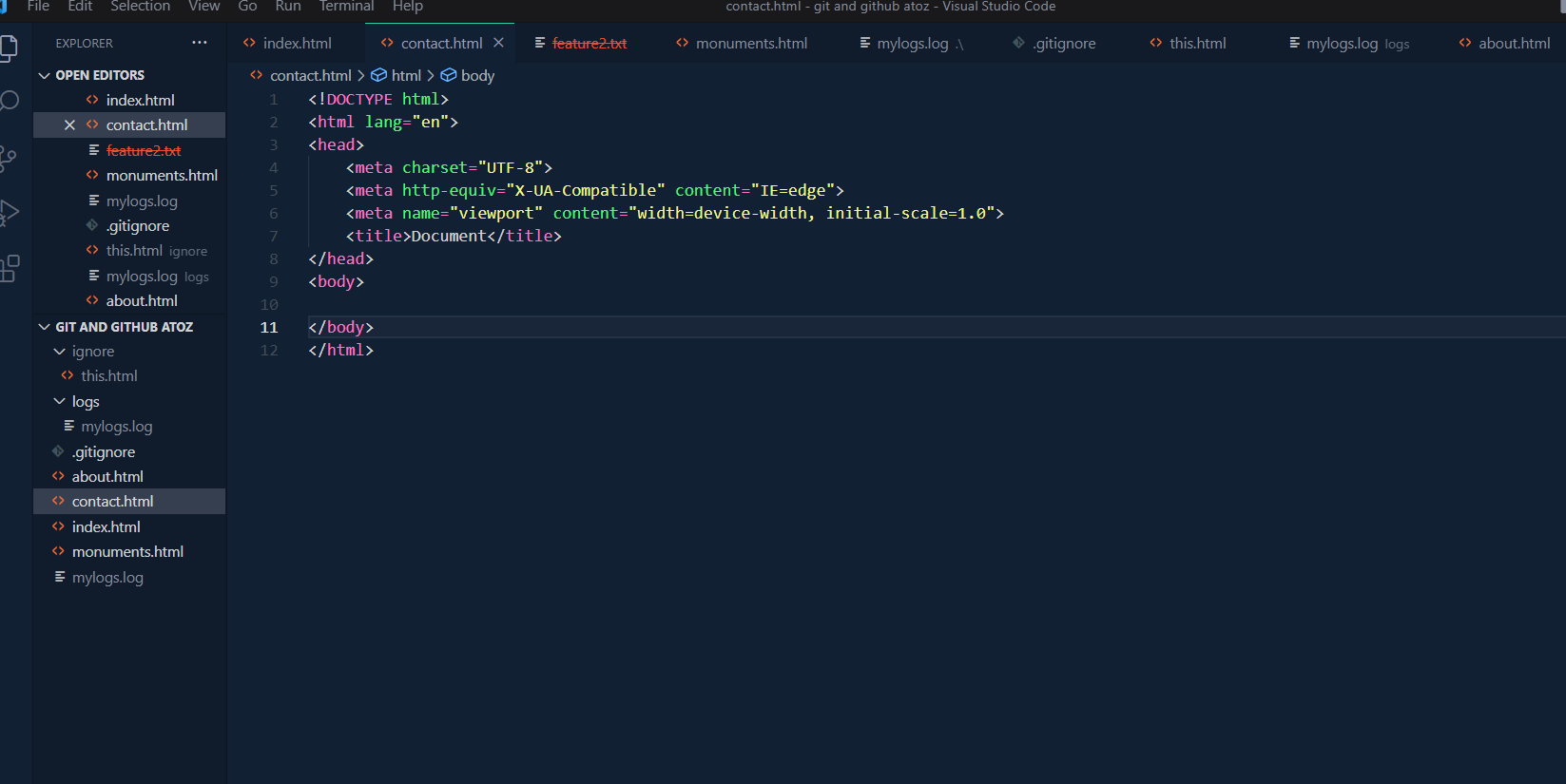
feature1

feature2

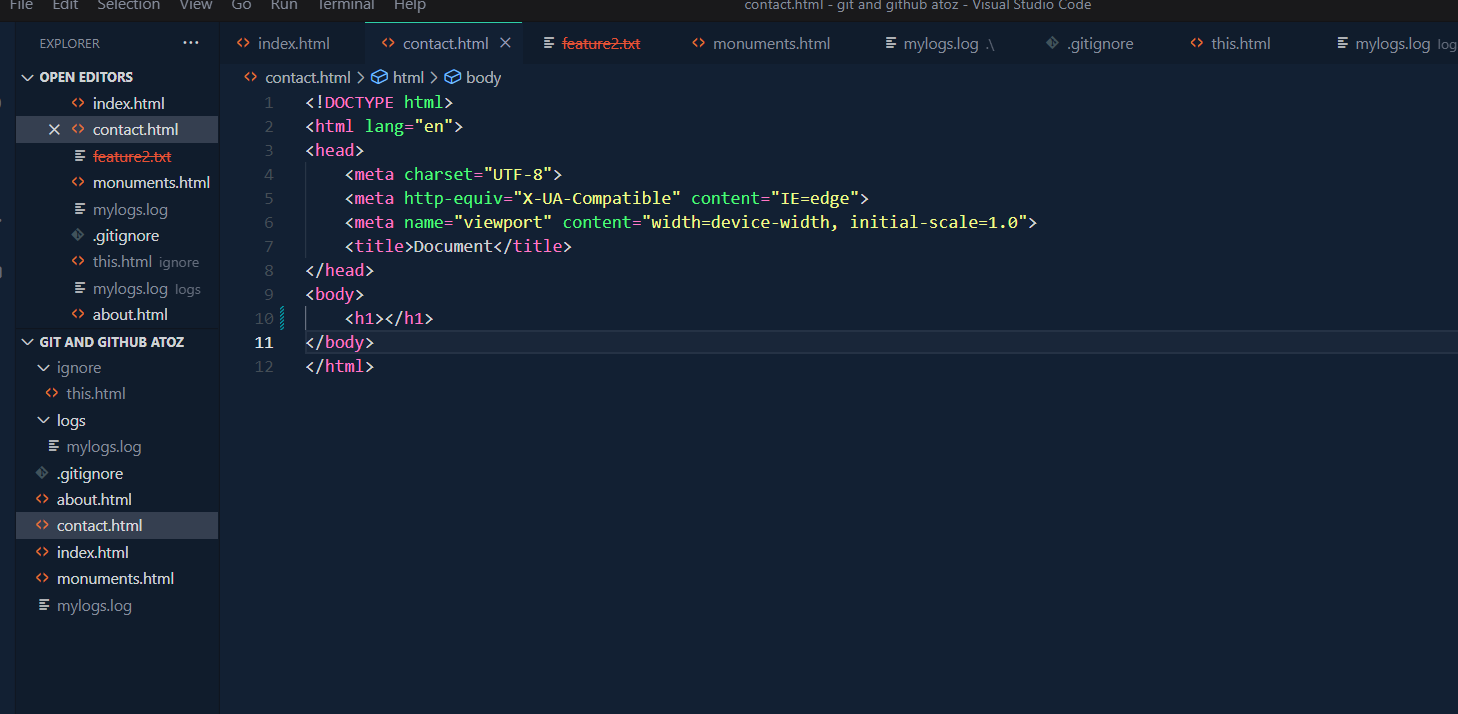
\* main

**3.5 Git merge to insert the modified code of branch in main branch**

**Before merge:**



**After merge: h1 tag added**



Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git merge feature1

Updating 742b1fb..389504c

Fast-forward

contact.html | 2 +-

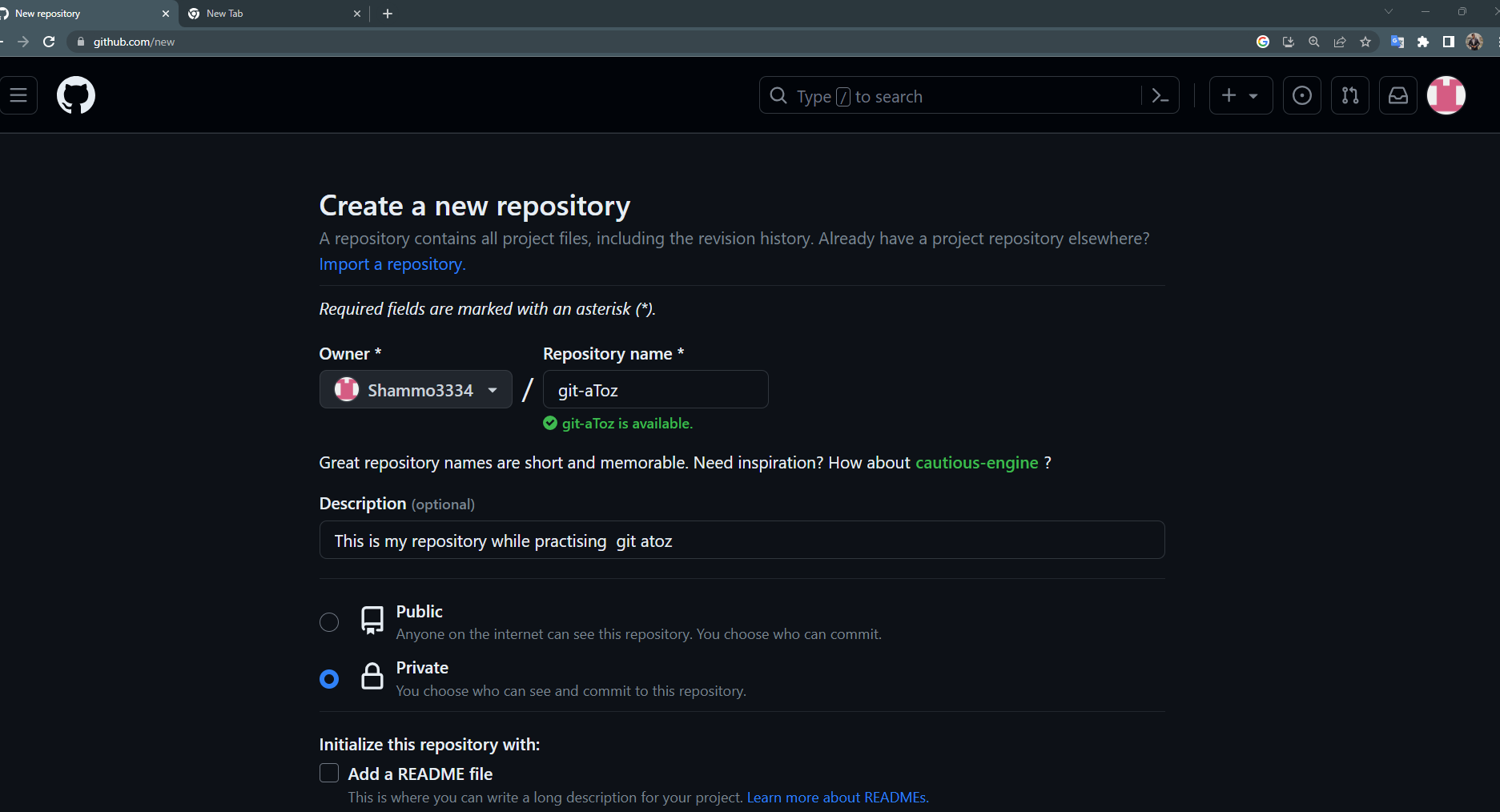
1 file changed, 1 insertion(+), 1 deletion(-)

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

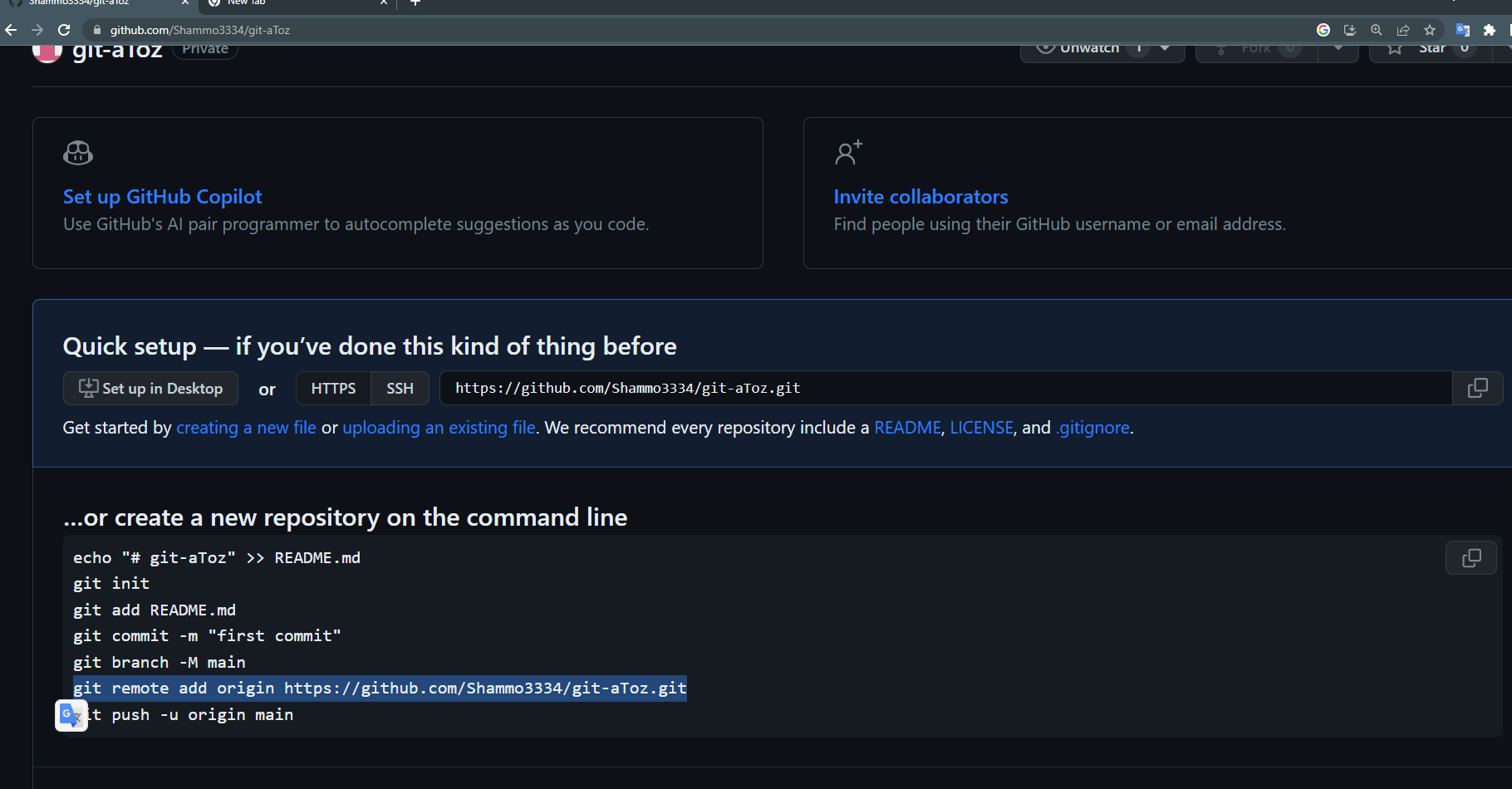
$

**Github:**

**4.1 Creating a private repository:**

****

4.2 URL to host repository ,To connect local repository to remote repository



4.3 Set remote as origin (name) for short name of URL

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git status

On branch main

nothing to commit, working tree clean

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git remote add origin https://github.com/Shammo3334/git-aToz.git

4.4 Check git remote and remote url

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git remote

origin

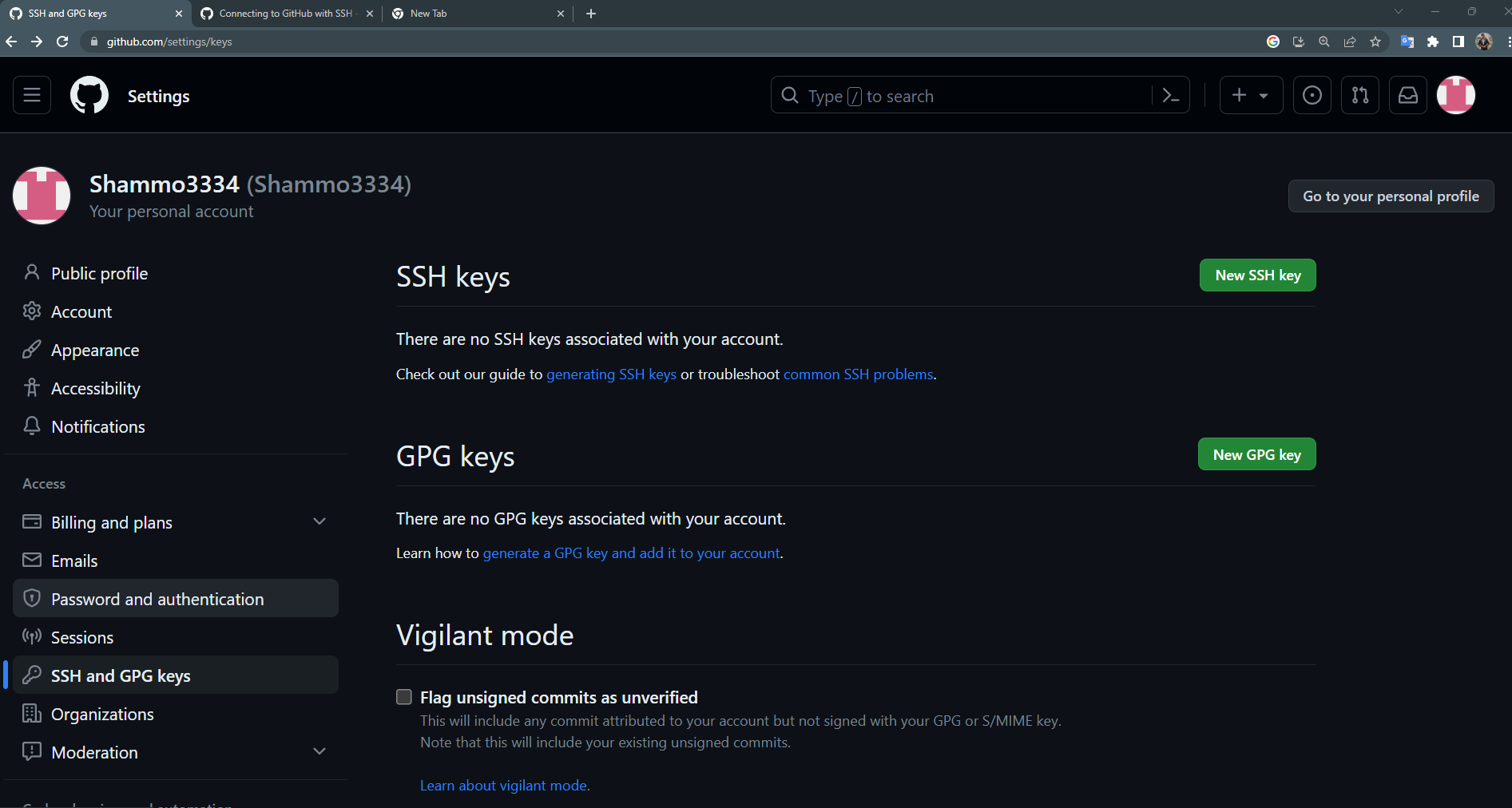
Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git remote -v

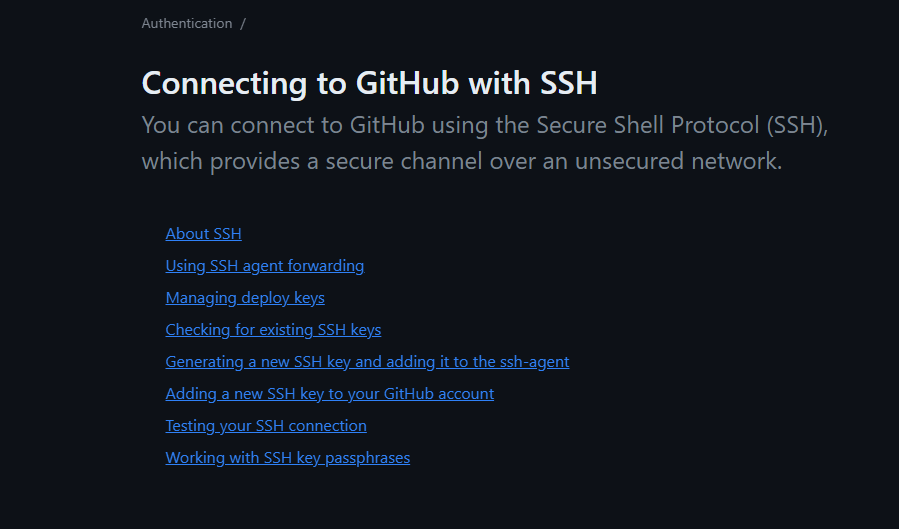
origin https://github.com/Shammo3334/git-aToz.git (fetch)

origin https://github.com/Shammo3334/git-aToz.git (push)

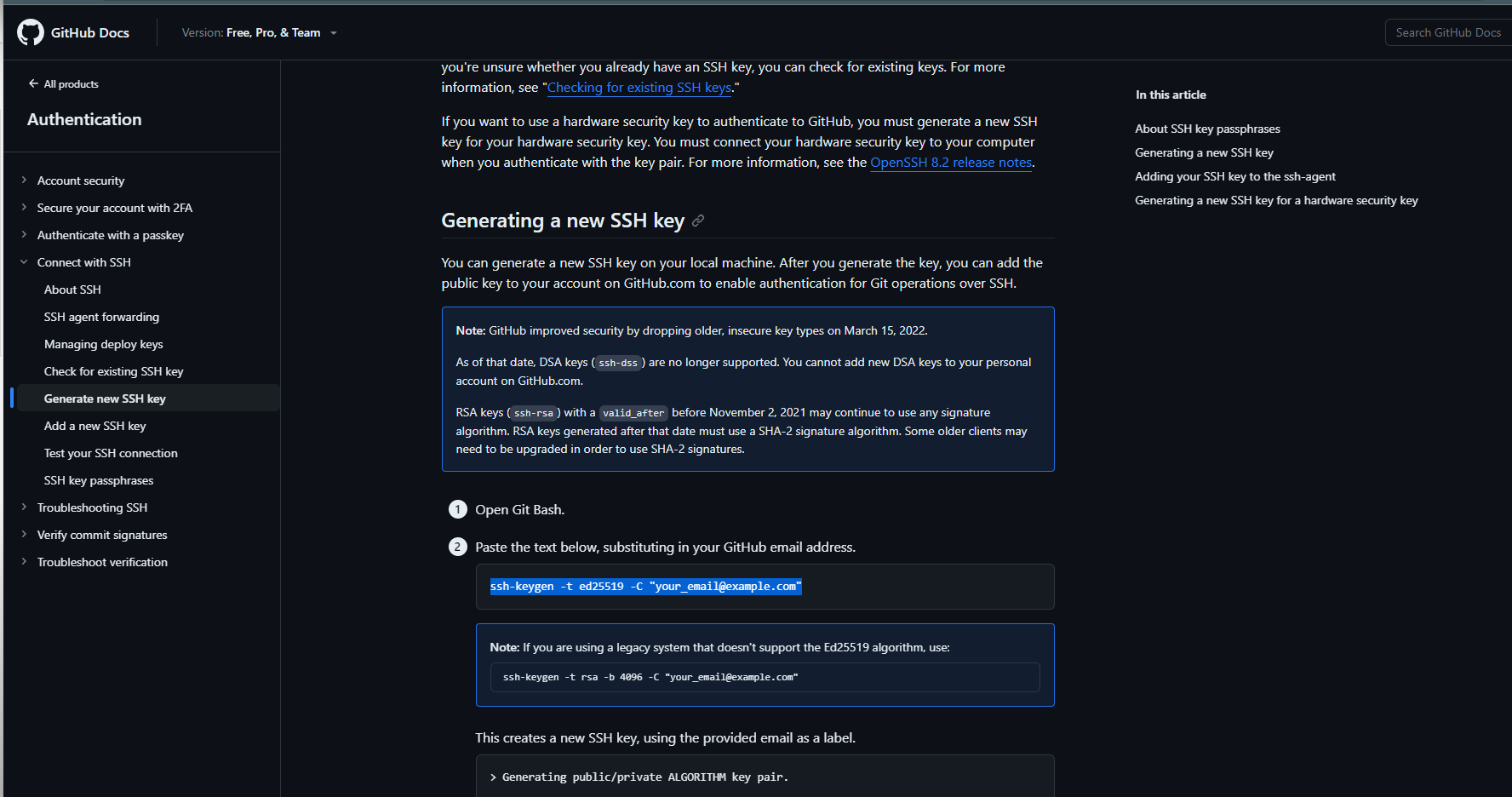
4.5 go to setting and SSH and GPG key for give access my github account of my pc: We have to deploy it



Step1



Step2

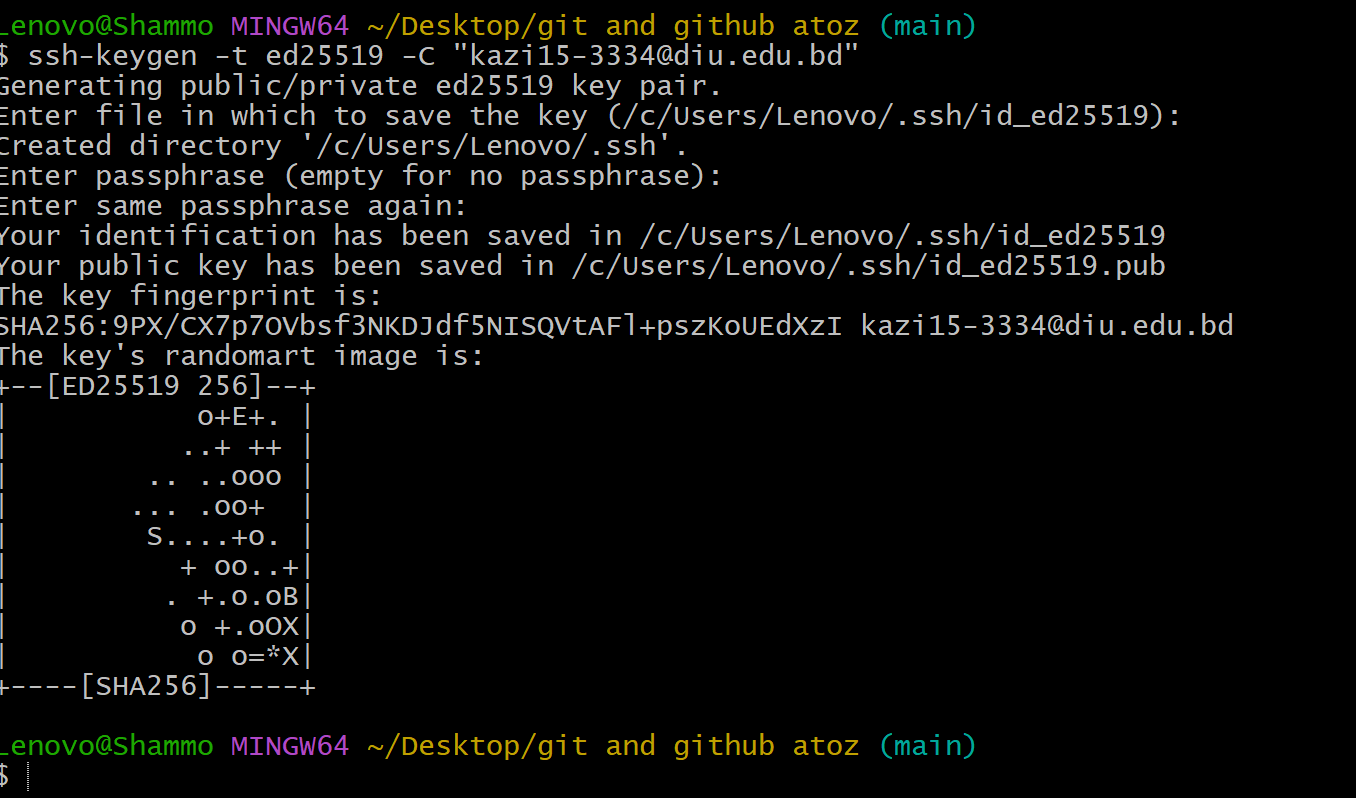


Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

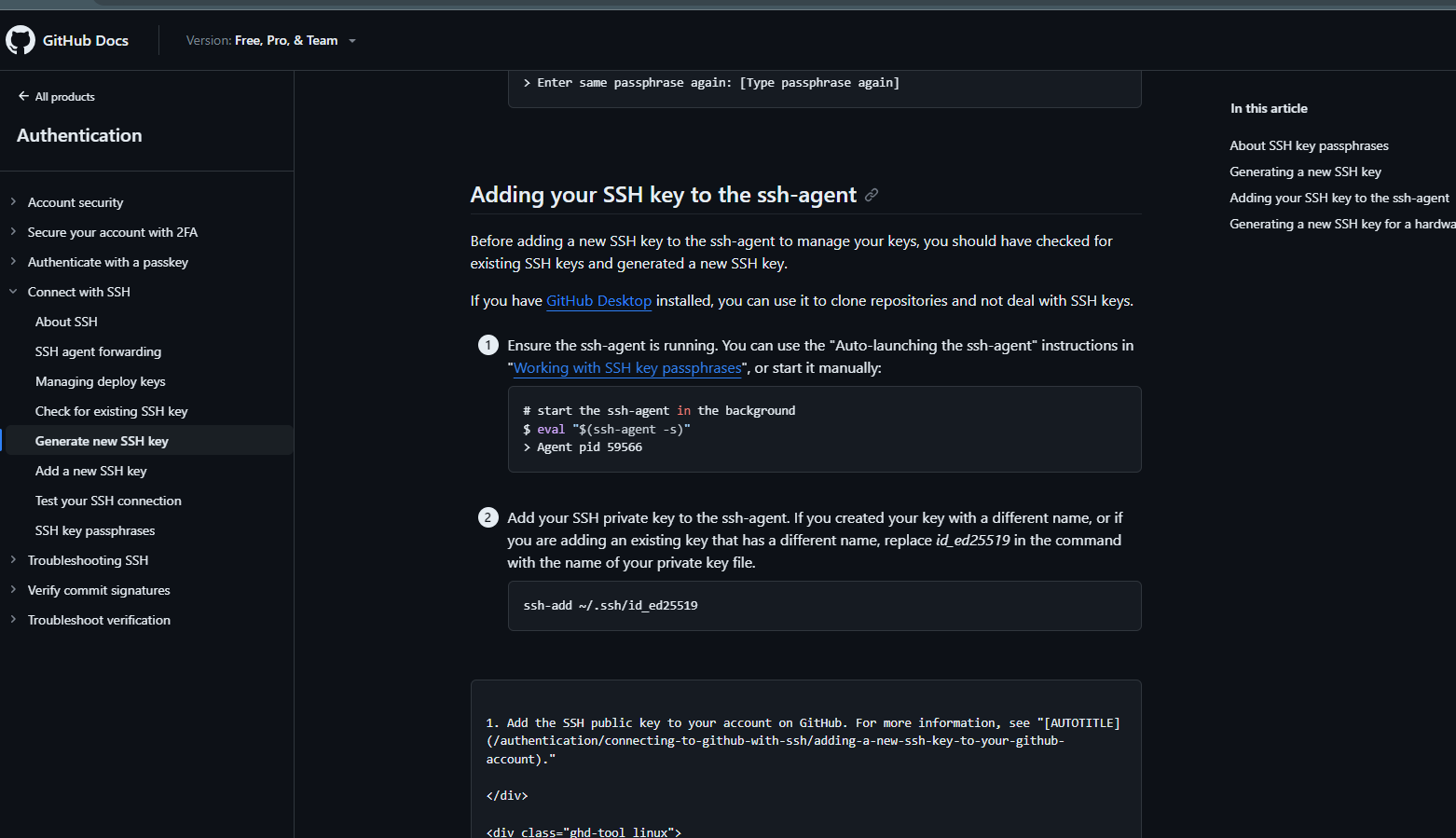
$ ssh-keygen -t ed25519 -C "kazi15-3334@diu.edu.bd"

Generating public/private ed25519 key pair.

Enter file in which to save the key (/c/Users/Lenovo/.ssh/id\_ed25519):

Step4: After pressing enter ,enter we get public private key which we have to deploy

Step5:Run eval (We will get agent pid)

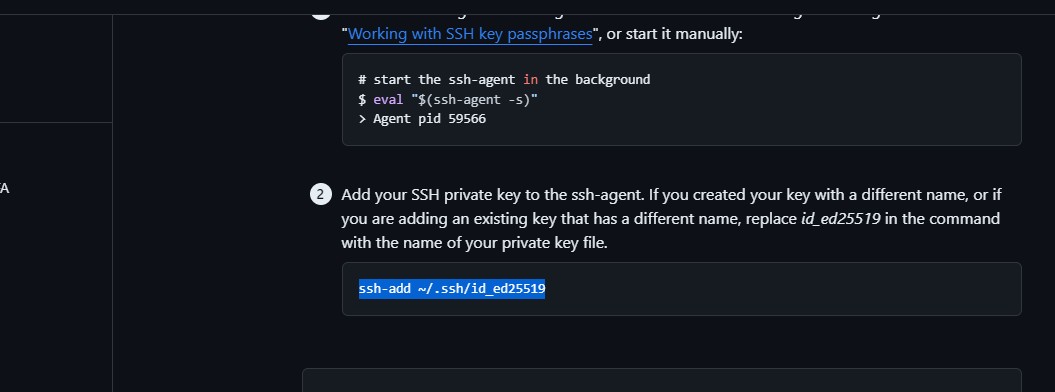


Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ eval $(ssh-agent -s)

Agent pid 1616

Step6:ssh-add



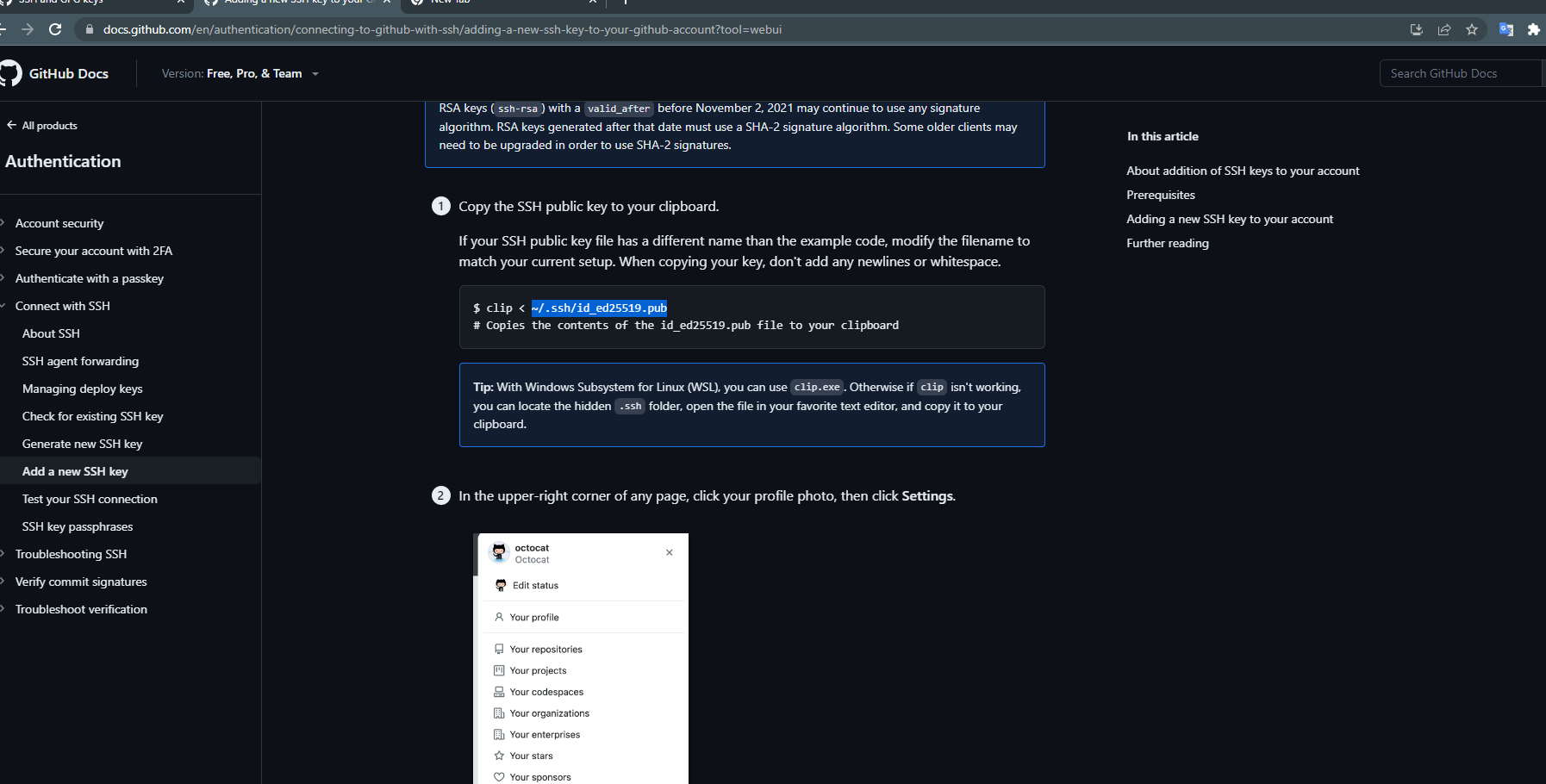
Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ ssh-add ~/.ssh/id\_ed25519

Identity added: /c/Users/Lenovo/.ssh/id\_ed25519 (kazi15-3334@diu.edu.bd)

Step7:go to add a new ssh key and copy the public key to run

($ cat ~/.ssh/id\_ed25519.pub shows what content I have)

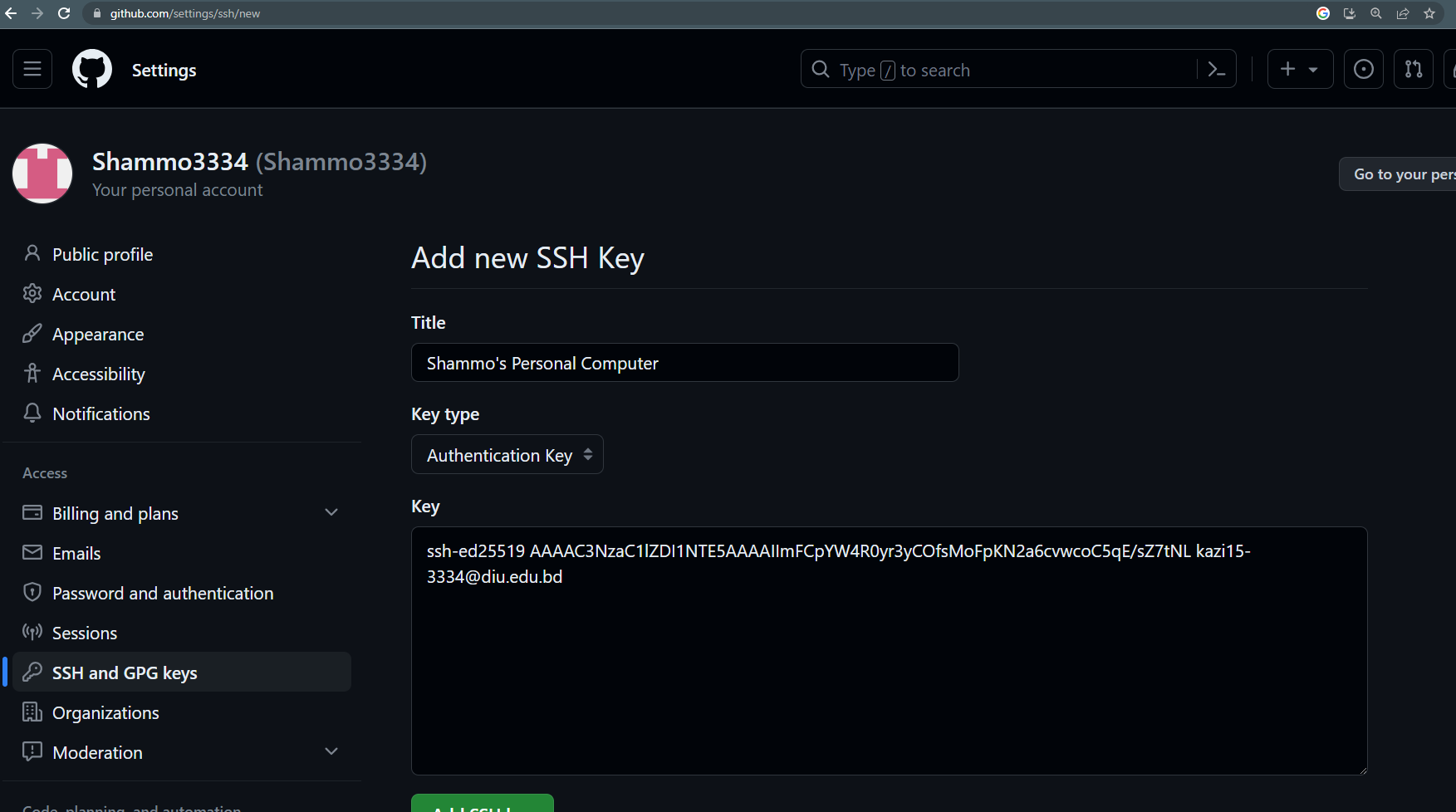


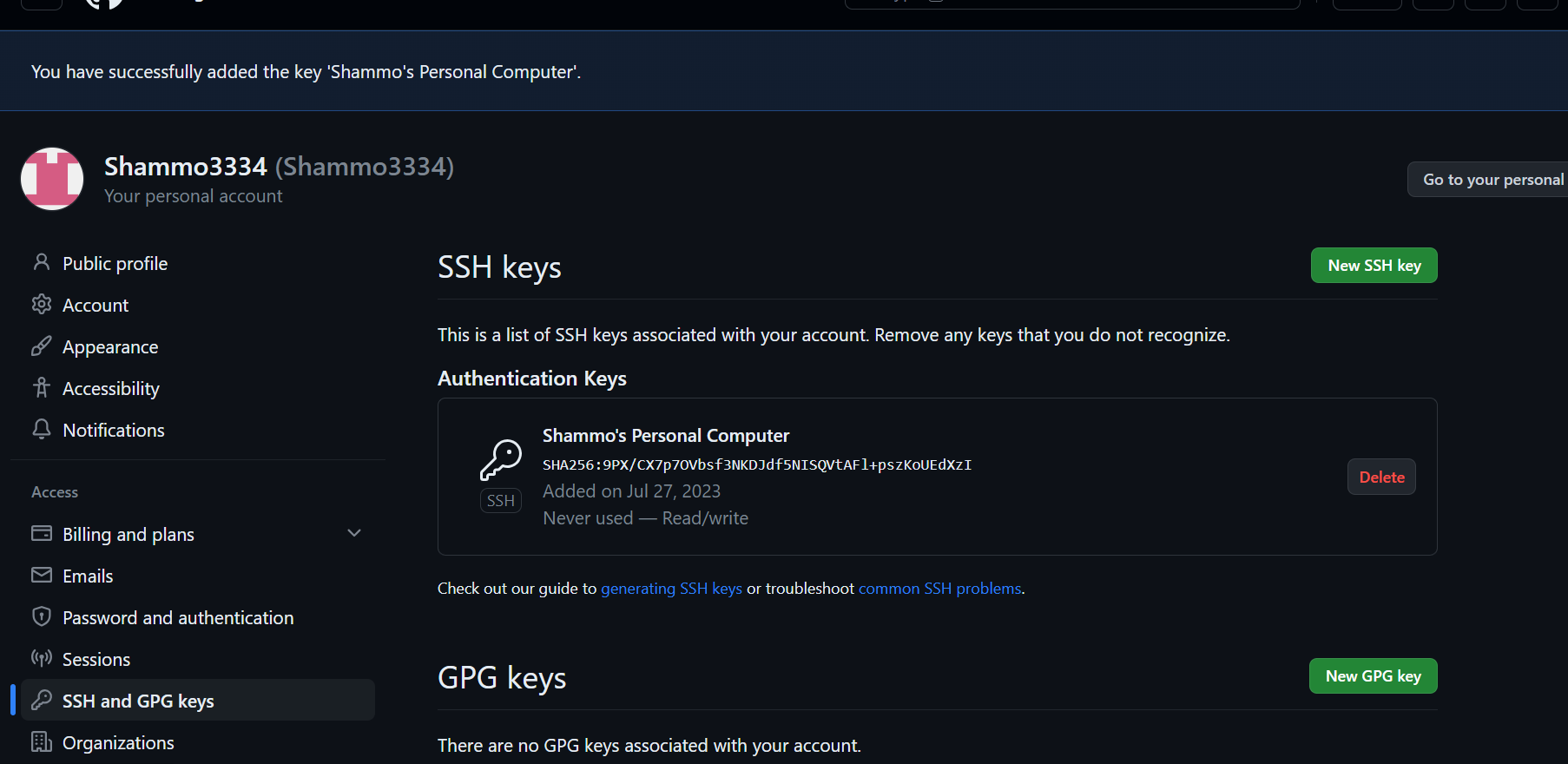
Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ cat ~/.ssh/id\_ed25519.pub

ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIImFCpYW4R0yr3yCOfsMoFpKN2a6cvwcoC5qE/sZ7tNL kazi15-3334@diu.edu.bd

Step8: copy the content and paste to add new SSH key and give title





Step 9:

Push branches:

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git push -u origin main

Everything up-to-date

branch 'main' set up to track 'origin/main'.

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (main)

$ git checkout feature1

Switched to branch 'feature1'

Lenovo@Shammo MINGW64 ~/Desktop/git and github atoz (feature1)

$ git push -u origin feature1

Total 0 (delta 0), reused 0 (delta 0), pack-reused 0

remote:

remote: Create a pull request for 'feature1' on GitHub by visiting:

remote: https://github.com/Shammo3334/git-aToz/pull/new/feature1

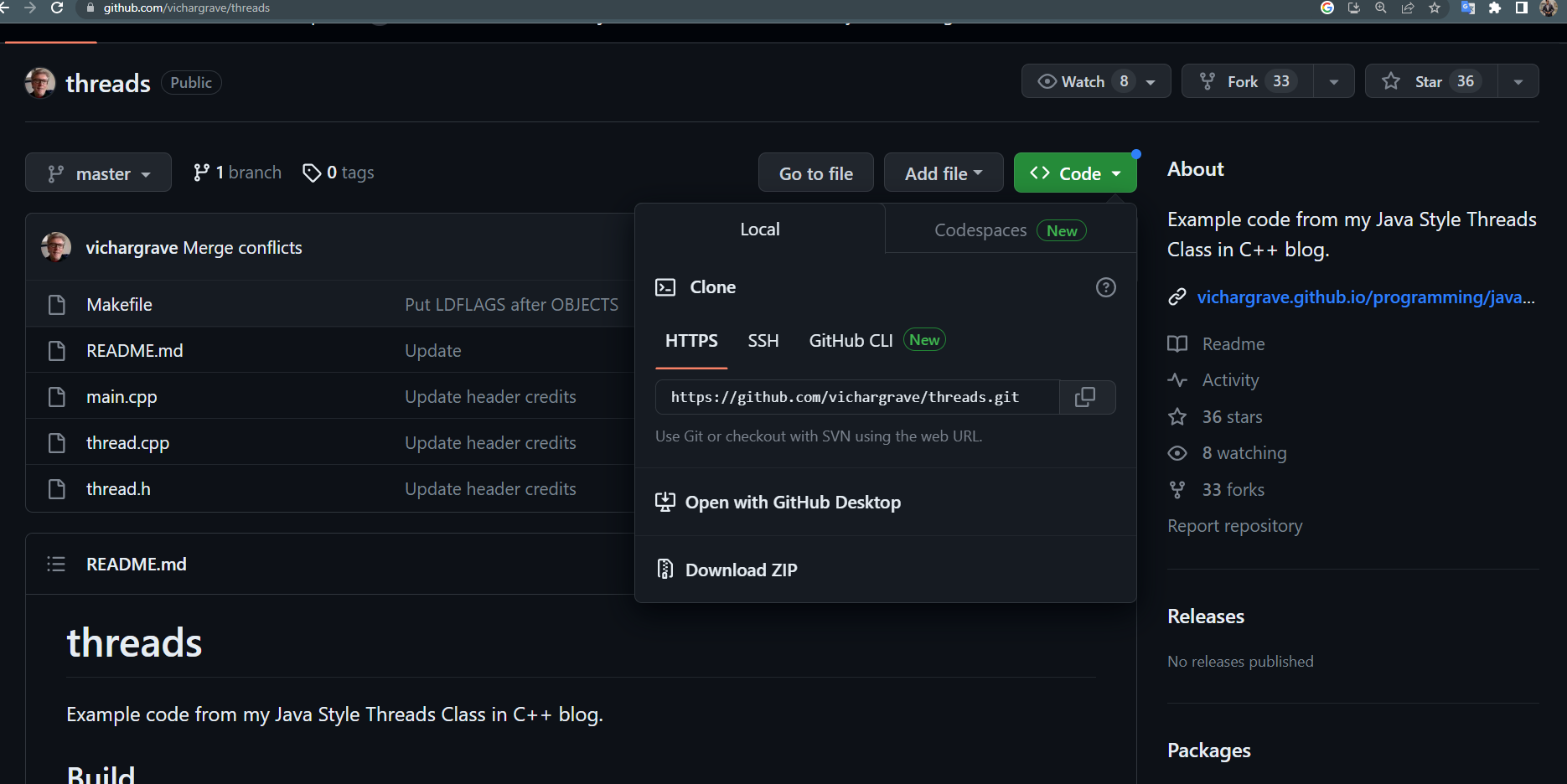
remote:

To https://github.com/Shammo3334/git-aToz.git

\* [new branch] feature1 -> feature1

branch 'feature1' set up to track 'origin/feature1'.

5.1 Git clone:



**\*\*\*Shammo is a folder name\*\*\***

Lenovo@Shammo MINGW64 ~/Desktop/clone

$ git clone https://github.com/vichargrave/threads.git shammo

Cloning into 'shammo'...

remote: Enumerating objects: 76, done.

remote: Total 76 (delta 0), reused 0 (delta 0), pack-reused 76

Receiving objects: 100% (76/76), 8.96 KiB | 4.48 MiB/s, done.

Resolving deltas: 100% (37/37), done.

Lenovo@Shammo MINGW64 ~/Desktop/clone

$ ls

shammo/

Lenovo@Shammo MINGW64 ~/Desktop/clone

$