1. **Git – HOL**

**Objectives:**

* Familiarize yourself with Git commands like git init, git status, git add, git commit, git push, and git pull.
* Set up your machine with Git Configuration.
* Integrate notepad++.exe to Git and make it a default editor.
* Add a file to a source code repository.

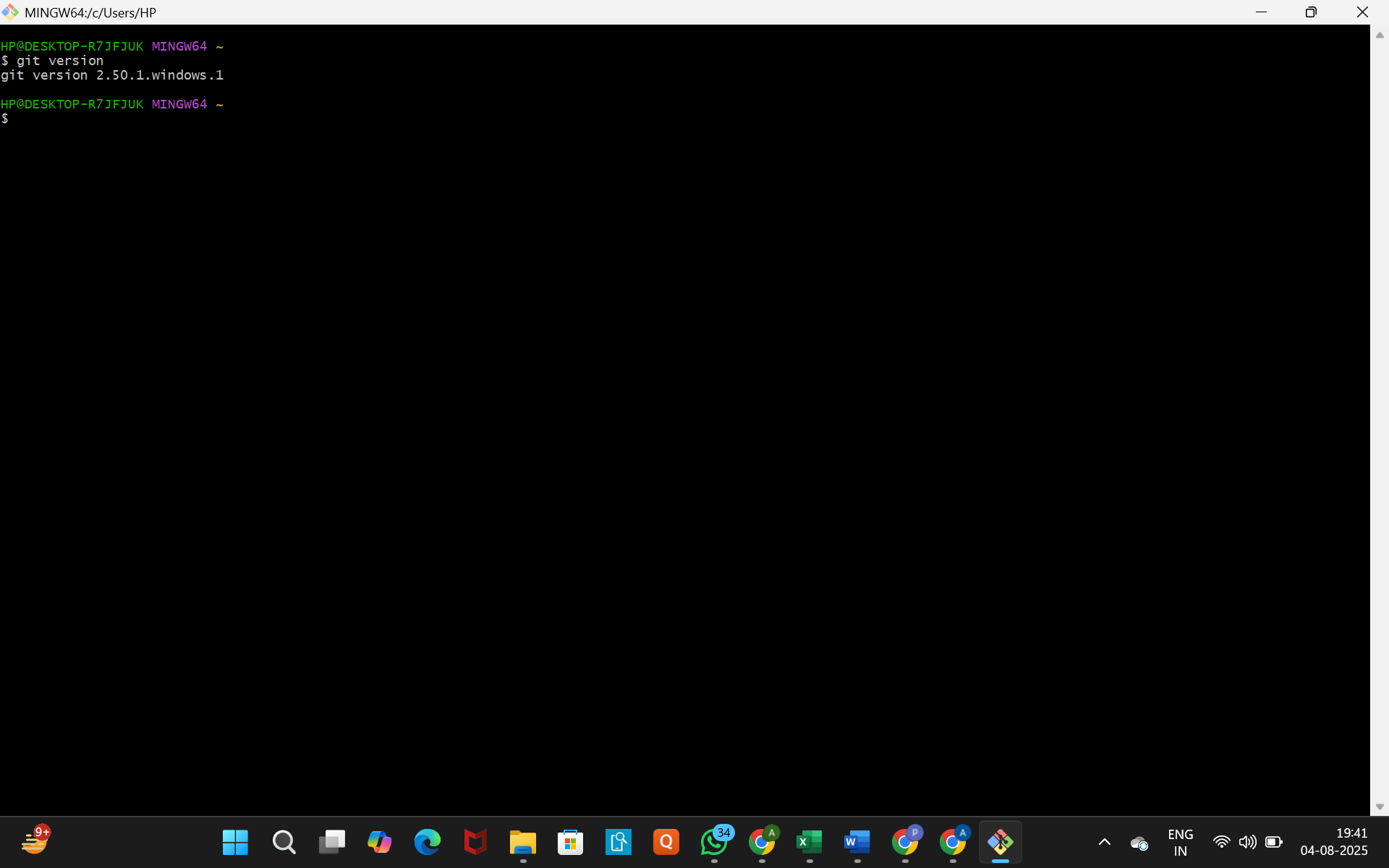
**Prerequisites:**

* Install Git Bash client on your machine.

**Step 1: Setup your machine with Git Configuration**

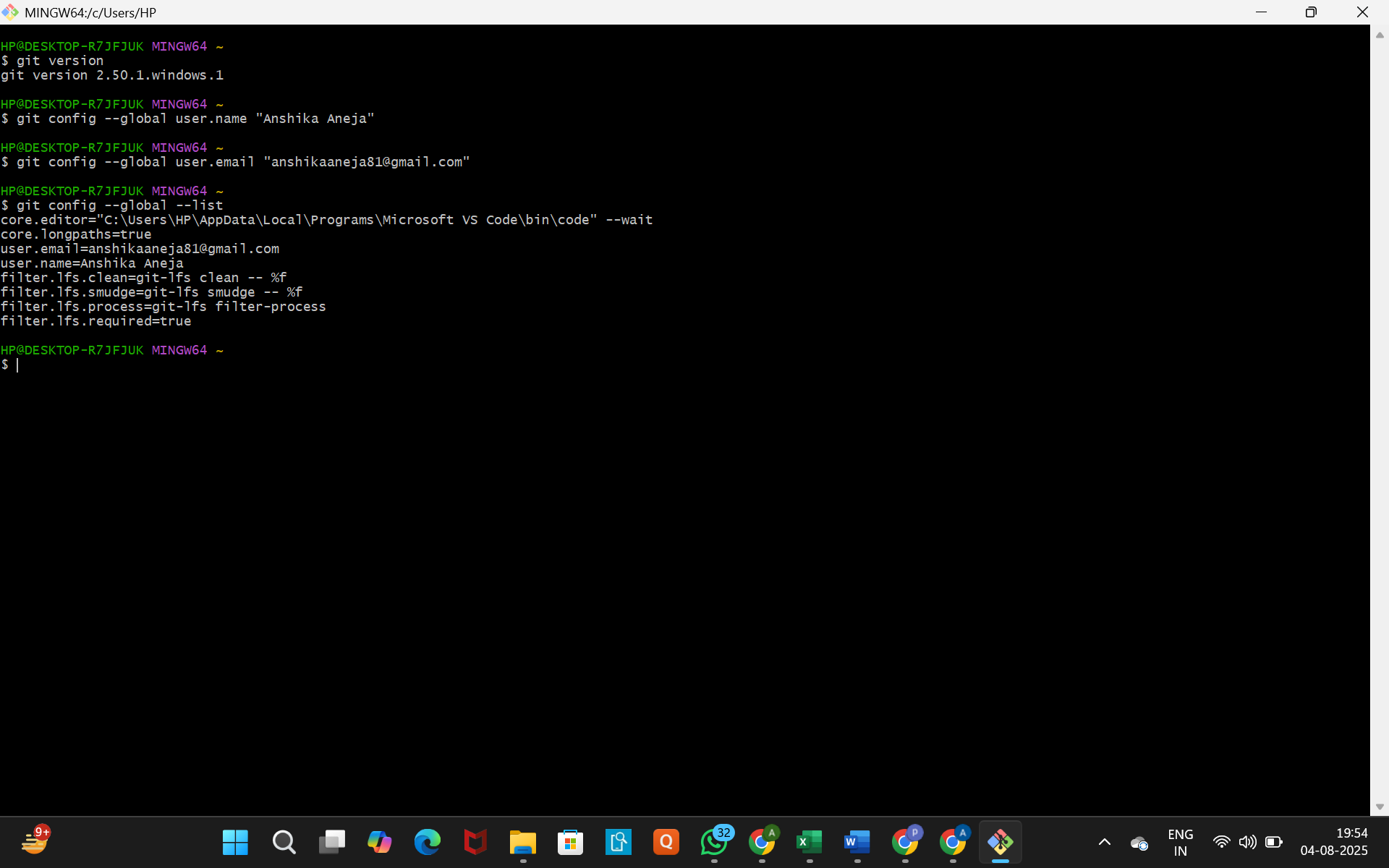
To create a new repository, sign up with GitLab and register your credentials. Login to GitLab and create a "GitDemo" project.

1. **To check if Git client is installed properly:**
   * Open Git Bash shell and execute the command:

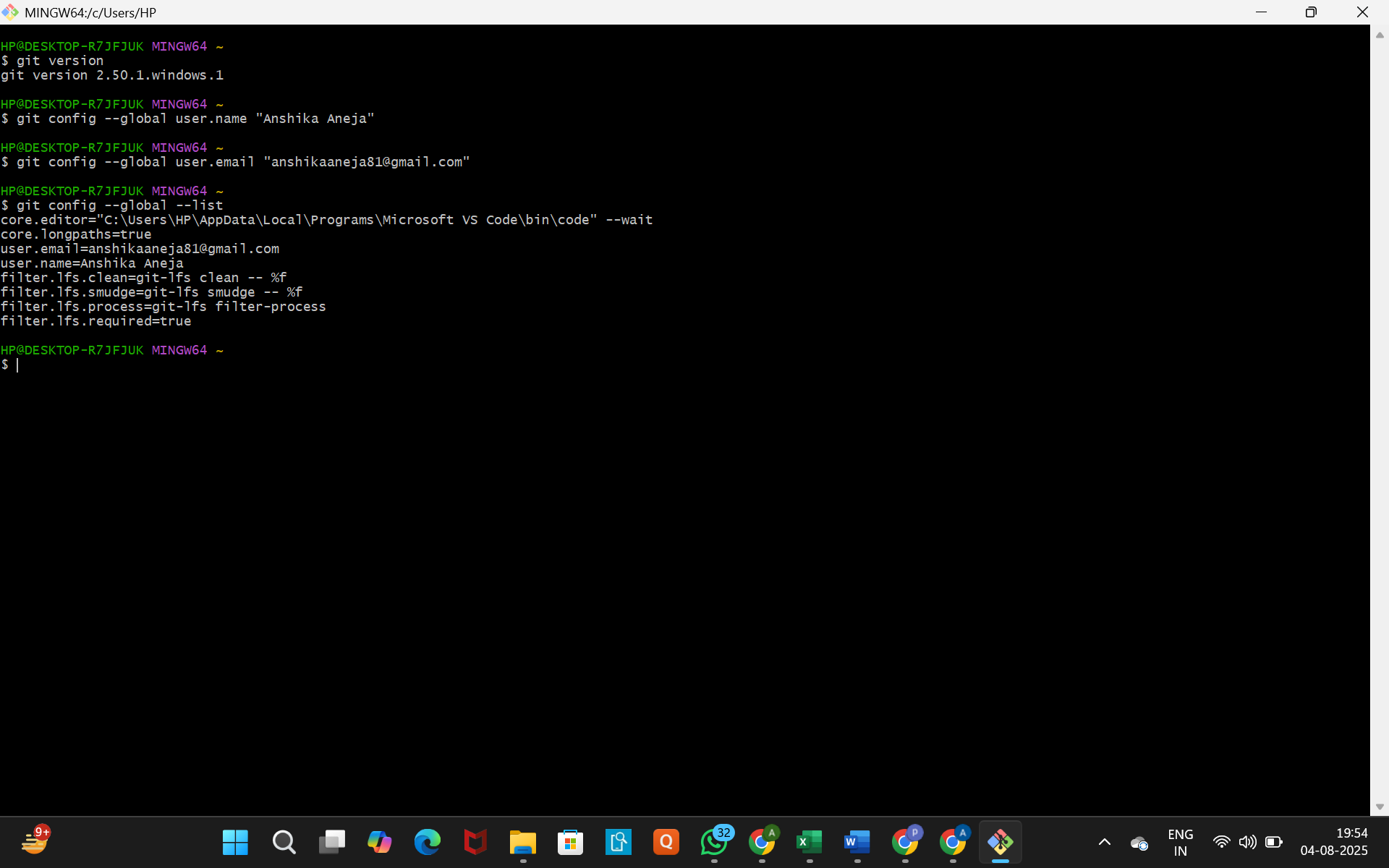


* + The output shows Git with its version information, it indicates that Git Client is installed properly.

1. **To configure user-level configuration of user ID and email ID:**
   * Execute the following commands, replacing "username" and "username@cognizant.com" with your actual Git username and email:



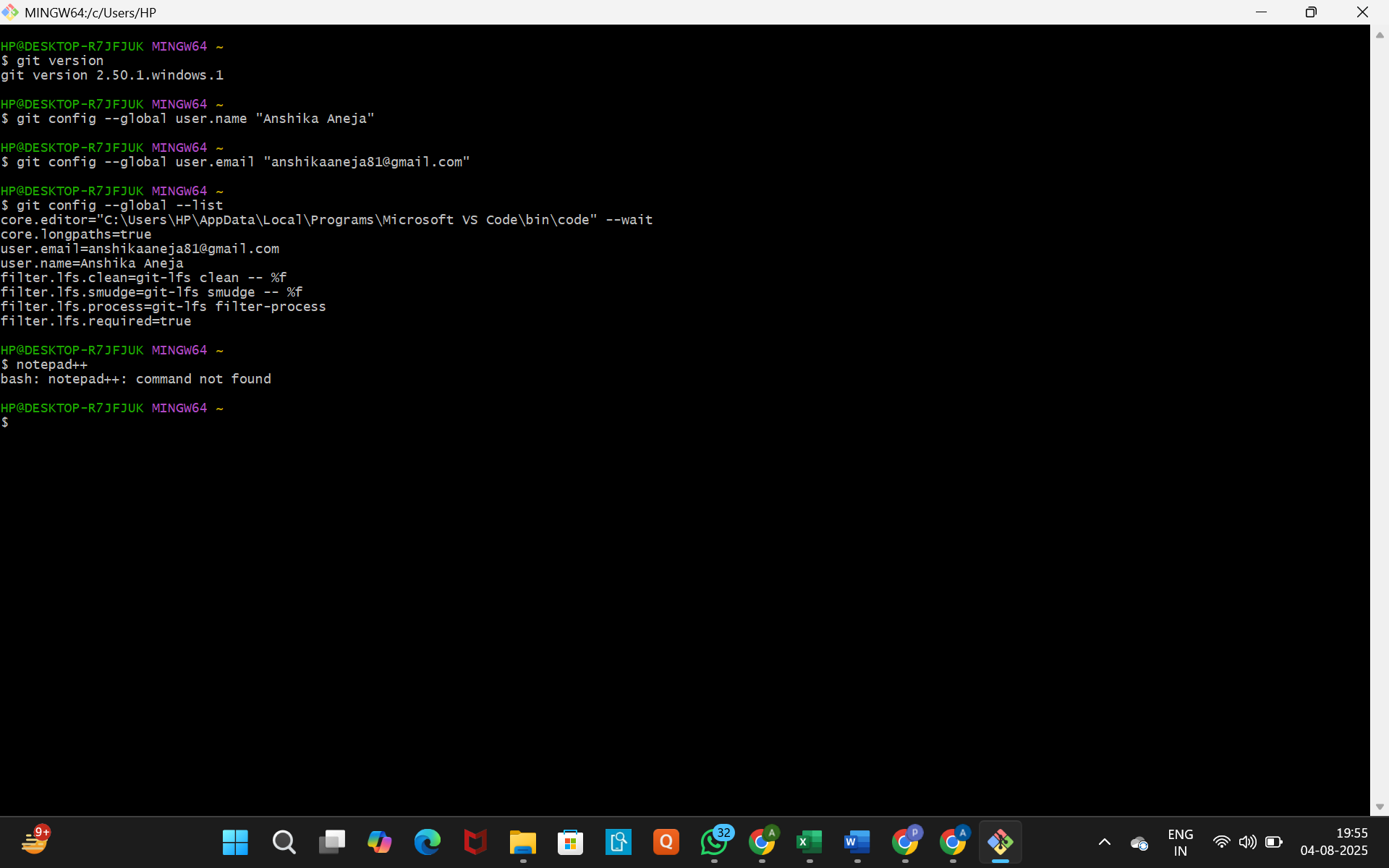
1. **To check if the configuration is properly set:**
   * Execute the following command:



* + You will see your user.name and user.email listed.

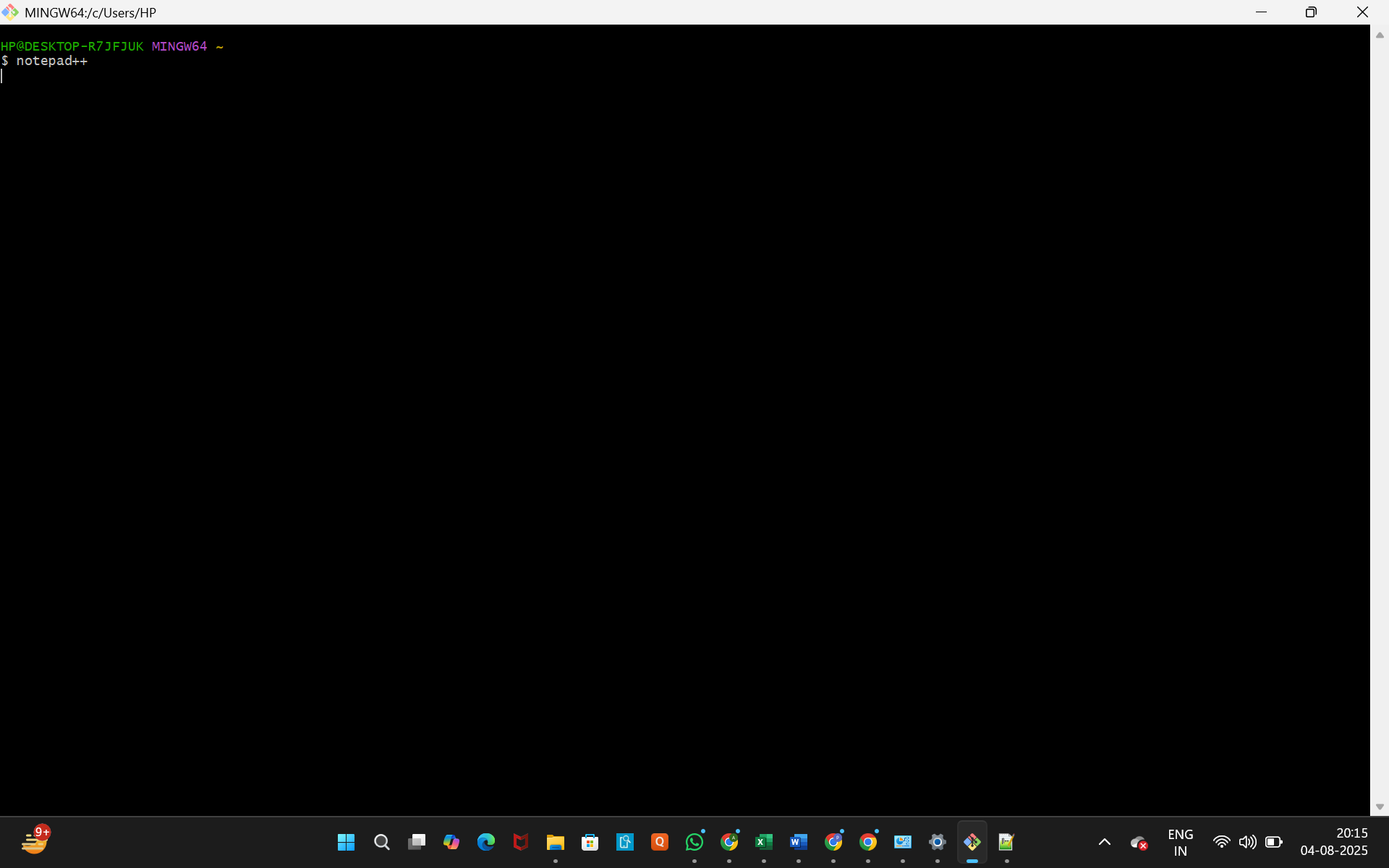
**Step 2: Integrate notepad++.exe to Git and make it a default editor**

1. **To check if notepad++.exe executes from Git Bash:**
   * Open Git Bash and type:

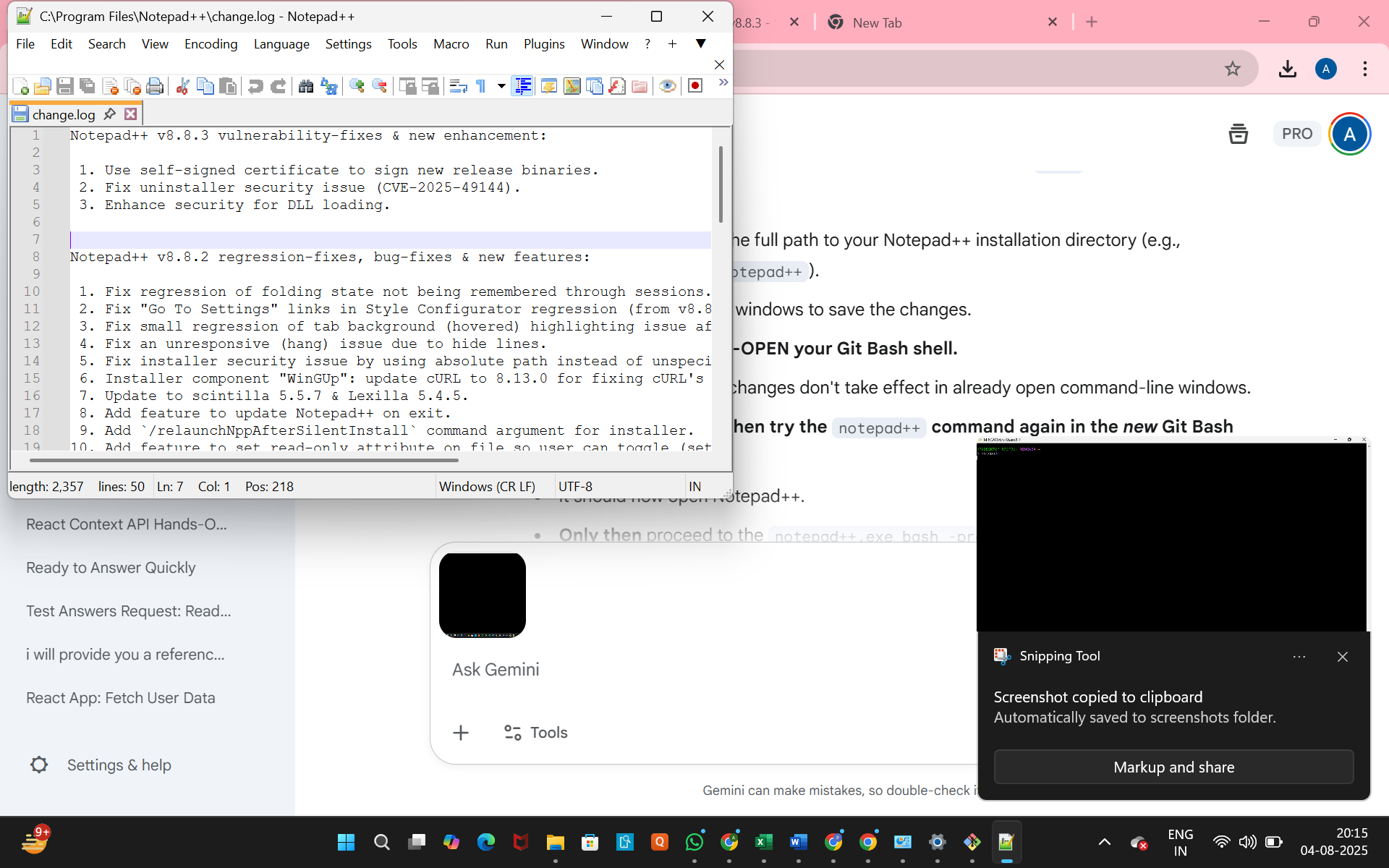


* + If you see bash: notepad++: command not found, it implies that notepad++.exe is not added to the environment PATH variable.
  + **To add the path of notepad++.exe to environment variables:**
    - Go to Control Panel -> System -> Advanced System settings.
    - Go to the Advanced tab -> Environment Variables.
    - In the "User variables" section, find the Path variable and click on "Edit".
    - Add the path to your notepad++.exe (e.g., C:\Program Files\Notepad++ if that's where it's installed) to the Path user variable. Make sure to separate paths with a semicolon ; if there are existing entries, or add it as a new line if the editor allows.

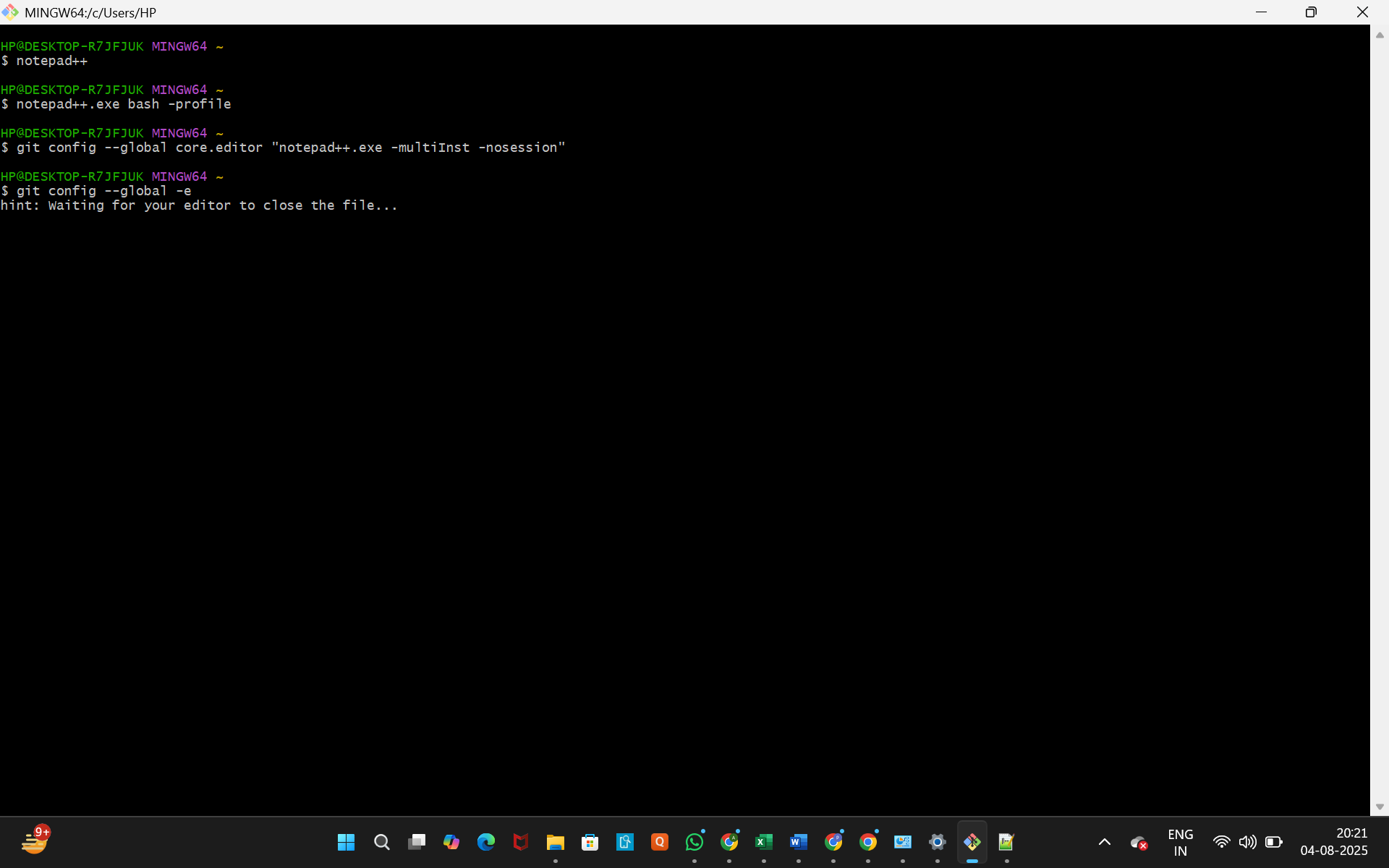
1. **Exit Git Bash shell, then open a new Git Bash shell and execute:**



* + Now, Notepad++ should open from Git Bash.



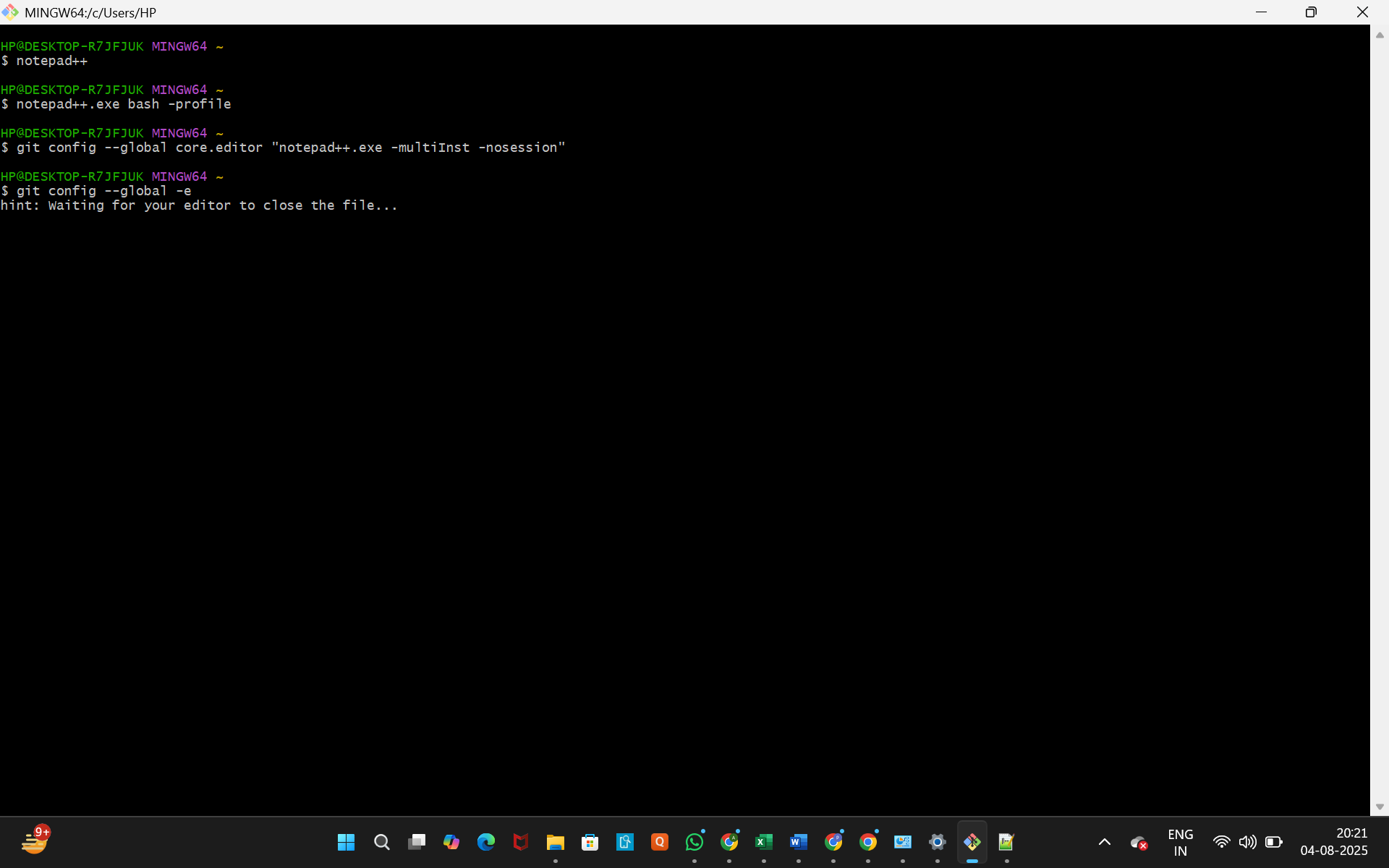
1. **To create an alias command for notepad++.exe:**
   * Execute the following command:



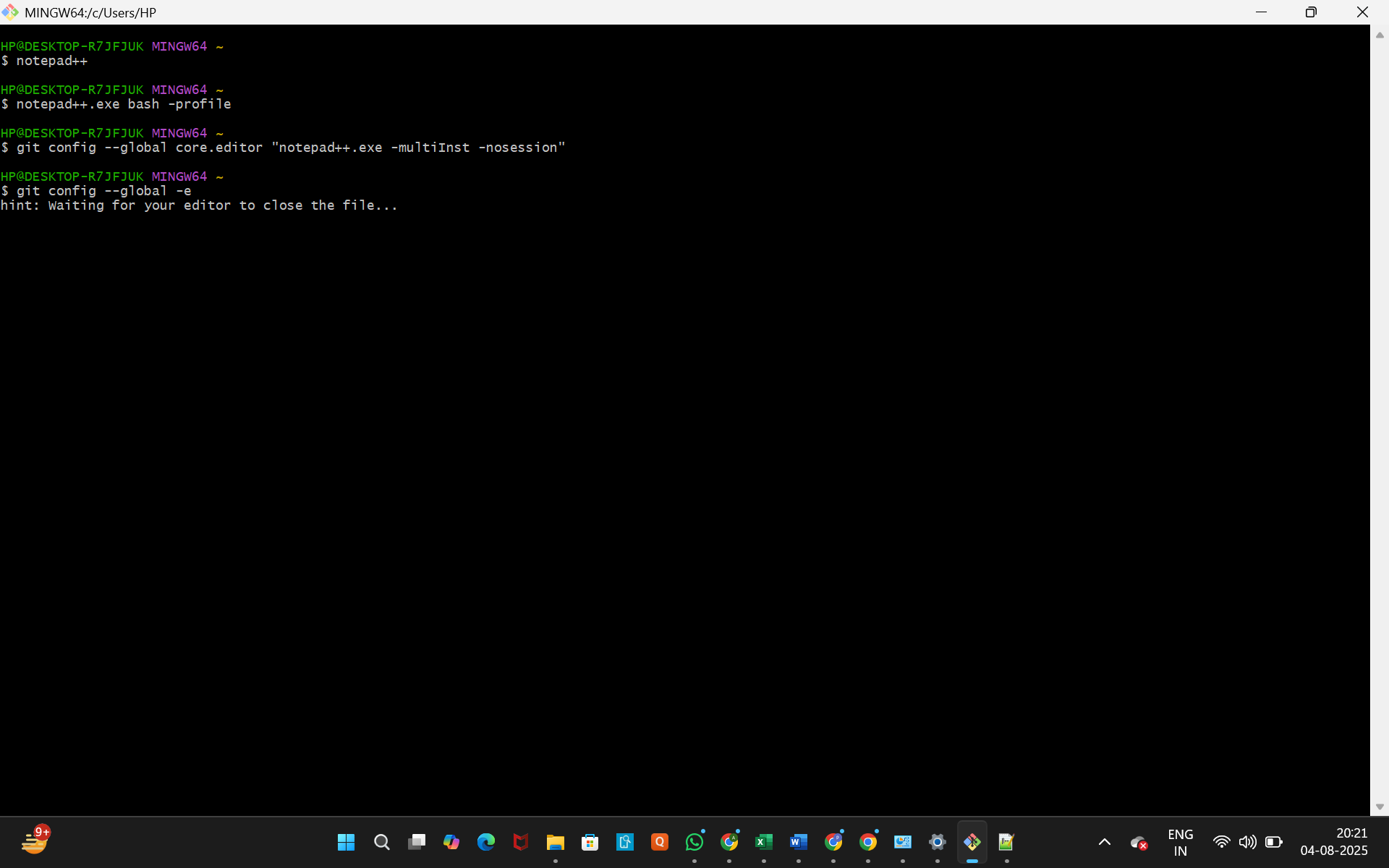
This will open Notepad++ from the bash shell and create a user profile by adding the line alias npp='notepad++.exe -multiInst -nosession' to your bash profile (e.g., .bashrc or .bash\_profile).



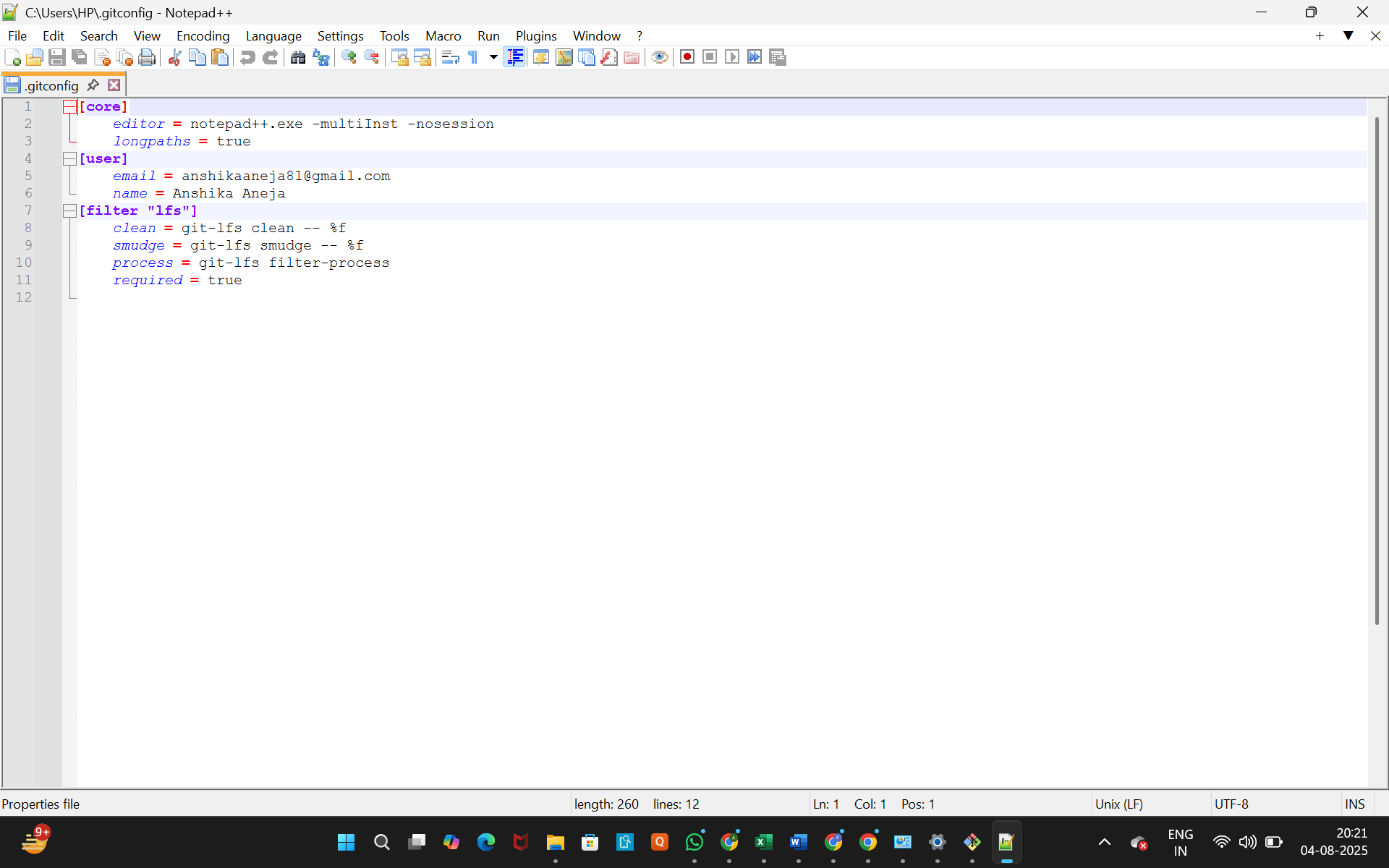
1. **To configure Notepad++ as the default editor for Git:**
   * Execute the command:



1. **To verify if Notepad++ is the default editor:**
   * Execute the command:

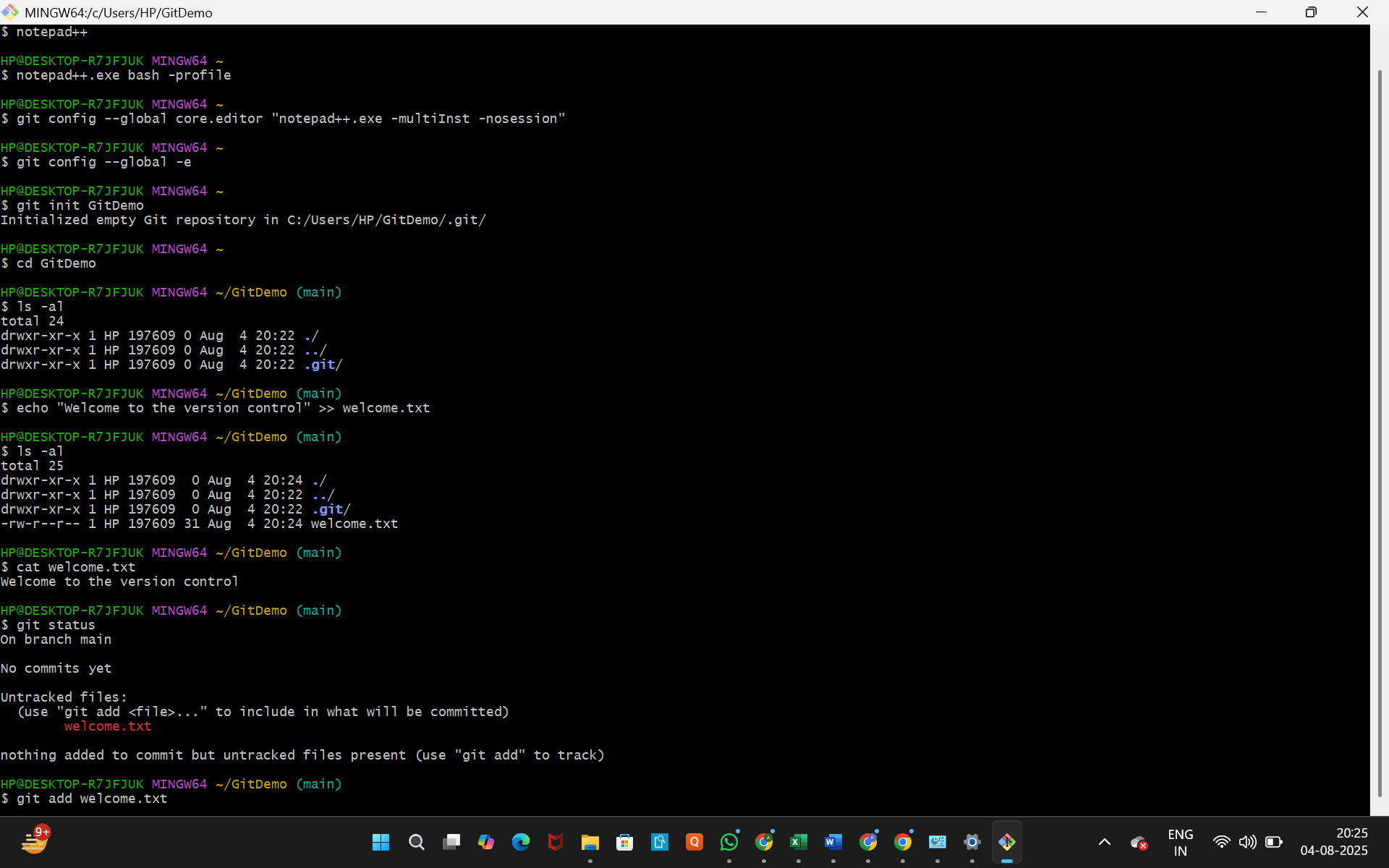


* + You might see a hint like Hint: Waiting for your editor to close the file....
  + This command will open the global Git configuration file in Notepad++. It should show the entire global configuration, including:



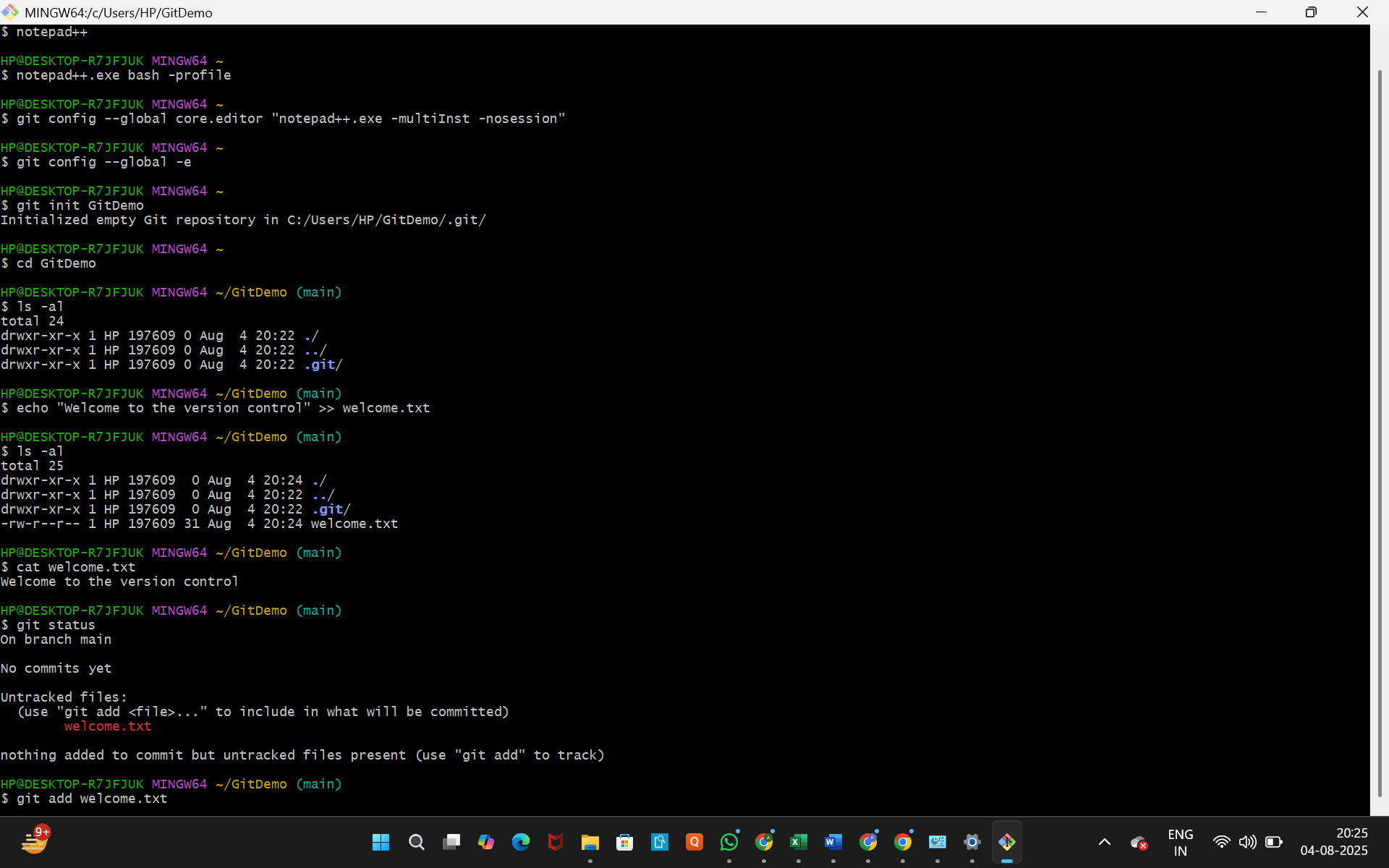
**Step 3: Add a file to source code repository**

1. **Open Git Bash shell and create a new project "GitDemo" by executing the command:**



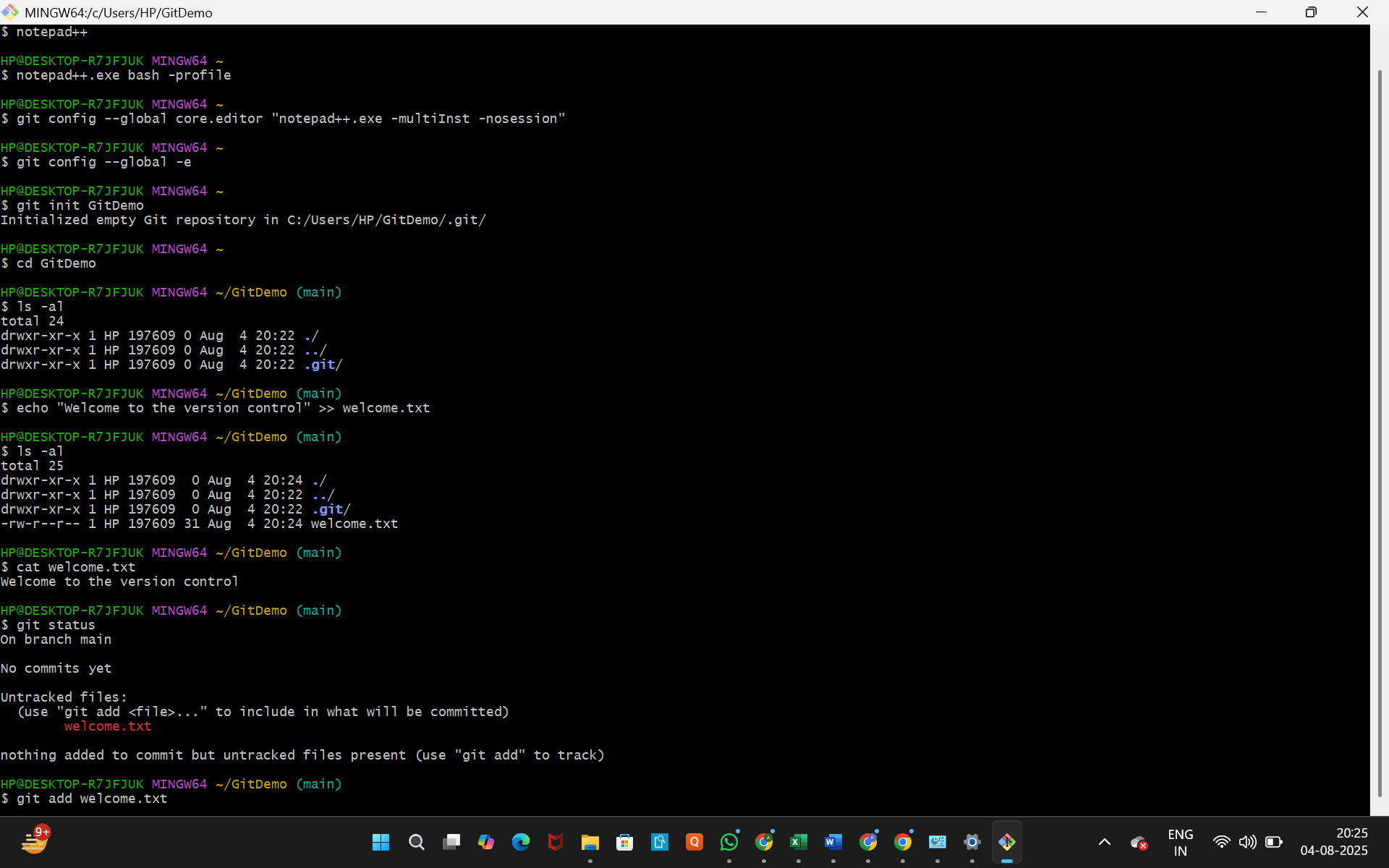
This will initialize an empty Git repository in the GitDemo directory.

1. **Git Bash initializes the "GitDemo" repository. To verify, navigate into the GitDemo directory and execute the command:**

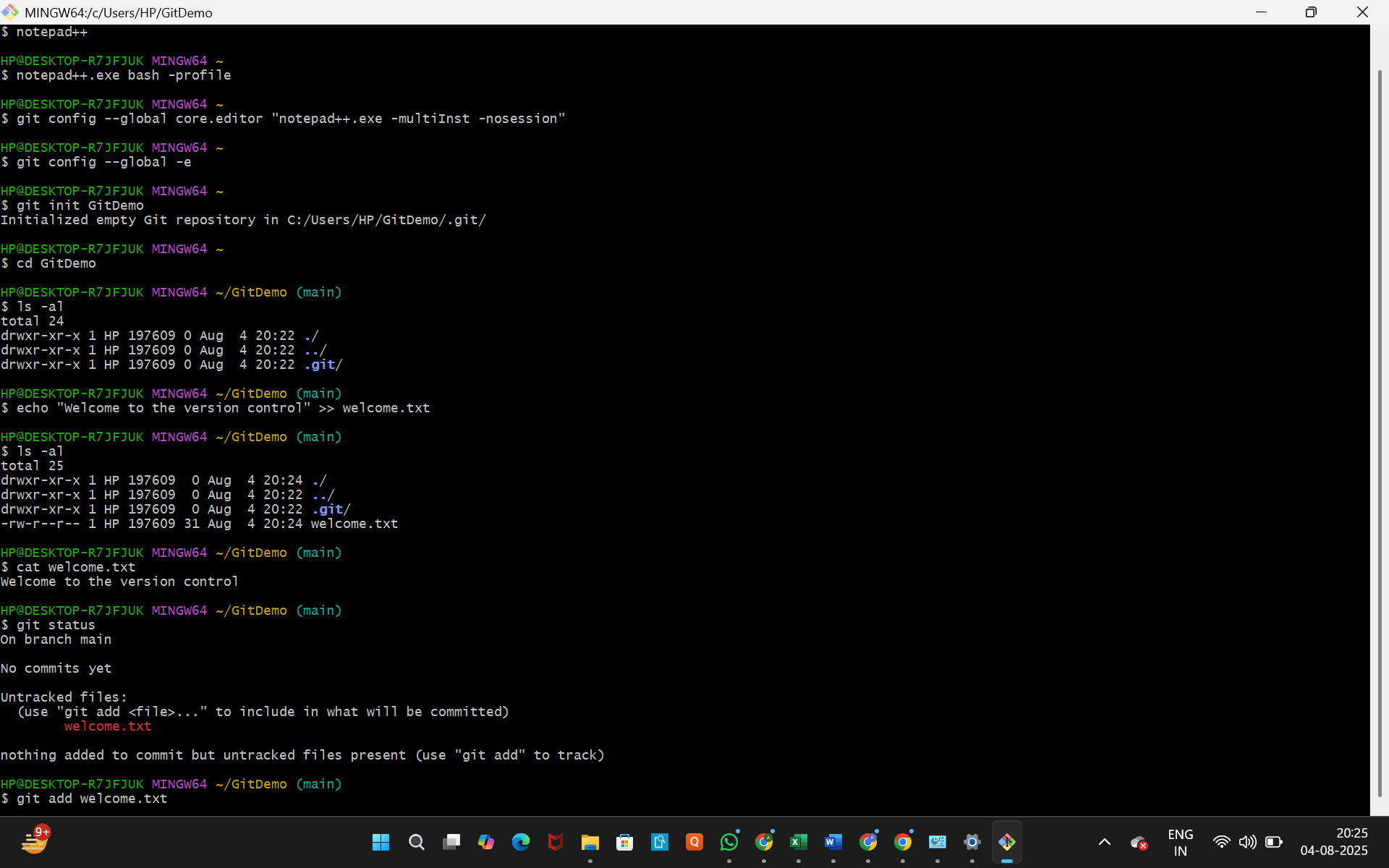


* + This will display all the hidden files in the Git "working directory", including the .git folder.

1. **To create a file "welcome.txt" and add content to the file:**
   * Execute the command:

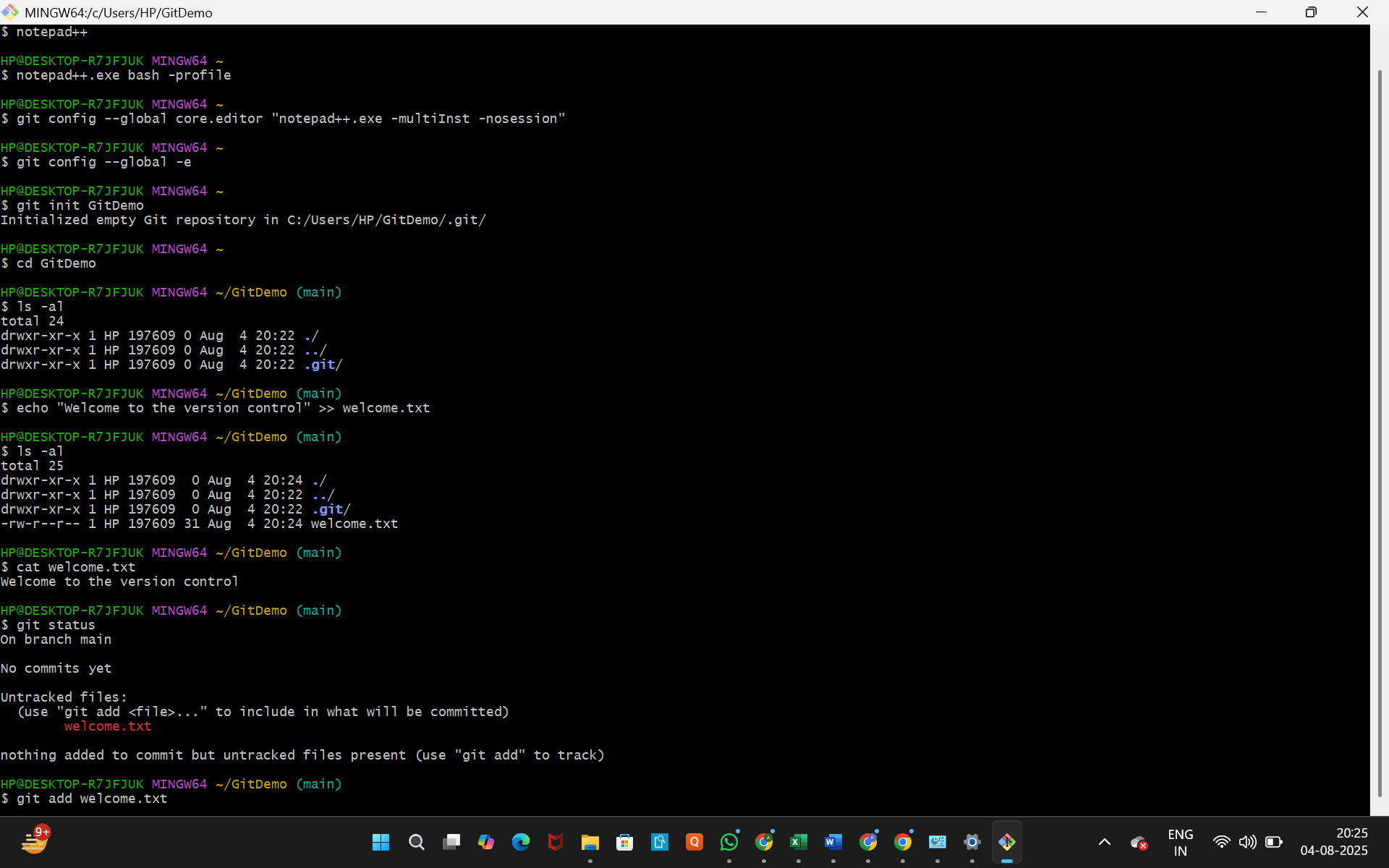


1. **To verify if the file "welcome.txt" is created:**
   * Execute the command:



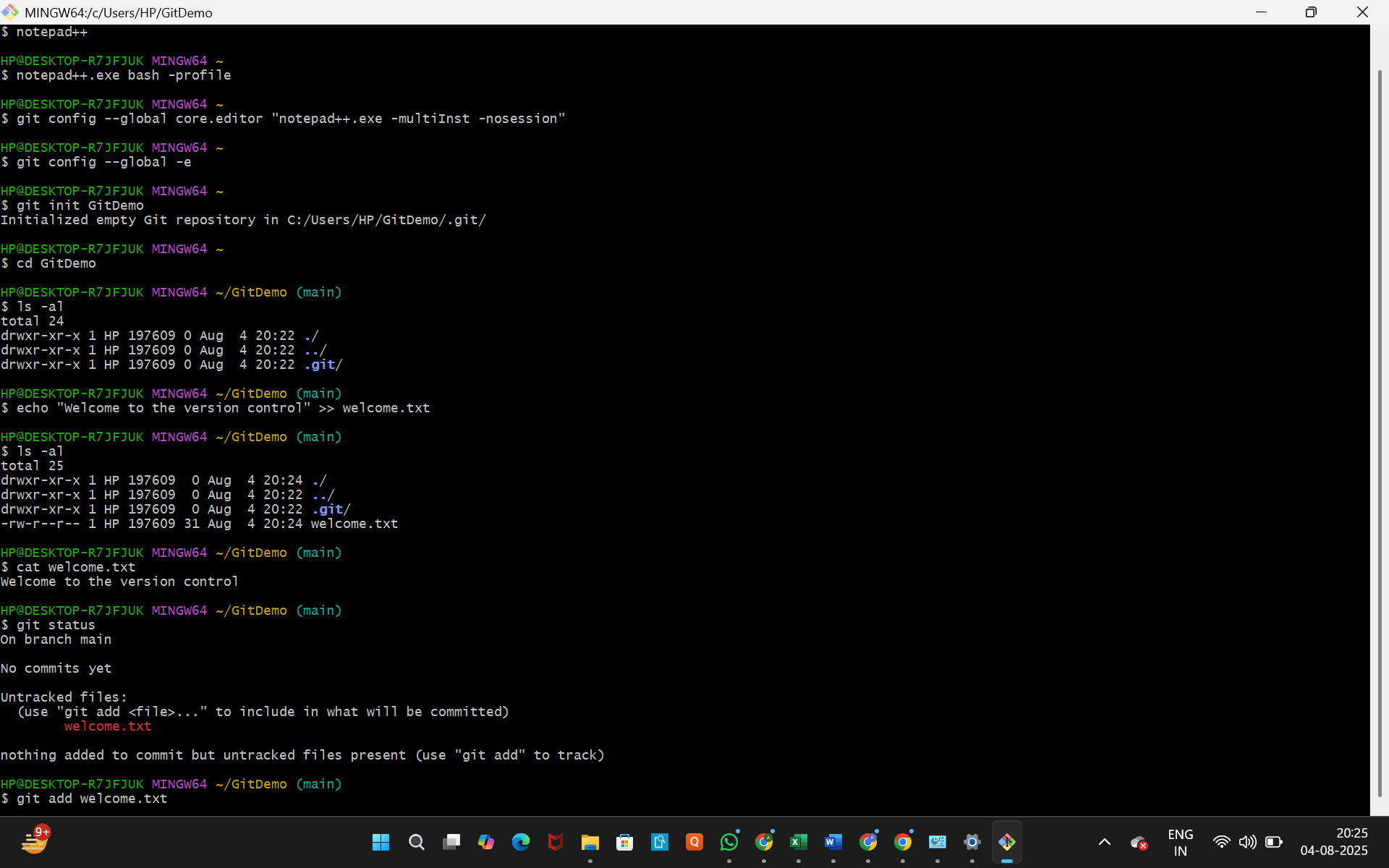
* + You should see welcome.txt listed.

1. **To verify the content:**
   * Execute the command:



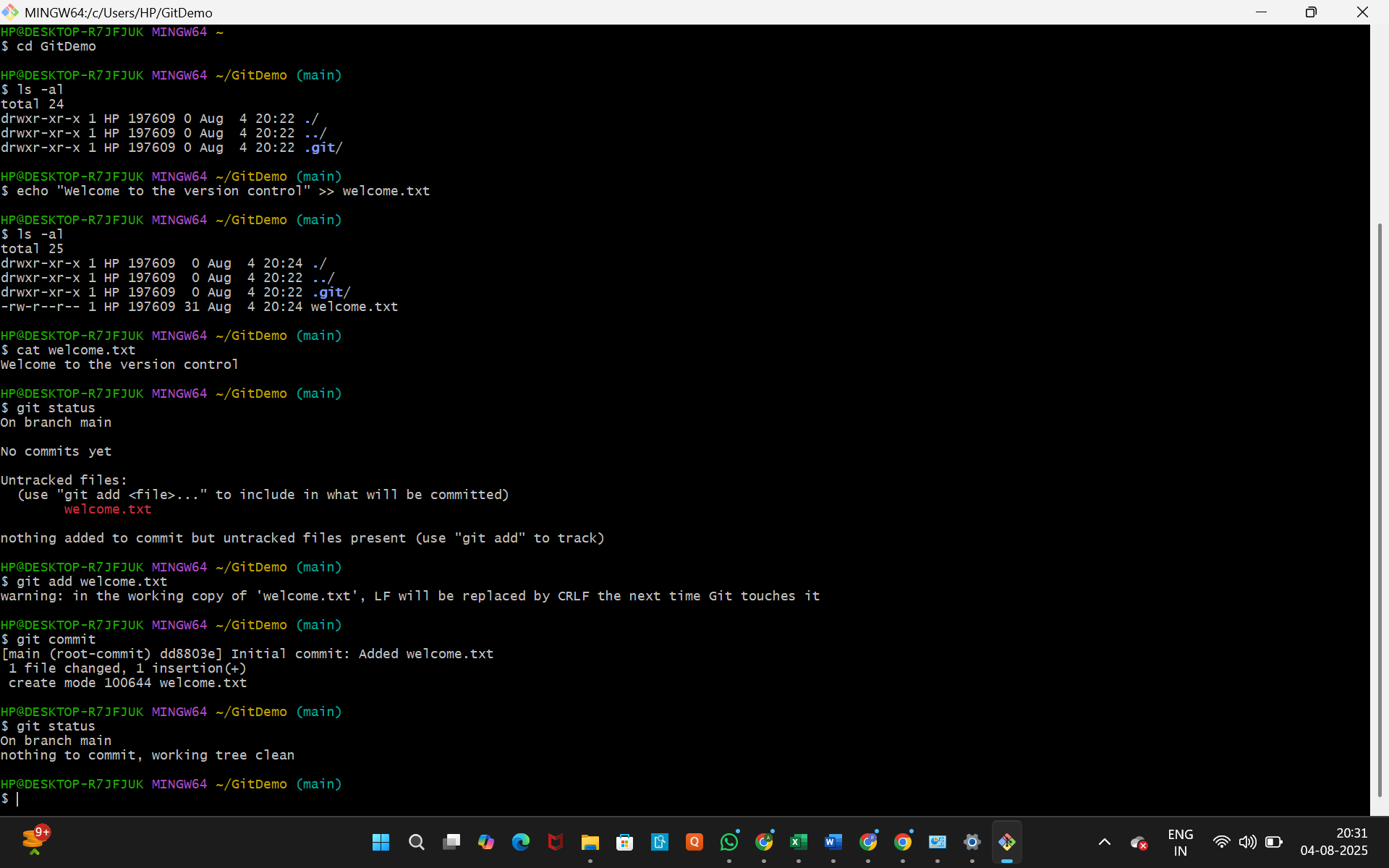
* + The output should be Welcome to the version control.

1. **Check the status by executing:**



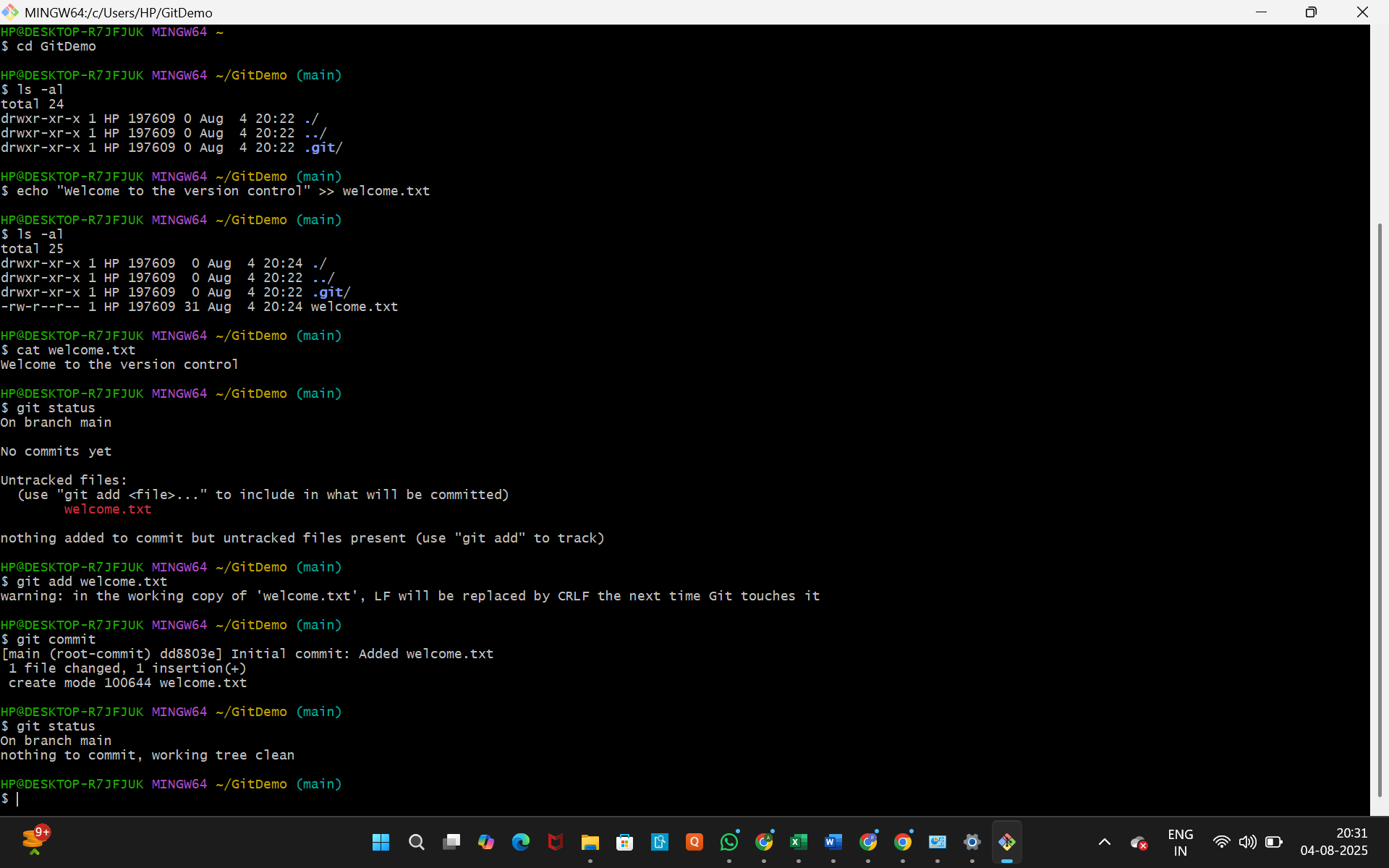
* + The output should show welcome.txt as an untracked file.
  + Now the file "welcome.txt" is available in Git "working directory".

1. **To make the file to be tracked by Git repository:**
   * Execute the command:



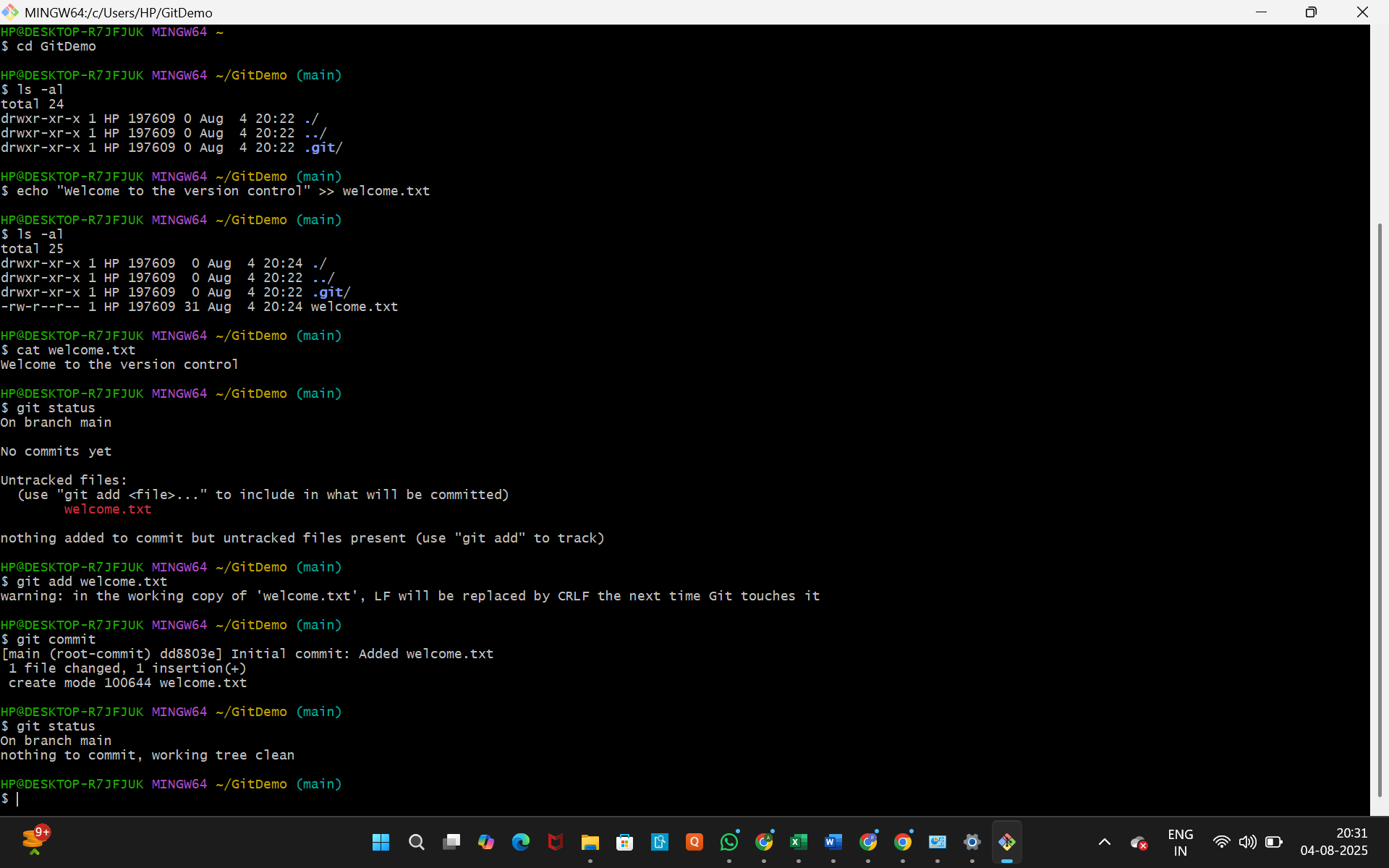
* + You might see a warning like warning: LF will be replaced by CRLF in welcome.txt. The file will have its original line endings in your working directory. This is generally harmless and relates to line ending conversions between operating systems.

1. **To add multi-line comments, we are opening the default editor to comment:**
   * Execute the command:



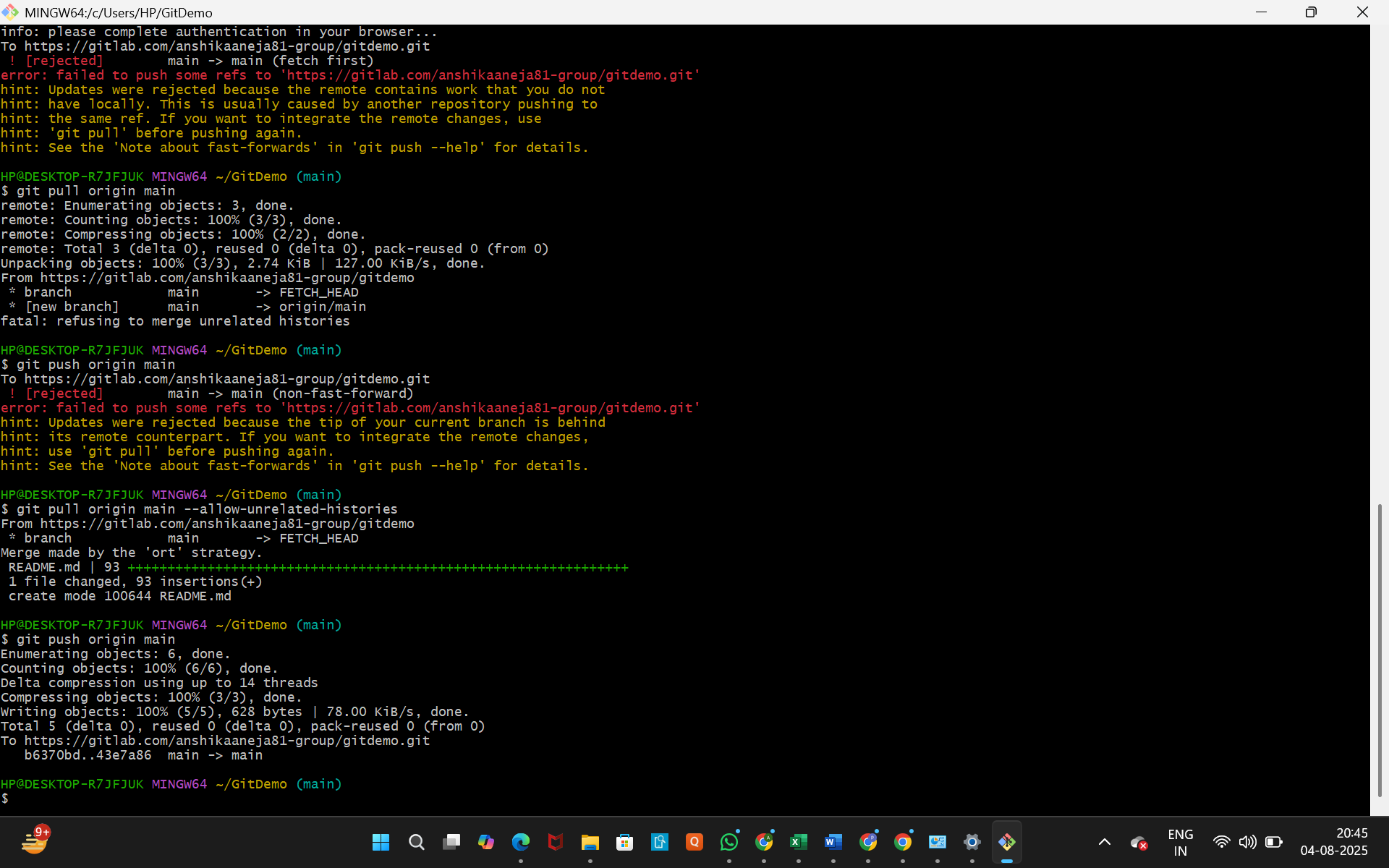
* + Notepad++ editor will open and allow you to add a multi-line commit message. Save and close the file after entering your message.

1. **To check if the local and "Working Directory" git repository are same:**
   * Execute git status again:



* + The output should show On branch master and nothing to commit, working tree clean.

1. **Signup with GitLab and create a remote repository "GitDemo".** (This step is external to Git Bash and needs to be done on the GitLab website).
2. **To pull the remote repository (if there are changes):**
   * Execute:



1. **To push the local repository to the remote repository:**
   * Execute:

