SCM REPORT

NAME : ABHISHEK THAKUR

ROLL NO.: 2110990070

GROUP : GO1-B

CLUSTER : BETA

|  |  |
| --- | --- |
| S NO. | EXPERIMENT NAME |
| 1. | INTRODUCTION |
| 2. | SETTING UP GIT CLIENT |
| 3. | SETTING UP GITHUB ACCOUNT |
| 4. | PROGRAM TO GENERATE LOGS |

|  |  |
| --- | --- |
| 5. | CREATE AND VISUALIZE BRANCHES |
|  |  |

**INTRODUCTION**

**WHAT IS GIT AND ITS IMPORTANCE**

*Git is a version control system that allows numerous developers to collaborate on non-linear development projects.*

*Git is an example of a distributed version control system (DVCS) (hence Distributed Version Control System).*

**WHAT IS GITHUB ?**

*GitHub is a version management and collaboration tool for programming. It allows you and others to collaborate on projects from any location.*

**WHAT IS REPOSITORY ?**

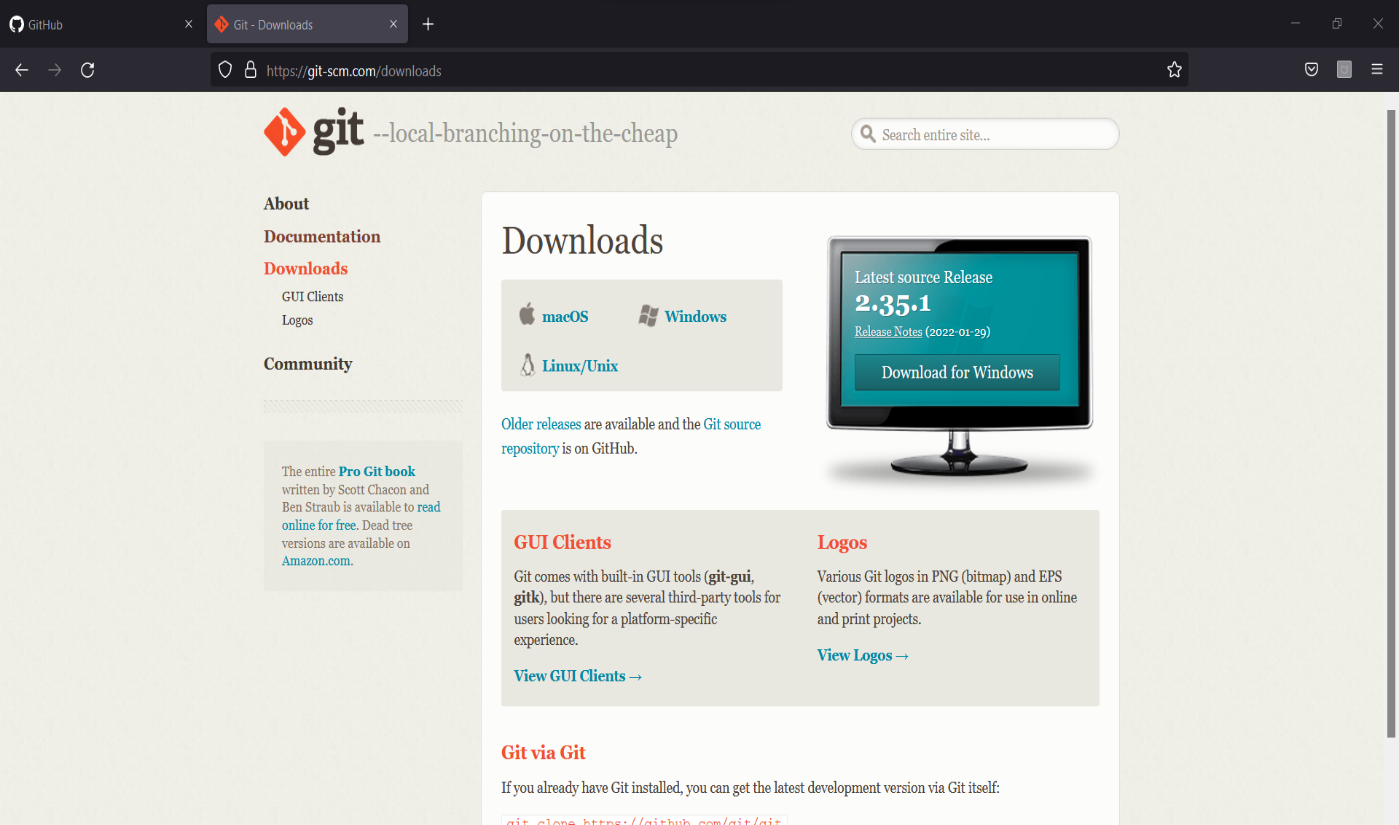
*A repository stores all of your project's files, as well as the revision history for each one. Within the repository, you may discuss and monitor your project's progress. The .git/ subdirectory within a project is a Git repository. This repository keeps track of any changes made to files in your project over time, creating a history.*

**EXPERIMENT NO.1**

**AIM : SETTING UP GIT CLIENT**

* For git installation on your system, go to the linked URL.

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

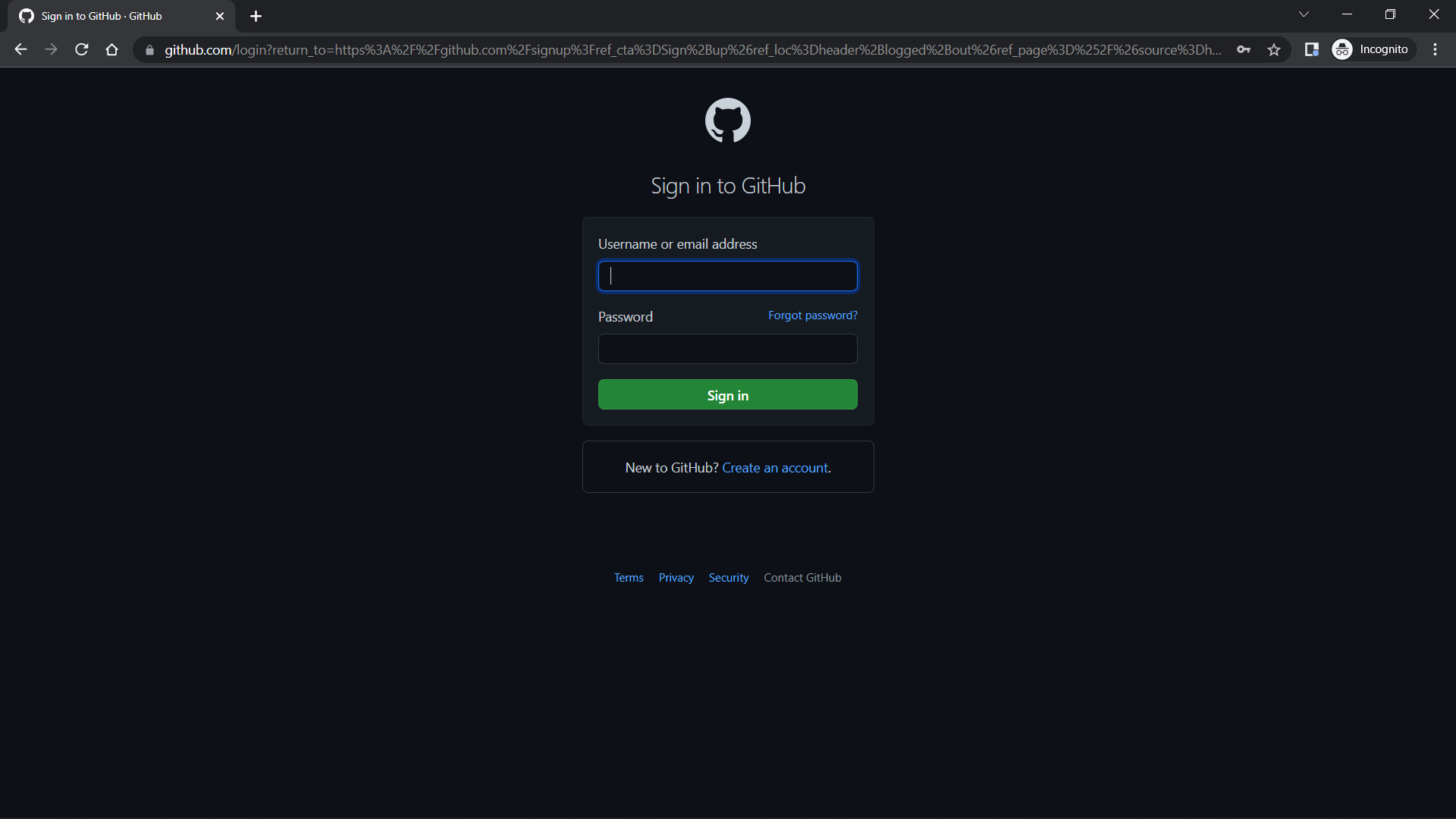


* You must first access this webpage and then choose your operating system by clicking on it. I'll walk you through the processes for the Windows operating system in this article.
* Select the CPU for your system now. (Most of the system now runs on 64-bit processors.) Your download will begin when you pick a processor.
* You must now open the Git folder.
* You will be asked if you want to enable this program to make modifications to your PC once you launch it.
* YES should be selected.
* Click on Next
* Continue clicking on next few times more.
* Now select the Install option.
* Click on Finish after the installation is finished.

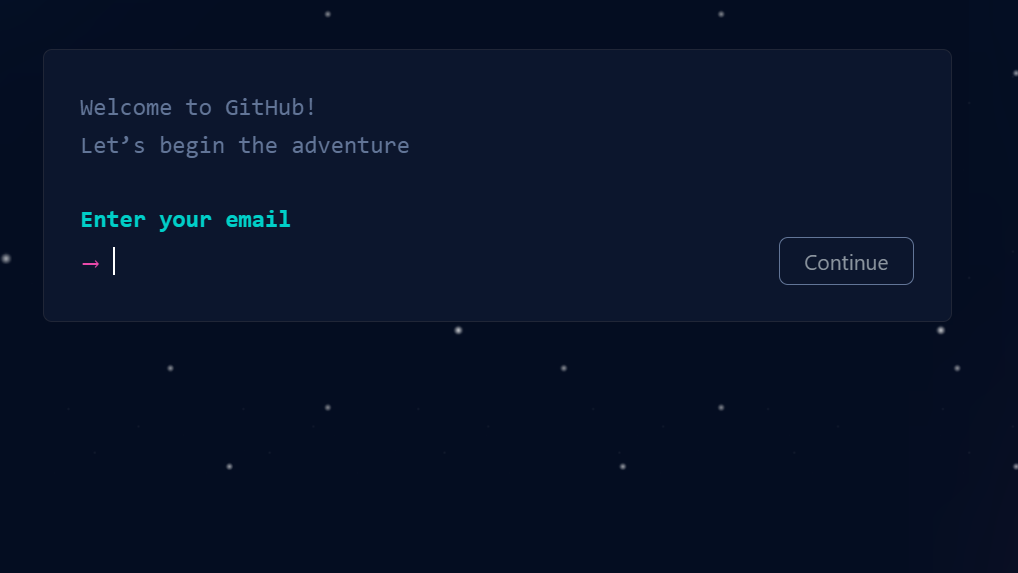
**EXPERIMENT NO.2**

**AIM : SETTING UP GITHUB ACCOUNT**

* *Open your web browser search GitHub login.*
* *Click on Create an account if you are a new user or if you have already an account, please login.*



* *After clicking on "Create a New Account," you will be sent to a new page where you must enter your email address for your account. Now type in the password you'd want to use for your GitHub account. Then you'll be prompted to enter your username.*

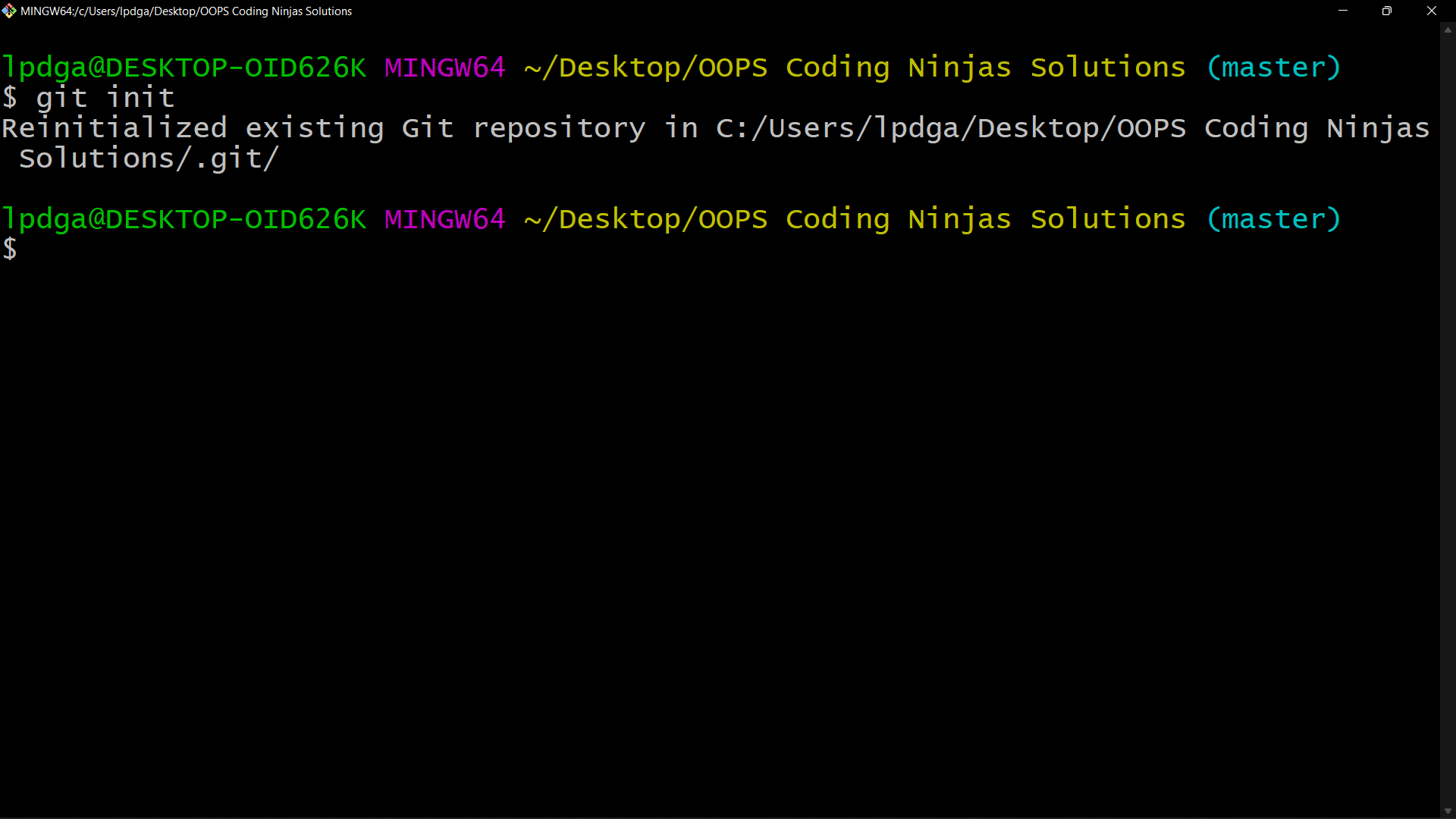


* *Now Click on Create Account.*
* *Verify it from your email and you are all set to go.*

**EXPERIMENT NO.3**

**AIM : PROGRAM TO GENERATE LOGS**

## ***First of all create a local repository using Git. For this, you have to make a folder in your device, right click and select “****Git Bash Here****”. This opens the Git terminal. To create a new local repository, use the command “****git init****” and it creates a folder*** *.git****.***



* *When we use GIT for the first time, we have to give the user name and email so that if I am going to change in project, it will be visible to all.*

*For this, we use command 🡪*

***“git config --global user.name Name”***

***“git config --global user.email email”***

*For verifying the user’s name and email, we use 🡪*

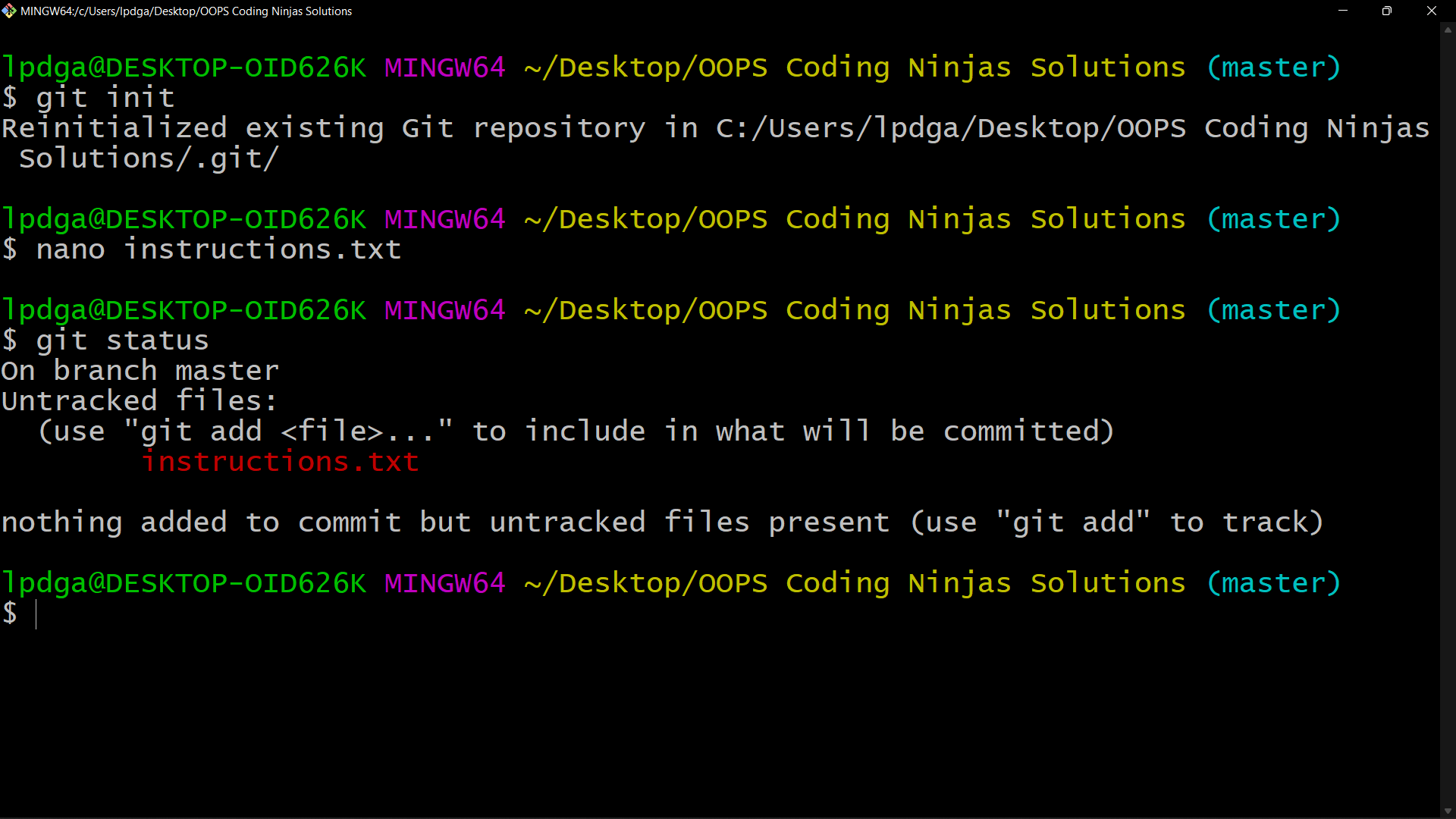
***“git config --global user.name”***

***“git config --global user.email”***

***IMPORTANT COMMANDS :***

* ***ls 🡪*** *It gives the file names in the folder.*
* ***ls -lart 🡪*** *Gives the hidden files also.*
* ***git status 🡪***  *Displays the state of the working directory and the staged snapshot.*
* ***touch filename 🡪*** *This command creates a new file in the repository.*
* ***Clear 🡪***  *It clears the terminal.*
* ***rm -rf .git 🡪*** *It removes the repository.*
* ***git log 🡪*** *displays all of the commits in a repository's history*
* ***git diff 🡪*** *It compares my working tree to staging area.*

*Now, we have to create some files in the repository. Suppose we created instructions.txt Now type git status:*



*You can see that instructions.txt is in red colour that means it is an untracked file.*

*Now firstly add the file in staging area and then commit the file.*

*For this, use command 🡪*

***git add -A [*** *For add all the files in staging area.****]***

***git commit -m “write any message” [*** *For commit the file****]***

* **git log:** The git log command displays a record of the commits in a Git repository. By default, the git log command displays a commit hash, the commit message, and other commit metadata

**EXPERIMENT NO.4**

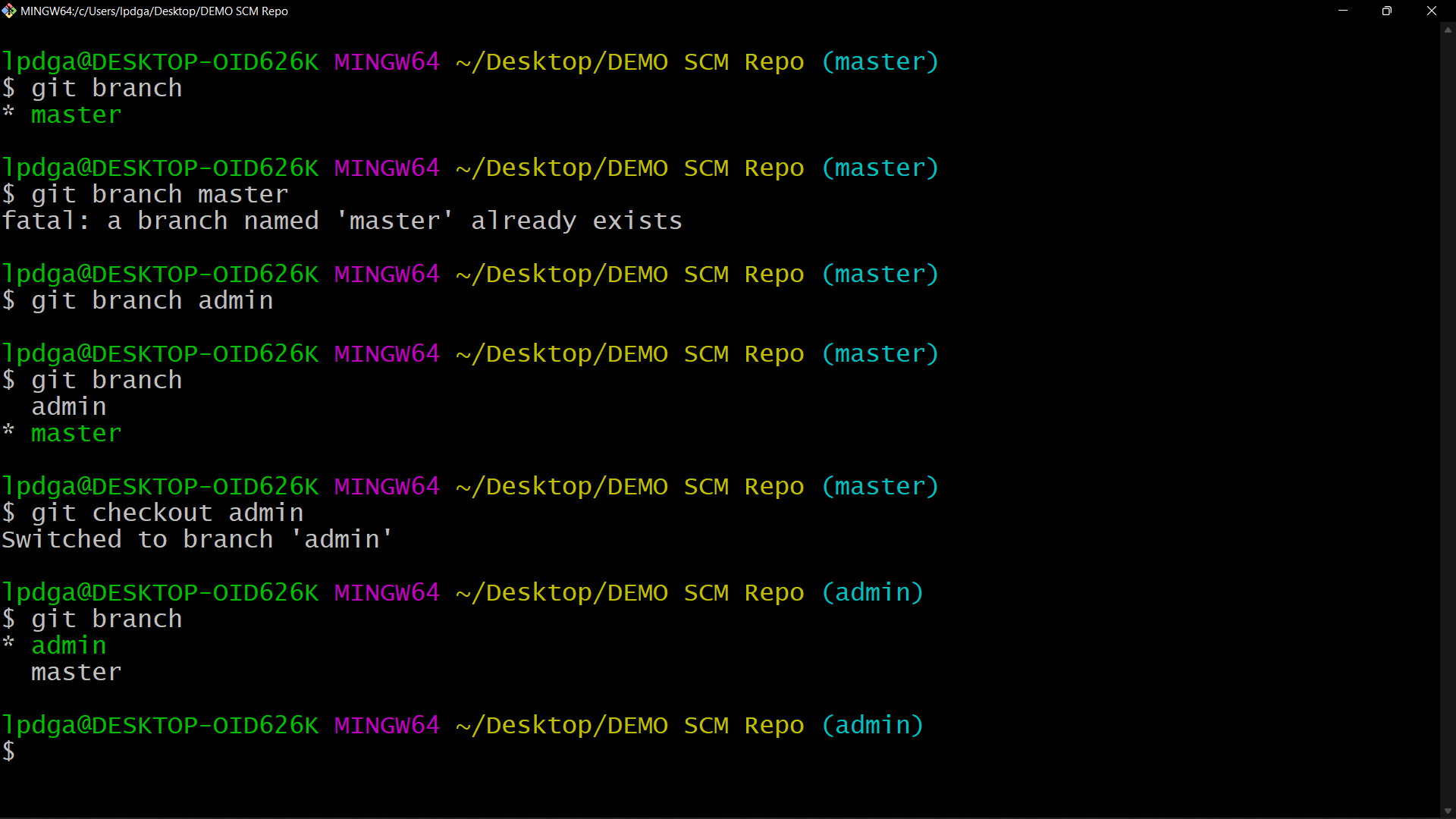
**AIM : CREATE AND VISUALIZE BRANCHES**

* ***Branching:*** *A branch in Git is an independent line of work (a pointer to a specific commit). It allows users to create a branch from the original code (master branch) and isolate their work. Branches allow you to work on different parts of a project without impacting the main branch.*

*Let us see the command of it:*

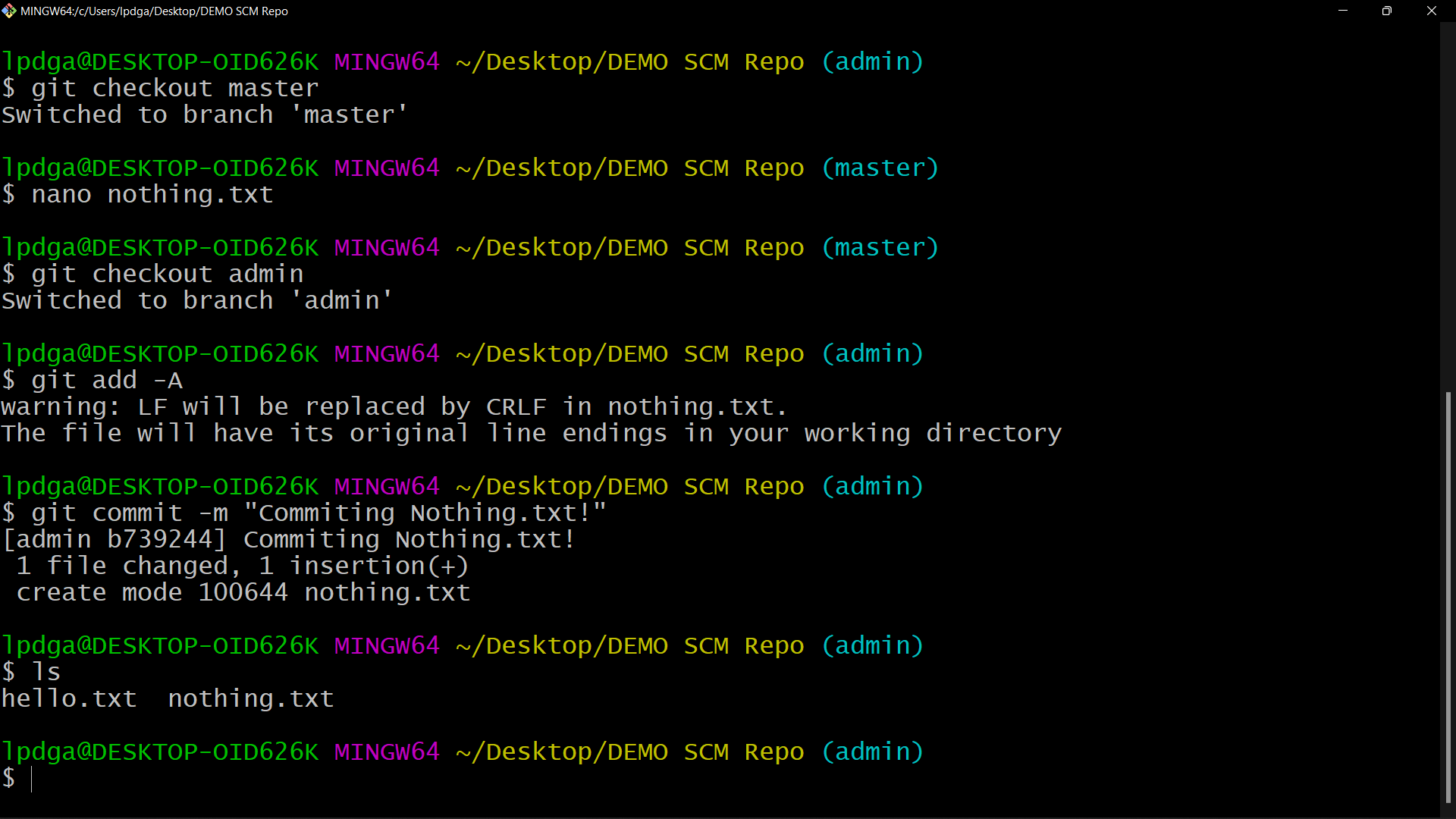
*Firstly, add a new branch, let us suppose the branch name is activity1.*

*For this use command 🡪*

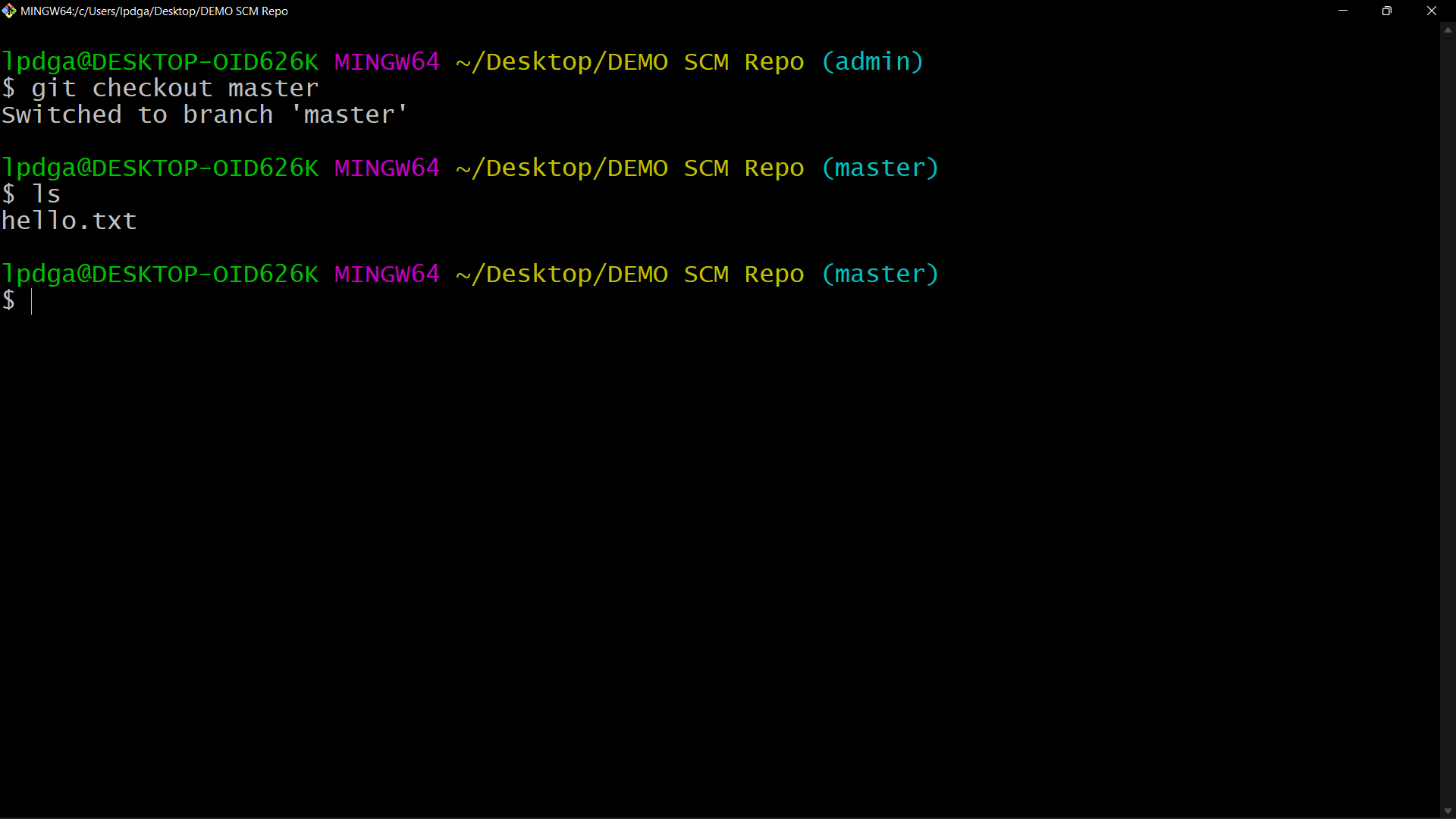
* ***git branch name [****adding new branch****]***
* ***git branch [****use to see the branch’s names****]***
* ***git checkout branch name [****use to switch to the given branch****]***
* 

*In this you can see that firstly ‘git branch’ shows only one branch in green colour but when we add a new branch using ‘git branch admin’, it shows 2 branches but the green colour and star is on master. So, we have to switch to admin by using ‘git checkout admin’. If we use ‘git branch’, now you can see that the green colour and star is on admin. It means you are in admin branch and all the data of master branch is also on admin branch. Use “ls” to see the files.*

*Now add a new file in admin branch, do some changes in file and commit the file.*



*If we switched to master branch, ‘nothing.txt’ file is not there. But the file is in admin branch.*



* *To add these files in master branch, we have to do merging. For this firstly switch to master branch and then use command 🡪*

***git merge branch-name [****use to merge branch****]***

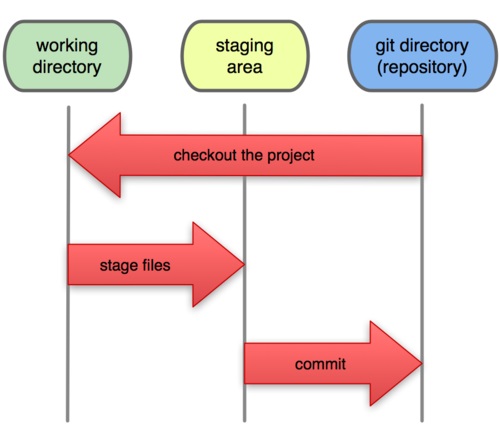


**EXPERIMENT NO.5**

**AIM:GIT LIFECYCLE DESCRIPTION**

# *Stages in GIT Life Cycle:*

*Now let’s understand the three-stage architecture of Git:*



* ***Working Directory:*** *This is the directory that we’ve initialized, and here all the changes are made to commit on GitHub.*
* ***Staging Area:*** *This is where we first put out code or files of the working repository. The command that we use to stage code is, “git add --a”, “git add File-Name” or “git add -A”. In simple terms, staging means telling Git what files we want to commit (new untracked files, modified files, or deleted files).*
* ***Git directory(repository):*** *This is where all the commits are stored whenever we make a commit. We can revert to an older version of or project using the “git checkout” command from this directory.*