Subject Name: **Source Code Management**

Subject Code: **24CSE0106**

Session: **2024-25**

Department: **CSE**

Submitted By:
Harshit Sharda
2410990331
G5

Submitted To:
Dr.Chetna Sharma

List of Programs

Practical 1

Aim: To install and configure Git Client on your local system.

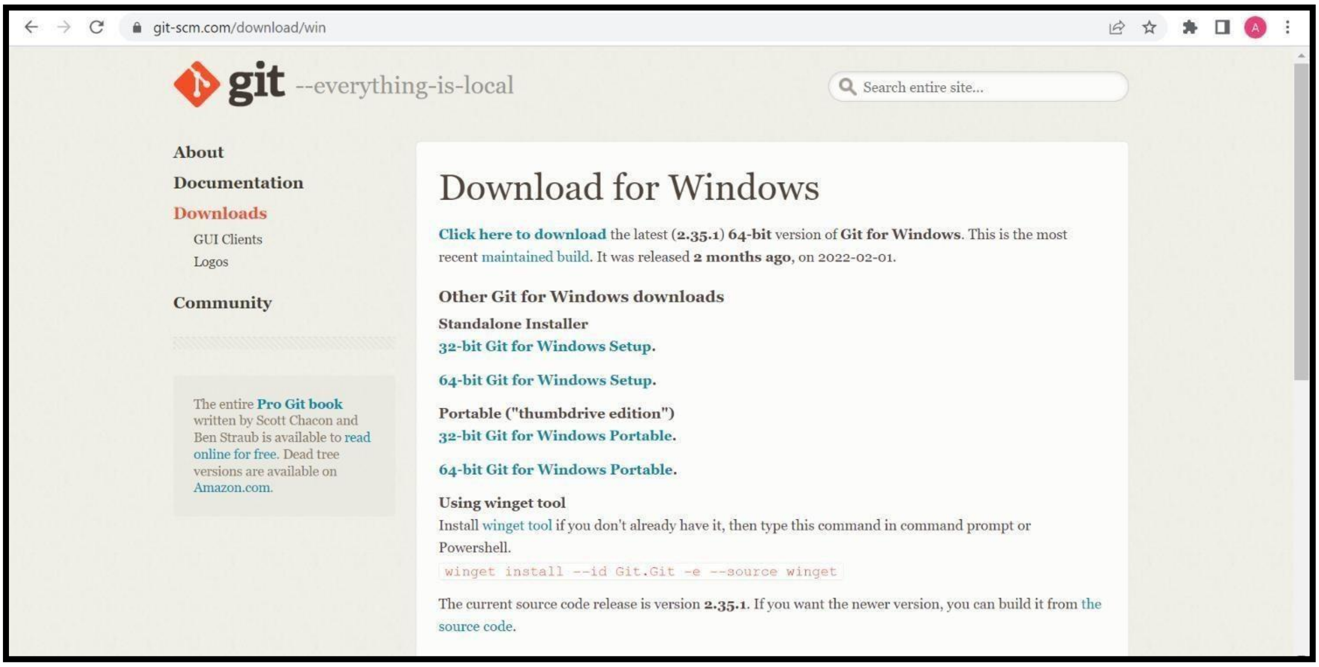
Theory:

Git is a distributed version control system used to track changes in source code. This practical focuses on setting up Git on your local system for effective version control.

Procedure:

○ Download Git from git-scm.com.
○ Install Git by following the setup wizard.
○ Open Git Bash and verify installation using the command: git--version.
○ Configure user details using the commands:
git config --global user.name "Your Name"
git config--global user.email "Your Email"

Snapshots of download:



The entire Pro Git book written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

Download for Windows

Click here to download the latest (2.35.1) 64-bit version of Git for Windows. This is the most recent maintained build. It was released 2 months ago, on 2022-02-01.

Other Git for Windows downloads

Standalone Installer
32-bit Git for Windows Setup.
64-bit Git for Windows Setup.

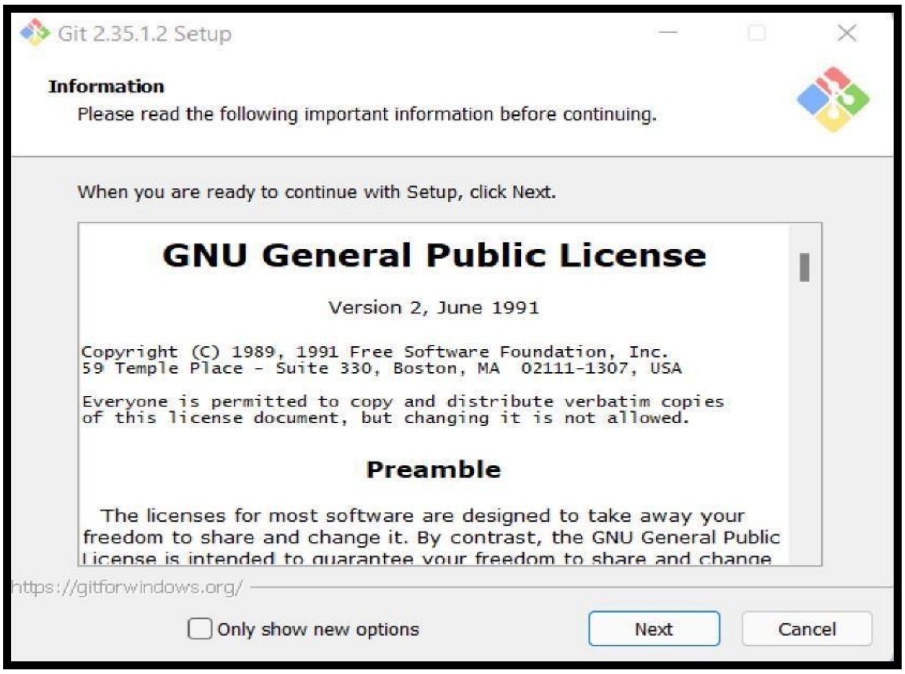
Portable ("thumbdrive edition")
32-bit Git for Windows Portable.
64-bit Git for Windows Portable.

Using winget tool
Install winget tool if you don't already have it, then type this command in command prompt or Powershell.

```
winget install --id Git.Git -e --source winget
```

The current source code release is version 2.35.1. If you want the newer version, you can build it from the source code.

Opted for "64-bit Git for Windows Setup"



Git 2.35.1.2 Setup

Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

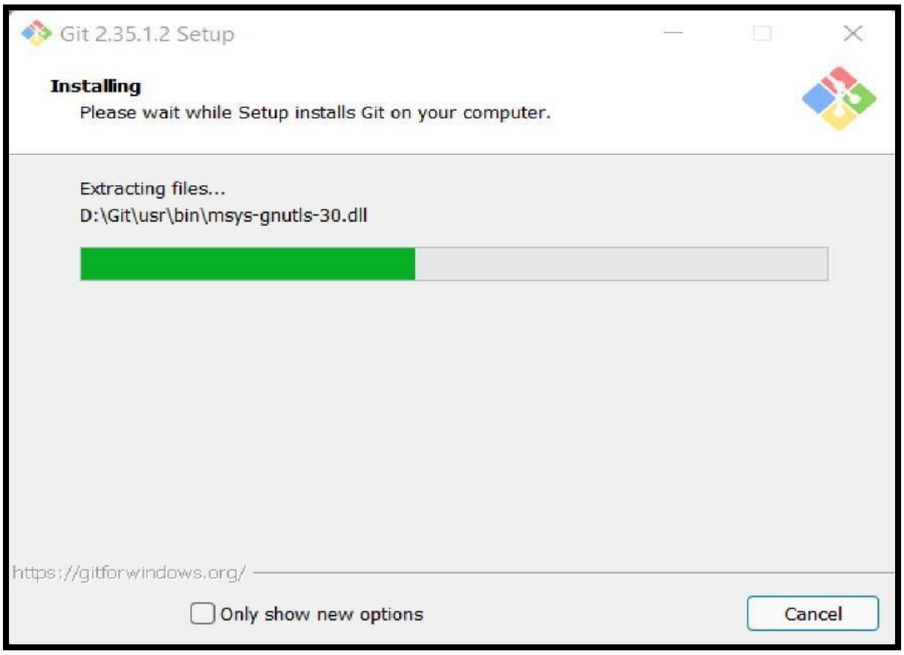
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change

Git Setup



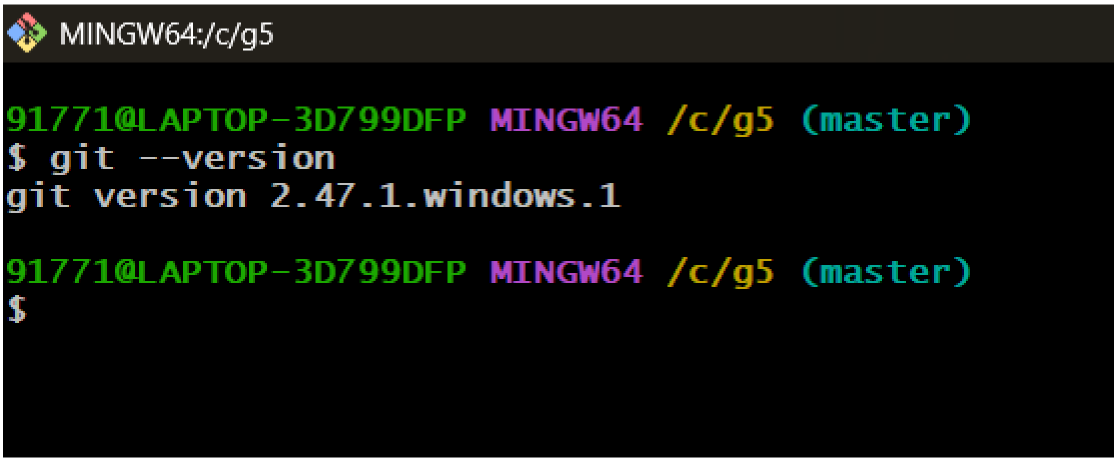
Git 2.35.1.2 Setup

Installing

Please wait while Setup installs Git on your computer.

Extracting files...
D:\Git\usr\bin\msys-gnutls-30.dll

Git Installation



```
MINGW64:/c/g5

91771@LAPTOP-3D799DFP MINGW64 /c/g5 (master)
$ git --version
git version 2.47.1.windows.1

91771@LAPTOP-3D799DFP MINGW64 /c/g5 (master)
$
```

Git Bash version

Practical 2

Aim: Setting up GitHub Account and Adding Collaborators on GitHub Repository

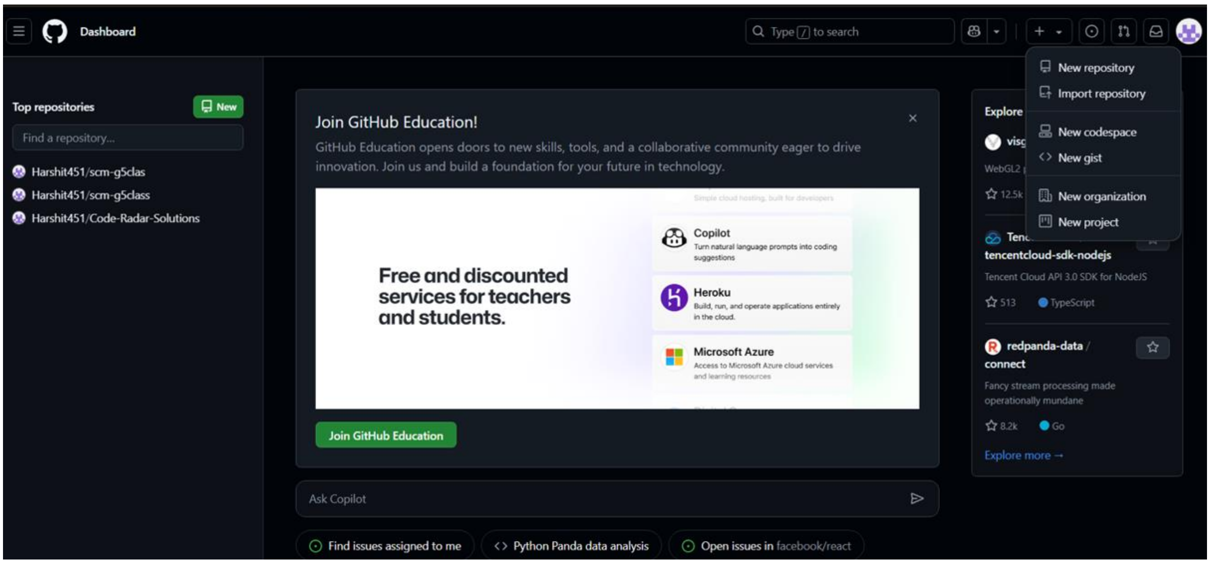
Theory:

Collaborators are individuals with write access to a repository. They can contribute to the project by pushing changes and merging pull requests.

Procedure:

○ Log in to your GitHub account and create a new repository.
○ Navigate to Settings > Manage Access in the repository.
○ Add collaborators by their GitHub usernames.
○ Collaborators will receive an invitation email,which they must accept.

Snapshots:

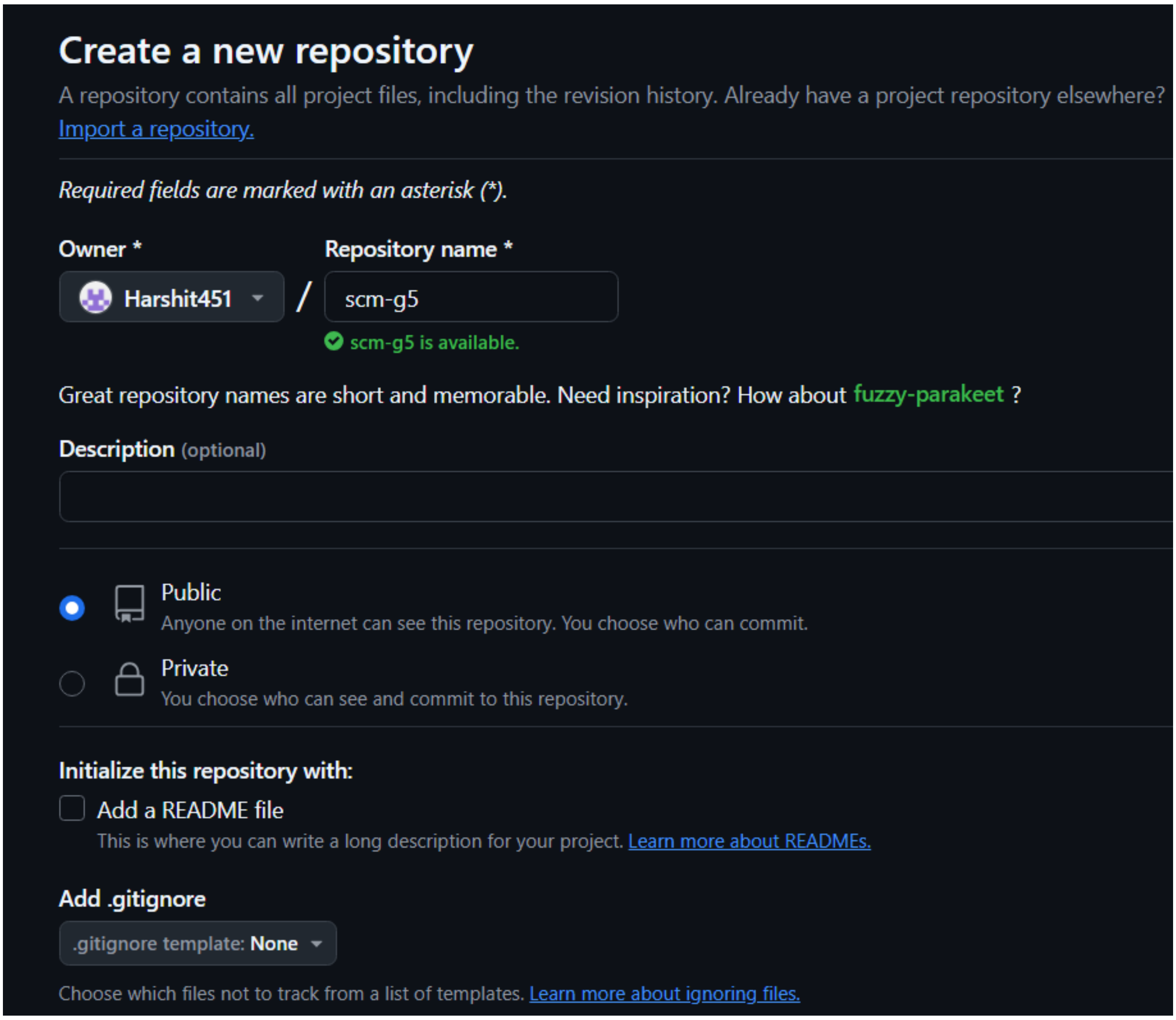


Join GitHub Education!

GitHub Education opens doors to new skills, tools, and a collaborative community eager to drive innovation. Join us and build a foundation for your future in technology.

Free and discounted services for teachers and students.

Dashboard
Type / to search
New repository
Import repository
New codespace
New gist
New organization
New project
Top repositories
New
Find a repository...
Harshit451/scm-g5clas
Harshit451/scm-g5class
Harshit451/Code-Radar-Solutions
Copilot
Turn natural language prompts into coding suggestions
Heroku
Build, run, and operate applications entirely in the cloud.
Microsoft Azure
Access to Microsoft Azure cloud services and learning resources
Join GitHub Education
Explore
WebGL2
12.5k
tencentcloud-sdk-nodejs
Tencent Cloud API 3.0 SDK for NodeJS
513
TypeScript
redpanda-data / connect
Fancy stream processing made operationally mundane
8.2k
Go
Explore more →
Ask Copilot
Find issues assigned to me
Python Panda data analysis
Open issues in facebook/react



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 *

Harshit451

/

Repository name *

scm-g5

✓ scm-g5 is available.

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

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

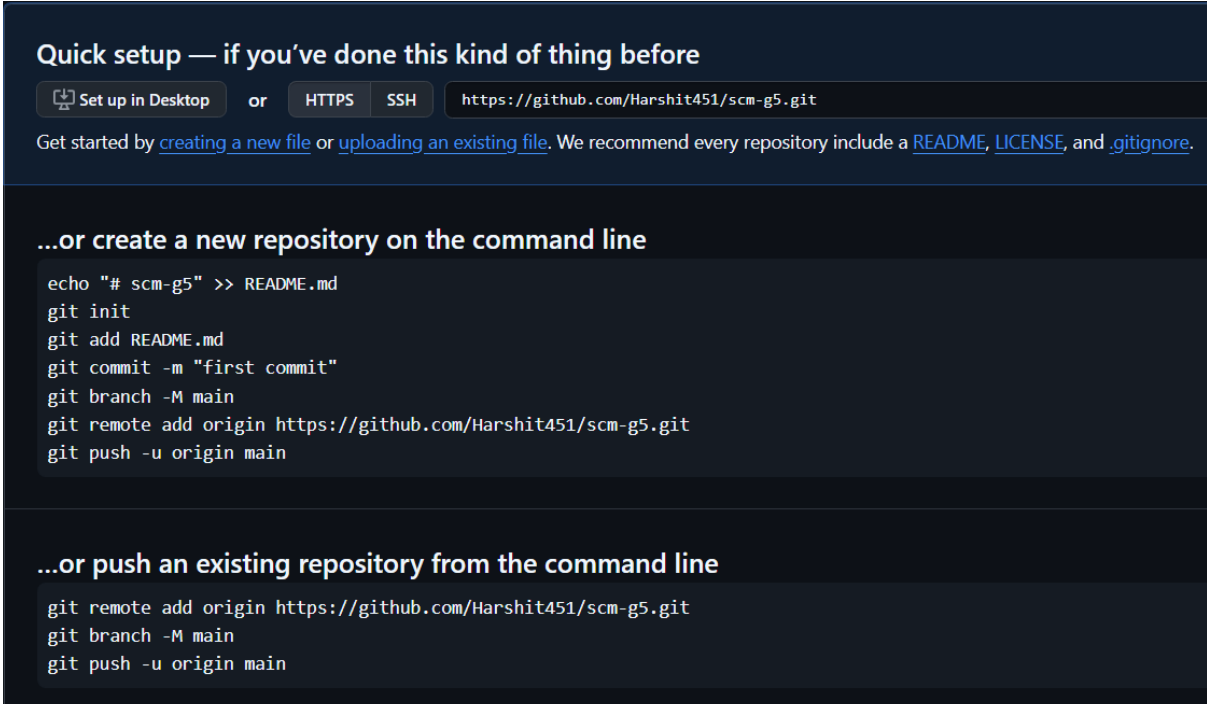
Initialize this repository with:

□ Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: **None**

Choose which files not to track from a list of templates. Learn more about ignoring files.



Quick setup — if you've done this kind of thing before

```
https://github.com/Harshit451/scm-g5.git
```

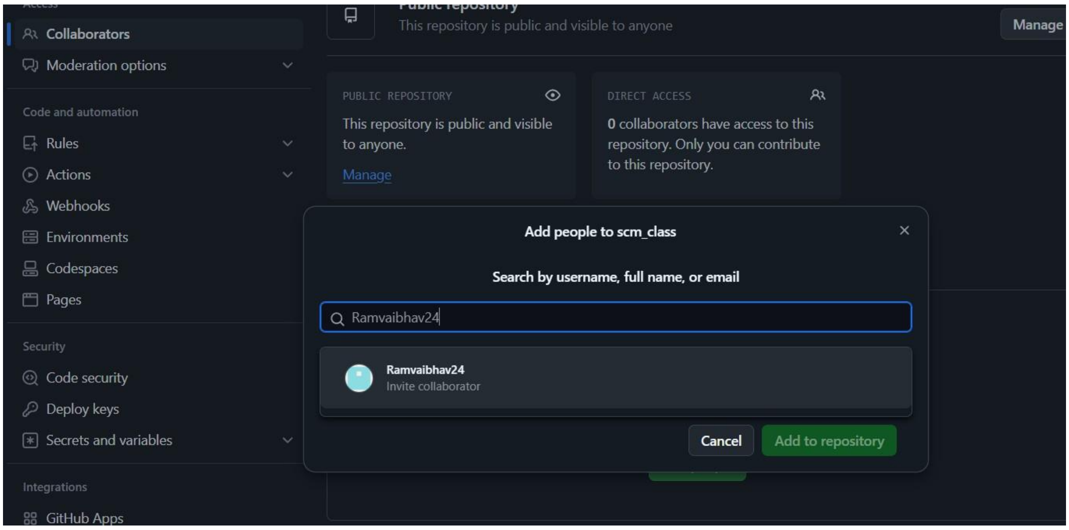
Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

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

```
echo "# scm-g5" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Harshit451/scm-g5.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/Harshit451/scm-g5.git
git branch -M main
git push -u origin main
```



This repository is public and visible to anyone

PUBLIC REPOSITORY

This repository is public and visible to anyone.

Manage

DIRECT ACCESS

0 collaborators have access to this repository. Only you can contribute to this repository.

Add people to scm_class

Search by username, full name, or email

Ramvaibhav24

Ramvaibhav24
Invite collaborator

Practical 3

Aim:
To merge two branches within a Git repository**.**

Theory:
Merging branches in Git allows you to combine changes from one branch into another. It is a fundamental process in collaborative workflows, ensuring all contributions are integrated into a single codebase.

Procedure:
1. Create a new branch and switch to it :

```
Git checkout-b new-branch
```

2. Make changes to a file in the newbranch and commit them:

```
echo "New content"> file.txt
git add file.txt
git commit-m"Addchanges in new branch"
```

3. Switch back to the main branch :

```
git checkout main
```

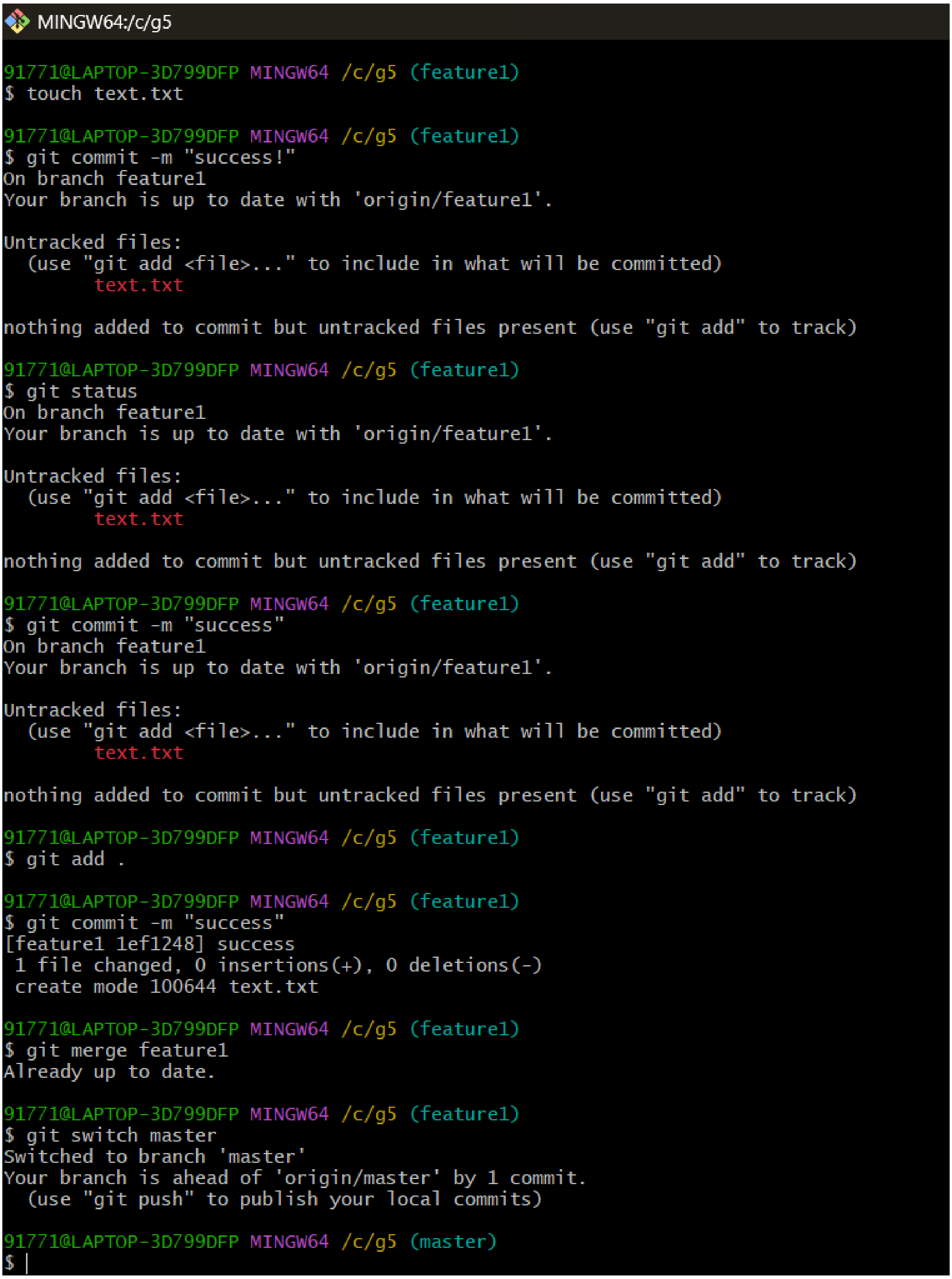
4. Modify another file in the main branch and commit the changes:

```
echo "Main branch changes"> another-file.txt
git add another-file.txt
git commit-m"Modify file in main branch"
```

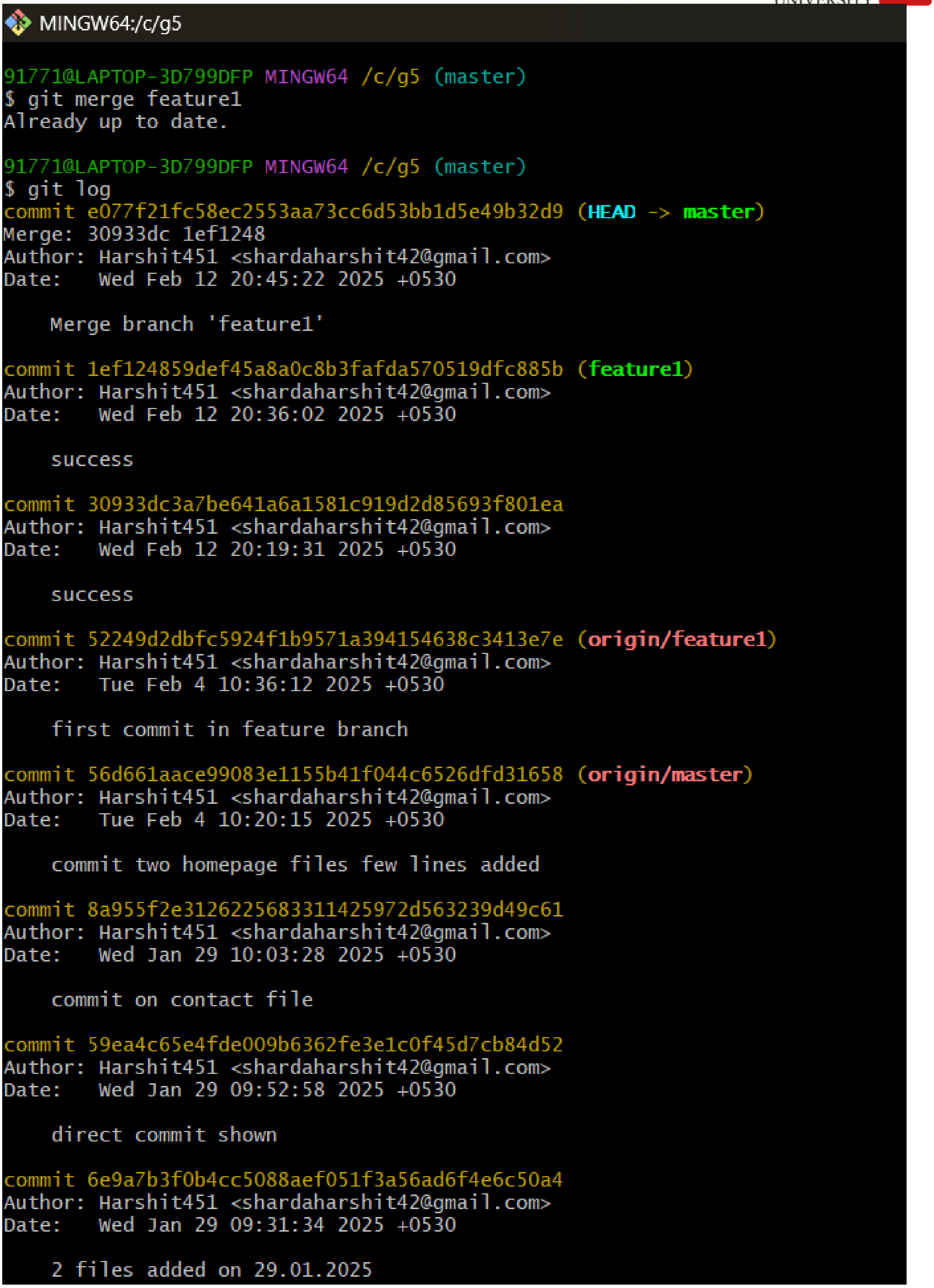
5. Merge the new branch into the main branch :

```
Git merge new-branch

git merge new-branch
```

Snapshots:



```
91771@LAPTOP-3D799DFP MINGW64 /c/g5 (feature1)
$ touch text.txt

91771@LAPTOP-3D799DFP MINGW64 /c/g5 (feature1)
$ git commit -m "success!"
On branch feature1
Your branch is up to date with 'origin/feature1'.

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

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

91771@LAPTOP-3D799DFP MINGW64 /c/g5 (feature1)
$ git status
On branch feature1
Your branch is up to date with 'origin/feature1'.

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

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

91771@LAPTOP-3D799DFP MINGW64 /c/g5 (feature1)
$ git commit -m "success"
On branch feature1
Your branch is up to date with 'origin/feature1'.

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

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

91771@LAPTOP-3D799DFP MINGW64 /c/g5 (feature1)
$ git add .

91771@LAPTOP-3D799DFP MINGW64 /c/g5 (feature1)
$ git commit -m "success"
[feature1 1ef1248] success
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 text.txt

91771@LAPTOP-3D799DFP MINGW64 /c/g5 (feature1)
$ git merge feature1
Already up to date.

91771@LAPTOP-3D799DFP MINGW64 /c/g5 (feature1)
$ git switch master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

91771@LAPTOP-3D799DFP MINGW64 /c/g5 (master)
$ |
```



```
91771@LAPTOP-3D799DFP MINGW64 /c/g5 (master)
$ git merge feature1
Already up to date.

91771@LAPTOP-3D799DFP MINGW64 /c/g5 (master)
$ git log
commit e077f21fc58ec2553aa73cc6d53bb1d5e49b32d9 (HEAD -> master)
Merge: 30933dc 1ef1248
Author: Harshit451 <shardaharshit42@gmail.com>
Date:   Wed Feb 12 20:45:22 2025 +0530

    Merge branch 'feature1'

commit 1ef124859def45a8a0c8b3fafda570519dfc885b (feature1)
Author: Harshit451 <shardaharshit42@gmail.com>
Date:   Wed Feb 12 20:36:02 2025 +0530

    success

commit 30933dc3a7be641a6a1581c919d2d85693f801ea
Author: Harshit451 <shardaharshit42@gmail.com>
Date:   Wed Feb 12 20:19:31 2025 +0530

    success

commit 52249d2dbfc5924f1b9571a394154638c3413e7e (origin/feature1)
Author: Harshit451 <shardaharshit42@gmail.com>
Date:   Tue Feb 4 10:36:12 2025 +0530

    first commit in feature branch

commit 56d661aace99083e1155b41f044c6526dfd31658 (origin/master)
Author: Harshit451 <shardaharshit42@gmail.com>
Date:   Tue Feb 4 10:20:15 2025 +0530

    commit two homepage files few lines added

commit 8a955f2e3126225683311425972d563239d49c61
Author: Harshit451 <shardaharshit42@gmail.com>
Date:   Wed Jan 29 10:03:28 2025 +0530

    commit on contact file

commit 59ea4c65e4fde009b6362fe3e1c0f45d7cb84d52
Author: Harshit451 <shardaharshit42@gmail.com>
Date:   Wed Jan 29 09:52:58 2025 +0530

    direct commit shown

commit 6e9a7b3f0b4cc5088aef051f3a56ad6f4e6c50a4
Author: Harshit451 <shardaharshit42@gmail.com>
Date:   Wed Jan 29 09:31:34 2025 +0530

    2 files added on 29.01.2025
```

Practical 4

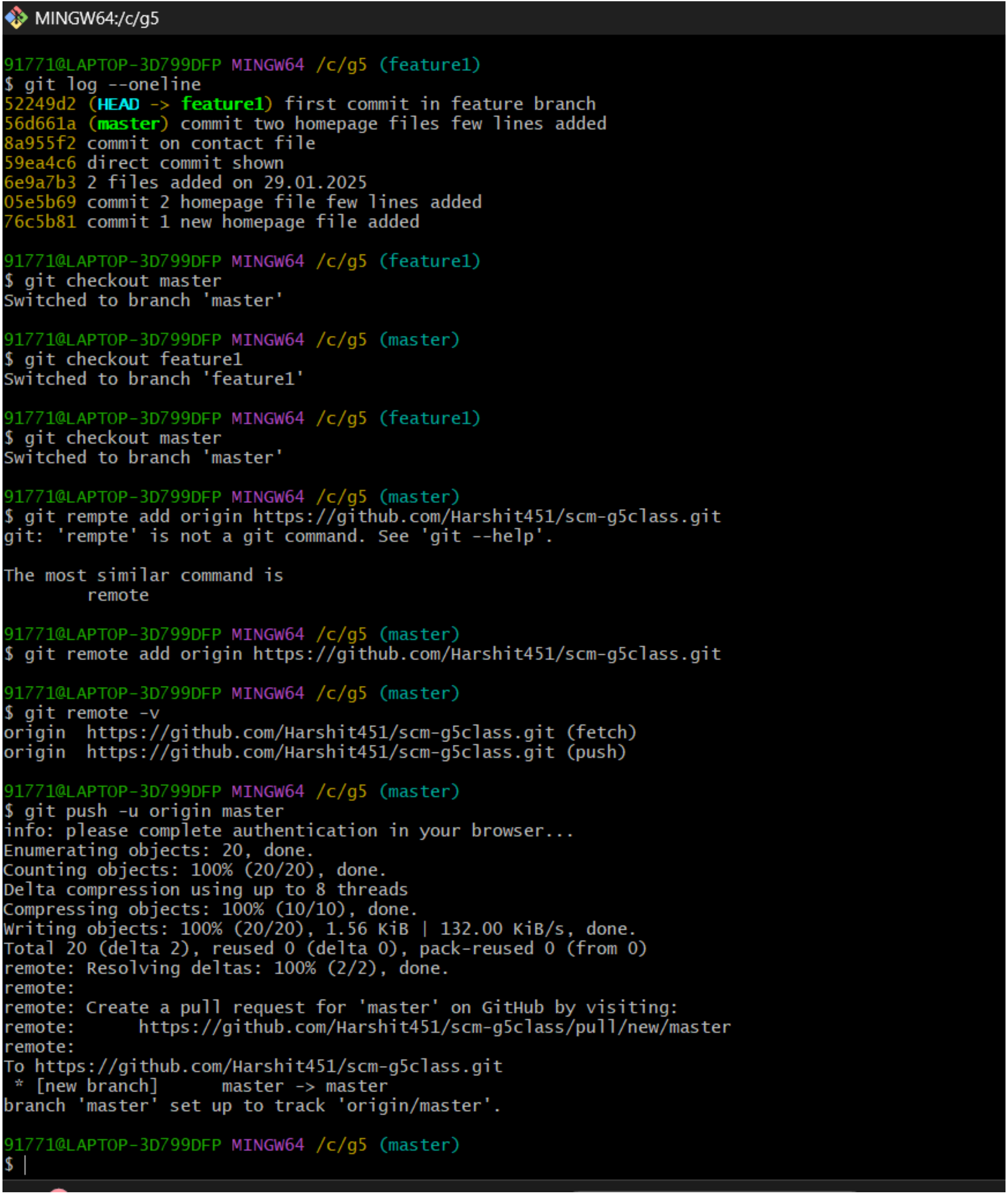
Aim:
To demonstrate push and pull operations in Git.

Theory:
Push transfers committed changes from the local repository to the remote repository, while pull retrieves updates from the remote repository.

Procedure:
○ Make changes in the local repository and commi them.
○ Push the changes to the remote repository using git push.
○ Make changes directly on the remote repository (e.g.,via GitHub interface).
○ Pull the changes to the local repository using git pull.

Tasks:
Provide screenshots of the push and pull operations.
Include the updated commit log.

<u>Screenshots</u>



```
91771@LAPTOP-3D799DFP MINGW64 /c/g5 (feature1)
$ git log --oneline
52249d2 (HEAD -> feature1) first commit in feature branch
56d661a (master) commit two homepage files few lines added
8a955f2 commit on contact file
59ea4c6 direct commit shown
6e9a7b3 2 files added on 29.01.2025
05e5b69 commit 2 homepage file few lines added
76c5b81 commit 1 new homepage file added

91771@LAPTOP-3D799DFP MINGW64 /c/g5 (feature1)
$ git checkout master
Switched to branch 'master'

91771@LAPTOP-3D799DFP MINGW64 /c/g5 (master)
$ git checkout feature1
Switched to branch 'feature1'

91771@LAPTOP-3D799DFP MINGW64 /c/g5 (feature1)
$ git checkout master
Switched to branch 'master'

91771@LAPTOP-3D799DFP MINGW64 /c/g5 (master)
$ git rempte add origin https://github.com/Harshit451/scm-g5class.git
git: 'rempte' is not a git command. See 'git --help'.

The most similar command is
        remote

91771@LAPTOP-3D799DFP MINGW64 /c/g5 (master)
$ git remote add origin https://github.com/Harshit451/scm-g5class.git

91771@LAPTOP-3D799DFP MINGW64 /c/g5 (master)
$ git remote -v
origin  https://github.com/Harshit451/scm-g5class.git (fetch)
origin  https://github.com/Harshit451/scm-g5class.git (push)

91771@LAPTOP-3D799DFP MINGW64 /c/g5 (master)
$ git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 20, done.
Counting objects: 100% (20/20), done.
Delta compression using up to 8 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (20/20), 1.56 KiB | 132.00 KiB/s, done.
Total 20 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/Harshit451/scm-g5class/pull/new/master
remote:
To https://github.com/Harshit451/scm-g5class.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

91771@LAPTOP-3D799DFP MINGW64 /c/g5 (master)
$ |
```