EXPERIMENT NO.1

AIM: Introduction to Devops

THEORY:

1. Evolution of DevOps

- Before DevOps ○ Traditionally, software development and IT operations worked in separate silos. ○ The Waterfall model led to long release cycles (months or years). ○ Developers focused on writing code, while Operations handled deployment, monitoring, and maintenance.
    - This separation often caused delays, miscommunication, and production failures.
- Agile Era
    - Agile methodologies (early 2000s) introduced iterative development and frequent releases. ○ While Agile improved development speed, operations still struggled to keep up with fast deployments.
- Birth of DevOps
    - Around 2009, the term DevOps (Development + Operations) emerged from the need for better collaboration. ○ Inspired by Agile principles and the lean manufacturing mindset, DevOps brought automation, continuous integration (CI), continuous delivery (CD), and monitoring into one culture. ○ It bridged the gap between development and operations using tools, processes, and a collaborative mindset.

2. **Need for DevOps**

- **Faster Time-to-Market:** Businesses needed to release features and updates rapidly to stay competitive.

- **High Deployment Frequency:** Users expect continuous improvements rather than big, rare updates.
- **Better Quality & Reliability:** Minimizing downtime and post-release issues became critical.
- **Efficient Collaboration:** Eliminate silos between dev, QA, and ops teams.
- **Automation to Reduce Manual Work:** Speed up builds, testing, deployments, and monitoring.
- **Scalability:** Modern applications needed to handle growing users without delays.

---

3. **About DevOps**

- **Definition:**
  DevOps is a **set of practices, principles, and tools** that integrates **software development (Dev)** and **IT operations (Ops)** to shorten the software development life cycle while delivering high-quality software continuously.
- **Core Principles:**
  1. **Collaboration & Communication** between teams.
  2. **Automation** of build, test, and deployment.
  3. **Continuous Integration (CI)** – merging code changes frequently.
  4. **Continuous Delivery (CD)** – making software ready for release anytime.
  5. **Monitoring & Feedback Loops** – to improve future releases.
- **Popular DevOps Tools:**
  Git, Jenkins, Docker, Kubernetes, Ansible, Terraform, Prometheus, Grafana, AWS, Azure DevOps.

---

4. **Advantages of DevOps**

1. **Faster Delivery of Software:** Frequent releases with minimal delays.
2. **Improved Quality:** Automated testing ensures fewer bugs in production.
3. **Better Collaboration:** Teams work as one unit with shared goals.

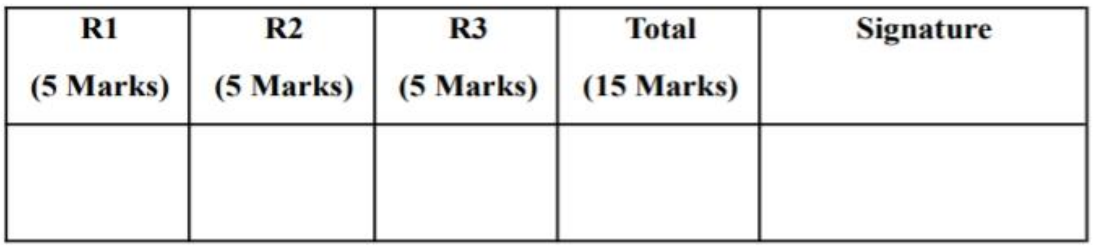
4. **Higher Efficiency:** Automation reduces repetitive manual tasks.
5. **Reduced Failures:** Continuous monitoring detects and fixes issues quickly.
6. **Scalability & Reliability:** Applications can handle large user bases without disruptions.
7. **Cost Optimization:** Fewer errors and efficient resource usage save money.

---

5. **Conclusion**

DevOps has transformed the software industry by bridging the gap between **development and operations**, fostering **collaboration, automation, and continuous delivery**.
It evolved as a response to the limitations of traditional software models and the fast-paced demands of modern businesses.

Marks & Signature:



R1 (5 Marks)	R2 (5 Marks)	R3 (5 Marks)	Total (15 Marks)	Signature

EXPERIMENT NO.2

AIM: To study and practice GIT commands for version control

THEORY:

Git is a free and open source distributed version control system designed to handle everything fromsmall to very large projects with speed and effciency. Git is easy to learn and has a tiny footprintwith lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, andClearCase with features like cheap local branching, convenient staging areas, and multiple workfows.

Some of the basic operations in Git are:

1.Initialize 2.Add 3.Commit 4.Pull 5.Push

Some advanced Git operations are: 1.Branching 2. Merging 3. Rebasing Installation

Installation of GIT

1)In windows, download GIT from https://git-scm.com/ and perform the straightforward installation.

2) In Ubuntu, install GIT using $sudo apt install git,

Confrm the version after installation $git –version Once installation is done,

open the terminal in Ubuntu and perform the following steps or in windows Right click and select Git bash here.

To perform version control, let us create a directory dvcs (Distributed version control system) and change directory to dvcs.

```
$ mkdir git-dvcs
```

```
$ cd git-dvcs/
```

Now check the user information using $ git confg –global

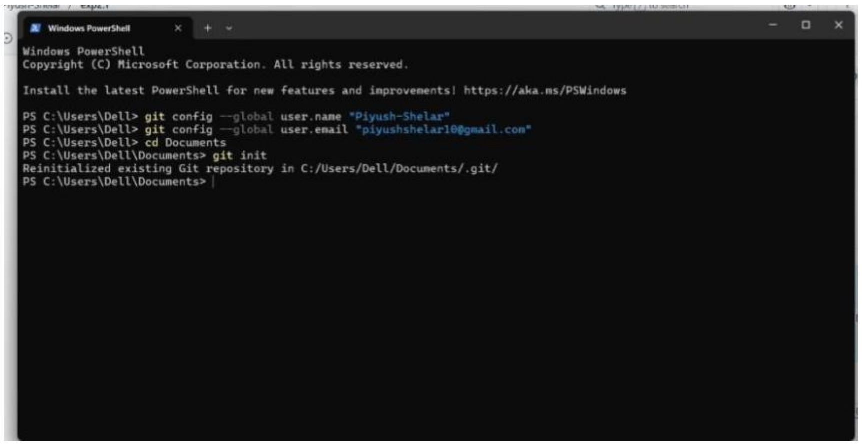
As there are no users defned, let us defne it using following two commands

$ git confg --global user.name "bhushan"

$ git confg --global user.email "bhushan,jadhav1@gmail.com"

Now, check the list of users

Commands: git init: The git init command creates a new Git repository. It can be used to convert an existing, unversioned project to a Git repository or initialize a new, empty repository. Most other Git commands are not available outside of an initialized repository, so this is usually the frst command you'll run in a new project.

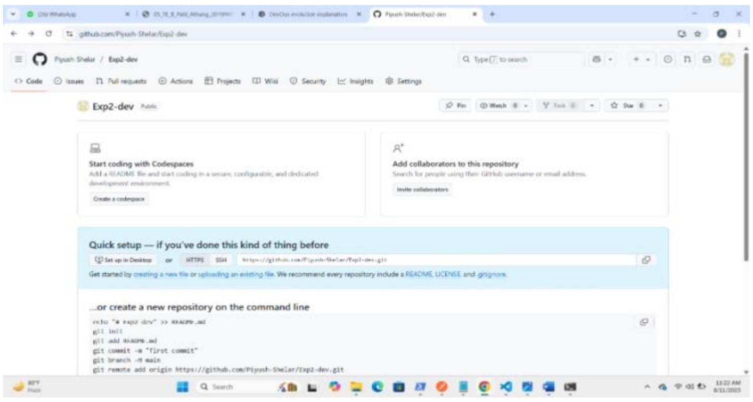


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Dell> git config --global user.name "Piyush-Shelar"
PS C:\Users\Dell> git config --global user.email "piyushshelar10@gmail.com"
PS C:\Users\Dell> cd Documents
PS C:\Users\Dell\Documents> git init
Reinitialized existing Git repository in C:/Users/Dell/Documents/.git/
PS C:\Users\Dell\Documents>
```

git clone: git clone is a Git command line utility which is used to target an existing repository and create a clone, or copy of the target repository. ... Cloning a local or remote repository. Cloning a bare repository. Using shallow options to partially clone repositories. Git URL syntax and supported protocols.



Exp2-dev Public

Start coding with Codespaces
Add a README file and start coding in a secure, configurable, and dedicated development environment.

Add collaborators to this repository
Search for people using their GitHub username or email address.

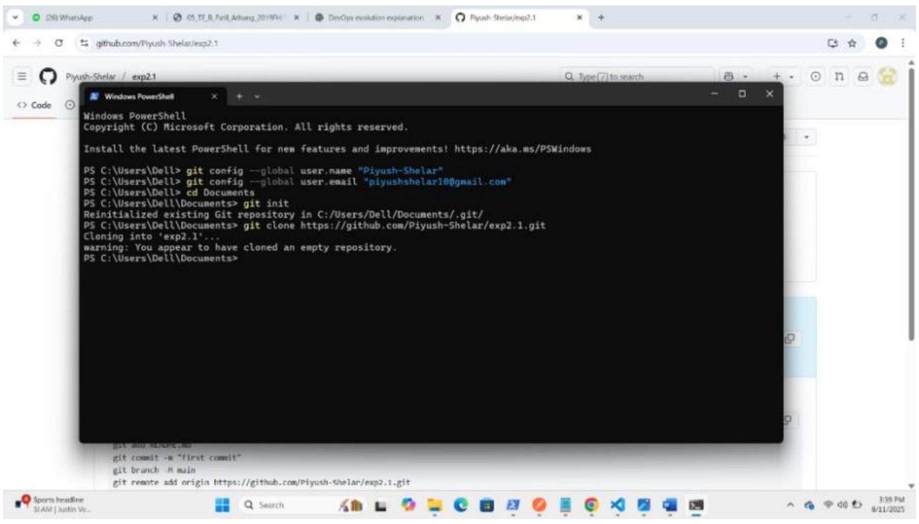
Quick setup — if you've done this kind of thing before

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 "# Exp2-dev" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Piyush-Shelar/Exp2-dev.git
```

Step 2: Paste it after git clone & hit enter



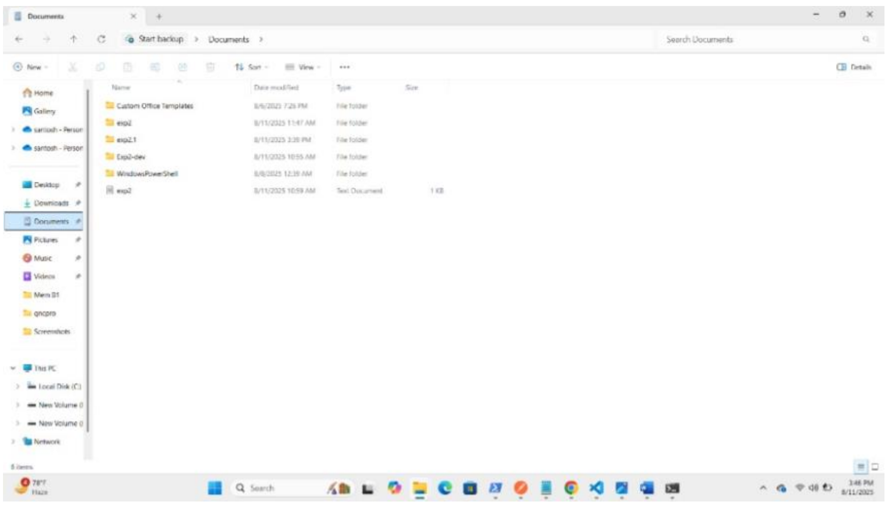
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Dell> git config --global user.name "Piyush-Shelar"
PS C:\Users\Dell> git config --global user.email "piyushshelar10@gmail.com"
PS C:\Users\Dell> cd Documents
PS C:\Users\Dell\Documents> git init
Reinitialized existing Git repository in C:/Users/Dell/Documents/.git/
PS C:\Users\Dell\Documents> git clone https://github.com/Piyush-Shelar/exp2.1.git
Cloning into 'exp2.1'...
warning: You appear to have cloned an empty repository.
PS C:\Users\Dell\Documents>
```

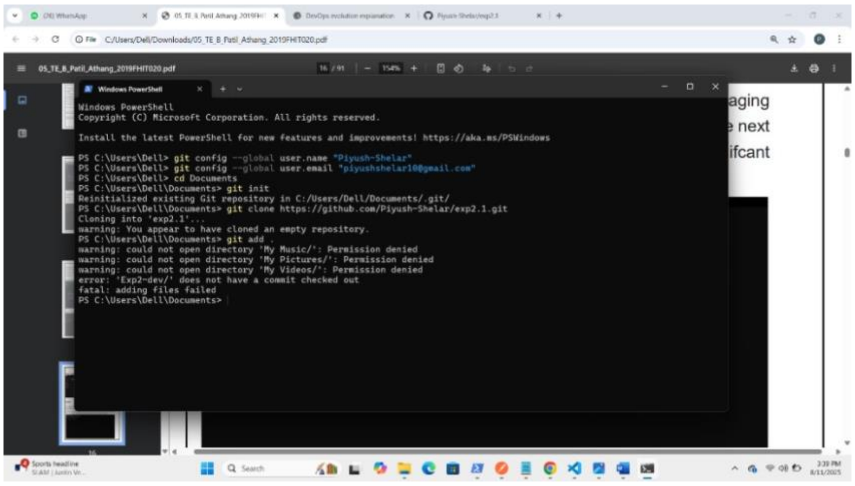
```
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Piyush-Shelar/exp2.1.git
```

Step 3: Repo folder is created in ur localhost



Name	Date modified	Type	Size
Custom Office Templates	8/6/2025 7:26 PM	File folder	
exp2	8/11/2025 11:47 AM	File folder	
exp2.1	8/11/2025 3:39 PM	File folder	
Exp2-dev	8/11/2025 10:55 AM	File folder	
WindowsPowerShell	8/8/2025 12:39 AM	File folder	
exp2	8/11/2025 10:59 AM	Text Document	1 KB

git add . The git add command adds a change in the working directory to the staging area. It tells Git that you want to include updates to a particular fle in the next commit. However, git add doesn't really affect the repository in any signifcant way—changes are not actually recorded until you run git commit.

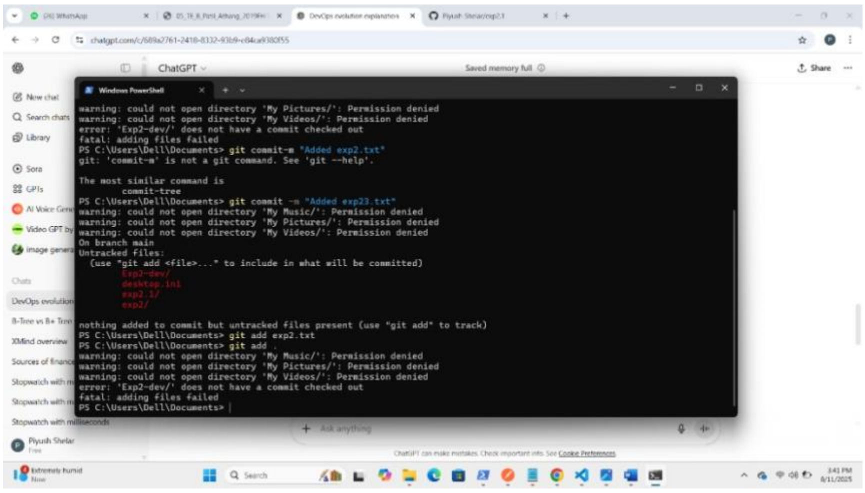


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Dell> git config --global user.name "Piyush-Shelar"
PS C:\Users\Dell> git config --global user.email "piyushshelar10@gmail.com"
PS C:\Users\Dell> cd Documents
PS C:\Users\Dell\Documents> git init
Reinitialized existing Git repository in C:/Users/Dell/Documents/.git/
PS C:\Users\Dell\Documents> git clone https://github.com/Piyush-Shelar/exp2.1.git
Cloning into 'exp2.1'...
warning: You appear to have cloned an empty repository.
PS C:\Users\Dell\Documents> git add .
warning: could not open directory 'My Music/': Permission denied
warning: could not open directory 'My Pictures/': Permission denied
warning: could not open directory 'My Videos/': Permission denied
error: 'Exp2-dev/' does not have a commit checked out
fatal: adding files failed
PS C:\Users\Dell\Documents>
```

git commit -m “message goes here” The "commit" command is used to save your changes to the local repository. ... Using the "git commit" command only saves a new commit object in the local Git repository. Exchanging commits has to be performed manually and explicitly (with the "git fetch", "git pull", and "git push" commands).

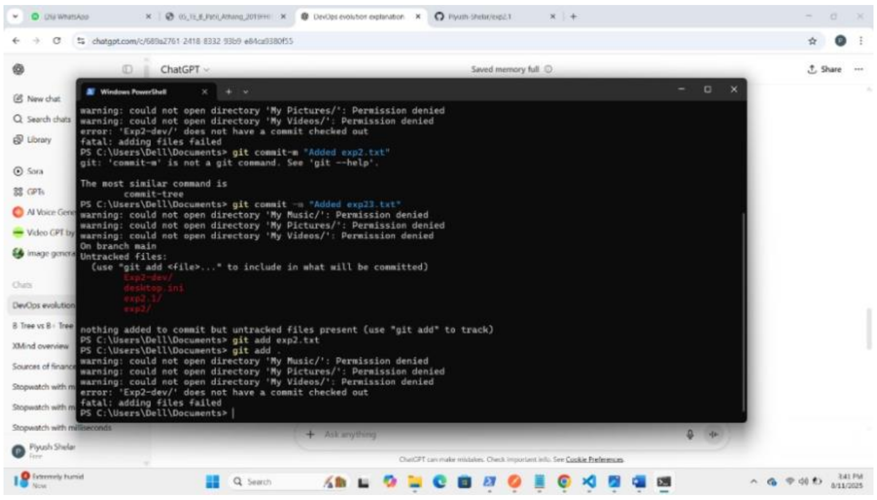


```
warning: could not open directory 'My Pictures/': Permission denied
warning: could not open directory 'My Videos/': Permission denied
error: 'Exp2-dev/' does not have a commit checked out
fatal: adding files failed
PS C:\Users\Dell\Documents> git commit-m "Added exp2.txt"
git: 'commit-m' is not a git command. See 'git --help'.

The most similar command is
        commit-tree
PS C:\Users\Dell\Documents> git commit -m "Added exp23.txt"
warning: could not open directory 'My Music/': Permission denied
warning: could not open directory 'My Pictures/': Permission denied
warning: could not open directory 'My Videos/': Permission denied
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Exp2-dev/
        desktop.ini
        exp2.1/
        exp2/

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\Dell\Documents> git add exp2.txt
PS C:\Users\Dell\Documents> git add .
warning: could not open directory 'My Music/': Permission denied
warning: could not open directory 'My Pictures/': Permission denied
warning: could not open directory 'My Videos/': Permission denied
error: 'Exp2-dev/' does not have a commit checked out
fatal: adding files failed
PS C:\Users\Dell\Documents>
```

git push origin main The git push command is used to upload local repository content to a remote repository. Pushing is how you transfer commits from your local repository to a remote repo. It's the counterpart to git fetch , but whereas fetching imports commits to local branches, pushing exports commits to remote branches

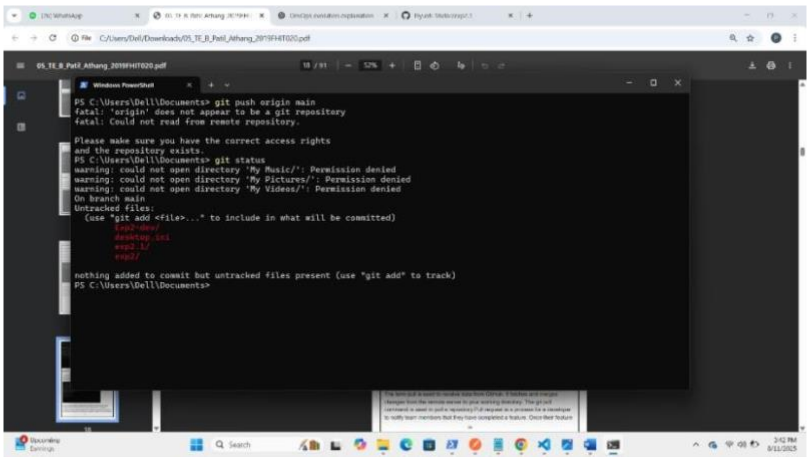


```
warning: could not open directory 'My Pictures/': Permission denied
warning: could not open directory 'My Videos/': Permission denied
error: 'Exp2-dev/' does not have a commit checked out
fatal: adding files failed
PS C:\Users\Dell\Documents> git commit-m "Added exp2.txt"
git: 'commit-m' is not a git command. See 'git --help'.

The most similar command is
        commit-tree
PS C:\Users\Dell\Documents> git commit -m "Added exp23.txt"
warning: could not open directory 'My Music/': Permission denied
warning: could not open directory 'My Pictures/': Permission denied
warning: could not open directory 'My Videos/': Permission denied
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Exp2-dev/
        desktop.ini
        exp2.1/
        exp2/

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\Dell\Documents> git add exp2.txt
PS C:\Users\Dell\Documents> git add .
warning: could not open directory 'My Music/': Permission denied
warning: could not open directory 'My Pictures/': Permission denied
warning: could not open directory 'My Videos/': Permission denied
error: 'Exp2-dev/' does not have a commit checked out
fatal: adding files failed
PS C:\Users\Dell\Documents> |
```

git status The git status command displays the state of the working directory and the staging area. It lets you see which changes have been staged, which haven't, and which fles aren't being tracked by Git.

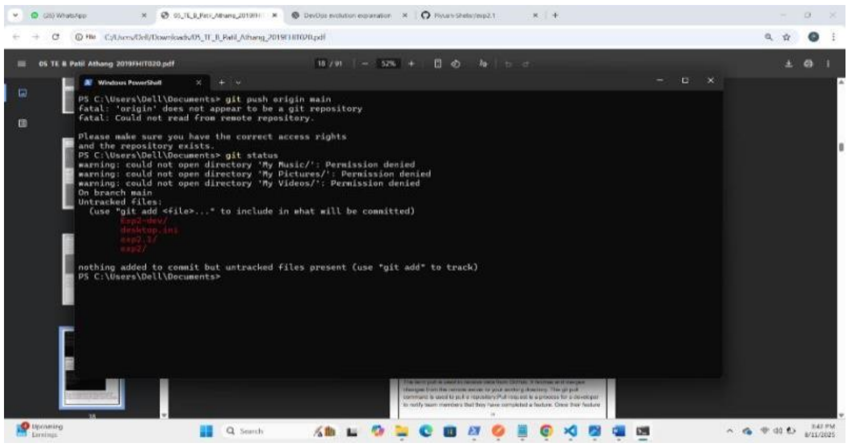


```
PS C:\Users\Dell\Documents> git push origin main
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
PS C:\Users\Dell\Documents> git status
warning: could not open directory 'My Music/': Permission denied
warning: could not open directory 'My Pictures/': Permission denied
warning: could not open directory 'My Videos/': Permission denied
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Exp2-dev/
        desktop.ini
        exp2.1/
        exp2/

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\Dell\Documents>
```

The term pull is used to receive data from GitHub. It fetches and merges changes from the remote server to your working directory. The git pull command is used to pull a repository.Pull request is a process for a developer to notify team members that they have completed a feature. Once their feature 15 branch is ready, the developer fles a pull request via their remote server account.



```
PS C:\Users\Dell\Documents> git push origin main
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
PS C:\Users\Dell\Documents> git status
warning: could not open directory 'My Music/': Permission denied
warning: could not open directory 'My Pictures/': Permission denied
warning: could not open directory 'My Videos/': Permission denied
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Exp2-dev/
        desktop.ini
        exp2.1/
        exp2/

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\Dell\Documents>
```

Conclusion:- Hence, basic commands of git version control was studied properly

Marks & Signature:

R1 (5 Marks)	R2 (5 Marks)	R3 (5 Marks)	Total (15 Marks)	Signature