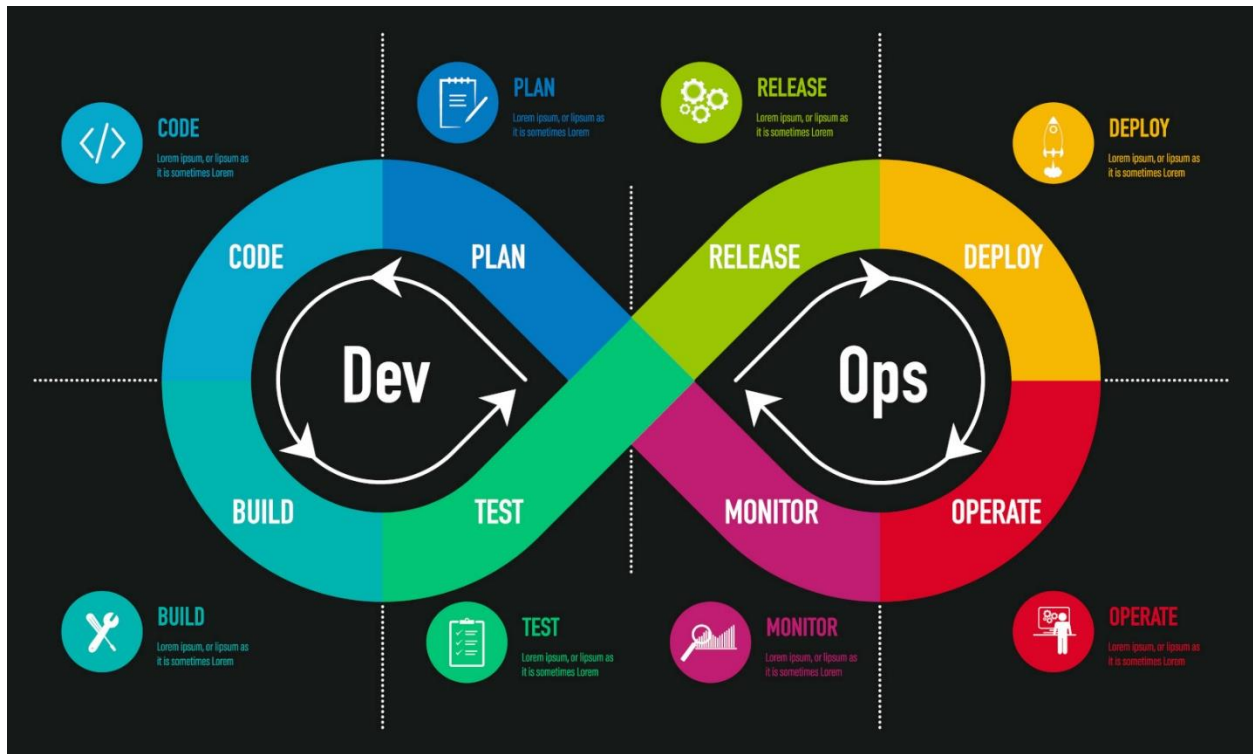Name: PRAKRITI RAWAT
SAP-ID: 500108285
BATCH: 2

DEVOPS

Name: Prakriti Rawat

SAP-ID: 500108285

Batch:2

Submitted to: Prateek Raj Gautam

Name: PRAKRITI RAWAT
SAP-ID: 500108285
BATCH: 2

Assignment – 1

Version Control Systems

Subversion, Mercurial, and GitHub

1. What is Version Control?

Version control is a system that records changes to files over time, allowing users to track modifications, revert to previous versions, and collaborate effectively. It is widely used in software development to manage source code and avoid conflicts between team members.

2. Why is Version Control Important?

- **Tracks changes**: Maintains a history of modifications.

- **Collaboration**: Allows multiple developers to work on the same project without conflicts.

- **Backup & Recovery**: Prevents data loss by storing previous versions.

- **Branching & Merging**: Enables experimentation with new features without affecting the main project.

Overview of SVN, Mercurial, and GitHub

1. Subversion (SVN)

Apache Subversion (SVN) is a centralized version control system where all project files are stored in a single repository on a server. Developers must **commit** and **update** changes through a central repository.

- **Pros**: Easy to set up, suitable for large enterprises.

- **Cons**: Requires an internet connection to access the repository, slower than distributed systems.

2. Mercurial (HG)

Mercurial is a distributed version control system (DVCS) designed for speed and scalability. Unlike SVN, every user has a complete copy of the repository.

- **Pros**: Fast, simple commands, efficient handling of large repositories.

- **Cons**: Less popular than Git, fewer hosting services.

3. GitHub (Git)

GitHub is a cloud-based platform that hosts Git repositories, allowing developers to collaborate on projects easily. Git itself is a distributed version control system similar to Mercurial but more widely used.

- **Pros**: Highly scalable, fast, supports branching and merging.

- **Cons**: Steeper learning curve than SVN and Mercurial.

💡 Key Difference:

Feature	SVN	Mercurial	Git (GitHub)
Type	Centralized	Distributed	Distributed
Performance	Slower	Faster than SVN	Fastest
Collaboration	Requires a central server	No central server needed	No central server needed
Popularity	Less common	Moderate	Most widely used

3. Subversion (SVN)

1. **Installed SVN** (TortoiseSVN-1.14.9)

Name: PRAKRITI RAWAT
SAP-ID: 500108285
BATCH: 2

2. Created a Repository

```
C:\Users\prakr>svnadmin create D:\DEVOPS\MySVNRepo

C:\Users\prakr>
```

3. Checked out the Repository

```
C:\Users\prakr>svn checkout file:///D:/DEVOPS/MySVNRepo D:\DEVOPS\MyProject
Checked out revision 0.
```

4. Add the file to version control

```
C:\Users\prakr>svn add D:\DEVOPS\MyProject\test.txt
A         D:\DEVOPS\MyProject\test.txt
```

5. Commit the file to SVN

```
C:\Users\prakr>svn commit -m "Added test.txt file" D:\DEVOPS\MyProject\test.txt
Adding         D:\DEVOPS\MyProject\test.txt
Transmitting file data .done
Committing transaction...
Committed revision 1.
```

6. View Repository History

```
C:\Users\prakr>svn log file:///D:/DEVOPS/MySVNRepo
------------------------------------------------------------------------
r1 | prakr | 2025-02-16 13:12:52 +0530 (Sun, 16 Feb 2025) | 1 line

Added test.txt file
------------------------------------------------------------------------
```

4. Mercurial (HG)

1. **Installed Mercurial**
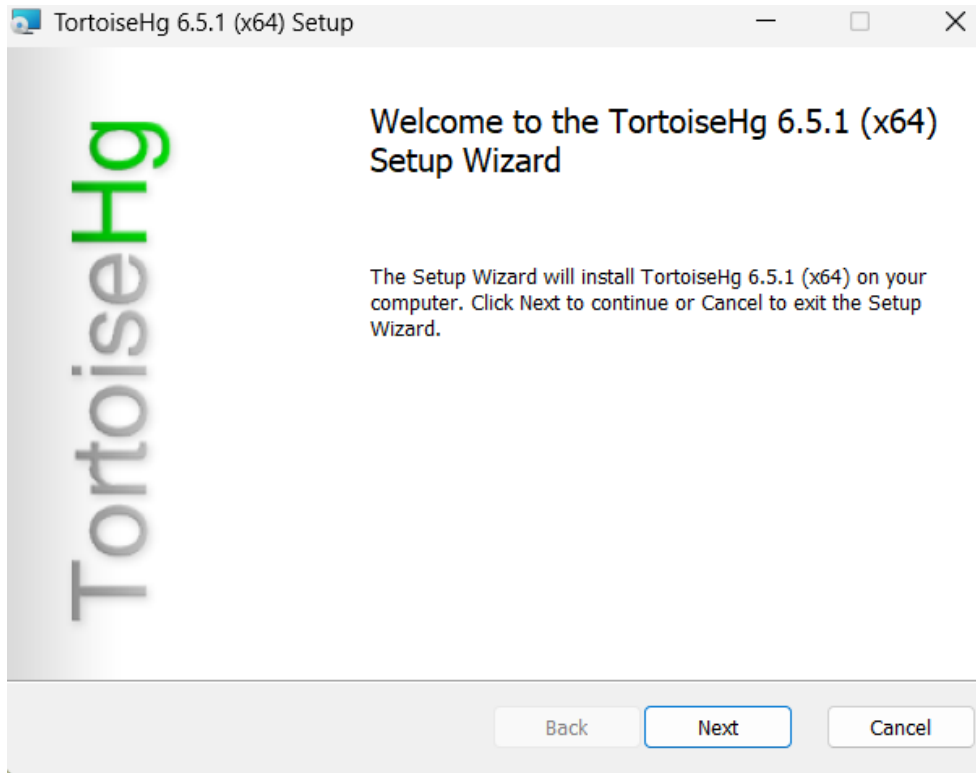

```
C:\Users\prakr>hg --version
Mercurial Distributed SCM (version 6.5.1)
(see https://mercurial-scm.org for more information)

Copyright (C) 2005-2023 Olivia Mackall and others
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

2. **Created a Repository**

```
C:\Users\prakr>mkdir D:\DEVOPS\MyHgRepo

C:\Users\prakr>cd D:\DEVOPS\MyHgRepo

C:\Users\prakr>hg init
```

3. **Set Username**

```
C:\Users\prakr>hg config --edit
```

```
PS C:\Users\prakr> hg config ui.username
prakriti <prakritirwtz@gmail.com>
PS C:\Users\prakr>
```

4. **Added files**

```
C:\Users\prakr>echo "This is a test file" > test.txt
```

```
C:\Users\prakr>hg add test.txt
```

5. **Commit files**

```
PS C:\Users\prakr> hg commit -m "Added test.txt file"
```

```
PS C:\Users\prakr> hg log
changeset:    0:5d42c50c61e4
tag:          tip
user:         prakriti <prakritirwtz@gmail.com>
date:         Sun Feb 16 13:34:43 2025 +0530
summary:      Added test.txt file
```

6. **View Repository History**

```
C:\Users\prakr>hg log
changeset:    0:5d42c50c61e4
tag:          tip
user:         prakriti <prakritirwtz@gmail.com>
date:         Sun Feb 16 13:34:43 2025 +0530
summary:      Added test.txt file
```
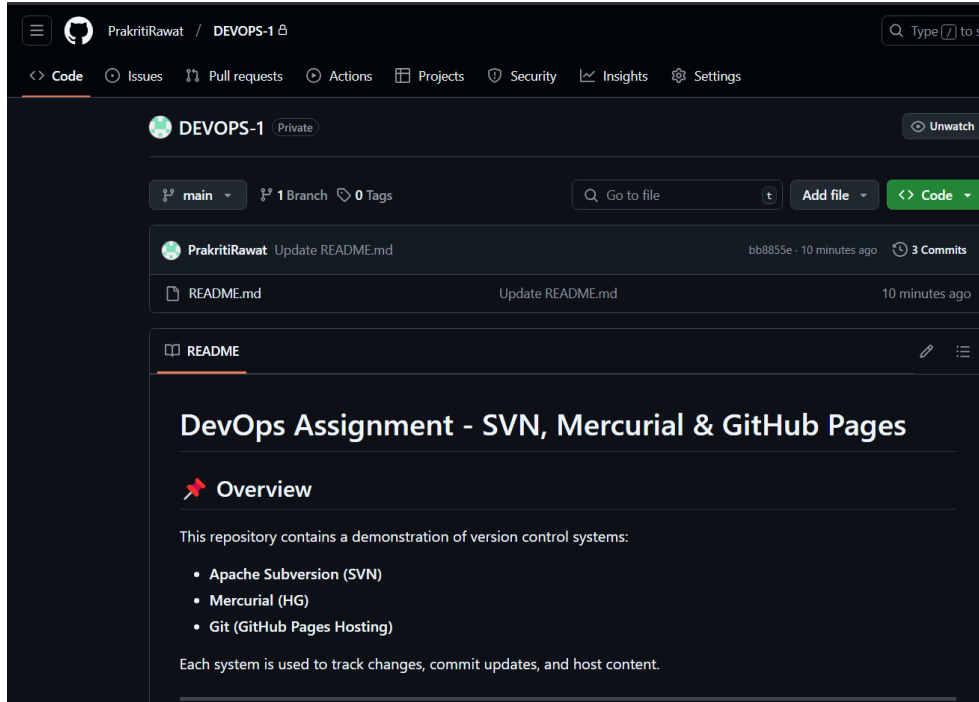
Name: PRAKRITI RAWAT
SAP-ID: 500108285
BATCH: 2

5. GitHub

1. **Created GitHub Repository**: DEVOPS-1

2. **Cloned the Repository**

```
PS C:\Users\prakr> cd D:\DEVOPS
PS D:\DEVOPS> git clone https://github.com/PrakritiRawat/DEVOPS-1.git
Cloning into 'DEVOPS-1'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 9 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (9/9), done.
PS D:\DEVOPS> cd DEVOPS-1
PS D:\DEVOPS\DEVOPS-1> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
PS D:\DEVOPS\DEVOPS-1>
```

3. **Navigated to Repository**

```
PS D:\DEVOPS\DEVOPS-1> cd D:\DEVOPS\DEVOPS-1
```

4. **Pull**

```
PS D:\DEVOPS\DEVOPS-1> cd D:\DEVOPS\DEVOPS-1
PS D:\DEVOPS\DEVOPS-1> git pull origin main
From https://github.com/PrakritiRawat/DEVOPS-1
 * branch            main       -> FETCH_HEAD
Already up to date.
PS D:\DEVOPS\DEVOPS-1>
```

Name: PRAKRITI RAWAT
SAP-ID: 500108285
BATCH: 2