Assignment 1

Subject: Devops

BY- HARDIK RAI

Sap id- 500102001

Batch 2 CCVT

Subversion (SVN) is a centralized version control system (VCS) that allows multiple users to collaborate on a project by storing all versions of files in a central repository. Every user must connect to this central repository to update their working copy and commit changes. Since SVN follows a client-server architecture, it ensures that all changes are stored in one location, making it easier to maintain control over the project.Step 1: Install SVN Command-Line Client

1. **Download** the **Slik SVN** client from:
   https://sliksvn.com/download/
2. **Install** it (select the option **"Add to PATH"** during installation).
3. **Verify Installation**
   Open **Command Prompt (cmd)** and run:

   svn –version

```
C:\Users\Hardik>svn --version
svn, version 1.14.2-SlikSvn (SlikSvn/1.14.2)
   compiled Mar 30 2023, 09:47:32 on x86_64-microsoft-windows10.0.20348

Copyright (C) 2022 The Apache Software Foundation.
This software consists of contributions made by many people;
see the NOTICE file for more information.
Subversion is open source software, see http://subversion.apache.org/

The following repository access (RA) modules are available:

* ra_svn : Module for accessing a repository using the svn network protocol.
  - with Cyrus SASL authentication
  - handles 'svn' scheme
* ra_local : Module for accessing a repository on local disk.
  - handles 'file' scheme
* ra_serf : Module for accessing a repository via WebDAV protocol using serf.
  - using serf 1.3.9 (compiled with 1.3.9)
  - handles 'http' scheme
  - handles 'https' scheme

The following authentication credential caches are available:

* Wincrypt cache in C:\Users\Hardik\AppData\Roaming\Subversion
```

Step 2: Create an SVN Repository

1. Open **Command Prompt (cmd)**.
2. Run the following command to **create an SVN repository**:
   svnadmin create C:\svn-repo

   - This will create a local repository at **C:\svn-repo.**

```
C:\Users\Hardik>svnadmin create C:\svn-repo
```

Step 3: Checkout the Repository (Clone)

1. **Create a working directory** for your project:
   mkdir C:\svn-project
   cd C:\svn-project

2. **Checkout the repository** (connect the working directory to SVN):
   svn checkout file:///C:/svn-repo

   - This downloads the repository into C:\svn-project.

```
C:\Users\Hardik>mkdir C:\svn-project

C:\Users\Hardik>cd C:\svn-project

C:\svn-project>svn checkout file:///C:/svn-repo
Checked out revision 0.
```

Step 4: Add and Commit Files

1. **Create a new file** inside C:\svn-project:
   echo "Hello SVN" > file.txt

2. **Add the file to SVN tracking**:
   svn add file.txt

3. **Commit the file** to the repository:
   svn commit -m "Added file.txt"

```
C:\svn-project>echo "Hello SVN" > file.txt
```

```
C:\svn-project>svn add file.txt
A         file.txt

C:\svn-project>svn commit -m "Added file.txt"
Adding         file.txt
Transmitting file data .done
Committing transaction...
Committed revision 1.
```

Step 5: Update Repository

If other team members make changes, you need to sync:
svn update

```
C:\svn-project>svn update
Updating '.':
At revision 1.
```

Step 6: View Commit History

To see all previous commits:
svn log

```
C:\svn-project>svn log
------------------------------------------------------------------------
r1 | Hardik | 2025-02-16 12:26:22 +0530 (Sun, 16 Feb 2025) | 1 line

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

Step 7: View File Changes

To check which files were modified:
svn status

```
C:\svn-project>svn status
?       svn-repo
```

Step 8: Delete Files

1. Remove a file from tracking:
   svn delete file.txt

2. Commit the changes:
   svn commit -m "Deleted file.txt"

```
C:\svn-project>svn delete file.txt
D         file.txt

C:\svn-project>svn commit -m "Deleted file.txt"
Deleting       file.txt
Committing transaction...
Committed revision 2.

C:\svn-project>
```

Mercurial (Hg), on the other hand, is a distributed version control system (DVCS) similar to Git. Unlike SVN, Mercurial does not rely on a central repository for managing versions. Instead, every developer maintains a full copy of the repository on their local machine, allowing them to work offline and commit changes without requiring an internet connection. This feature provides greater flexibility, especially for large, distributed teams working across different time zones**.**

Step 1: Install Mercurial

1. **Download Mercurial** from: https://www.mercurial-scm.org/downloads
2. **Install** it (select **"Add to PATH"**).
3. **Verify Installation**
   Open **Command Prompt (cmd)** and run:

hg --version

```
C:\Users\Hardik>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.
```

Step 2: Create a Mercurial Repository

1. **Create a new folder** for your project:

mkdir C:\hg-project
cd C:\hg-project

2. **Initialize a new repository**:

hg init

- This creates a Mercurial repository in C:\hg-project.

```
C:\Users\Hardik>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.

C:\Users\Hardik>mkdir C:\hg-project

C:\Users\Hardik>cd C:\hg-project

C:\hg-project>hg init
```

Step 3: Add and Commit Files

echo "Hello Mercurial" > file.txt

1. **Add the file to version control**:

hg add file.txt

2. **Commit the file**:

hg commit -m "Added file.txt"

```
C:\hg-project>echo "Hello Mercurial" > file.txt

C:\hg-project>hg add file.txt
```

Step 4: View Log and Status

- **Check commit history**:

hg log

- **Check which files are modified**: hg status

```
C:\hg-project>hg status
A file.txt
```

Step 5: Undo Changes

To undo changes before committing:

hg revert file.txt