Walchand College of Engineering, Sangli

Computer Science & Engineering

Third Year

**Course Name:**

**Software Engineering Tools**

**Course Code: 5CS351**

**Assignment No-5**

**Name : Akash Babu Misal**

**PRN: 21520005**

**Q1) What is version control system and why it is important?**

version control system (VCS) is a software tool that allows developers to manage changes to their codebase over time. It tracks every modification made to the codebase, making it easy to see who made changes, when they made them, and what those changes were. VCS also allows developers to collaborate on a codebase, providing a way to manage conflicts that arise when multiple people are working on the same codebase. There are two main types of VCS: centralized and distributed. In centralized VCS, there is a single repository that serves as the authoritative source of code. Developers check out files from the repository, make changes, and then check them back in. In distributed VCS, every developer has their own local repository, and changes are shared between repositories as needed. VCS is important for several reasons.

First, it provides a way to keep track of changes to a codebase over time. This makes it easier to understand how the codebase has evolved and who made which changes. Second, it allows multiple developers to work on the same codebase without overwriting each other's changes. VCS provides a way to merge changes from different developers and resolve conflicts that arise. Third, it provides a way to revert changes if something goes wrong. If a change causes a bug, it's easy to roll back to a previous version of the codebase. Finally, VCS can be used to manage releases of a codebase. By tagging specific versions of the codebase, it's easy to keep track of which version is running in production and to roll back to a previous version if needed.

**Q 2. Illustrate different types of version control system with example.**

Version Control System (VCS) is a software tool that helps in managing changes made to source code, documents, and other files. It enables tracking and managing changes, collaboration, and facilitates easy rollback of changes to a previous version. There are primarily two types of Version Control Systems - Centralized Version Control System (CVCS) and Distributed Version Control System (DVCS).

1. Centralized Version Control System (CVCS): In a CVCS, there is a central repository where all the files and changes are stored. Developers can check out a copy of the code, make changes to their copy, and then commit it back to the central repository. Examples: SVN (Apache Subversion): It is an open-source CVCS that allows developers to work collaboratively on a centralized repository. Perforce: It is a commercial version control system that provides centralized version control for large-scale projects.
2. Distributed Version Control System (DVCS): In a DVCS, every developer has a complete copy of the repository on their local system, which allows them to work independently and without the need for a central server.

Examples: Git: It is an open-source DVCS that allows developers to work offline and collaboratively on code. It is widely used for managing source code and version control in software development. Mercurial: It is also an open-source DVCS that is similar to Git and used for version control of source code and other files. It is widely used in the Python community. Both CVCS and DVCS have their advantages and disadvantages, and the choice of VCS depends on the requirements of the project and the team's preference.

**Q 3. Perform below operations using CVS**

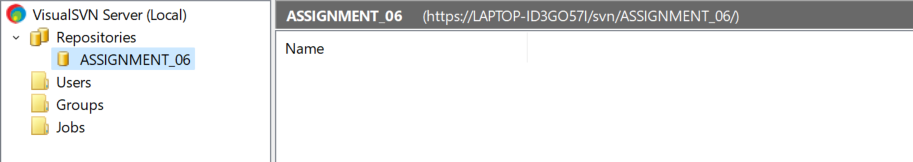
**a. cvs checkout**

**b. cvs update**

**c. cvs add**

**d. cvs remove**

**e. cvs commit**

****

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text

Description automatically generated

A screenshot of a computer

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Q 4. Differentiate Between The Git & SVN Repository?**

|  |  |
| --- | --- |
| GIT | SVN |
| Git is open source distributed vice control system developed by Linus Torvalds in 2005. It emphasis on speed and data integrity | Apache Subversion is an open source software version and revision control system under Apache license. |
| Git has a Distributed Model. | SVN has a Centralized Model. |
| In git every user has their own copy of code on their local like their own branch. | In SVN there is central repository has working copy that also make changes and committed in central repository. |
| In git we do not required any Network to perform git operation. | In SVN we required Network for runs the SVN operation. |
| Git is more difficult to learn. It has more concepts and commands. | SVN is much easier to learn as compared to git. |
| Git deals with large number of files like binary files that change quickly that why it become slow. | SVN control the large number of binary files easily. |
| In git we create only .git directory. | In SVN we create .svn directory in each folder. |
| It does not have good UI as compared to SVN. | SVN has simple and better user interface . |

**Q 5. What is “branch”, “tag” And “trunk” In SVN?**

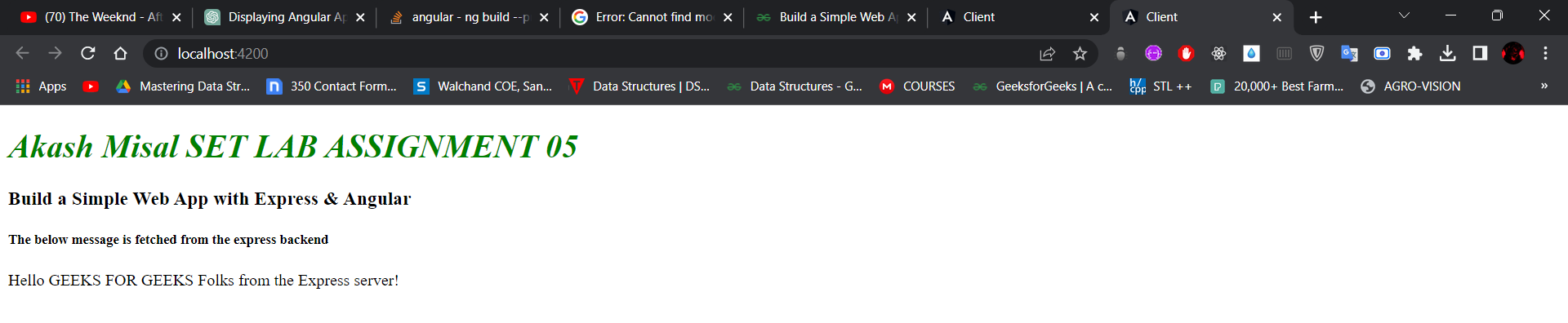
In SVN (Subversion), "branch", "tag", and "trunk" are three important concepts used to organize and manage source code. Branch: A branch is a copy of the source code that is created from the trunk or another branch. It allows developers to work on a separate copy of the codebase without affecting the main trunk. Branches are typically used to develop new features or make major changes to the codebase that may take a long time to complete. Once the work on the branch is complete, it can be merged back into the trunk or another branch. Tag: A tag is a label that is applied to a specific version of the source code. Tags are typically used to mark important releases or milestones in the project history. They allow developers to easily refer to specific versions of the codebase for debugging or testing purposes. Trunk: The trunk is the main line of development in SVN. It represents the latest version of the codebase that is stable and ready for release. Developers typically commit changes to the trunk on a regular basis, and the code is built and tested to ensure that it is stable and bug-free.

**Q.6. How CVS is different from SVN?**

|  |  |
| --- | --- |
| CVS | SVN |
| It is Concurrent Versions System. | It is SubVersioN. |
| CVS was developed by Dick Grune. | SVN was developed by CollabNet Inc. |
| It is open source and released with GNU General Public License. | It is open source and released with Apache License. |
| CVS uses RFC format to store repository. | SVN uses binary format to store repository |
| CVS has a systematic mechanism of maintaining tags and branches. | SVN uses branch, area and trunk to manage repository and there is no tag feature. |
| CVS is slow as compared to SVN. | SVN is fast due to all the files are completed backed up onto computer. |

**Q.7 Demonstrate a display the app version in angular.**

**Q.8 Build a simple web app with Express and Angular.**

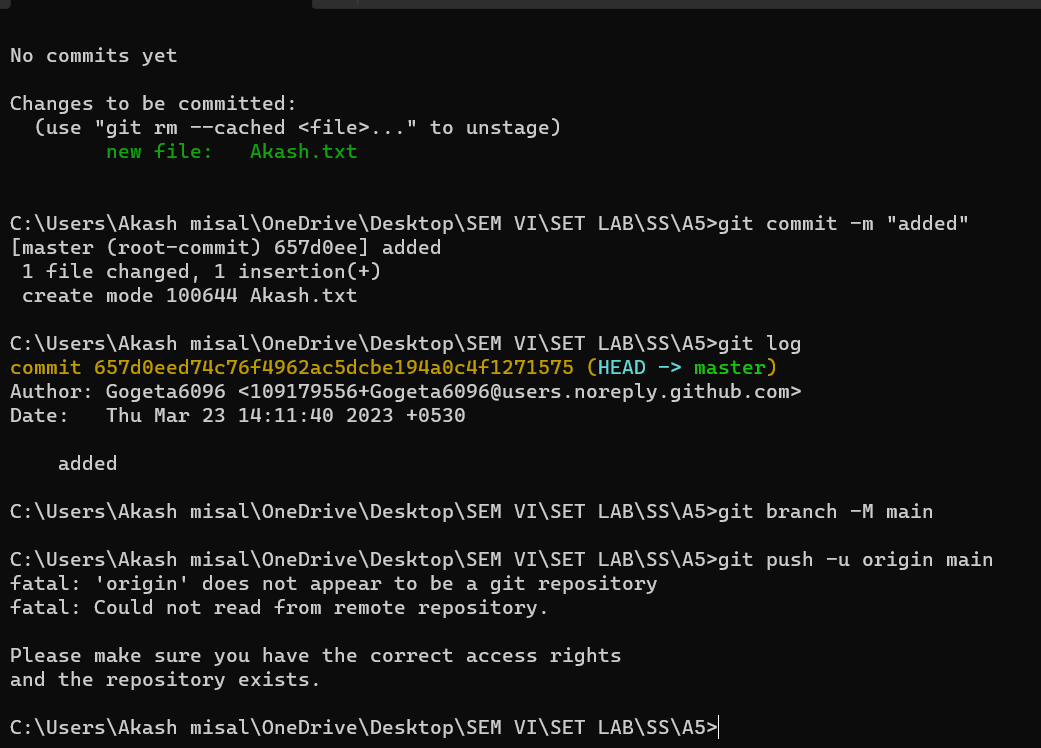
****

**Q.9 What is git version control?**

Git is a distributed version control system that allows developers to track changes in their code over time. With Git, developers can collaborate on projects, work on different versions of the same codebase, and easily manage and merge changes made by multiple people. Git works by creating snapshots of the codebase at different points in time, called commits. Each commit represents a set of changes to the codebase, along with a message describing those changes. Git also maintains a history of all the commits made to the codebase, allowing developers to track the evolution of the code over time.

One of the key advantages of Git is that it allows developers to work on different versions of the same codebase simultaneously. Developers can create branches of the codebase, which represent different versions of the code, and work on those branches independently. Git makes it easy to merge changes made on different branches back into the main codebase, ensuring that everyone is working with the latest version of the code. Git is widely used in the software development industry and has become the de facto standard for version control. It is open source and can be used on a wide range of platforms, including Windows, macOS, and Linux.

**Q.10 Demonstrate creation of repository in git.**

****