**Name:** Pranav Dilip Hingankar

**Batch:** T6

**PRN:** 2020BTECS00040

Assignment 5 - Understanding version control using VSS and Managing code using SVN

1. What is version control system and why it is important?

🡪

A version control system (VCS) is a software tool that helps developers manage changes to their code over time. It provides a way to keep track of changes to files, collaborate with other developers, and revert to previous versions of the code if necessary.

VCS is important because it helps to manage codebase efficiently and effectively. With version control, developers can work on the same codebase without worrying about conflicts, keeping track of changes becomes easier, and collaboration between developers is streamlined. It also helps in bug tracking, code review, and release management.

1. Illustrate different types of version control system with example.

🡪

There are several types of version control systems, including:

1. Centralized Version Control System (CVCS) - Examples include CVS, Subversion (SVN)
2. Distributed Version Control System (DVCS) - Examples include Git, Mercurial, Bitbucket, and GitHub.
3. Perform below operations using CVS
4. cvs checkout
5. cvs update
6. cvs add
7. cvs remove
8. cvs commit
9. Differentiate Between the Git & SVN Repository?

🡪

Git is a distributed version control system, while SVN is a centralized version control system. This means that with Git, each developer has their own copy of the repository, and changes can be made and committed locally. With SVN, there is a single central repository, and changes must be made and committed to that repository.

Git uses a branching and merging model, which allows developers to work on multiple versions of the code simultaneously, while SVN uses a trunk-based model, where all changes are made to a single trunk of the code. Git is known for its speed and performance, while SVN is known for its stability and ease of use.

1. What is “branch”, “tag” And “trunk” In SVN?

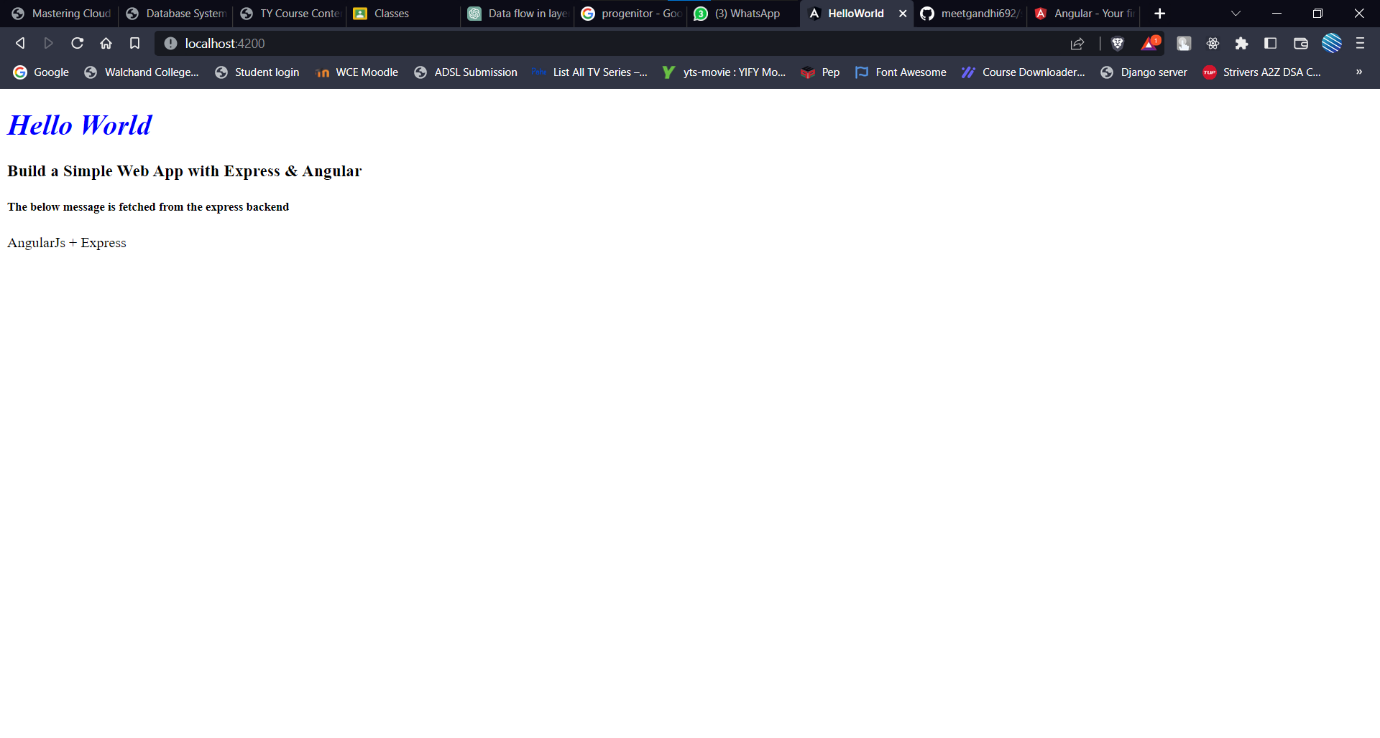
🡪

1. In SVN, a branch is a copy of the codebase that can be developed independently of the main codebase. This allows developers to work on new features or bug fixes without affecting the main codebase. When the work on a branch is complete, the changes can be merged back into the main codebase.
2. A tag is a specific version of the codebase that is marked as being significant, such as a release or a milestone. Tags are used to create snapshots of the codebase at specific points in time.
3. A trunk is the main branch of the codebase in SVN. It represents the current state of the codebase and serves as the basis for creating branches and tags. All development work is typically done on the trunk, and changes made to the codebase are committed to the trunk. The trunk can also be branched to create development branches or tags can be created from specific revisions of the trunk to create a stable release.
4. How CVS is different from SVN?

CVS (Concurrent Versions System) and SVN (Subversion) are both centralized version control systems, but there are some differences between them.

1. CVS uses a "lock-modify-unlock" model, where files are locked before they can be modified. SVN uses a "copy-modify-merge" model, where each developer has their own copy of the codebase and changes can be merged together.
2. SVN supports atomic commits, meaning that changes to multiple files can be committed as a single transaction. CVS does not have this feature.
3. SVN supports renaming and moving of files without losing the history of the file. CVS does not support this feature.
4. SVN has better support for binary files, while CVS is better suited for text files.
5. Demonstrate a display the app version in angular.
6. Build a simple web app with Express and Angular.

<https://github.com/meetgandhi692/set-angularjs.git>



1. What is git version control?

🡪

Git is a distributed version control system that is used to manage source code and track changes made to files over time. It allows developers to work on the same codebase and merge changes made by multiple developers. With Git, each developer has a complete copy of the repository on their local machine, which means they can work on the codebase even when they don't have an internet connection. Git also allows developers to create branches, which are separate versions of the codebase that can be developed independently of the main codebase.

1. Demonstrate creation of repository in git.

🡪

To create a new Git repository, follow these steps:

1. Open the command prompt or terminal and navigate to the directory where you want to create the repository.
2. Type the command "git init" and press enter. This will create a new Git repository in the current directory.
3. Add the files you want to track to the repository using the "git add" command. For example, to add all files in the current directory, use the command "git add ."
4. Commit the changes using the "git commit" command. For example, to commit all changes with a message, use the command "git commit -m 'Initial commit'"
5. Optionally, connect the repository to a remote server using the "git remote add" command. For example, to connect to a repository on GitHub, use the command "git remote add origin <repository url>"
6. Push the changes to the remote repository using the "git push" command. For example, to push the changes to the master branch on the remote repository, use the command "git push -u origin master".

