Project Title: Version control system

Category: Command line interface (CLI) application, utility application.

PURPOSE

The purpose of developing this “version control system” is to help people organise their files, maintain multiple versions of their files and keep track of the changes in their files.

SCOPE

It will help users to organise files, maintain multiple versions of files and primarily keep track of changes in files. It will allow developers working on large software projects to keep track of new lines of code added to the project, changes made in the current code and if needed, revert back to the previous version/release of software. It will facilitate and ease the collaboration process of multiple developers working on a large software project.

INTRODUCTION

Introduction contain the following sub categories:

Existing System

Existing system is a manual system and is very ineffective. Manual systems involve no automation, and keeping track of changes to file is manual and error prone. Maintaining, organising and tracking changes in the files in a large project is a tedious process under the manual system. Existing system has the following disadvantages:

- It is manual in nature - User has to take care of every aspect of organising, maintaining and tracking of changes in files.

- Tracking of changes in files is very difficult and error prone.

- Existing system is time consuming since a lot of manual work is required. Thus for developers working on large projects, it leads to reduced productivity.

- Maintaining multiple versions of files is difficult, tedious and memory intensive.

- Existing system doesn’t provide a way to keep track of changes, and who made the changes to files in a software project on which multiple developers are working together.

- Debugging in large software projects under the existing system is very difficult and time consuming.

- Maintaining a full backup of the project under the current system is memory intensive, especially for large software projects.

Proposed System

The version control system aims to make management, organisation and tracking of changes in files more efficient and easier than the existing system. The proposed system will consist of the following modules which will help to overcome the limitations of existing manual system:

- **Intuitive command line interface:** The version control system will have an intuitive command line interface and a set of commands which will help users to interact with the VCS with ease.

- **Automation:** Users can automate on top of already provided automation by version control system (VCS) using custom command line scripts in languages such as bash, python etc.

- **Log:** Version control system will maintain a log of changes made to various files tracked by the VCS. Users will be able to view the history of changes made in a chronological order.

- **Branch:** Version control system will allow users to make multiple versions or copies of the same file in a memory efficient way. Users can then independently make changes in each of the copies.

- **Diff:** Version control system will help developers to debug their code easily and spot bugs in the code easily by comparing files before and after the change. This will help developers to easily spot the change which caused the bug.

- **Memory efficiency:** Only changes made to a file are stored rather than multiple copies of a single file. Thus reducing the memory requirement as compared to the existing system.

ADVANTAGES

The advantages of using a version control system are:

- **Time saving and productivity increase:** Maintaining, organising and tracking of changes to files in a large project becomes easy, fast and efficient, thus saving developer time and enhancing developer productivity.

- **Easy collaboration:** Collaboration becomes easy and smooth in projects using version control systems.

- **Accountability:** In large projects, where multiple developers are working together, version control systems help establish accountability. Using the version control system, anyone can know which developer is responsible for certain changes in codebase.

- **Easy Rollback:** Rollback to the previous version/release of the codebase is easy.

- **Authentication:** The system ensures any data corruption due to some unwanted external factor is detected.

FUNCTIONAL REQUIREMENTS

Functional requirements of the version control system are:

- Version control system should have an intuitive command line interface.

- Commands should be easy to use and can be used by other programs for automation.

- **init:** The user should be able to initialize the version control system in any directory using a single easy command.

- **Log:** The user should be able to obtain the history log of changes made to a file or a whole directory using a single command. The history log should provide useful and descriptive information such as date and time at which commit was made, commit message, user name and email address who made the commit and the changes which were made in the commit.

- **Hash-object:** The system should be able to compute the object ID of a file using the contents of a file and efficient hashing mechanism. The object ID generated for each file should be unique throughout the VCS tracked directory.

- **Error checking:** The system should be able to detect unwanted corruption of files using the object ID generated.

- **Checkout:** User should be able to create a new branch of the directory. Any changes made to the new branch will not affect the original branch.

- A user should be able to temporarily change to the previous version of a codebase,ie, state before certain changes were made.

NON FUNCTIONAL REQUIREMENTS

- Fast detection of unwanted data corruption of files tracked by the version control system.

- Maximum time availability
- Easy and smooth scalability
- Scalable and better API design will allow easy and robust integration with other programs/softwares.
- Error free backup of files.

SOFTWARE TOOLS

- **Programming Language:** Python, Bash
- **Client:** Any unix based command line, windows powershell.
- **Development Tools:** Visual Studio Code, Vim.

DEPLOYMENT

Operating System: Linux, UNIX, Windows 8 and above
Interpreter: Python 3.8 and above

HARDWARE SPECIFICATION

Processor: Intel Core i5
RAM: 8 GB
Hard Disk: 1 TB