What is a VCS (Version Control System)?

There are several Version Control Systems (VCS) available, including Git, Subversion (SVN), and Mercurial. In this project, we will be using Git, which is one of the most popular and widely used VCS tools.

A VCS works by tracking changes to files over time, allowing developers to keep a complete history of modifications. It stores every version of the project, making it possible to revert to previous states, compare versions, and see who made which changes. This is particularly useful when working in teams, as multiple developers can collaborate simultaneously without overwriting each other's work. In Git, changes are stored in commits, which represent snapshots of the project at specific points in time.

To manage these versions, each project or file will have version numbers. These are typically in the format major.minor.patch (e.g., 1.0.0). The major version should increase when there are significant changes or overhauls, the minor version when new features are added that remain backward compatible, and the patch number for bug fixes or small updates. Version numbers help maintain clarity on the state of the project, making it easy to see when updates occurred and what type of change was made.

Why Use a VCS?

A VCS offers several critical benefits:

Collaboration: Developers can work simultaneously on the same codebase, using branches to separate different features or tasks. This enables seamless teamwork without conflicts.

History & Backup: Every change is logged, providing a clear history of all modifications. This means developers can undo mistakes, restore previous versions, or track down bugs that were introduced.

Experimentation: Developers can create branches to test new features or ideas without impacting the main project. Once the feature is stable, the changes can be merged back into the main version.

Transparency: A VCS makes it easy to see who made specific changes, which is important for accountability and understanding the evolution of the code.

By using Git and proper version control practices, the team can work more efficiently, track progress, and ensure that the project remains stable and organized throughout its lifecycle.