index.html

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Version Control System (Git & GitHub)</title>
</head>

<body>

    <h1 id="main-heading">Version Control System</h1>

    <p id="vcs-para">Version Control System (VCS), also known as Source Code Management (SCM) is a fundamental tool used in software development and other collaborative projects. It is a system that records changes to a file or set of files over a period of time. It allows allows to track the history of changes and manage multiple versions of a project.</p>

    <h2 id="sub-heading">Working of Version Control System:</h2>

    <ul>

        <li><strong>Repository</strong>: A VCS starts with a central storage location called a "repository". This repository holds all the project files and their entire history.</li>

        <li><strong>Local Copy</strong>: Each developer working on the project creates a personal "local copy" of the repository on their computer. This local copy contains all the project files and their history.</li>

        <li><strong>Making Changes</strong>: Developers can make changes to the project files in their local copy. These changes can include adding, modifying or deleting files.</li>

        <li><strong>Committing Changes</strong>: After making changes, developers create a "commit". A commit is like taking a snapshot of the project at that moment. It records what changes were made and includes a message explaining the purpose of the changes.</li>

        <li><strong>Tracking Changes</strong>: The VCS continuously tracks the differences between commits, creating a history of changes over time. This history helps developers understand how the project evolved and why specific changes were made.</li>

        <li><strong>Branching and Merging</strong>: Developers can create separate "branches" to work on specific features of fixes independently. Once a branch is complete, its changes can be "merged" back into the main project, combining the changes from different branches.</li>

        <li><strong>Conflict Resolution</strong>: When multiple developers make conflicting changes (example: editing the same line of code), the VCS helps identify these conflicts, allowing developers to resolve them systematically.</li>

        <li><strong>Collaboration</strong>: Developers can share their commits with others by "pushing" their changes to the central repository. They can also "pull" changes made by others to update their local copies, enabling collaboration on a shared codebase.</li>

        <li><strong>History and Versioning</strong>: The VCS maintains a detailed history of all commits and changes made to the project. Developers can use this history to understand the context behind each modification and to roll back to a previous state if needed.</li>

        <li><strong>Tagging and Releases</strong>: Developers can "tag" specific commits to mark significant milestones or releases. These tags help identify stable versions of the projects</li>

        <li><strong>Remote Repository</strong>: In many cases, VCS systems support "remote repositories" hosted on servers. These remote reposito&ies allows for distributed collaborations, enabling developers to share their work with others even if they are not in the same physical location.</li>

    </ul>

    <h1 id="git-heading">GIT (Global Information Tracker)</h1>

    <p id="git-para">Git is a widely used distributed version control system (DVCS) that is essential in modern software development. It was created by Linus Torvalds in 2005 and has since become the industry standard for source code management. Git is known for its speed, flexibility, and powerful branching capabilities.</p>

    <img src="git-lifecycle.png" alt="GIT Lifecycle Diagram">

</body>

</html>
```

style.css

```css
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: cursive;
}

#main-heading {
    text-align: center;
    color: 🟥 red
}

#vcs-para {
    font-size: 18px;
    text-align: justify;
    margin: 10px 15px;
}

ul {
    margin: 10px 15px;
}

li {
    margin: 10px 15px;
    font-size: 18px;
}

#sub-heading {
    text-align: center;
    color: 🟦 blue
}

strong {
    color: 🟩 green
    font-size: 18px;
}
```

```css

#git-heading {
    text-align: center;
    color: 🟥 red
}

#git-para {
    font-size: 18px;
    text-align: justify;
    margin: 10px 15px;
}

img {
    display: block;
    margin: 15px auto;
    width: 50%;
}
```

Version Control System

Version Control System (VCS), also known as Source Code Management (SCM) is a fundamental tool used in software development and other collaborative projects. It is a system that records changes to a file or set of files over a period of time. It allows allows to track the history of changes and manage multiple versions of a project.

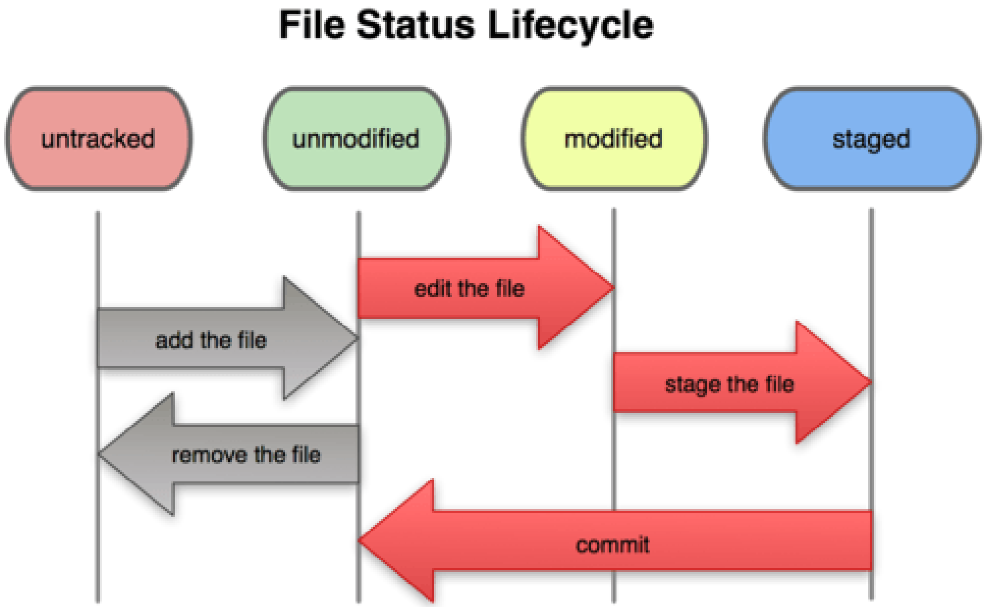
Working of Version Control System:

- **Repository**: A VCS starts with a central storage location called a "repository". This repository holds all the project files and their entire history.
- **Local Copy**: Each developer working on the project creates a personal "local copy" of the repository on their computer. This local copy contains all the project files and their history.
- **Making Changes**: Developers can make changes to the project files in their local copy. These changes can include adding, modifying or deleting files.
- **Committing Changes**: After making changes, developers create a "commit". A commit is like taking a snapshot of the project at that moment. It records what changes were made and includes a message explaining the purpose of the changes.
- **Tracking Changes**: The VCS continuously tracks the differences between commits, creating a history of changes over time. This history helps developers understand how the project evolved and why specific changes were made.
- **Branching and Merging**: Developers can create separate "branches" to work on specific features of fixes independently. Once a branch is complete, its changes can be "merged" back into the main project, combining the changes from different branches.
- **Conflict Resolution**: When multiple developers make conflicting changes (example: editing the same line of code), the VCS helps identify these conflicts, allowing developers to resolve them systematically.

- **Collaboration**: Developers can share their commits with others by "pushing" their changes to the central repository. They can also "pull" changes made by others to update their local copies, enabling collaboration on a shared codebase.
- **History and Versioning**: The VCS maintains a detailed history of all commits and changes made to the project. Developers can use this history to understand the context behind each modification and to roll back to a previous state if needed.
- **Tagging and Releases**: Developers can "tag" specific commits to mark significant milestones or releases. These tags help identify stable versions of the projects
- **Remote Repository**: In many cases, VCS systems support "remote repositories" hosted on servers. These remote reposito&ies allows for distributed collaborations, enabling developers to share their work with others even if they are not in the same physical location.

GIT (Global Information Tracker)

Git is a widely used distributed version control system (DVCS) that is essential in modern software development. It was created by Linus Torvalds in 2005 and has since become the industry standard for source code management. Git is known for its speed, flexibility, and powerful branching capabilities.


File Status Lifecycle
untracked
unmodified
modified
staged
edit the file
add the file
stage the file
remove the file
commit