**Implementing Basic Version Control for a Project Using Git**

**Project Selection**

For this, we will create a simple "To-Do List" application. The project will consist of HTML, CSS, and JavaScript files.

**Steps**

1*. Set Up the Project Directory*

First, we create a directory for our project:

=> mkdir todo-list-app

=>cd todo-list-app

2. *Initialize a Git Repository*

Initialize the Git repository in the project directory:

* git init

3. *Create Initial Project Files*

Create the initial files for the project:

=>touch index.html styles.css app.js

Add some basic content to these files:

index.html

styles.css

app.js

4. *Stage and Commit Initial Files*

Stage the initial files for commit:

* git add index.html styles.css app.js

Commit the changes:

* git commit -m "Initial commit with basic project structure"

5. *Implement Basic Features*

Update app.js to allow adding new items:

Stage and commit the changes:

* git add app.js
* git commit -m "Add feature to add items to the to-do list"

6. *Create a Branch for a New Feature*

To add a new feature (e.g., removing items), create a new branch:

* git checkout -b remove-items-feature

7. *Implement and Commit the New Feature*

Update app.js to add item removal functionality:

Stage and commit the changes:

* git add app.js
* git commit -m "Add feature to remove items from the to-do list"

8. *Merge the Feature Branch into Main*

Switch back to the main branch and merge the new feature:

* git checkout main
* git merge remove-items-feature

9. *Push Changes to a Remote Repository*

Add the remote repository:

* git remote add origin <repository-url>

Push the changes:

* git push -u origin main

**Benefits of Version Control in Project Management and Collaboration**

* **Collaboration**: Git enables multiple developers to work on the same project simultaneously without interfering with each other’s work.
* **History Tracking**: Every change is recorded, providing a detailed history of the project’s evolution, which is useful for debugging and understanding the development process.
* **Branching and Merging:** Developers can create branches to work on new features or bug fixes independently, then merge them into the main codebase once they are ready.
* **Backup and Recovery:** The repository serves as a backup, allowing recovery of previous versions of the project.
* **Accountability:** Commits are linked to authors, making it clear who made specific changes and facilitating accountability within the team.