Experiment 08: Change specification and use any SCM Tool to make different versions

Learning Objective: Students will able to use SCM Tool to make handle versioning of projects

Theory

Software configuration management: The traditional software configuration management (SCM) process is looked upon by practitioners as the best solution to handling changes in software projects. It identifies the functional and physical attributes of software at various points in time, and performs systematic control of changes to the identified attributes for the purpose of maintaining software integrity and traceability throughout the software development life cycle.

The SCM process further defines the need to trace changes, and the ability to verify that the final delivered software has all of the planned enhancements that are supposed to be included in the release. It identifies four procedures that must be defined for each software project to ensure that a sound SCM process is implemented. They are:

1. Configuration identification
2. Configuration control
3. Configuration status accounting
4. Configuration audits

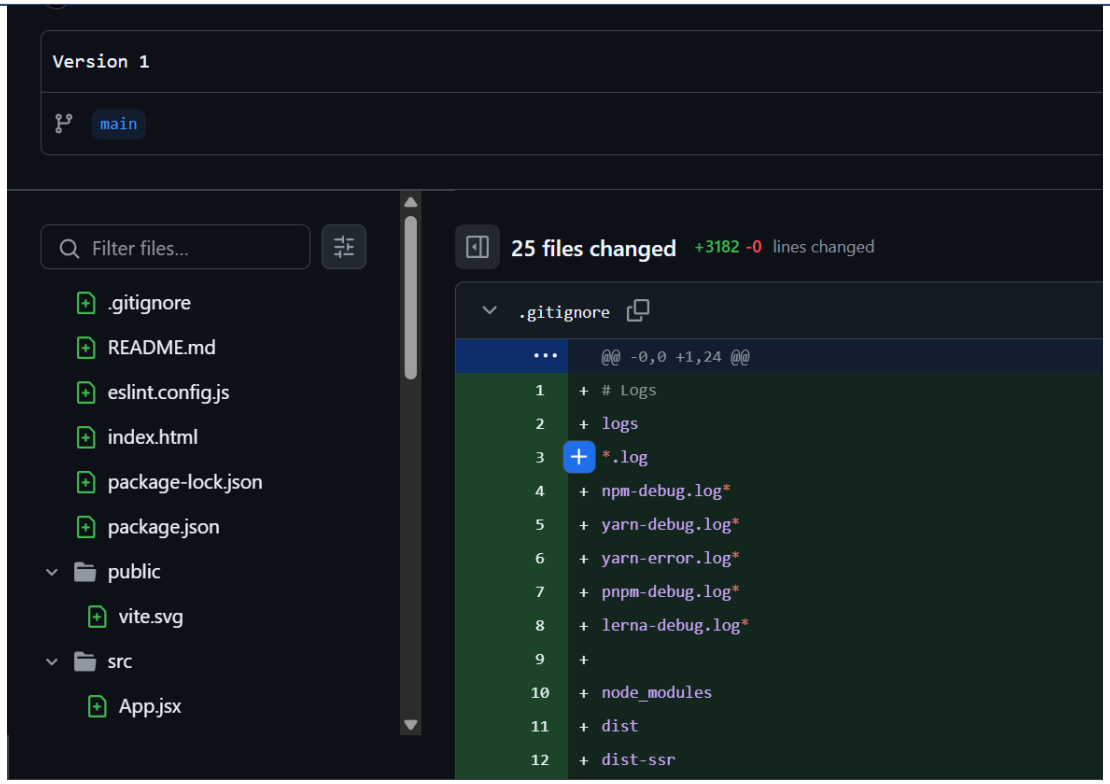
These terms and definitions change from standard to standard, but are essentially the same.

- Configuration identification is the process of identifying the attributes that define every aspect of a configuration item. A configuration item is a product (hardware and/or software) that has an end-user purpose. These attributes are recorded in configuration documentation and baselined. Baselining an attribute forces formal configuration change control processes to be effected in the event that these attributes are changed.
- Configuration change control is a set of processes and approval stages required to change a configuration item's attributes and to re-baseline them.
- Configuration status accounting is the ability to record and report on the configuration baselines associated with each configuration item at any moment of time.
- Configuration audits are broken into functional and physical configuration audits. They occur either at delivery or at the moment of effecting the change. A functional configuration audit ensures that functional and performance attributes of a configuration item are achieved, while a physical configuration audit ensures that a configuration item is installed in accordance with the requirements of its detailed design documentation.

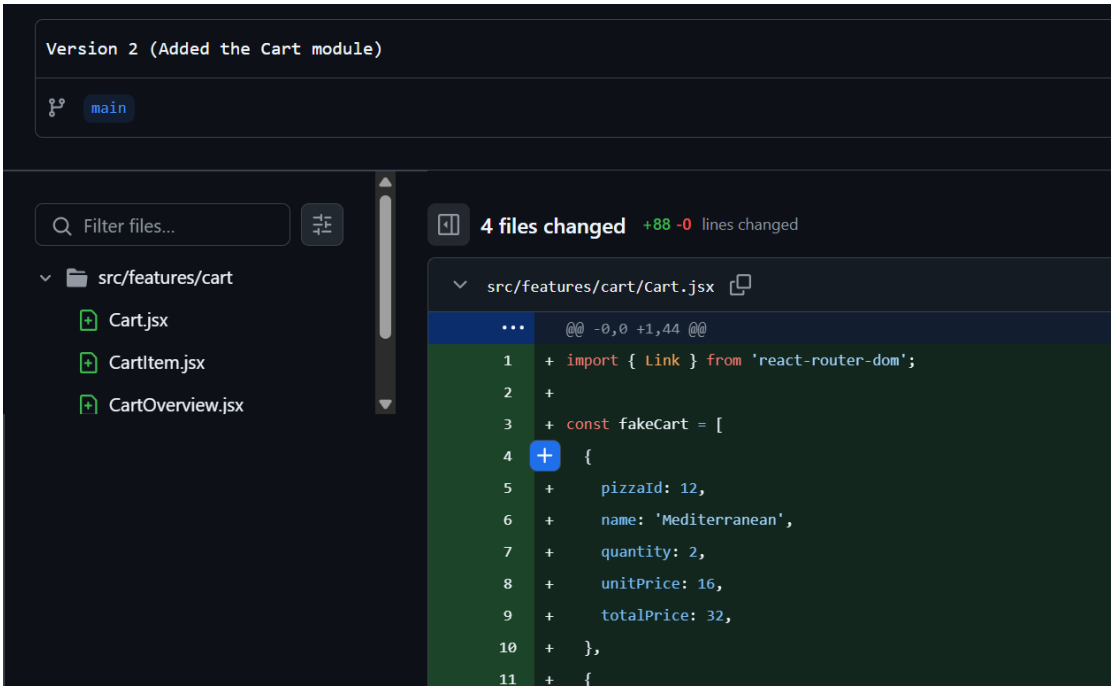
GitHub offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features. Unlike Git, which is strictly a command- line tool, GitHub provides a Web-based graphical interface and desktop as well as mobile integration. It also provides access control and several collaboration features such as bug tracking, feature requests, task management for every project.

Result and Discussion:

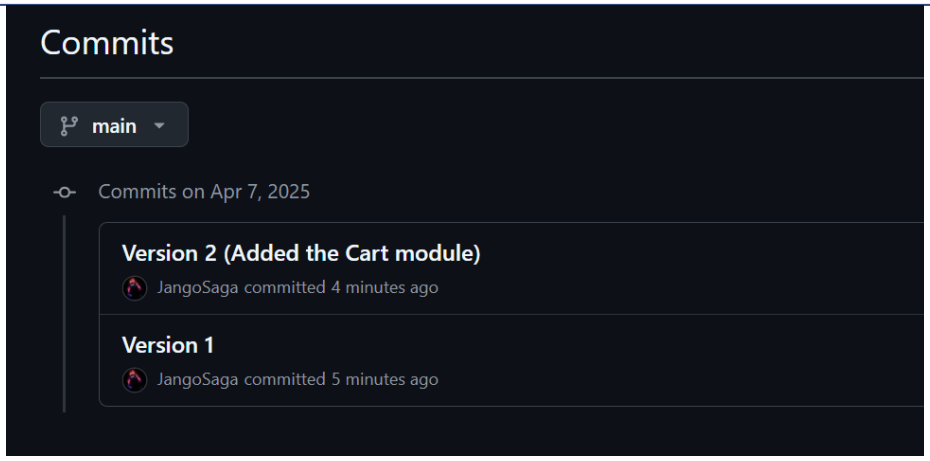
Version 1 of Restaurant Application:



Version 2 of Restaurant Application:



SCM using Git and GitHub: Version history and commits



Software Configuration Management (SCM) provides a structured process for handling, organizing, and controlling changes to software, requirements, code, etc throughout the software development lifecycle, ensuring consistency and traceability.

As an example, we tried to manage the version of our Restaurant App application using the Git and Github tool.GitHub offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features

The fist version of the Restaurant App is pushed to the github repository named “Version 1”. It was the initial application. Later the Restaurant App is developed further with following:

- **Added the Cart module with its features, etc.**

The updated version of the Restaurant App is committed and pushed to the repository again. Finally, the changes in the application are reflected in the github repository. By using the SCM tool like Github, we can review the commit history of the project, code or document and can even rollback to previous version thereby empowering the overall Software development and management process.

Course outcomes: Upon completion of the course students will be able to use any SCM Tool to make different versions.

Conclusion:

For Faculty Use

Correction Parameters	Formative Assessment [40%]	Timely completion of Practical [40%]	Attendance / Learning Attitude [20%]	
Marks Obtained				