**GIT Handson – 2**

* Explain git ignore

**Solu-**

The .gitignore file functions as a text document that instructs Git about which files and directories to exclude from project tracking. Git will avoid monitoring these unnecessary files so your repository focuses solely on essential code elements.

The standard location for this file exists at the base level of your repository directory. When Git scans the .gitignore file it uses this information to decide what files should remain untracked. The tool benefits from ignoring automatically created files including compiled binaries and dependency directories like node\_modules as well as configuration files containing sensitive API key information.

* Explain how to ignore unwanted files using git ignore

**Solu-**

To ignore unwanted files, you simply create or edit the .gitignore file in your project's root directory. The file uses a simple pattern-matching syntax.

* **Create the file**: If it doesn't already exist, create a file named .gitignore in the root of your Git repository.
* **Add patterns**: Add patterns to the file, one per line, to specify which files or directories to ignore.

Here are some common patterns you can use:

* **Ignoring a specific file**:

config.log

* **Ignoring all files with a certain extension**:

\*.log

* **Ignoring a specific directory**:

/temp

* **Ignoring a directory and all its contents**:

build/

* **Ignoring a file with a specific name in any directory**:

secrets.txt

After saving the .gitignore file, Git will automatically start ignoring the specified files. If you have already committed some of the files you now want to ignore, you'll need to remove them from Git's tracking first using git rm --cached <file> before adding the pattern to your .gitignore.