**2. GIT-HOL**

**GIT IGNORE COMMAND**

When working on a project, certain files like logs, temporary files, and build outputs don’t need to be tracked by Git. Using .gitignore, we can tell Git to ignore these unnecessary files and folders. This helps keep the repository clean and avoids pushing unwanted data to remote repositories.

**Objectives:**

* Understand the purpose of .gitignore
* Learn how to ignore specific files and folders using .gitignore
* Apply the .gitignore rule to ignore .log files and log/ folders
* Confirm that ignored files are not tracked using git status.

**Prerequisites**

• Git environment setup  
 • Integration of Notepad++ as the default Git editor  
 • A Git repository available in the local system  
 • A remote repository on GitHub or GitLab  
 • A free GitHub account (Do not use Cognizant credentials)

**Step 1:** Create a GitHub Account

1. Go to https://github.com.  
 2. Sign up for a free account using your personal email   
 3. Verify your email and log in.

**Step 2:** Set Up Local Repository

1. Open Git Bash or Command Prompt.  
 2. Navigate to your working directory:  
 cd path/to/your/project  
 3. Initialize Git if not already done:  
 git init

**Step 3:** Create .log File and log Folder

echo "This is a log file"> test.log  
 mkdir log  
 echo "Folder log created"> log/info.txt

**Step 4:** Configure .gitignore

1. Create a .gitignore file in the root of your repository:  
 notepad++ .gitignore  
 2. Add the following lines to ignore .log files and the log folder:  
 \*.log  
 log/

**Step 5:** Verify .gitignore

Run:  
 git status  
 You should not see test.log or the log folder in the tracked changes.  
 Example output:  
 On branch main  
  
 No commits yet  
 nothing to commit (create/copy files and use "git add" to track)

**Step 6:** Commit Other Files

1. Add all non-ignored files:  
 git add .  
 2. Commit changes:  
 git commit -m "Added .gitignore and project setup"

Verification

• .log files and the log folder remain untracked.  
 • Other project files are committed successfully.

