

SET-222

Software Operations & Maintenance

Experiment # 01

**Experiment Title**

**Introduction to Version Control with Git and GitHub**

**Assessment of CLO(s): 03**

**Performed on**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name:** |  | | |
| **Roll No.** |  | **Group** |  |
| **Semester** |  | **Session** |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S. No.** | **Perf. Level**  **Criteria** | **Excellent**  **(2.5)** | **Good**  **(2)** | **Satisfactory**  **(1.5)** | **Needs Improvement**  **(0 ~ 1)** | **Marks Obtained** |
| **1** | Project Execution & Implementation | Fully functional, optimized, and well-structured. | Minor errors, mostly functional. | Some errors, requires guidance. | Major errors, non-functional, or not Performed. |  |
| **2** | Results & Debugging  Or Troubleshooting | Accurate results with effective debugging  Or Troubleshooting. | Mostly correct, some debugging Or Troubleshooting needed. | Partial results, minimal debugging  Or Troubleshooting. | Incorrect results, no debugging Or Troubleshooting, or not attempted. |  |
| **3** | Problem-Solving & Adaptability  (VIVA) | Creative approach, efficiently solves challenges. | Adapts well, minor struggles. | Some adaptability, needs guidance. | Lacks innovation or no innovation, unable to solve problems. |  |
| **4** | Report Quality & Documentation | Clear, structured, with detailed visuals. | Mostly clear, minor gaps. | Some clarity issues, missing details. | Poorly structured, lacks clarity, or not submitted. |  |
| **Total Marks Obtained Out of 10** | | | | | |  |

**Experiment evaluated by**

|  |  |  |  |
| --- | --- | --- | --- |
| **Instructor’s Name** | **Ms. Shagufta Aftab** | | |
| **Date** |  | **Signature** |  |

## Copyright © Department of Engineering & Technology – UIT University Karachi

**Objective:**

* To understand the basics of version control.
* To learn how to install and set up Git.
* To practice using Git for version control and GitHub for remote repository management.

### **Theory:**

Version control systems (VCS) help track changes to files over time, facilitating collaboration among developers. Git is a distributed version control system that allows multiple developers to work on a project simultaneously. GitHub is a cloud-based platform that provides repository hosting and collaboration features for Git.

**Lab Exercise:**

**Step 1: Installing Git**

1. Download and install Git from <https://git-scm.com/downloads>.
2. Verify the installation by running the following command in the terminal:

git --version

**Step 2: Configuring Git**

1. Set up your username and email:
2. git config --global user.name "Your Name"

git config --global user.email "your-email@example.com"

1. Verify the configuration:

git config --list

**Step 3: Initializing a Git Repository**

1. Create a new directory and navigate into it:
2. mkdir my-git-project

cd my-git-project

1. Initialize an empty Git repository:

git init

1. Create a new file and add content:

echo "# My First Git Repository" > README.md

1. Add the file to the staging area:

git add README.md

1. Commit the file to the repository:

git commit -m "Initial commit"

**Step 4: Creating a GitHub Repository**

1. Log in to [GitHub](https://github.com/).
2. Click on "New Repository" and name it "my-git-project".
3. Copy the repository URL.

**Step 5: Connecting Local Repository to GitHub**

1. Add the remote repository:

git remote add origin <repository-url>

1. Push the local commits to GitHub:

git push -u origin main

**Step 6: Cloning a Repository**

1. To clone an existing GitHub repository, run:

git clone <repository-url>

**Step 7: Basic Git Operations**

* Check the status of your repository:

git status

* View commit history:

git log

* Create a new branch:

git branch feature-branch

* Switch to the new branch:

git checkout feature-branch

* Merge branches:

git checkout main

git merge feature-branch

* Pull the latest changes from GitHub:

git pull origin main

**Assessment & Exercises:**

1. Explain the difference between Git and GitHub.
2. List the commands used to check repository status and commit history.
3. What command is used to push local changes to a remote repository?
4. Write the command to create and switch to a new branch.
5. Clone a repository from GitHub and take a screenshot of the output.

**Conclusion:**

In this lab, students successfully installed and configured Git, initialized a repository, made commits, and pushed changes to GitHub. Understanding Git and GitHub helps in managing code efficiently and collaborating on software projects.