

# **EXPERIMENT: 9**

## **(2K17/SE/79 PARV GUPTA)**

**AIM:** Demonstrate how Git can be used to perform version control?

### **THEORY:**

Version control is important to keep track of changes and keep every team member working off the latest version. We should use version control software for all code, files, and assets that multiple team members will collaborate on.

It needs to do more than just manage and track files. It should help you develop and ship products faster. This is especially important for teams practicing DevOps.

That's because using the right one:

- Improves visibility.
- Helps teams collaborate around the world.
- Accelerates product delivery.

Git is a distributed version control software which you need to install on your local system in order to use it.

### **GIT FILE WORKFLOW:**

**git init:** Usage: git init [repository name]

This command is used to start a new repository.

**Workspace Copy:** Users' active directory simply creates new files in this space and this will be tracked by the Git.

**Stage Area:** It is a place where all the modified files marked to be committed are placed.

**Local Repository:** User's copy of the version database or file and access all the files through local repos and push the change made to remote

**Remote Repository** It is a server where all the collaborators upload changes made to files.

## **THE WORKFLOW:**

The GitHub workflow can be summarised by the commit-pull-push" mantra.

- **Commit** Once you've saved your files, you need to commit them - this means the changes you have made to files in your repo will be saved as a version of the repo, and your changes are now ready to go up on GitHub (the online copy of the repository).
- **Pull** Now, before you send your changes to Github, you need to pull, i.e. make sure you are completely up to date with the latest version of the online version of the files - other people could have been working on them even if you haven't. You should always pull before you start editing and before you push.
- **Push** Once you are up to date, you can push your changes - at this point in time your local copy and the online copy of the files will be the same.

The diagram illustrates the GitHub Workflow. It shows the sequence of actions: fork, clone, commit, push, pull request, and merge. The diagram includes icons for GitHub, a repository, a forked repository, a pull request, and a merged pull request. A green box labeled 'GitHub' and a purple box labeled 'Workflow' are also present.

```

graph TD
    GH[GitHub] -- fork --> FR[Forked Repository]
    FR -- clone --> RE[Repository]
    RE -- commit --> CR[Commit]
    CR -- push --> PR[Pull Request]
    PR -- pull request --> MR[Merge]
    MR -- merge --> GH
  
```

www.petanikode.com



In this Experiment we learned about how to work with git and understand its workflow.