Name: Param Shukla
Roll No: C107
Sap ID: 60004220046
Div: C-2

Experiment No 2

What is Version Control?

Version control software enables collaboration, coordination, and version management in DevOps, helping teams work asynchronously, track changes, and resolve merge conflicts across code and artifacts.

Why use Version Control?

Version control software enables collaboration, coordination, and version management in DevOps, helping teams work asynchronously, track changes, and resolve merge conflicts across code and artifacts.

What is a Version Control System?

Version Control Systems (VCS) track file changes, enabling collaboration and version history management. Centralized VCS (CVCS) stores all versions on a single server, allowing developers to check out, modify, and commit files. Distributed VCS (DVCS), like Git, provides each developer with a full project history, enabling offline work, branching, and merging. VCS ensures efficient, organized software development by safeguarding progress and facilitating teamwork.

Types of Version Control Systems

Distributed VCS (DVCS) stores files across multiple repositories, allowing remote collaboration and multi-device access

.

Centralized VCS (CVCS) uses a single central repository for all users, commonly found in team-based software development.

Lock-based VCS prevents conflicts by locking files, ensuring only one user edits at a time.

Optimistic VCS allows users to work independently and later merge changes through a server-managed process.

Main Version Control Systems

Git

Git is a powerful distributed version control system (DVCS) designed for speed, flexibility, and efficiency. It allows developers to track changes, collaborate seamlessly, and manage code versions efficiently.

Subversion (SVN)

Subversion (SVN) is a centralized VCS with a single codeline, making branching impossible but offering scalability and folder security.

Mercurial

Mercurial is a distributed VCS with simple branching, merging, and an intuitive command-line interface for scalable, collaborative development.

Benefits of Version Control

Quality

Encourages peer review, tracks changes, and ensures adherence to best practices, leading to better code and early bug detection.

Acceleration

Supports concurrent development with efficient branching and merging, speeding up iteration and issue resolution.

Visibility

Provides a central repository for tracking changes, improving project transparency, planning, and collaboration

What is GitHub?

GitHub is a cloud-based platform built on Git that enables developers to store, share, and collaborate on code through repositories. By keeping code in a repository, users can track changes, manage versions, and showcase their work while allowing others to review, suggest improvements, and contribute without disrupting the main project. GitHub's collaborative features ensure that multiple developers can work together seamlessly, integrating changes only when they are ready, making it an essential tool for open-source and team-based software development.

Git Starter Commands

1. git init

Initializes a new Git repository in the current directory

2. git remote add origin <link>

Connects your local repository to a remote GitHub repository

3. git remote -v

Verifies the remote repository connection

4. git branch

Shows all local branches in your repository

5. git add <file-name> or git add

Stages changes for commit (. stages all changes)

6. git commit -m "Your message"

Creates a new commit with the staged changes and a descriptive message

7. git push origin master

Uploads your commits to the remote repository on the master branch

Other Git Commands

1. git push <remote> <branch>

Uploads local branch commits to a remote repository.

2. git pull <remote> <branch>

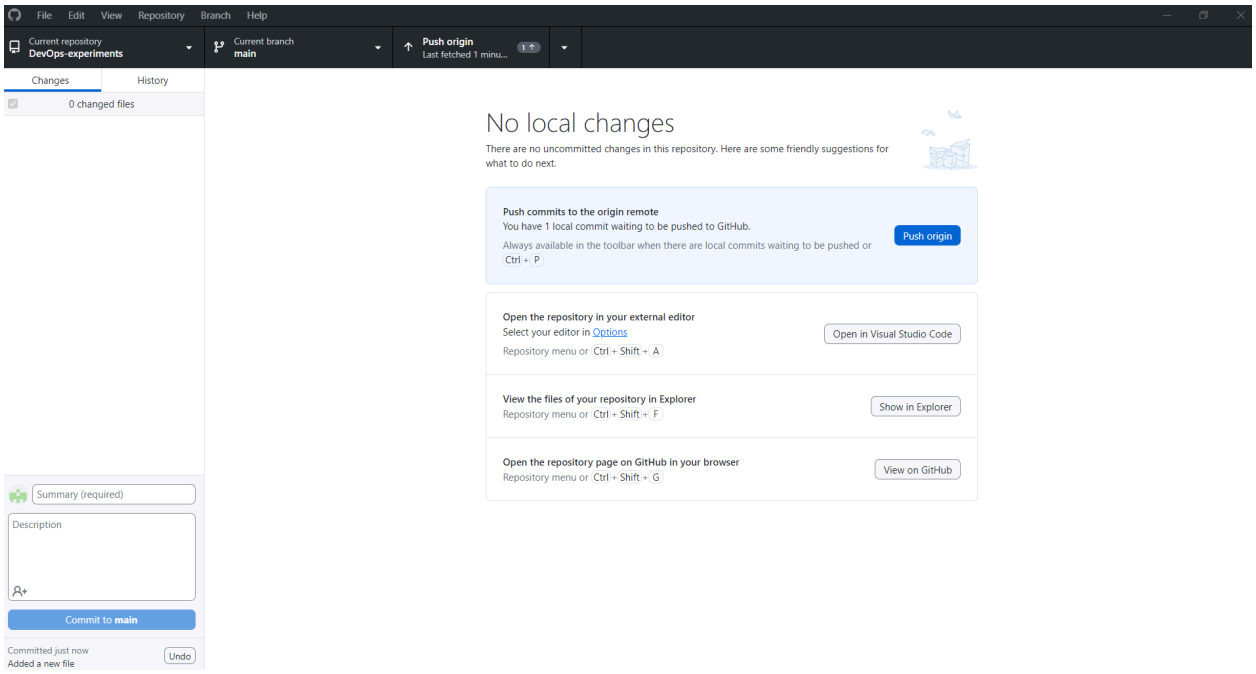
Fetches changes from a remote repository and merges them into your local branch.

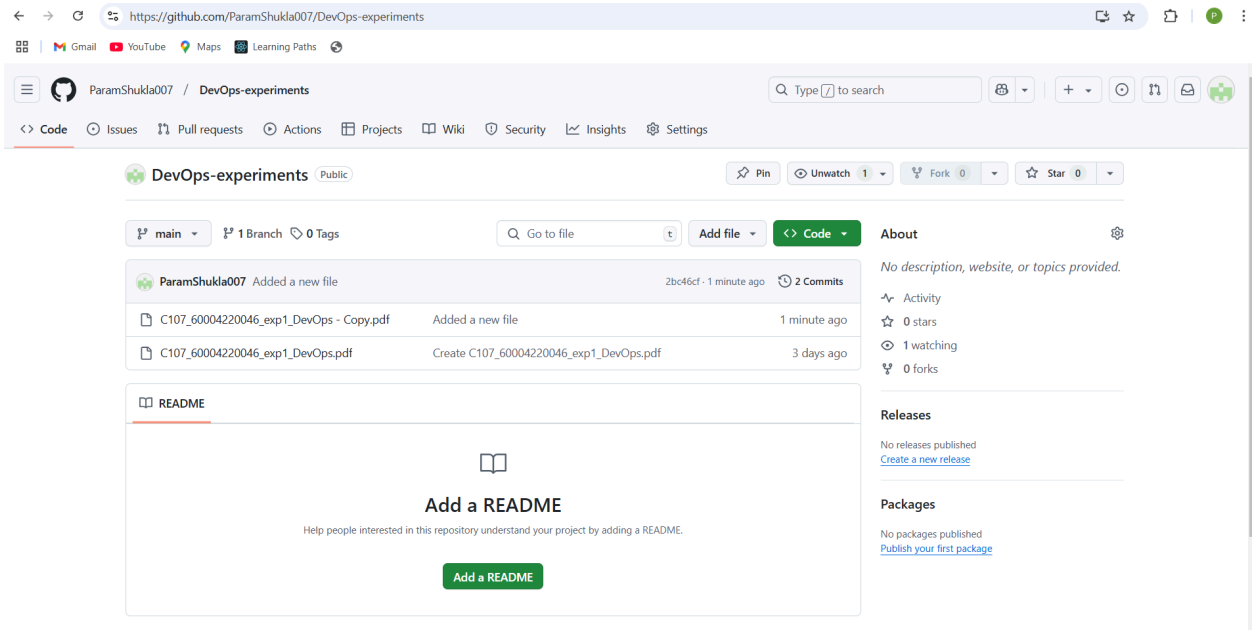
3. git revert <commit-hash>

It undoes changes from a specific commit while preserving history by creating a new commit that reverses the original changes.

Screenshots

Created a repository & Pushed the file into the repository through Git Desktop





Pulling the file from the GitHub cloud to the local folder