**FSD Laboratory 01**

**Name : Rajdeep Chauhan**

**Roll no. : PA-14**

**Batch : A1**

**PRN : 1032211100**

**Aim:** Version control with Git.

**Objectives:**

1. To introduce the concepts and software behind version control, using the example of Git.
2. To understand the use of 'version control' in the context of a coding project.
3. To learn Git version control with Clone, commit to, and push, pull from a git repository.

**Theory:**

1. **What is Git? What is Version Control?**

* Git is a distributed version control system (DVCS) that is commonly used for tracking changes in computer files and coordinating work among multiple people on a software development project. It was created by Linus Torvalds in 2005 and has since become one of the most widely used version control systems in the world.
* Version control is a system that helps manage changes to a set of files over time. It allows multiple people to collaborate on a project, track changes, and maintain a history of those changes. Version control systems are used in a wide range of fields, including software development, document management, and even in creative industries like graphic design.

**2. How to use Git for version controlling?**

Using Git for version control involves a series of steps and commands to track changes, collaborate with others, and manage your project's history. Here's a step-by-step guide on how to use Git:

1. **Install Git:** If you haven't already, download and install Git on your computer from the official Git website (<https://git-scm.com/downloads>). Once installed, open a terminal or command prompt to start using Git.

2**. Configure Git:** Before you start using Git, you should configure it with your name and email address.

3. **Create a Git Repository:** You can either initialize a new Git repository or clone an existing one.

4**. Add Files to the Staging Area**: Before you commit changes, you need to add the files you want to track to the staging area.

FAQ:

1. **What is branching in Git?**

Branching in Git is a fundamental concept that allows you to diverge from the main line of development (often referred to as the "master" or "main" branch) and work on new features, bug fixes, or experiments independently. It's one of the key features that makes Git a powerful and flexible version control system. Here's a closer look at what branching is in Git**:**

* **Main Branch:** The main branch typically represents the stable and production-ready version of your project. It's where the code is reliable and has passed all necessary tests.
* **Branches:** A branch is a separate line of development that stems from the main branch or another existing branch. Each branch has its own copy of the project files, and changes made in one branch do not affect other branches until you decide to merge them.
* **Creating a Branch:** You can create a new branch using the git branch command.

1. **How to create and merge branches in Git? Write the commands used.**

**The four commonly used commands to create a Git branch are:**

* git branch <branchname>
* git checkout -b <branchname>
* git branch <branchname> <tag>
* git branch <branchname> <commit id>

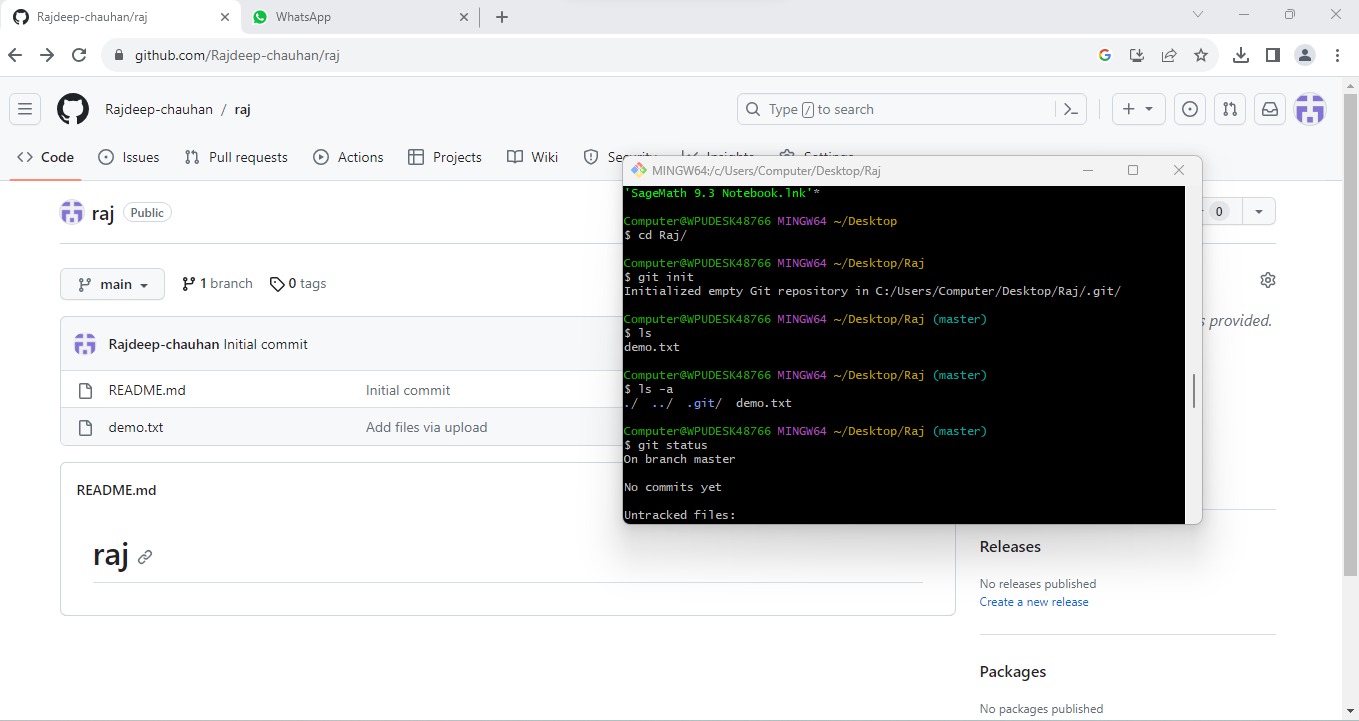
**Merge Two Git Repositories**

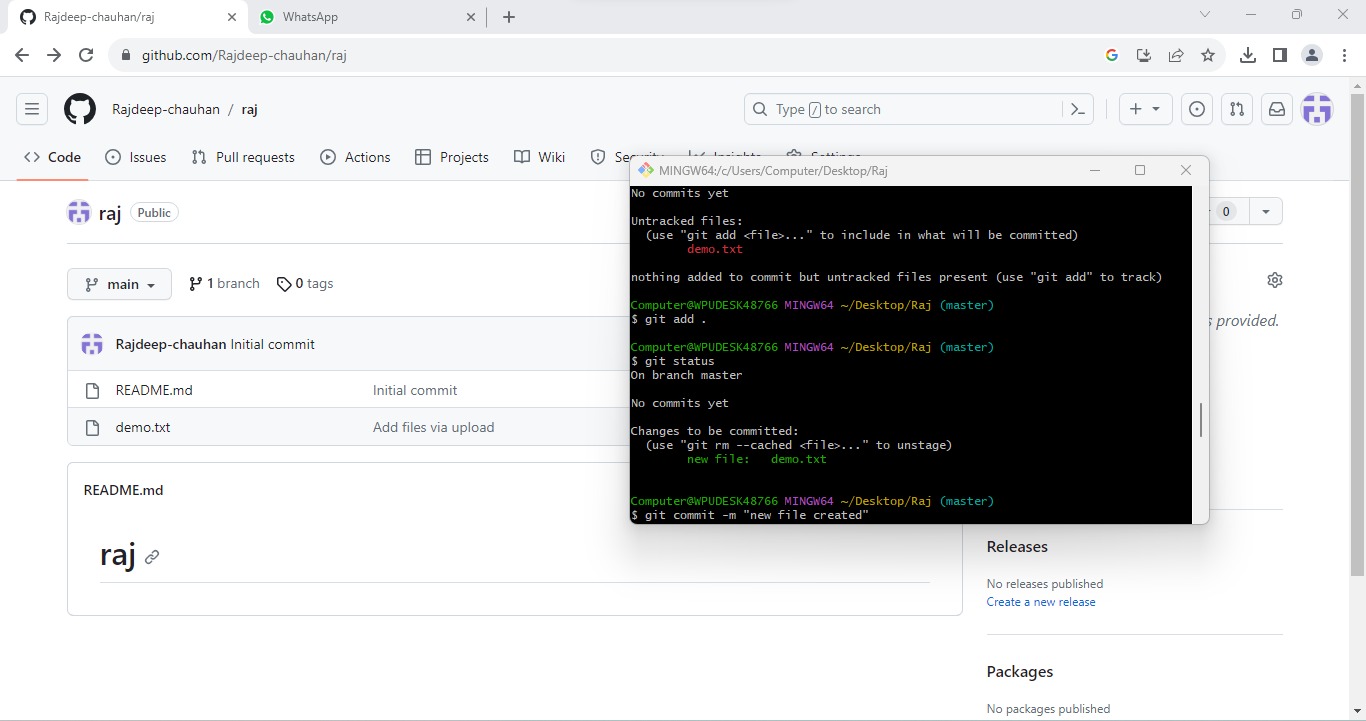
1. Cloning the First Repository. ...
2. Creating a New Branch for the Second Repository. ...
3. Adding the Second Repository as a Remote. ...
4. Fetching the Second Repository. ...
5. Merging the Second Repository into the New Branch.

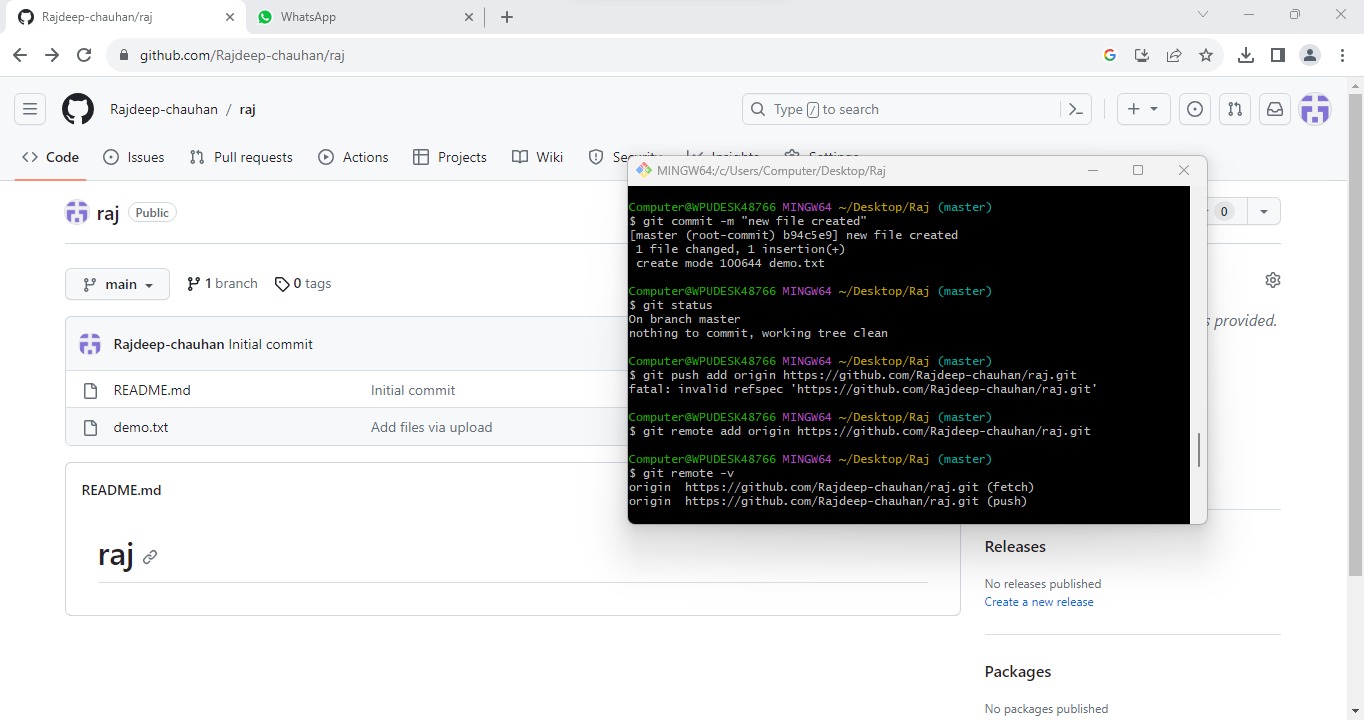
**Problem Statement:**

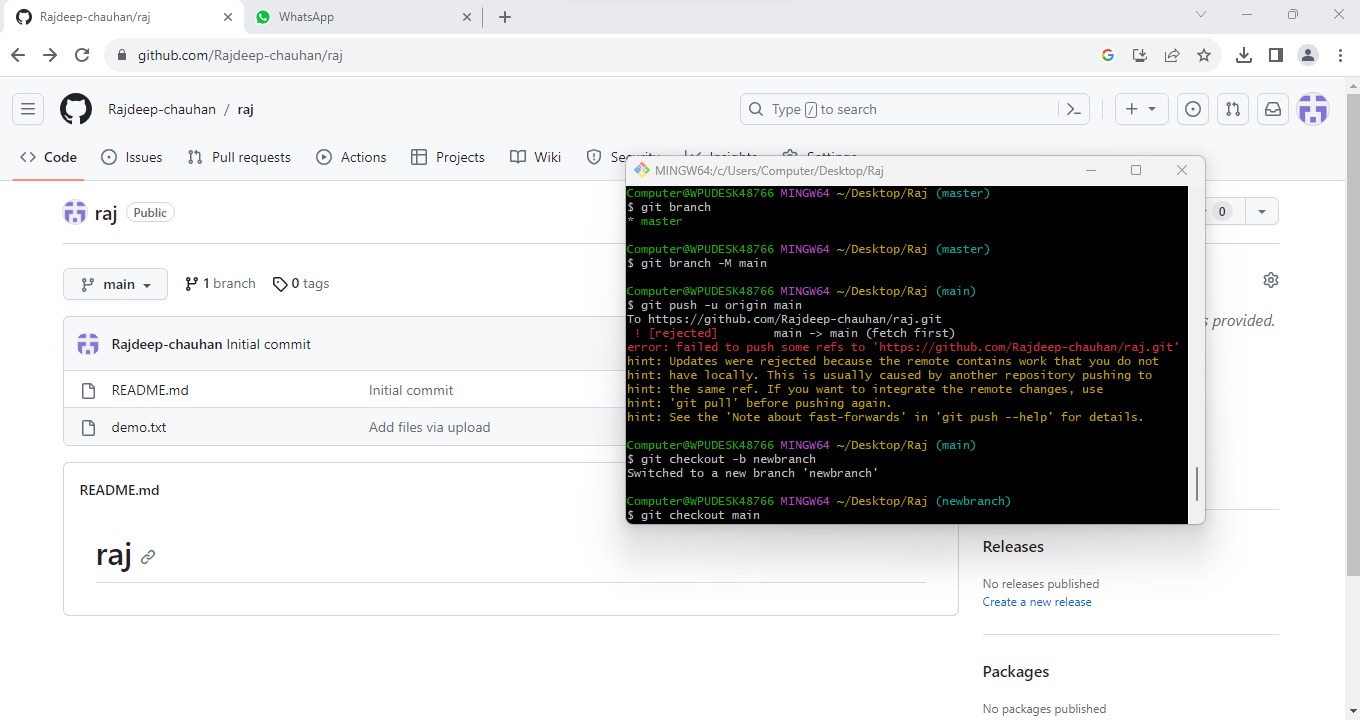
Created a public git repository for your team and submit the repo URL as a solution to this assignment, Learn Git concept of Local and Remote Repository, Push, Pull, Merge and Branch.

**Output: Screenshots of the output to be attached.**









URL : <https://github.com/Rajdeep-chauhan/raj>