**ASSIGNMENT-3**

**Version Control with Git: Branching and Merging**

**Creating a New Branch and Making Changes**

1. **Create a New Branch**:

To create a new branch named "feature-branch" in your Git repository, use the following commands in your terminal or command prompt:

bash

Copy code

git checkout -b feature-branch

This command creates a new branch named "feature-branch" and switches to it.

1. **Make Changes and Commit**:

Make changes to a file (e.g., edit README.md) in the "feature-branch" and commit the changes:

bash

Copy code

# Edit README.md file

# Save changes

git add README.md

git commit -m "Added feature X to README"

This command stages the changes in README.md and commits them to the "feature-branch".

**Merging the "feature-branch" into the Main Branch**

1. **Merge Process**:

To merge the changes from "feature-branch" into the main branch, follow these steps:

bash

Copy code

# Switch to the main branch

git checkout main

# Merge the feature-branch into main

git merge feature-branch

Git will attempt to automatically merge the changes. If there are no conflicts, the merge will be successful, and Git will create a merge commit.

1. **Handling Merge Conflicts**:

If there are conflicts between the changes made in "feature-branch" and the main branch:

* + Git will pause the merge process and indicate the conflicting files.
  + Open the conflicting files in a text editor and manually resolve the conflicts.
  + After resolving conflicts, stage the resolved files:

bash

Copy code

git add <conflicting-file>

* + Continue the merge process:

bash

Copy code

git commit -m "Resolved merge conflicts"

* + Git will create a merge commit with the resolved changes.

**Summary**

Branching and merging are essential concepts in Git for managing project development effectively. Branches allow you to work on features or fixes independently, while merging integrates changes back into the main codebase. Git automates much of the merging process, but conflicts can arise when changes conflict with each other. Resolving conflicts involves manually adjusting files and committing the resolved changes, ensuring a cohesive codebase with contributions from multiple collaborators.