ASSIGNMENT#1

Git Branch Handeling:

I’ve got to learn alot about git branches like how to create a branch, how to switch a branch,how to merge a branch and various other techniques of branching and I’ll let you how I got to learn the process of them;

### **Step 1: Understanding Branches**

* **Concept** : A branch in Git is essentially a lightweight movable pointer to one of the commits. The default branch name in Git is main (or master in older repositories).

### **Step 2: Creating a Branch**

1. **Command** : git branch <branch\_name>
   * Example: git branch feature-xyz
   * This creates a new branch called feature-xyz but does not switch to it.

### **Step 3: Listing Branches**

1. **Command** : git branch
   * This lists all the branches in the repository. The current branch is highlighted with an asterisk (\*).

### **Step 4: Switching Branches**

1. **Command** : git checkout <branch\_name>
   * Example: git checkout feature-xyz
   * This switches to the branch feature-xyz.
2. Alternatively, from Git version 2.23.0 onwards, you can use:
3. **Command** : git switch <branch\_name>
   * Example: git switch feature-xyz

### **Step 5: Creating and Switching to a Branch**

1. **Command** : git checkout -b <branch\_name>
   * Example: git checkout -b feature-xyz
   * This creates a new branch and switches to it.
2. Alternatively, from Git version 2.23.0 onwards, you can use:
3. **Command** : git switch -c <branch\_name>
   * Example: git switch -c feature-xyz

### **Step 6: Merging Branches**

1. **Switch to the target branch** (e.g., main):
   * git checkout main
2. **Merge the branch** (e.g., feature-xyz):
   * git merge feature-xyz
   * This merges feature-xyz into main.

### **Step 7: Resolving Conflicts**

* **Concept** : Sometimes, Git cannot automatically resolve conflicts between branches. When this happens, you’ll need to manually resolve the conflicts.
  + Open the conflicting files and decide how to merge the changes.
  + After resolving, mark the conflicts as resolved:
    - git add <resolved\_file>
  + Commit the changes:
    - git commit

### **Step 8: Deleting a Branch**

1. **Command** : git branch -d <branch\_name>
   * Example: git branch -d feature-xyz
   * This deletes the branch feature-xyz.
2. **Force delete a branch** (if the branch hasn’t been merged):
   * git branch -D <branch\_name>
   * Example: git branch -D feature-xyz

### **Step 9: Remote Branches**

1. **Push a branch to a remote repository**:
   * git push origin <branch\_name>
   * Example: git push origin feature-xyz
2. **Fetch branches from a remote repository**:
   * git fetch
3. **Check out a remote branch**:
   * git checkout -t origin/<branch\_name>
   * Example: git checkout -t origin/feature-xyz
4. **Delete a remote branch**:
   * git push origin --delete <branch\_name>
   * Example: git push origin --delete feature-xyz

### **Step 10: Branching Best Practices**

* **Keep branches short-lived**: Merge them back to the main branch frequently.
* **Use meaningful branch names**: Reflect the purpose of the branch, such as feature-xyz, bugfix-issue123, or hotfix-critical.

========================================================================