Assignment 2: Branch Creation and Switching

Create a new branch named 'feature' and switch to it. Make changes in the 'feature' branch and commit them.

A)

Certainly! Here's how you can create a new branch named 'feature', switch to it, make changes, and commit them:

1.Create a new branch named 'feature':

git branch feature

2.Switch to the 'feature' branch:

git checkout feature

3.Alternatively, you can use a single command to create the branch and switch to it:

git checkout -b feature

4.Make changes in the 'feature' branch:

Edit the files in your project directory as needed.

5.Add the changes to the staging area:

git add .

This command adds all changes in the current directory and its subdirectories to the staging area. If you only want to stage specific files, you can replace . with the filenames.

Commit the changes:

git commit -m "Made changes in the feature branch"

Replace "Made changes in the feature branch" with an appropriate commit message describing the changes you made.

Now, you've created a new branch named 'feature', switched to it, made changes, and committed them in the 'feature' branch

Script for implementation:

```
#!/bin/bash

# Create a new branch named 'feature' and switch to it

git checkout -b feature

# Make changes in the 'feature' branch

echo "Adding new feature" > feature.txt

# Add the changes to the staging area

git add feature.txt

# Commit the changes

git commit -m "Added new feature in feature branch"
```

Save this script with a .sh extension (e.g., create_feature_branch.sh) and make it executable with the command chmod +x create_feature_branch.sh. Then you can run the script by executing ./create_feature_branch.sh. It will create a new branch named 'feature', make changes in the 'feature' branch, and commit them automatically.