Version Control Day 2 Assignment

Mohamed Emary

March 4, 2025

1 Git Lab 2

1.1 Stashing Scenario

You made some changes to `script.js` but need to switch branches without committing your changes. Use Git's stashing feature to save your work temporarily.

1.1.1 Questions:

1. What command will you use to stash your changes?

```
git stash save "WIP: Changes to script.js"
```

> WIP (Work In Progress)

1.2 Creating and Working with Branches

> Repository Link For All Next Questions

You need to add a new feature to your project but don't want to disrupt the `main` branch.

1.2.1 Tasks:

1. Create a branch named `feature-login`
2. Merge the `feature-login` branch into `main`
3. Push the updated `main` branch to the remote repository

```
git checkout -b feature-login
git add .
git commit -m "Add login feature"
git checkout main
git merge feature-login
git push origin main
```

1.3 Branching Scenario

You need to add a new feature to your project but don't want to disrupt the main branch. Create a branch named `feature-login`.

1.3.1 Questions:

1. How will you create a new branch?
2. How will you switch to the newly created branch?

```
git checkout -b feature-login
```

1.4 Merging Scenario

After completing the `feature-login` branch, merge it back into the main branch.

1.4.1 Questions:

1. What command will you use to merge the `feature-login` branch into the main branch?
2. What will you do if there are merge conflicts?

```
git checkout main
git merge feature-login
```

If there are merge conflicts, we will resolve them manually by editing the conflicted files and then commit the changes, then:

```
git add .
git commit -m "Resolve merge conflicts"
```