Experiment 2: Devops Lab

To Perform various GIT operations on local and Remote repositories using GIT Cheat-Sheet

Given we have performed in previous lab, setting up a Git account along with Cloning other repo on our local desktop . After cloning we pushed the clone repo on our remote repository. The cloned repo is deployed on remote server(Netlify or vercel)

In this experiment our Goal is to practice local & remote Git operations including setup, version control, branching, merging, and remote sync.

.

Git Cheat-Sheet for Experiment 2

1.Initial Setup:

```
# Set your identity for commits
git config --global user.name "Your Name"
git config --global user.email you@example.com

# Optional: colorize output
git config --global color.ui auto
```

2.Creating Repositories:

```
git init
# ➤ Initialize current directory as a Git repo

git clone <url>
# ➤ Clone a remote repository locally
```

3.Tracking Changes:

```
git status
# ➤ View tracked/untracked changes

git add <file>
git add .
# ➤ Stage files for commit

git commit -m "Message"
# ➤ Commit staged changes locally

git commit -am "Msg"
# ➤ Add tracked files and commit in one go
```

4:Viewing History & Differences:

```
git log
# ➢ Show commit history

git log --oneline --graph
# ➢ Condensed visual history

git diff
# ➢ Show changes not yet staged
```

5. **Branching & Checkout:**

```
git branch
# ➢ List branches

git checkout -b new_branch
# ➢ Create and switch to new branch

git checkout existing_branch
# ➢ Switch branch

git branch -d branch
# ➢ Delete a merged branch
```

6. Merging & Conflict Management:

```
git merge feature_branch
# ➢ Merge into current branch

git log --graph
# ➢ Visual confirmation of merge

# If conflicts arise, edit files, then:
git add <conflicted-files>
git commit
```

7. Remote Repositories:

```
git remote add origin <url>
# ➢ Link to remote repository

git fetch origin
# ➢ Download latest branches
```

```
git pull origin main
# ➢ Fetch and merge remote changes

git push origin main
# ➢ Upload local commits to remote
```

8. Undoing Changes:

```
git reset HEAD <file>
# ➢ Unstage file without deleting changes

git checkout -- <file>
# ➢ Discard unsaved local changes

git revert <commit>
# ➢ Reverse a commit by creating a new one
```

9.Advanced Tools:

```
git stash
# ➢ Temporarily stash changes

git stash pop
# ➢ Restore stashed changes

git reflog
# ➢ View history of HEAD positions

git clean -n
git clean -f
# ➢ Preview and delete untracked files
```

For our experiment 2 try to execute atleast 7 git commands from the above git cheatsheets.
In my absence please utilize the time to complete the experiment 2 use the repo you deployed in experiment 1. Everyone should perform their experiment on their own repo.

I want writeup for experiment 1 and 2 next week or you'll lose marks.
And again never settle for mediocrity. See you next week - Rohaan Khan