TP: Version Control System

Say OL

September 27, 2024

Version Control with Git: Problem Set

This problem set is designed to help you practice using Git as a version control system. Follow the instructions for each problem and document your progress.

Problem 1: Setting Up Git

a. Install Git on your local machine.

b. Configure your user name and email in Git. Verify the configuration by running:

```
git config --global user.name "Your Name"
git config --global user.email "youremail@example.com"
git config --list
```

Problem 2: Creating a Repository

a. Create a new directory for a project called `version_control_practice`.

b. Initialize a Git repository in this directory by running:

```
git init
```

c. Create a new file named `README.md` and add a brief description of the project.

Problem 3: Basic Commands

a. Add the `README.md` file to the staging area by running:

```
git add README.md
```

b. Commit the changes with a message explaining what you did:

```
git commit -m "Add README.md with project description"
```

Problem 4: Making Changes

a. Edit `README.md` to include a list of your favorite programming languages.

b. Stage and commit this change.

Problem 5: Branching

a. Create a new branch called `feature-1`:

```
git branch feature-1
```

b. Switch to this branch:

```
git checkout feature-1
```

Problem 6: Further Changes

a. Add a new file named `languages.txt` that lists your favorite programming languages.

b. Stage and commit this change in the `feature-1` branch.

Problem 7: Merging

a. Switch back to the `main` branch:

```
git checkout main
```

b. Merge the `feature-1` branch into `main`:

```
git merge feature-1
```

c. Resolve any merge conflicts if they arise.

Problem 8: Remote Repositories

a. Create a new repository on GitHub named `version_control_practice`.

b. Link your local repository to the GitHub repository:

```
git remote add origin https://github.com/username/version_control_practice.git
```

c. Push your `main` branch to GitHub:

```
git push origin main
```

Problem 9: Collaborating

a. Ask a peer to fork your GitHub repository.

b. Have them create a new branch, make changes, and submit a pull request.

c. Review the pull request and merge it into your main branch.

Problem 10: Tags and Releases

a. Create a tag for your current version:

```
git tag -a v1.0 -m "Version 1.0"
```

b. Push the tags to the remote repository:

```
git push origin --tags
```

Optional Challenges

a. **Challenge 1:** Implement a feature where you track changes in a specific file (e.g., `languages.txt`). Show how to view the change history.

b. **Challenge 2:** Use Git commands to roll back to a previous commit and explain the steps taken.

Submission

Submit your local repository as a zip file, along with screenshots of relevant commands and outputs from your terminal, as well as links to your GitHub repositories.

GitHub CLI: Problem Set

This problem set is designed to help you practice using the GitHub CLI ('gh'). Follow the instructions for each problem and document your progress.

Problem 1: Setting Up GitHub CLI

a. Install the GitHub CLI on your local machine. Follow the instructions on the official website: https://cli.github.com/.

b. Authenticate your GitHub account with the CLI:

```
gh auth login
```

Problem 2: Creating a New Repository

a. Use the GitHub CLI to create a new repository named `gh_cli_practice`.

```
gh repo create gh_cli_practice
```

b. Clone the new repository to your local machine.

Problem 3: Managing Issues

a. Create a new issue in your repository:

```
gh issue create --title "Fix bug in README" --body "The README file has outdated informatic
```

b. List all issues in your repository:

```
gh issue list
```

Problem 4: Creating and Managing Pull Requests

a. Create a new branch in your local repository for a feature:

```
git checkout -b feature-1
```

b. Make some changes and commit them.

c. Push the branch to the remote repository:

```
git push origin feature-1
```

d. Use the GitHub CLI to create a pull request for this branch:

```
gh pr create --base main --head feature-1 --title "Add feature 1" --body "This PR adds feat
```

Problem 5: Reviewing Pull Requests

a. List open pull requests in your repository:

```
gh pr list
```

b. Review one of the pull requests using the CLI:

```
gh pr review [PR_NUMBER] --approve
```

c. Merge the pull request:

```
gh pr merge [PR_NUMBER]
```

Problem 6: Managing Releases

a. Create a new release using the GitHub CLI:

```
gh release create v1.0 --title "Version 1.0" --notes "Initial release"
```

b. List all releases in your repository:

```
gh release list
```

Problem 7: Adding Collaborators

a. Invite a collaborator to your repository:

```
gh repo add-collaborator username
```

Problem 8: Working with Gists

a. Create a new gist using the GitHub CLI:

```
gh gist create my_script.py --public --description "A simple Python script."
```

b. List your gists:

```
gh gist list
```

Optional Challenges

a. **Challenge 1:** Use the GitHub CLI to search for repositories that match a specific topic.

b. **Challenge 2:** Write a script that automates the creation of issues in a repository using the ‘gh‘ CLI.

Submission

Submit your local repository as a zip file, along with screenshots of relevant commands and outputs from your terminal, as well as links to your GitHub repositories.