Ghulam Ishaq Khan Institute (GIKI)

DevOps Assignment # 1

Subject: DevOps	**Course Code:** CS - 328 – Spring - 2025
Class: BS CY **Batch:** Fall – 2022	**Submission Deadline:** 11 Feb 2025 Tuesday (09:00 AM)
Course Instructor: Muhammad Ahmad Nawaz	**Total Marks:** 10
Course TA:	

Note (Read notes & instructions first)

- First, read the instructions and statements of each exercise/question carefully then write the solution.
- *For Assignment:*
  - The name of your pdf file should contain your assignment number and your roll number as shown in following example, For Example if your roll number is 2022532 and you have done assignment number 1 then the name of file should be as ---> 2022532_1.pdf
  - Then upload that pdf file at Microsoft teams. Remember the sequence of pages should be right.
  - **Drop off** the physical copy at my office **11 Feb 2025 Tuesday (09:00 AM**), before Tuesday Class.

CHEATING/COPY CASE or LATE SUBMISSION will be graded as STRAIGHT ZERO MARKS.

Practical Assignment: Mastering Git and GitHub for Collaboration

This assignment is designed to help you understand the core concepts of Git and GitHub, including collaboration, branching, forking, cloning, reverting, and resetting. By the end of this assignment, you should be comfortable using Git and GitHub for version control and teamwork.

Assignment Overview

You will work on a simulated team project where you will:

1. Create a repository.
2. Collaborate with others using branches.
3. Fork a repository.
4. Clone a repository.
5. Use revert and reset to manage changes.

Step 1: Set Up Your Repository

1. **Create a Repository on GitHub:**
   - Go to GitHub and create a new repository named team-project.
   - Add a README.md file and a .gitignore file for Python (or any language of your choice).
2. **Clone the Repository Locally:**
   - Open your terminal or Git Bash.
   - Run the command:

     bash

     ```
     git clone https://github.com/your-username/team-project.git
     ```
   - Navigate into the cloned repository:

     bash

     ```
     cd team-project
     ```
3. **Make Your First Commit:**
   - Create a new file called main.py (or any other file relevant to your project).
   - Add some code or text to the file.
   - Stage and commit your changes:

     bash

```
git add main.py
git commit -m "Added main.py with initial code"
```

- Push your changes to GitHub:

```bash
git push origin main
```

Step 2: Collaborate Using Branches

1. **Create a New Branch:**
   - Create a branch for a new feature:

   ```bash
   git checkout -b feature/new-feature
   ```

   - Make changes to the main.py file or add new files.
2. **Commit and Push the Branch:**
   - Stage and commit your changes:

   ```bash
   git add .
   git commit -m "Added new feature"
   ```

   - Push the branch to GitHub:

   ```bash
   git push origin feature/new-feature
   ```

3. **Create a Pull Request (PR):**
   - Go to your GitHub repository.
   - Click on the "Pull Requests" tab and create a new PR for the feature/new-feature branch.
   - Add a description of the changes and request a review from a teammate (if working in a team).
4. **Merge the PR:**
   - Once the PR is approved, merge it into the main branch.

Step 3: Fork a Repository

1. **Fork a Repository:**
   - Go to a public repository on GitHub (e.g., a classmate's repository or an open-source project).
   - Click the "Fork" button to create a copy of the repository under your GitHub account.
2. **Clone the Forked Repository:**
   - Clone the forked repository to your local machine:

   ```bash
   git clone https://github.com/your-username/forked-repo.git
   ```

3. **Make Changes and Push:**
   - Make changes to the code or documentation.
   - Commit and push your changes to your forked repository.
4. **Create a PR to the Original Repository:**
   - Go to your forked repository on GitHub.
   - Click "Contribute" and then "Open Pull Request" to submit your changes to the original repository.

Step 4: Revert and Reset

1. **Revert a Commit:**
   - Make a commit with some changes.
   - To revert the commit:

   ```bash
   ```

```
    git revert <commit-hash>
    ```
    - This creates a new commit that undoes the changes.
2. **Reset a Commit:**
    - Make a commit with some changes.
    - To reset to a previous commit (use with caution):

    bash

    ```
    git reset --hard <commit-hash>
    ```
    - This will discard all changes after the specified commit.

Step 5: Collaboration Workflow

1. **Simulate Team Collaboration:**
    - Pair up with a classmate or teammate.
    - Clone each other's repositories.
    - Create branches, make changes, and submit PRs to each other's repositories.
    - Review and merge the PRs.
2. **Resolve Merge Conflicts:**
    - Intentionally create a merge conflict by editing the same file in two different branches.
    - Resolve the conflict by editing the file, staging it, and committing the changes.

Deliverables

1. A GitHub repository with:
    - Multiple branches.
    - At least one merged PR.
    - A forked repository with a PR to the original repository.
2. A document or README file explaining:
    - The steps you followed.
    - Screenshots of your GitHub activity (e.g., PRs, branches, commits).
    - Challenges you faced and how you resolved them.

Grading Criteria

- **Functionality (40%):** All Git and GitHub features are implemented correctly.
- **Collaboration (30%):** Evidence of teamwork (e.g., PRs, reviews, resolved conflicts).
- **Documentation (20%):** Clear and detailed explanation of the process.
- **Code Quality (10%):** Clean and well-organized code.

This assignment will give you hands-on experience with Git and GitHub, preparing you for real-world collaboration and version control. Good luck!