**Practical Git Workflow**

**Objective:** Gain hands-on experience with Git version control by performing realistic tasks that include creating, merging, and resolving conflicts in branches, as well as handling pull requests and code reviews. This assignment will enhance your ability to manage complex workflows in software development projects.

**Instructions:**

- Complete the tasks outlined below, documenting each step and decision in a report.
- Ensure all your work is pushed to a public GitHub repository named `git-assignment`.
- Your final submission should include a link to the repository and a brief report.

**Setup:**

- Create a new GitHub repository called `git-assignment`.
- Clone the repository to your local machine.
- Create an initial `README.md` file describing the repository's purpose and commit it.

**Tasks:**

1. **Branch Creation and Content Addition**
    - **Branch 1: `feature/about`**
        - Create a branch from `master` called `feature/about`.
        - In this branch, add a file named `about.txt`.
        - Inside `about.txt`, write the following paragraph:
          "Git is a distributed version control system. It allows multiple developers to work on a project without requiring constant connection to a central repository."
        - Commit the changes.
    - **Branch 2: `feature/contact`**
        - Create another branch from `master` called `feature/contact`.
        - In this branch, add a file named `contact.txt`.
        - Inside `contact.txt`, write the following content:
          "For more information, visit our GitHub Page or contact our support team."
    - **Branch 3: master**
        - Keep as is for now (only contains README.md) and publish it.
2. **README Updates**
    - **In `feature/about`:**

    - Modify `README.md` to include this line at the end:
      "This project is an exercise in mastering Git functionalities."
    - Commit the change.
- **In `feature/contact`:**
    - Modify `README.md` similarly, but add this slightly different line:
      "This project serves as a practical exercise to understand Git operations."
    - Commit the change.

3. **Merging and Conflict Resolution**
    - Merge `feature/about` into `main` and push the changes.
    - Attempt to merge `feature/contact` into `main`. This will result in a merge conflict in `README.md`.
    - Manually resolve the conflict by integrating both lines in a coherent paragraph.
    - Complete the merge and push the final version.

4. **Pull Requests and Code Reviews**
    - **Creating Pull Requests:**
        - Create a pull request for `feature/about` to merge into `master`. Summarize the changes in the pull request description.
        - Add a classmate as a reviewer. Ensure they review the pull request, provide comments, and approve it before merging.
    - **Reviewing and Merging:**
        - After approval, merge the pull request into the `master` branch.

**Report Requirements:**

- Provide a link to the GitHub repository.
- Discuss the importance of pull requests in team-based projects and how they help maintain code quality.

**Submission:**

- Submit your report in PDF format through discord by the **Feb 06**.

---