Assignment 3: Collaborative Development Using Git - Fork an existing repository (this can be a simulated action if no actual repository is available), clone it

locally, and demonstrate managing updates from the original repository. Create a pull request to the original repository with changes made in the forked repository,

and outline a simple code review process.

SOLUTION:

1. **Fork an existing repository:**

   - If you're using a real repository, you can fork it on the platform where it's hosted (like GitHub, GitLab, Bitbucket, etc.).

   - If you don't have a real repository to work with, you can simulate forking by creating a copy of a local repository.

2. **Clone the forked repository locally:**

   ```bash

   # Clone the forked repository to your local machine

   git clone <URL_of_your_forked_repository>

   ```

3. **Manage updates from the original repository:**

   - Add the original repository as a remote to your local clone:

     ```bash

     git remote add upstream <URL_of_original_repository>

     ```

   - Fetch the latest changes from the original repository:

     ```bash

     git fetch upstream

     ```

- Merge the changes from the original repository into your local branch (e.g., main or master):

  ```bash
  git checkout main  # switch to your main branch
  git merge upstream/main  # merge changes from the original repository into your main branch
  ```

4. **Create a pull request to the original repository with changes made in the forked repository:**

   - Push your changes to your forked repository:

     ```bash
     git push origin main  # push changes to your forked repository
     ```

   - Go to the original repository's platform (e.g., GitHub) and create a pull request from your forked repository to the original repository.

   - Provide a meaningful description of the changes in the pull request and submit it.

5. **Outline a simple code review process:**

   - Review the changes made in the pull request:

     - Check for code quality, readability, and adherence to coding standards.

     - Test the changes locally if possible.

     - Leave comments or suggestions for improvement.

   - Discuss any concerns or questions with the author of the pull request.

   - Once the code review is complete and any necessary changes have been made, approve the pull request.

   - Optionally, if your project requires it, you can set up automated tests to run on pull requests to ensure code quality.

This process allows for collaborative development, where contributors can fork a repository, make changes, and submit pull requests to contribute back to

the original project. Code reviews help ensure that changes are of high quality and align with the

project's standards.