**Raising a Pull Request (PR) with Reviewers Flow**

A **Pull Request (PR)** is a mechanism for developers to propose changes to a codebase. It enables collaboration, code review, and quality assurance before merging the changes into the main branch.

**Steps to Raise a Pull Request (PR)**

**1. Clone the Repository**

* Clone the repository to your local system:
* git clone https://github.com/user/repository.git
* Navigate to the repository directory:
* cd repository

**2. Create a New Branch**

* Always create a new branch for your changes:
* git checkout -b feature/branch-name

**Example:**

git checkout -b feature/add-login-page

**3. Make Your Changes**

* Modify files, add features, or fix bugs as required.
* Track the changes:
* git add .
* Commit the changes with a meaningful message:
* git commit -m "Added login page with authentication"

**4. Push Your Changes**

* Push the branch to the remote repository:
* git push origin feature/branch-name

**Example:**

git push origin feature/add-login-page

**5. Open a Pull Request (PR)**

1. Go to the repository on GitHub/GitLab.
2. Click on **Pull Requests** (or Merge Requests for GitLab).
3. Select **New Pull Request**.
4. Choose:
   * **Base branch**: The branch you want to merge into (e.g., main or develop).
   * **Compare branch**: The branch with your changes (e.g., feature/add-login-page).

**Review Process**

**6. Assign Reviewers**

* Assign specific team members to review your PR.
* Add reviewers in the "Reviewers" section of the PR interface.

**7. Reviewers' Tasks**

* **Review Code**:
  + Check for bugs, performance issues, or code that doesn’t adhere to standards.
  + Ensure the changes align with the project's goals.
* **Provide Feedback**:
  + Comment on specific lines of code.
  + Suggest improvements or highlight areas that need clarification.

**8. Address Feedback**

* If reviewers request changes:
  1. Make the necessary updates in your branch.
  2. Push the updated code:
  3. git add .
  4. git commit -m "Addressed feedback on login logic"
  5. git push origin feature/add-login-page

**9. Approval and Merging**

* Once reviewers approve the PR:
  + The author or repository maintainer merges it into the base branch.
  + Optionally, delete the feature branch.

**Flow Example: Adding a Login Page**

**1. Developer Actions**

1. Create a branch: feature/add-login-page
2. Add login functionality:
   * Modify login.html and auth.js.
3. Commit changes:
4. git commit -m "Added login page with user authentication"
5. Push changes:
6. git push origin feature/add-login-page

**2. Raise PR**

* Title: Feature: Add Login Page
* Description:
* - Implemented a login page.
* - Added user authentication logic in auth.js.
* - Updated styling for consistency.

**3. Reviewer Actions**

* **Reviewer A**:
  + Comments: “Please add error handling for incorrect credentials.”
* **Reviewer B**:
  + Suggests: “Consider using bcrypt for hashing passwords.”

**4. Developer Updates**

1. Add error handling in auth.js.
2. Update PR with:
3. git add auth.js
4. git commit -m "Added error handling and bcrypt for password hashing"
5. git push origin feature/add-login-page

**5. Merge**

* Reviewers approve the changes.
* PR is merged into the main branch.

**Benefits of PR Workflow**

1. **Code Quality**: Ensures peer review for bugs and standards.
2. **Collaboration**: Encourages team feedback and knowledge sharing.
3. **Traceability**: Tracks all changes and discussions for future reference.