**PMS Repository – CI/CD & PR Workflow Setup**

**1. Repository Setup**

* Repository: **PMS-ai-org/PMS**
* Structure:
* PMS/
* ├── PMS\_UI → Angular application
* ├── PMS.WebAPI → .NET Core Web API application
* └── README.md
* Default branch: **main**

**2. Branch Protection**

* Enabled **branch protection rules** for main:
  + Require pull request before merging.
  + Require at least 1 approval review.
  + Require all status checks (pipelines) to pass before merging.
  + Disallow direct pushes to main.

**3. GitHub Actions – CI/CD Pipelines**

We created **separate GitHub Action workflows** for each project.

**3.1 Angular Pipeline**

* File: .github/workflows/angular-ci.yml
* Trigger: Runs on every PR to main and on push to main.
* Steps:
  1. Checkout repository.
  2. Set up Node.js.
  3. Install dependencies from PMS\_UI/package.json.
  4. Run ng lint for linting.
  5. Run ng build --configuration production to build the Angular app.
  6. Run unit tests (if configured).

**3.2 .NET Core Web API Pipeline**

* File: .github/workflows/dotnet-ci.yml
* Trigger: Runs on every PR to main and on push to main.
* Steps:
  1. Checkout repository.
  2. Set up .NET Core SDK.
  3. Restore NuGet packages from PMS/PMS.WebAPI/PMS.WebAPI.csproj.
  4. Build the project.
  5. Run unit tests (if configured).

**4. Pull Request Workflow**

Since we only have main branch, we create feature branches for all changes.

**4.1 Steps to Raise a Pull Request**

1. **Create a branch** from main:
2. git checkout main
3. git pull origin main
4. git checkout -b feature/your-feature-name
5. Make changes and commit:
6. git add .
7. git commit -m "Add: your change description"
8. Push branch:
9. git push origin feature/your-feature-name
10. On GitHub, click **Compare & Pull Request**.
11. Fill in:
    * PR title (clear and descriptive).
    * PR description (what was changed, why, and any testing notes).
12. Assign reviewers.
13. Wait for:
    * **Pipeline checks** (Angular & .NET CI jobs) to pass.
    * **Reviewer approval**.
14. Once approved & checks pass → Merge to main.

**5. Merge Policy**

* Only PR merges allowed (no direct pushes to main).
* All pipelines must pass before merging.
* At least 1 reviewer approval required.

**6. Benefits**

* Prevents broken builds from entering main.
* Keeps a clear audit trail of changes.
* Automated build verification for both Angular and .NET projects.
* Enforces code review culture.