**Branching Strategy**

**Task 1: Create a GitHub repository and demonstrate branching, pull requests, and code reviews.**

This document describes the Git branching strategy to be followed in the project.

**1. Main Branch (main or master)**

* The primary branch always contains stable, production-ready code.
* Only merged changes that have been reviewed and tested should be present.
* Protected from direct commits.

**2. Development Branch (develop/dev)**

* Serves as an integration branch for features and fixes.
* All new features and bug fixes are merged here before they reach main.
* May not always be stable but should be functional.

**3. Feature Branches (feature/<feature-name>)**

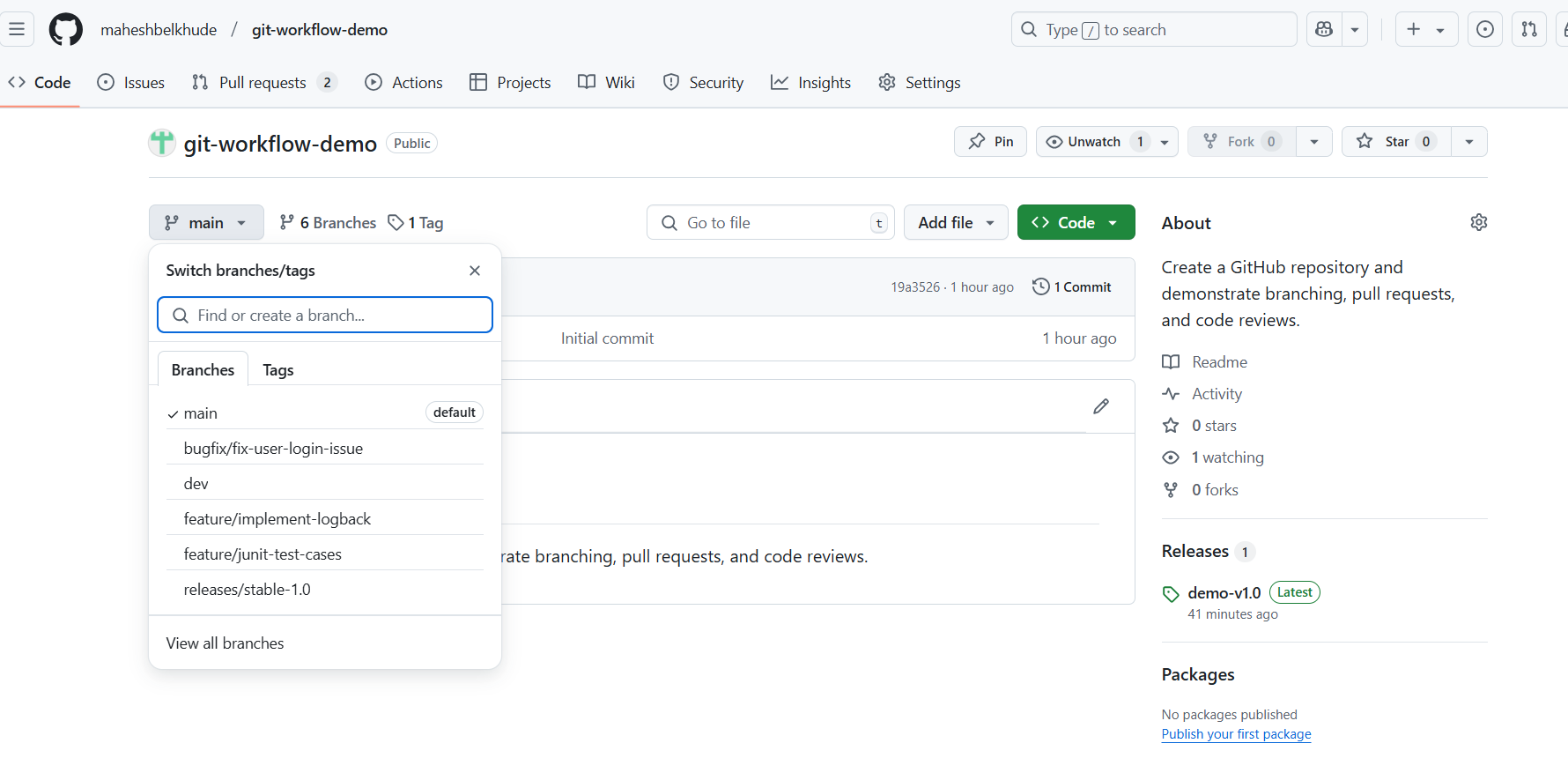
* Created for developing new features.
* Branched from develop/dev.
* Merged back into develop/dev after completion and review.
* Naming convention: **feature/feature-name**.

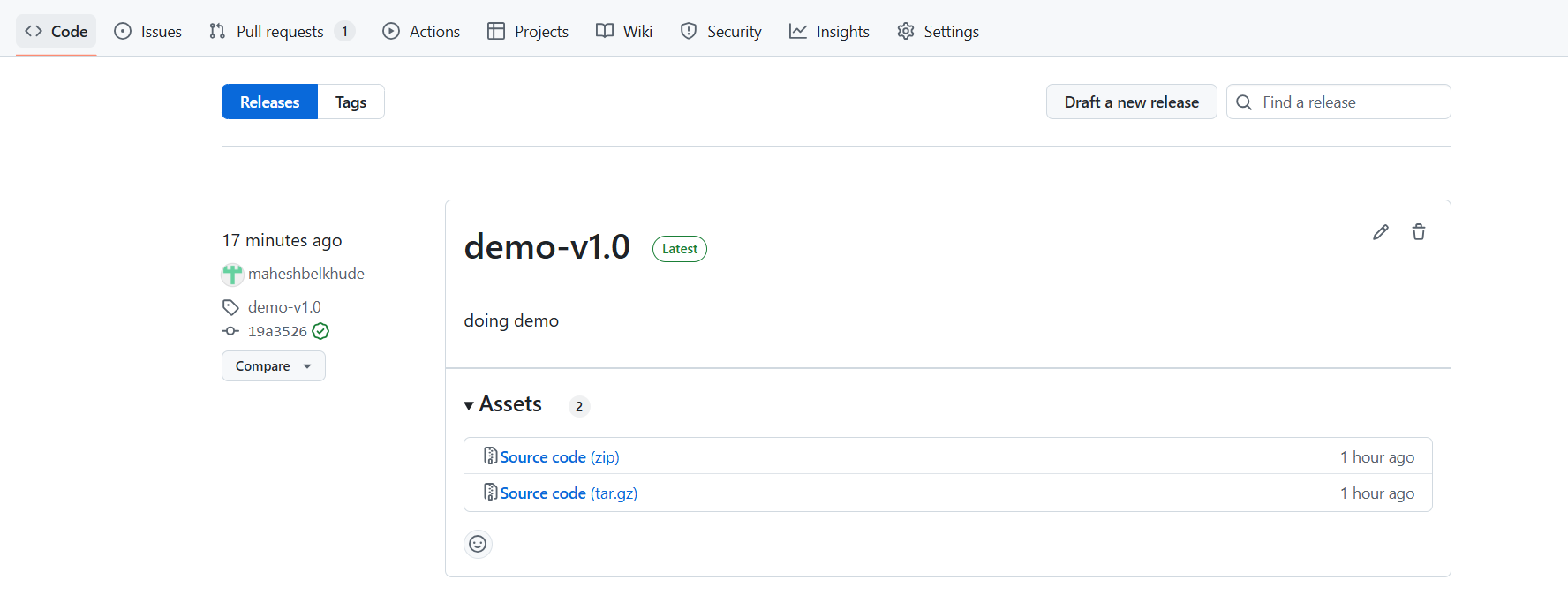
**4. Bugfix Branches (bugfix/<bug-id>)**

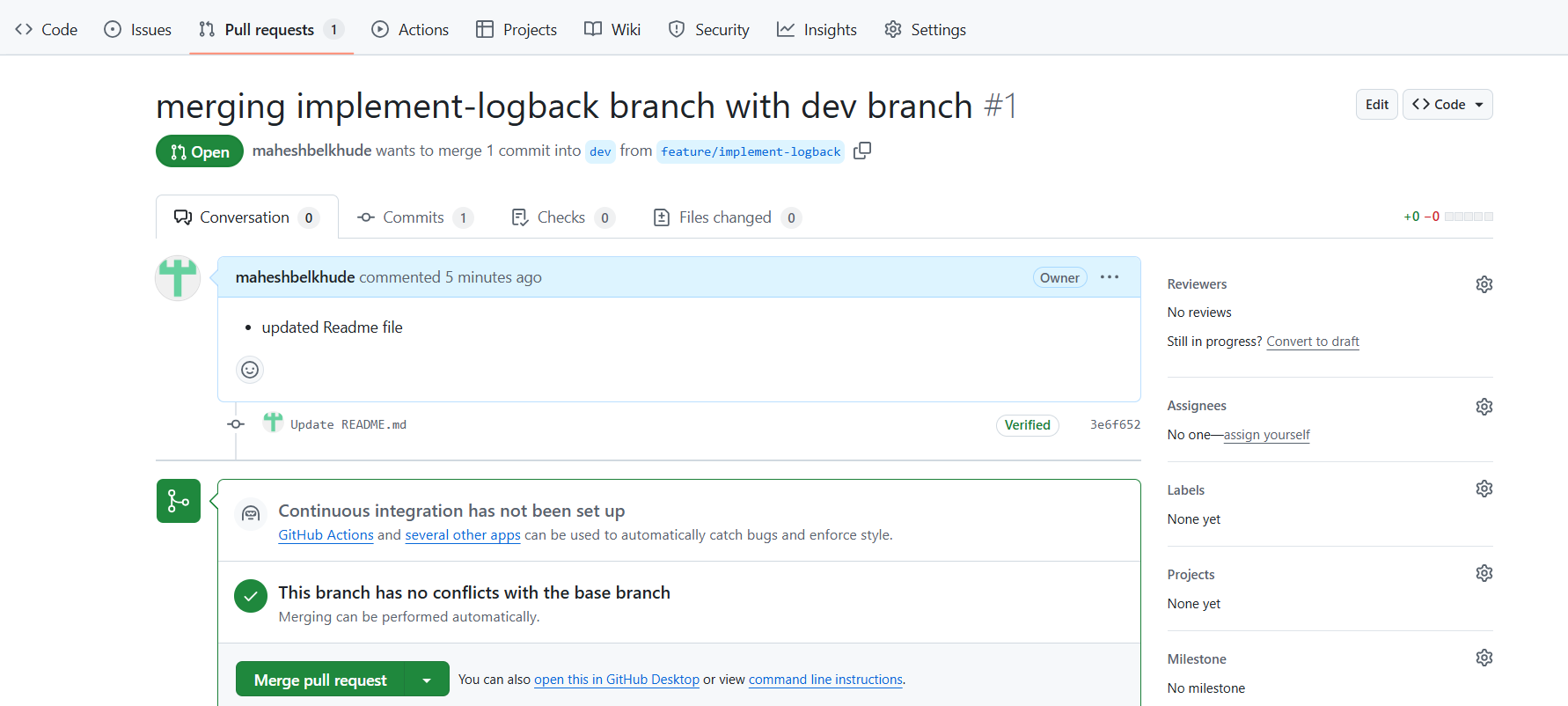
* Created for fixing bugs in the develop/dev branch.
* Branched from develop/dev.
* Merged back into develop/dev after fixing and testing.
* Naming convention: bugfix/bug-id or bug-name.

**5. Release Branches (release/<version>)**

* Created when preparing for a new production release.
* Branched from develop/dev.
* Used for final testing, minor fixes.
* Merged into dev or main after release.
* Naming convention **release/version-number** (e.g., release/1.0.0) or **release/stable-1.0**.







**Step: Code Review**

* Assign reviewers.
* Reviewers can add comments and request changes.
* If changes are requested
* Make updates locally.
* Commit and push the changes

