

Source Control

1. Purpose This document provides guidelines for managing version control to ensure consistent and efficient collaboration, tracking, and auditing of project changes.

2. Scope This policy applies to all team members, developers, and other members of the project

3. Tools Used

- **Version Control System (VCS):** github
- **Repository Hosting Platform:** github

4. Repository Structure

- **Branches:**
 - Main: first viable release until project is finished
 - Development sprint: Each sprint will have its own separate branch that is not merged into main.
 - Feature branches: Split off and merged back into sprint branch
 - Hotfix branches: Should not be needed.
- **Directory Structure:** Clearly define the folder structure and file organization.

5. Workflow

- **Commit Guidelines:**
 - Write meaningful commit messages.
 - Group related changes into one commit.
 - Avoid committing sensitive data (e.g., passwords, API keys).

6. Code Review Process

- All changes must go through a code review by at least 2 team members.
- Use pull requests or merge requests for reviewing changes.

7. Conflict Resolution

- Steps to handle merge conflicts:
 - Identify conflicting files.
 - Discuss with collaborators if necessary.
 - Test changes thoroughly after resolution.

8. Backup and Recovery

- Regularly backup repositories to a secure location.
- Use the version history for recovery if data is lost or corrupted.

9. Access Control

- Define user roles (e.g., Admin, Contributor, Reviewer).
- Set permissions for branches or repositories.

10. Best Practices

- Commit often and in small increments.
- Keep branches up-to-date with main/master.
- Use tags for marking releases.