THISARA KANDAGE

UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

CI/CD Pipeline for Automated Code Quality Checks Using GitHub Actions

Project Description:

This project demonstrates the implementation of a **Continuous Integration (CI)** pipeline using **GitHub Actions** to automatically enforce coding standards and maintain code quality across Python projects. The core objective was to integrate a **code linting workflow** that is triggered on every code push to the repository. This was achieved using **GitHub Super-Linter**, an open-source GitHub Action that bundles multiple popular linters into a single action.

The workflow was designed to:

- **Automatically trigger** on each push event to the repository.
- **Check out the code** from the repository using the actions/checkout step.
- **Run Super-Linter**, which executed multiple linters including:
    - flake8 for Python code style.
    - black for Python code formatting.
    - isort for import order.
    - mypy for static type checking.
    - yamllint for YAML file validation.

The setup ensures that all code committed to the repository adheres to widely accepted Python standards (PEP8 and PEP257), formatted cleanly, and structured properly before merging into the main branch.

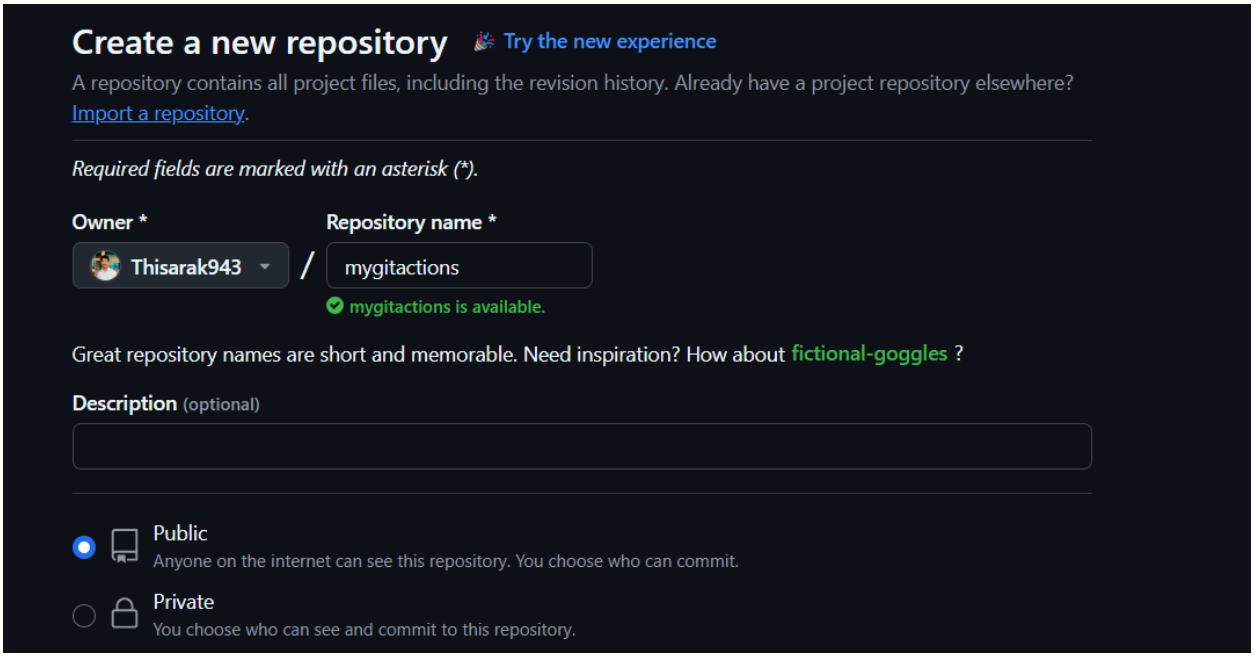
The project also included writing multiple Python files (main.py and greeting.py) and integrating them into the CI pipeline to validate the linting process and ensure the system works reliably for multiple files and commits.

THISARA KANDAGE

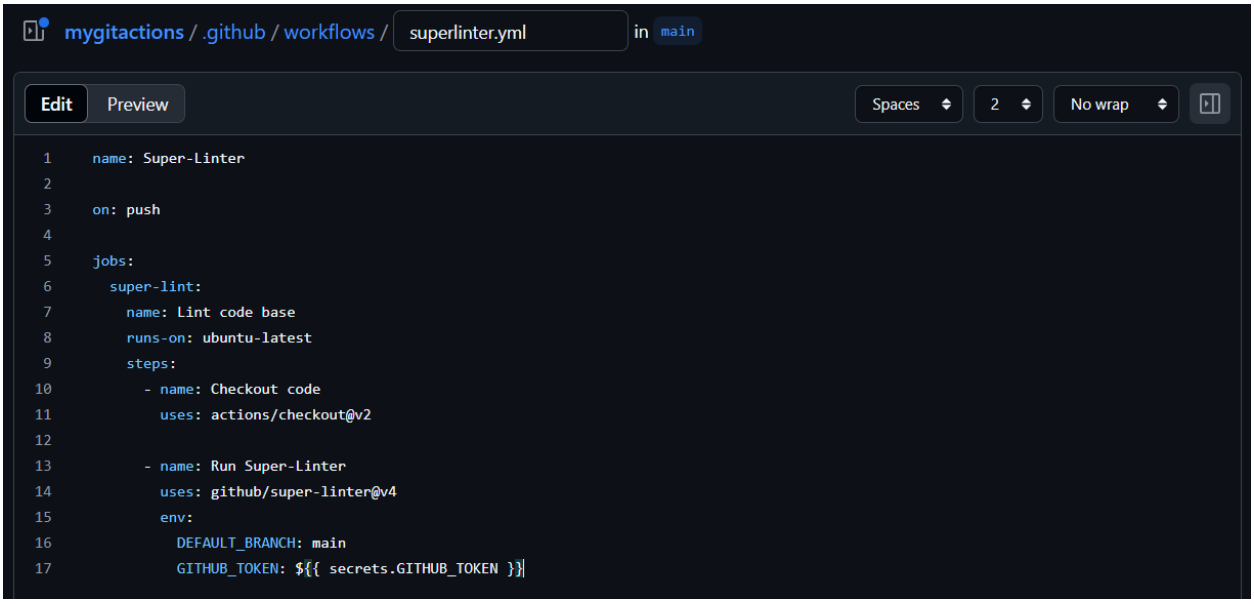
UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

- ## GitHub repo creation



- ## Create workflow steps

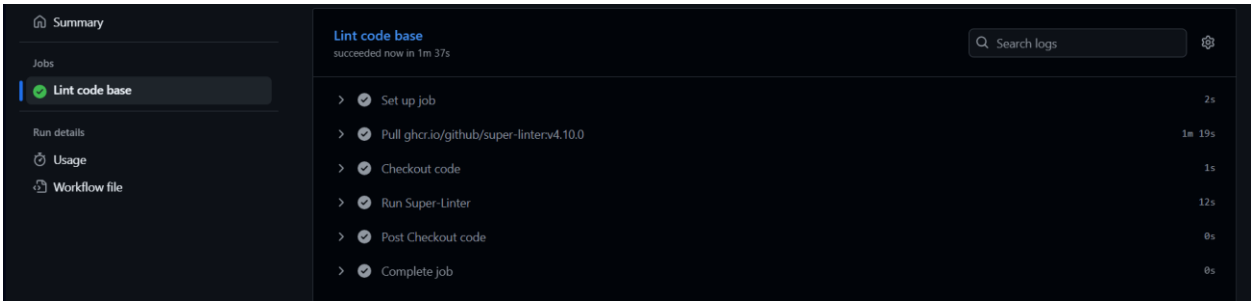


THISARA KANDAGE

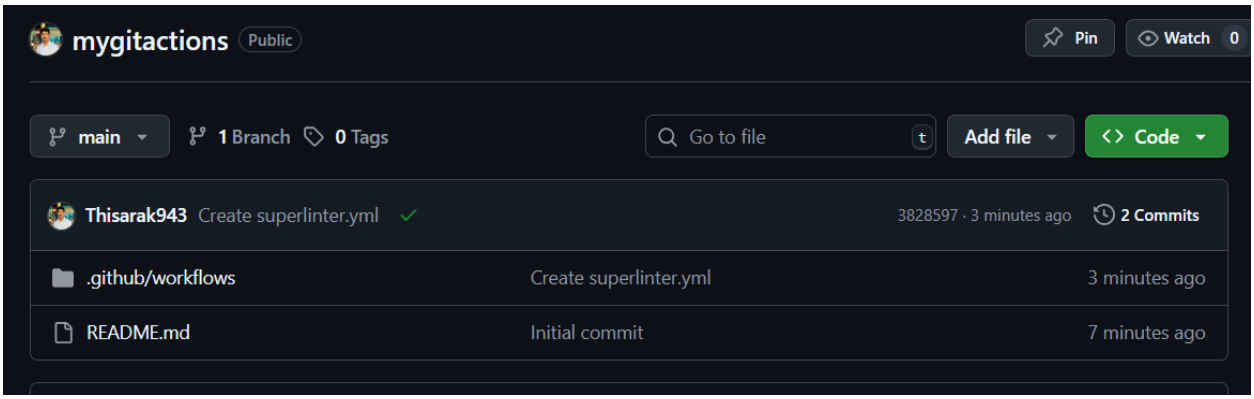
UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

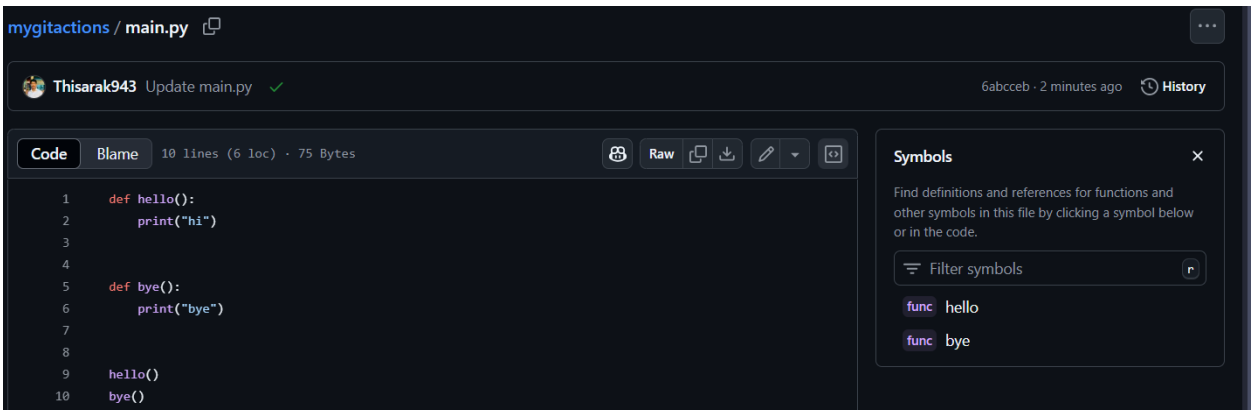
- **After running actions in GitHub**



- **All the checks have been passed**



- **The tried with push some codes**

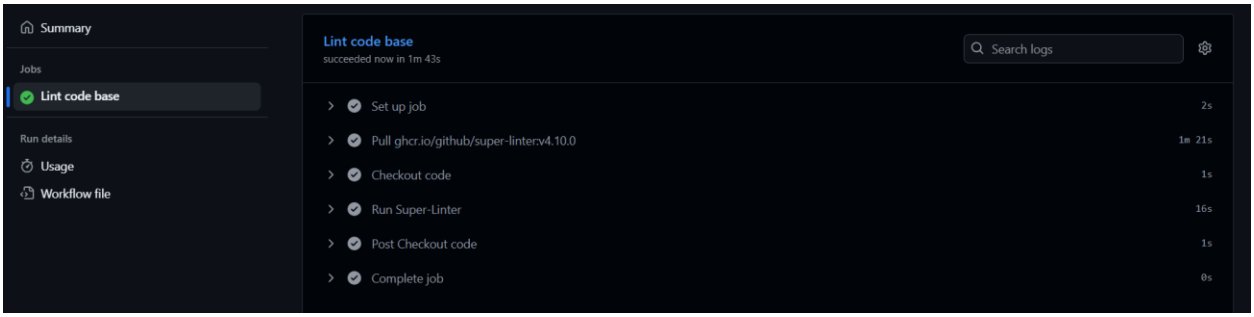


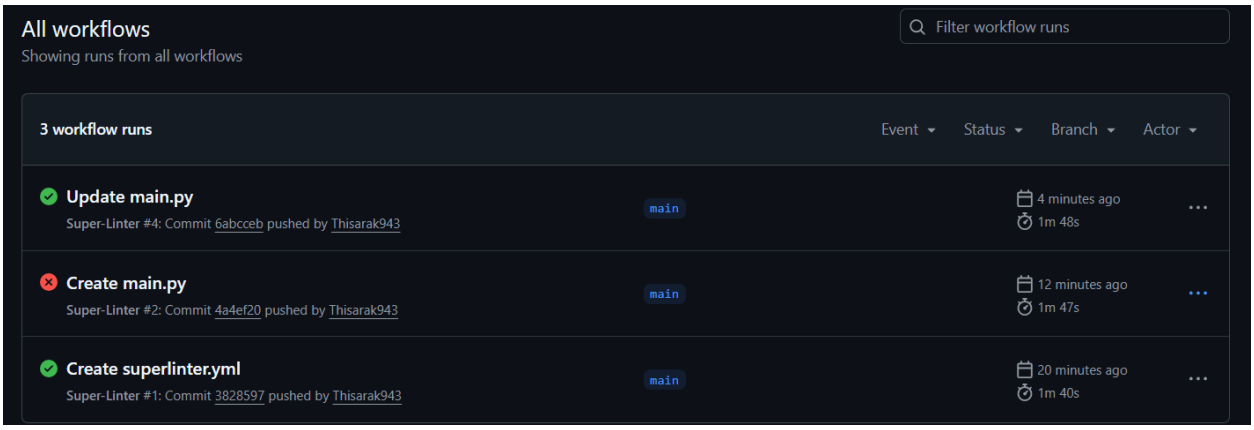
THISARA KANDAGE

UNDERGRADUATE - SLIIT

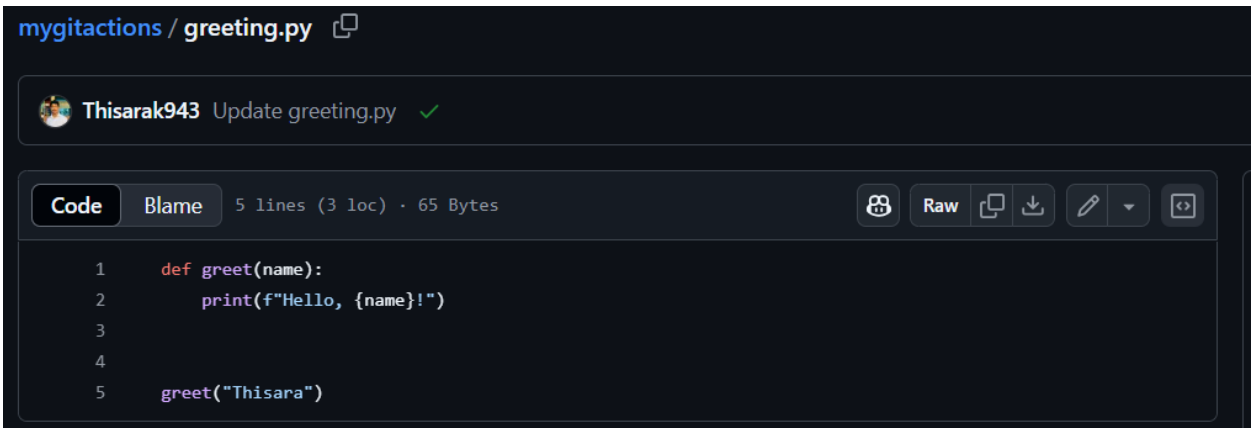
E-mail LinkedIn GitHub Website

- **After triggers running**





- **Check git actions again for another file**



THISARA KANDAGE

UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

- **After pushing code**

What I Learned:

1. GitHub Actions CI/CD Concepts

- Gained practical experience in setting up and customizing GitHub Actions workflows.
- Learned how to configure event triggers (like on: push) to automate tasks.
- Understood the job structure: defining jobs, steps, and using actions.

THISARA KANDAGE

UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

2. Using Super-Linter for Code Quality

- Understood the role of Super-Linter in automating code analysis with tools like:
  - flake8 (Python style guide enforcement),
  - black (auto-formatting),
  - isort (organizing imports),
  - mypy (type hint checking),
  - yamllint (YAML validation).
- Learned how Super-Linter simplifies code quality checks with minimal configuration.
- Explored how Super-Linter logs errors clearly, helping developers fix issues quickly.

3. YAML Configuration and Workflow Best Practices

- Learned the YAML syntax and structure for defining workflows.
- Understood the importance of correctly structuring YAML files and including document start (---).
- Practiced best practices in writing maintainable and scalable workflow files.

4. Clean Code Practices

- Learned how to write Python code that complies with standard style guides (PEP8, PEP257).
- Understood the importance of consistent formatting and spacing (e.g., blank lines after functions).
- Gained familiarity with using linters to catch subtle issues that improve readability and maintainability.

5. End-to-End Automation Mindset

- Understood how CI/CD helps enforce coding discipline in team environments.
- Realized how integrating linting tools in the pipeline reduces manual code review overhead.
- Learned how automation tools support scalability and quality control in real-world development workflows.

THISARA KANDAGE

UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website