CI/CD Pipeline Integration Report

Moumoni Roy

February 23, 2025

1 Introduction

This report documents the integration of automated tests into a CI/CD pipeline using GitHub Actions for a Flask-based API.

2 Repository Overview

The project repository contains the following key files:

- **app.py** - The main Flask API application.
- **test_app.py** - A set of automated tests using pytest.
- **requirements.txt** - Dependencies required for the application.
- **.github/workflows/cd-ci.yml** - GitHub Actions workflow configuration.

3 CI/CD Workflow

The GitHub Actions workflow is triggered on:

- Push events to the `main` branch.
- Pull requests targeting the `main` branch.

It performs the following steps:

1. Checks out the repository.
2. Sets up Python 3.12.
3. Installs required dependencies.
4. Starts the Flask application.
5. Runs the automated tests with pytest.
6. Uploads the test results as artifacts.

4 Test Results

The pipeline executed tests with the following results:

```
<?xml version="1.0" encoding="utf-8"?>
<testsuites>
    <testsuite name="pytest" errors="0" failures="0" skipped="0" tests="3">
        <testcase classname="test_app" name="test_home" time="0.005" />
        <testcase classname="test_app" name="test_add" time="0.003" />
        <testcase classname="test_app" name="test_invalid_input" time="0.002" />
    </testsuite>
</testsuites>
```

5 Conclusion

The CI/CD pipeline was successfully set up and executed tests automatically.
All test cases passed, ensuring the reliability of the Flask API.