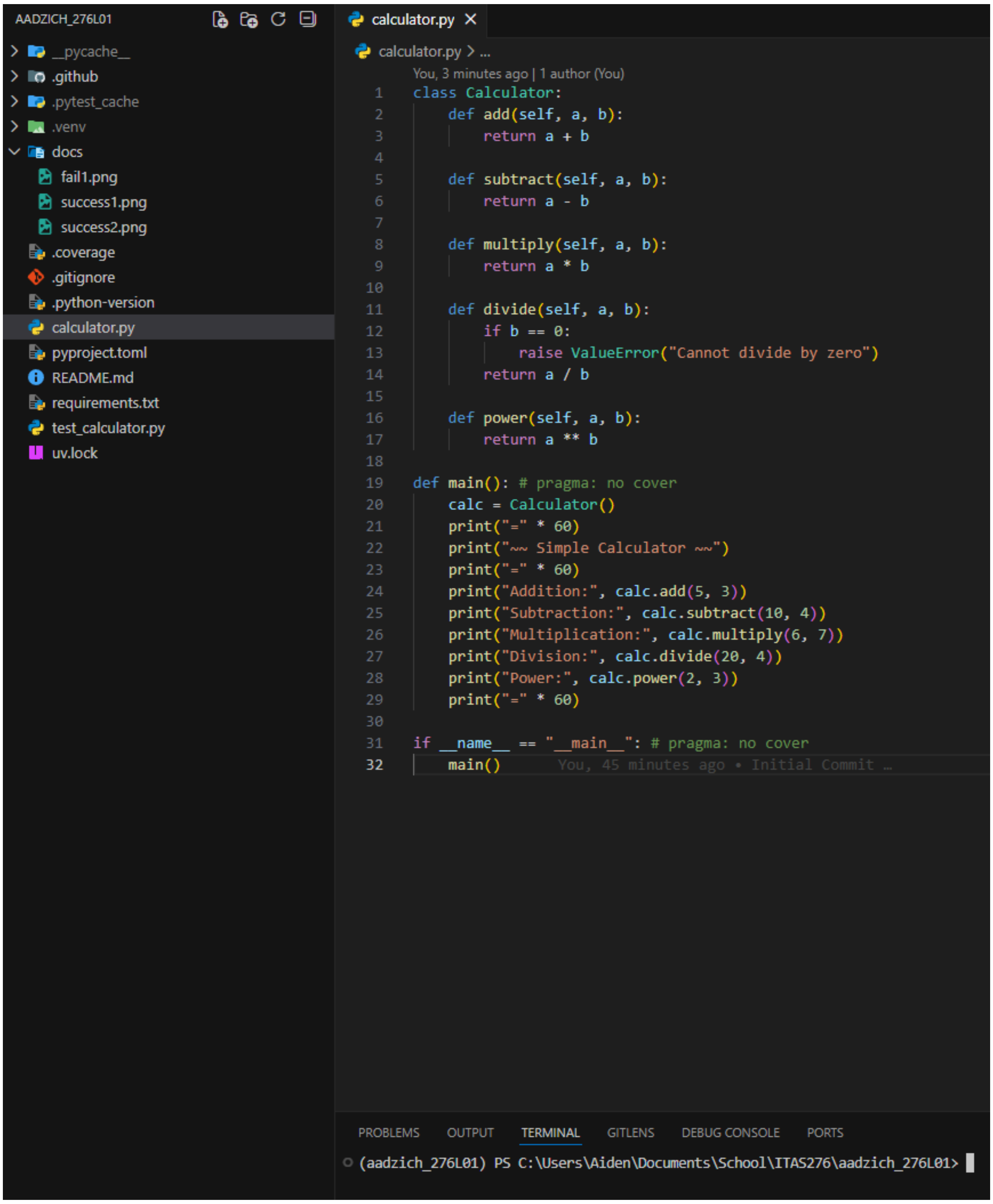
ITAS 276
DevSecOps

Lab #2
Using GitHub Actions for CI

Aiden Adzich
2026/02/08

Github Repo:
https://github.com/aidenadzich/calculator-ci

Figure 1
Local Development Environment

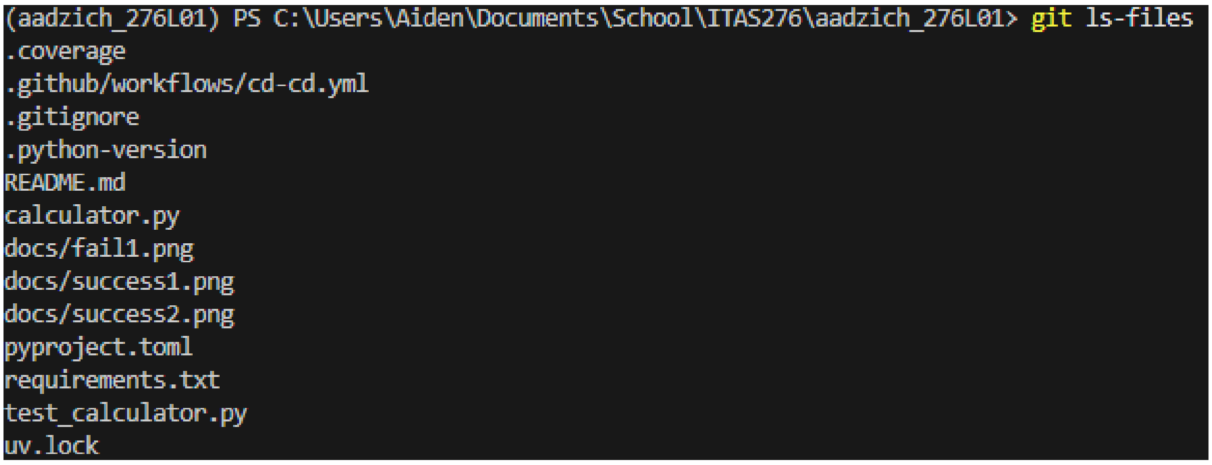


```python
class Calculator:
    def add(self, a, b):
        return a + b

    def subtract(self, a, b):
        return a - b

    def multiply(self, a, b):
        return a * b

    def divide(self, a, b):
        if b == 0:
            raise ValueError("Cannot divide by zero")
        return a / b

    def power(self, a, b):
        return a ** b

def main(): # pragma: no cover
    calc = Calculator()
    print("=" * 60)
    print("~~ Simple Calculator ~~")
    print("=" * 60)
    print("Addition:", calc.add(5, 3))
    print("Subtraction:", calc.subtract(10, 4))
    print("Multiplication:", calc.multiply(6, 7))
    print("Division:", calc.divide(20, 4))
    print("Power:", calc.power(2, 3))
    print("=" * 60)

if __name__ == "__main__": # pragma: no cover
    main()        You, 45 minutes ago • Initial Commit …
```

```
(aadzich_276L01) PS C:\Users\Aiden\Documents\School\ITAS276\aadzich_276L01>
```

Figure 2

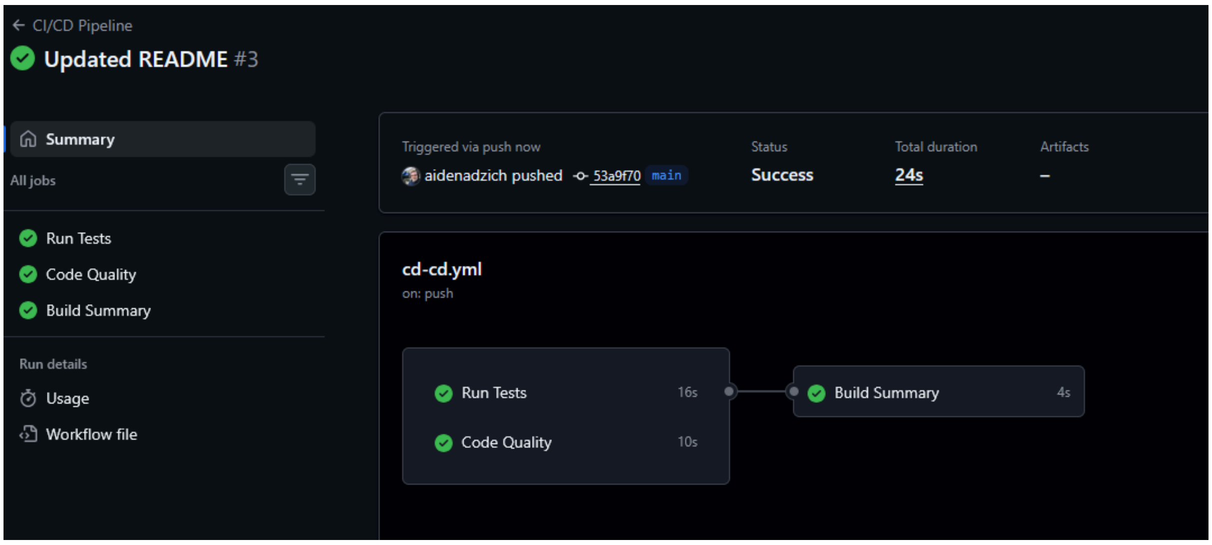
Git ls Results



```
(aadzich_276L01) PS C:\Users\Aiden\Documents\School\ITAS276\aadzich_276L01> git ls-files
.coverage
.github/workflows/cd-cd.yml
.gitignore
.python-version
README.md
calculator.py
docs/fail1.png
docs/success1.png
docs/success2.png
pyproject.toml
requirements.txt
test_calculator.py
uv.lock
```

Figure 3

GitHub action success



← CI/CD Pipeline

Updated README #3

Summary

All jobs

Run Tests

Code Quality

Build Summary

Run details

Usage

Workflow file

Triggered via push now

aidenadzich pushed 53a9f70 main

Status: Success

Total duration: 24s

Artifacts: –

cd-cd.yml

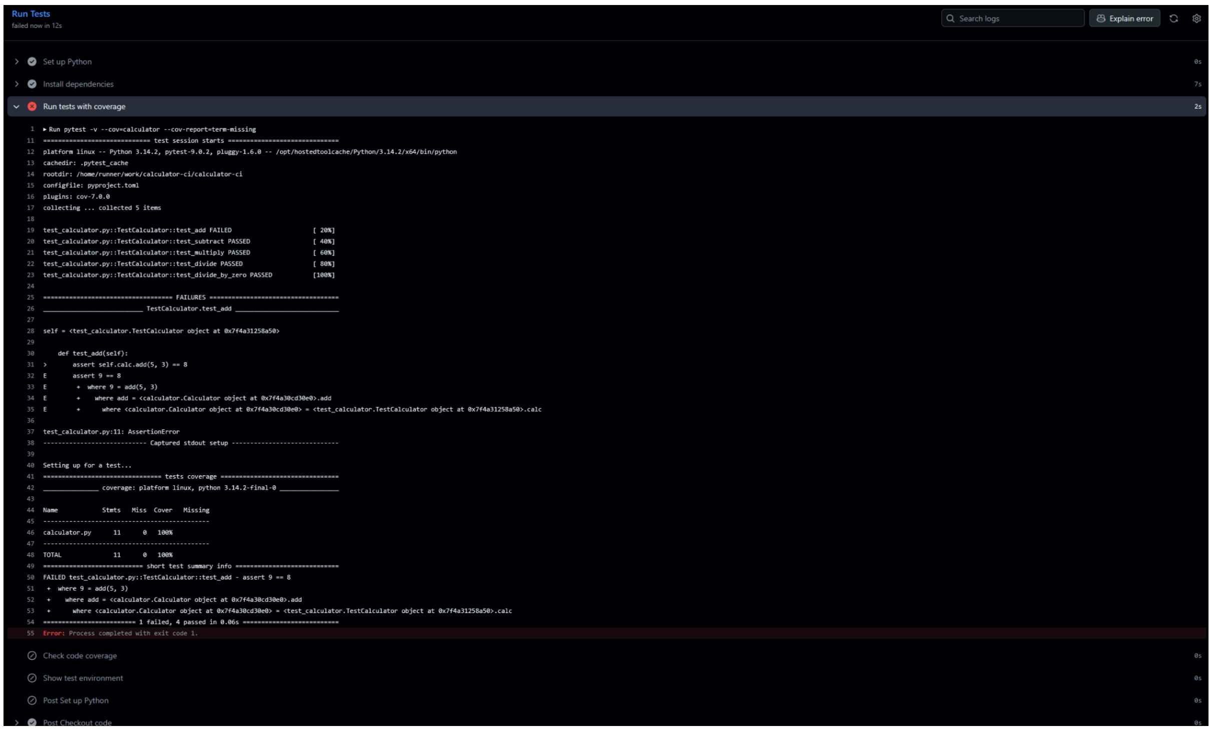
on: push

Run Tests 16s

Code Quality 10s

Build Summary 4s

Figure 4
GitHub action fail



```
▸ Run pytest -v --cov=calculator --cov-report=term-missing
============================= test session starts ==============================
platform linux -- Python 3.14.2, pytest-9.0.2, pluggy-1.6.0 -- /opt/hostedtoolcache/Python/3.14.2/x64/bin/python
cachedir: .pytest_cache
rootdir: /home/runner/work/calculator-ci/calculator-ci
configfile: pyproject.toml
plugins: cov-7.0.0
collecting ... collected 5 items

test_calculator.py::TestCalculator::test_add FAILED                      [ 20%]
test_calculator.py::TestCalculator::test_subtract PASSED                 [ 40%]
test_calculator.py::TestCalculator::test_multiply PASSED                 [ 60%]
test_calculator.py::TestCalculator::test_divide PASSED                   [ 80%]
test_calculator.py::TestCalculator::test_divide_by_zero PASSED           [100%]

=================================== FAILURES ===================================
___________________________ TestCalculator.test_add ____________________________

self = <test_calculator.TestCalculator object at 0x7f4a31258a50>

    def test_add(self):
>       assert self.calc.add(5, 3) == 8
E       assert 9 == 8
E        +  where 9 = add(5, 3)
E        +    where add = <calculator.Calculator object at 0x7f4a30cd30e0>.add
E        +      where <calculator.Calculator object at 0x7f4a30cd30e0> = <test_calculator.TestCalculator object at 0x7f4a31258a50>.calc

test_calculator.py:11: AssertionError
---------------------------- Captured stdout setup -----------------------------

Setting up for a test...
================================ tests coverage ================================
_______________ coverage: platform linux, python 3.14.2-final-0 ________________

Name            Stmts   Miss  Cover   Missing
---------------------------------------------
calculator.py      11      0   100%
---------------------------------------------
TOTAL              11      0   100%
=========================== short test summary info ============================
FAILED test_calculator.py::TestCalculator::test_add - assert 9 == 8
 +  where 9 = add(5, 3)
 +    where add = <calculator.Calculator object at 0x7f4a30cd30e0>.add
 +      where <calculator.Calculator object at 0x7f4a30cd30e0> = <test_calculator.TestCalculator object at 0x7f4a31258a50>.calc
========================= 1 failed, 4 passed in 0.06s ==========================
Error: Process completed with exit code 1.
```

Reflection

The CI/CD pipeline acts as a safety net by automatically catching bugs on every push before they reach the production environment. Without it, we would need to rely on manual testing, which is both slower and prone to human error. We ran tests and linting in parallel to save time. By having both tasks execute simultaneously, we were able to check the code as it was being tested, cutting down the wait time for results. 100% code coverage ensures that every bit of logic in our code is being verified and tested during the build. This is enforced using the --cov-fail-under flag, which will automatically fail the test if the code coverage drops below the given threshold. For a team of developers, this setup would ensure that one persons broken or messy code won't interfere with the work of the rest of the team.