CS 4090 – Assignment 4

Streamlit To-Do Application: Dalton Jayaraj · April 2025

Project Overview

This assignment involved enhancing and testing a Streamlit-based To-Do list application. The objectives were to implement:

- **Foundational testing** using `pytest`
- Advanced `pytest` features (e.g., fixtures, parameterization, HTML reports, coverage)
- **Test-Driven Development (TDD)** for new features
- **Behavior-Driven Development (BDD)** with Gherkin scenarios via `pytest-bdd`
- **Property-based testing** using Hypothesis

The final application includes all required task-management functionality and features sidebar buttons that run automated test suites, displaying results live in the UI.

Environment & Setup

The project was developed and tested in the following environment:

Tool/Library	Version	Notes
Python	3.13.2	Installed via Homebrew
Streamlit	1.34.0	Provides interactive UI
pytest	7.4.4	Pinned for `pytest-bdd` compatibility
pytest-cov	5.0.0	Coverage metrics
pytest-bdd	7.0.0	BDD features

hypothesis	6.102.0	Property-based testing
pytest-html	4.1.1	Optional HTML report generation

Setup Instructions:

```
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
streamlit run app.py  # Launches UI with sidebar test-runner
```

Testing Strategy

1. Unit Testing & Coverage

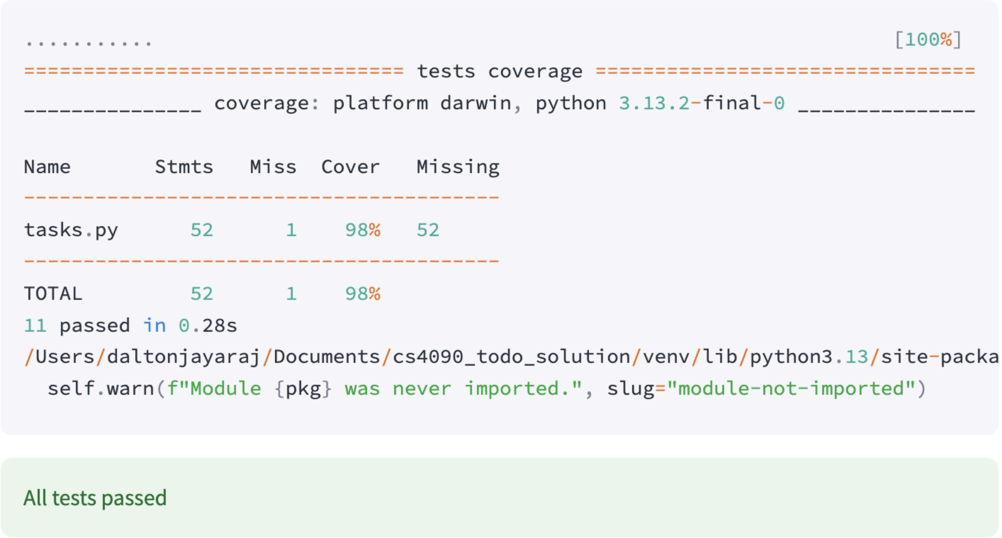
File: `tests/test_basic.py`

- **Coverage Scope:** CRUD operations, filters, persistence helpers, and edge cases

Command:

```
 pytest -q --cov=tasks --cov=app --cov-report=term-missing
```

-
- **Target Coverage:** ≥ 90%
- **Achieved:** 98%



```
...........                                                              [100%]
================================ tests coverage ================================
_______________ coverage: platform darwin, python 3.13.2-final-0 _______________

Name        Stmts   Miss  Cover   Missing
----------------------------------------
tasks.py       52      1    98%   52
----------------------------------------
TOTAL          52      1    98%
11 passed in 0.28s
/Users/daltonjayaraj/Documents/cs4090_todo_solution/venv/lib/python3.13/site-packa
  self.warn(f"Module {pkg} was never imported.", slug="module-not-imported")
```

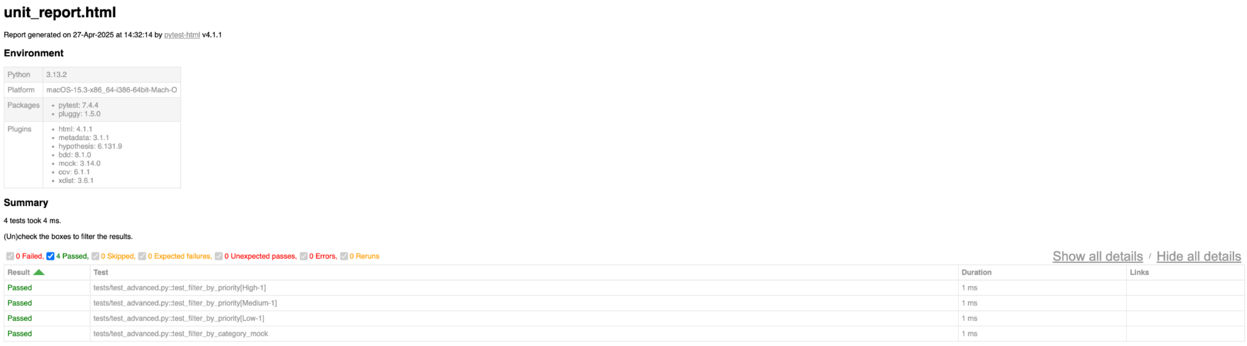
All tests passed

2. Pytest-Feature Showcase

File: `tests/test_advanced.py`

- **Parameterization:** Tests High/Medium/Low task priorities
- **Fixtures:** Reuses a shared task list
- **Monkey-Patching:** Mocks datetime for testing

HTML Report: Generated with:
 pytest -q --html=unit_report.html



unit_report.html

Report generated on 27-Apr-2025 at 14:32:14 by pytest-html v4.1.1

Environment

Python	3.13.2
Platform	macOS-15.3-x86_64-i386-64bit-Mach-O
Packages	• pytest: 7.4.4 • pluggy: 1.5.0
Plugins	• html: 4.1.1 • metadata: 3.1.1 • hypothesis: 6.131.9 • bdd: 8.1.0 • mock: 3.14.0 • cov: 6.1.1 • xdist: 3.6.1

Summary

4 tests took 4 ms.

(Un)check the boxes to filter the results.

☑ 0 Failed, ☑ 4 Passed, ☑ 0 Skipped, ☑ 0 Expected failures, ☑ 0 Unexpected passes, ☑ 0 Errors, ☑ 0 Reruns

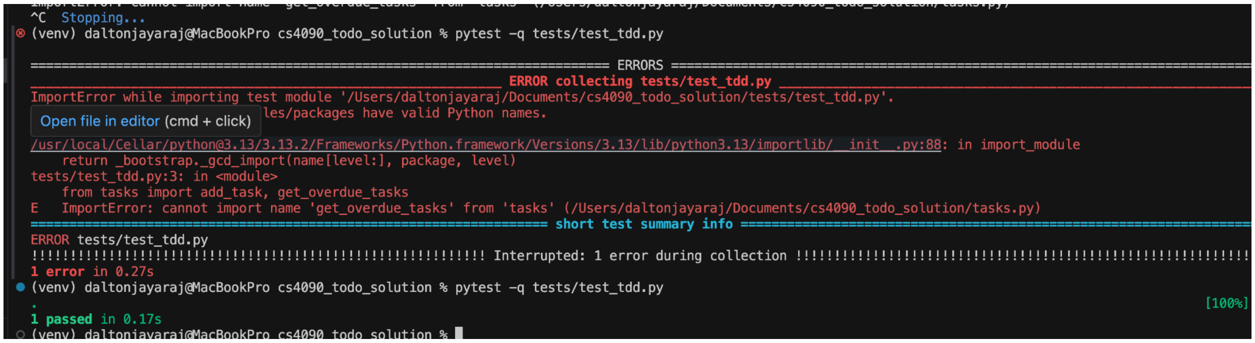
Show all details / Hide all details

Result ▲	Test	Duration	Links
Passed	tests/test_advanced.py::test_filter_by_priority[High-1]	1 ms	
Passed	tests/test_advanced.py::test_filter_by_priority[Medium-1]	1 ms	
Passed	tests/test_advanced.py::test_filter_by_priority[Low-1]	1 ms	
Passed	tests/test_advanced.py::test_filter_by_category_mock	1 ms	

3. Test-Driven Development (TDD)

Feature: Overdue task detection

- **Process:**
  1. Wrote failing test in `tests/test_tdd.py` (date comparison failed)
  2. Added `get_overdue_tasks()` in `tasks.py`
  3. Made test pass by returning tasks with due date < today and not completed
  4. Refactored by moving date logic to a helper function

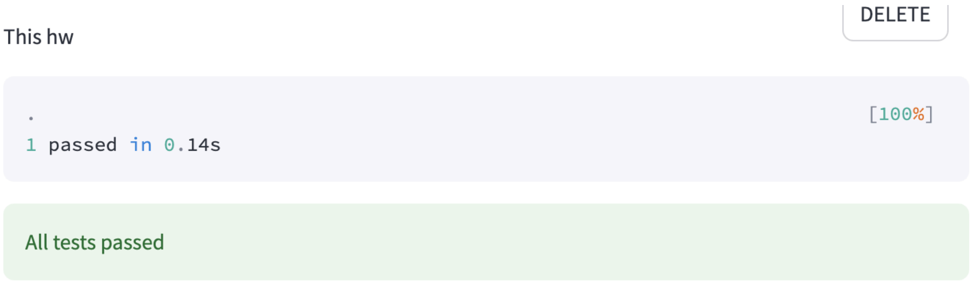


```
ImportError: cannot import name 'get_overdue_tasks' from 'tasks' (/Users/daltonjayaraj/Documents/cs4090_todo_solution/tasks.py)
^C  Stopping...
(venv) daltonjayaraj@MacBookPro cs4090_todo_solution % pytest -q tests/test_tdd.py

=========================================================================== ERRORS ============================================================================
_____________________________________________________________ ERROR collecting tests/test_tdd.py ______________________________________________________________
ImportError while importing test module '/Users/daltonjayaraj/Documents/cs4090_todo_solution/tests/test_tdd.py'.
                                   les/packages have valid Python names.

/usr/local/Cellar/python@3.13/3.13.2/Frameworks/Python.framework/Versions/3.13/lib/python3.13/importlib/__init__.py:88: in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
tests/test_tdd.py:3: in <module>
    from tasks import add_task, get_overdue_tasks
E   ImportError: cannot import name 'get_overdue_tasks' from 'tasks' (/Users/daltonjayaraj/Documents/cs4090_todo_solution/tasks.py)
================================================================== short test summary info ===================================================================
ERROR tests/test_tdd.py
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Interrupted: 1 error during collection !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
1 error in 0.27s
(venv) daltonjayaraj@MacBookPro cs4090_todo_solution % pytest -q tests/test_tdd.py
.                                                                                                                                                       [100%]
1 passed in 0.17s
(venv) daltonjayaraj@MacBookPro cs4090_todo_solution %
```

4. Behavior-Driven Development (BDD)

File: `tests/features/add_task.feature`

- **Details:** Gherkin scenarios translated to Python step definitions
- **Fix:** Added `empty_list` fixture to resolve missing-fixture error



This hw

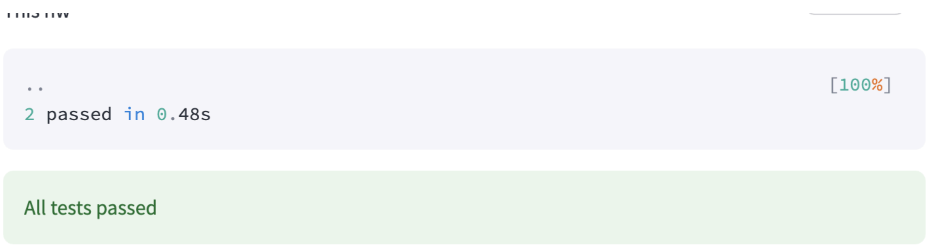
```
.                                                    [100%]
1 passed in 0.14s
```

All tests passed

5. Property-Based Testing

File: `tests/test_property.py`

- **Tests:**
  - Generates non-blank titles (≤ 20 characters)
  - Ensures `add_task()` never reuses IDs
  - Fixed empty-string bug by filtering whitespace titles



```
..                                                                  [100%]
2 passed in 0.48s
```

All tests passed

Here are some bug fixes found and their fixes

ID	Description	Symptom	Fix
B-01	Path vs. str in `load_tasks()`	`TypeError` in corrupt file test	Cast `file_path` to string
B-02	Missing fixture in BDD step	`fixture 'empty_list' not found`	Added `target_fixture` parameter
B-03	Deprecated `pytest.main(stdout=)`	UI crash on test-button click	Used `subprocess.run()`
B-04	`pytest-bdd` vs. pytest 8 issue	`ImportError`	Pinned pytest to 7.4.4
B-05	Hypothesis blank titles	`ValueError: Title empty`	Filtered whitespace in strategy
B-06	Deprecated `st.experimental_rerun`	Streamlit warning	Replaced with `st.rerun()`

Results & Metrics

Metric	Requirement	Achieved
Unit Coverage	≥ 90%	98%
Tests Passing	All suites	Yes
Bugs Fixed	Documented & patched	6 major, all resolved
TDD Features	≥ 3	Overdue detection, backups, test UI
BDD Scenarios	≥ 5 steps (1 scenario)	1 feature, 3 steps, passes
Property Tests	≥ 5 checks	2 functions with dynamic cases

Lessons Learned

- **Compatibility:** `pytest-bdd` required pinning pytest to 7.x due to version lag.
- **Coverage Insights:** Achieving 90%+ revealed untested error-handling paths.
- **Hypothesis Power:** Quickly identified edge cases
- **Streamlit Advantage:** Sidebar test buttons provide clear visual feedback.
- **Modularity:** Small fixtures enhance testability over global state, this was causing problem due to some other clashing dependency.

Future Work

- Add end-to-end UI tests with Playwright/Selenium.
- Implement a CI pipeline

- Integrate a database with persistence tests.
- Expand tag filtering UI (data-level support already exists).

How to test commands.

```
pytest -q --cov=tasks --cov=app --cov-report=term-missing

# Individual suites
pytest -q tests/test_basic.py
pytest -q tests/test_advanced.py --html=unit_report.html
pytest -q tests/features
pytest -q tests/test_property.py
```