

Stephen Kautt  
Capstone 1 Assignment 4  
27 April 2025

## PART 1: UNIT TESTS

Approach to Unit Testing:

1. Figure out what the most important features are of the application.
2. Write the unit tests to ensure that said features operate as intended, without having to go into the application and manually testing them all myself.
3. Use unique test cases that could cause edge case errors, to ensure that I accommodate those and that my application does not break after encountering one.

My 5 Unit Test Cases:

1. Testing the ability to Load Tasks
2. Testing the ability to Save Tasks
3. Testing the ability to Filter by Priority
4. Testing the ability to Filter by Category
5. Testing the ability to Filter by Completion

## PART 2: BUG REPORTING AND FIXING

Bugs Found:

1. ID Assignment was using `len(tasks) + 1` to determine the id of the task, this could cause duplicate ideas if a task was deleted.
2. JSON corruption had the potentiality of causing crashing, there was also not really good error check for it.
3. Search functionality was missing.
4. Overdue tasks functionality was missing.
5. There was no error checking on the undo button, which has the potential to cause crashes if not done properly.
6. Edit functionality was missing.

Fixes:

1. Switched to using the `generate_unique_ids()` function instead, to ensure that the id in the list is unique, removes the risk of duplicate ids.
2. JSON corruption now overwrites the `tasks.json` file with an empty list instead of causing the app to keep crashing.
3. Search functionality was now implemented to search tasks by name.
4. Overdue tasks functionality was now implemented to track tasks that have not been done by their due date.
5. Added error checking to the undo button to prevent crashes.
6. Added the ability to edit tasks with different names, dates, etc.

## PART 3: PYTEST FEATURES

1. All buttons working on streamlit. I did need to insert my own marker for pytest, which is not included in the default ones, but those are just warnings, the rest of the coverage and everything else works.

## PART 4: TEST DRIVEN DEVELOPMENT

Three New Features:

1. Mark Tasks as Important
2. Count the number of Completed Tasks
3. Clear Completed Tasks

Summary of Process:

1. Initial Test Creation
  - a. Wrote tests for three new features: marking as important, counting completed tasks, and clearing completed tasks.
2. Test Failure Demonstration
  - a. Tests failed because the features did not exist yet (ImportError).
3. Feature Implementation
  - a. Implemented `mark_as_important()`, `count_completed()`, `clear_completed()`.
4. Test Passing Verification
  - a. Ran pytest and all tests passed after implementing features.

Implemented Button in Streamlit App to run the tests

## PART 5: BEHAVIOR DRIVEN DEVELOPMENT

Here are the 5 tests that I have decided to implement:

1. Saving an empty list creates an empty JSON
2. Filter by High Priority Tasks
3. Getting Overdue Tasks
4. Making sure the generating unique ids is incremental
5. Loading a corrupt JSON returns an empty list

The BDD button passes all of my test cases.

## PART 6: PROPERTY BASED TESTING

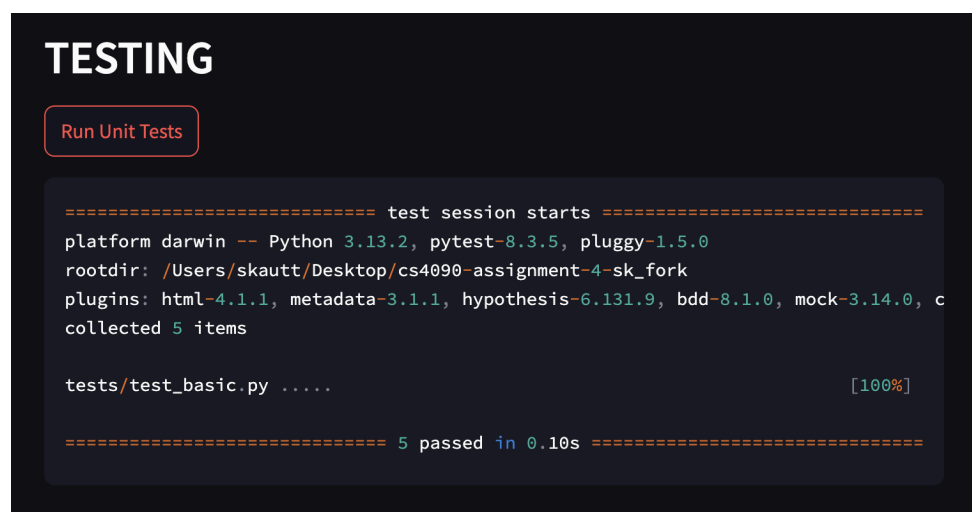
Here are the 5 hypothesis tests I have decided to do:

1. `test_generate_unique_id_property()`
  - a. New ID isn't in existing tasks and is positive
2. `test_filter_tasks_by_priority_property()`
  - a. Only tasks with "High" priority are returned
3. `test_filter_tasks_by_completion_property()`
  - a. Only completed tasks are returned
4. `test_filter_tasks_by_completion_false_property()`
  - a. Only incomplete tasks are returned
5. `test_task_creation_property()`
  - a. Each field of a new task has the right type

Button works on Streamlit under the Property Tests Section

TEST RESULTS:

Unit Tests:



The screenshot shows a Streamlit interface with a dark background. At the top, the word "TESTING" is displayed in large, bold, white capital letters. Below it, there is a red button with the text "Run Unit Tests". Underneath the button, a terminal window displays the output of a test session. The output includes the platform (darwin), Python version (3.13.2), and various installed packages (pytest-8.3.5, pluggy-1.5.0, etc.). It also shows the root directory and the number of collected items (5). The test results indicate that all 5 tests passed in 0.10s, with a progress bar showing 100% completion.

```
===== test session starts =====
platform darwin -- Python 3.13.2, pytest-8.3.5, pluggy-1.5.0
rootdir: /Users/skautt/Desktop/cs4090-assignment-4-sk_fork
plugins: html-4.1.1, metadata-3.1.1, hypothesis-6.131.9, bdd-8.1.0, mock-3.14.0, c
collected 5 items

tests/test_basic.py ..... [100%]

===== 5 passed in 0.10s =====
```

Coverage Report:

```

PYTEST FEATURES

Run Coverage Report

===== test session starts =====
platform darwin -- Python 3.13.2, pytest-8.3.5, pluggy-1.5.0
rootdir: /Users/skautt/Desktop/cs4090-assignment-4-sk_fork
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
collected 31 items

tests/test_advanced.py ..... [ 41%]
tests/test_basic.py ..... [ 58%]
tests/test_bdd.py ..... [ 74%]
tests/test_property.py ..... [ 90%]
tests/test_tdd.py ... [100%]

===== warnings summary =====

tests/test_advanced.py:19
/Users/skautt/Desktop/cs4090-assignment-4-sk_fork/tests/test_advanced.py:19:
PytestUnknownMarkWarning: Unknown pytest.mark.mock - is this a typo? You can register custom marks
to avoid this warning - for details, see https://docs.pytest.org/en/stable/how-to/mark.html
    @pytest.mark.mock

tests/test_advanced.py:33
/Users/skautt/Desktop/cs4090-assignment-4-sk_fork/tests/test_advanced.py:33:
PytestUnknownMarkWarning: Unknown pytest.mark.mock - is this a typo? You can register custom marks
to avoid this warning - for details, see https://docs.pytest.org/en/stable/how-to/mark.html
    @pytest.mark.mock

tests/test_advanced.py:43
/Users/skautt/Desktop/cs4090-assignment-4-sk_fork/tests/test_advanced.py:43:
PytestUnknownMarkWarning: Unknown pytest.mark.mock - is this a typo? You can register custom marks
to avoid this warning - for details, see https://docs.pytest.org/en/stable/how-to/mark.html
    @pytest.mark.mock

tests/test_advanced.py:58
/Users/skautt/Desktop/cs4090-assignment-4-sk_fork/tests/test_advanced.py:58:
PytestUnknownMarkWarning: Unknown pytest.mark.mock - is this a typo? You can register custom marks
to avoid this warning - for details, see https://docs.pytest.org/en/stable/how-to/mark.html
    @pytest.mark.mock

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
===== tests coverage =====
_____ coverage: platform darwin, python 3.13.2-final-0 _____

Name      Stmts  Miss  Cover   Missing
-----
src/tasks.py   41    2    95%   105-106

TOTAL        41    2    95%

===== 31 passed, 4 warnings in 1.34s =====
```

## Parameterized Tests:

### PYTEST FEATURES

Run Coverage Report

Run Parameterized Tests

```
===== test session starts =====
platform darwin -- Python 3.13.2, pytest-8.3.5, pluggy-1.5.0
rootdir: /Users/skautt/Desktop/cs4090-assignment-4-sk_fork
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
collected 31 items / 29 deselected / 2 selected

tests/test_advanced.py ..                                [100%]

===== warnings summary =====

tests/test_advanced.py:19
/Users/skautt/Desktop/cs4090-assignment-4-sk_fork/tests/test_advanced.py:19:
PytestUnknownMarkWarning: Unknown pytest.mark.mock - is this a typo? You can register custom marks
to avoid this warning - for details, see https://docs.pytest.org/en/stable/how-to/mark.html
  @pytest.mark.mock

tests/test_advanced.py:33
/Users/skautt/Desktop/cs4090-assignment-4-sk_fork/tests/test_advanced.py:33:
PytestUnknownMarkWarning: Unknown pytest.mark.mock - is this a typo? You can register custom marks
to avoid this warning - for details, see https://docs.pytest.org/en/stable/how-to/mark.html
  @pytest.mark.mock

tests/test_advanced.py:43
/Users/skautt/Desktop/cs4090-assignment-4-sk_fork/tests/test_advanced.py:43:
PytestUnknownMarkWarning: Unknown pytest.mark.mock - is this a typo? You can register custom marks
to avoid this warning - for details, see https://docs.pytest.org/en/stable/how-to/mark.html
  @pytest.mark.mock

tests/test_advanced.py:58
/Users/skautt/Desktop/cs4090-assignment-4-sk_fork/tests/test_advanced.py:58:
PytestUnknownMarkWarning: Unknown pytest.mark.mock - is this a typo? You can register custom marks
to avoid this warning - for details, see https://docs.pytest.org/en/stable/how-to/mark.html
  @pytest.mark.mock

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
===== 2 passed, 29 deselected, 4 warnings in 0.10s =====
```

## Mocking Tests:

### PYTEST FEATURES

Run Coverage Report

Run Parameterized Tests

Run Mocking Tests

```
===== test session starts =====
platform darwin -- Python 3.13.2, pytest-8.3.5, pluggy-1.5.0
rootdir: /Users/skautt/Desktop/cs4090-assignment-4-sk_fork
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
collected 31 items / 27 deselected / 4 selected

tests/test_advanced.py .... [100%]

===== warnings summary =====

tests/test_advanced.py:19
/Users/skautt/Desktop/cs4090-assignment-4-sk_fork/tests/test_advanced.py:19:
PytestUnknownMarkWarning: Unknown pytest.mark.mock - is this a typo? You can register custom marks
to avoid this warning - for details, see https://docs.pytest.org/en/stable/how-to/mark.html
  @pytest.mark.mock

tests/test_advanced.py:33
/Users/skautt/Desktop/cs4090-assignment-4-sk_fork/tests/test_advanced.py:33:
PytestUnknownMarkWarning: Unknown pytest.mark.mock - is this a typo? You can register custom marks
to avoid this warning - for details, see https://docs.pytest.org/en/stable/how-to/mark.html
  @pytest.mark.mock

tests/test_advanced.py:43
/Users/skautt/Desktop/cs4090-assignment-4-sk_fork/tests/test_advanced.py:43:
PytestUnknownMarkWarning: Unknown pytest.mark.mock - is this a typo? You can register custom marks
to avoid this warning - for details, see https://docs.pytest.org/en/stable/how-to/mark.html
  @pytest.mark.mock

tests/test_advanced.py:58
/Users/skautt/Desktop/cs4090-assignment-4-sk_fork/tests/test_advanced.py:58:
PytestUnknownMarkWarning: Unknown pytest.mark.mock - is this a typo? You can register custom marks
to avoid this warning - for details, see https://docs.pytest.org/en/stable/how-to/mark.html
  @pytest.mark.mock

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
===== 4 passed, 27 deselected, 4 warnings in 0.12s =====
```

## HTML Report:

```
PYTEST FEATURES

Run Coverage Report

Run Parameterized Tests

Run Mocking Tests

Run HTML Report

===== test session starts =====
platform darwin -- Python 3.13.2, pytest-8.3.5, pluggy-1.5.0
rootdir: /Users/skautt/Desktop/cs4090-assignment-4-sk_fork
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
collected 31 items

tests/test_advanced.py ..... [ 41%]
tests/test_basic.py ..... [ 58%]
tests/test_bdd.py ..... [ 74%]
tests/test_property.py ..... [ 90%]
tests/test_tdd.py ... [100%]

===== warnings summary =====
tests/test_advanced.py:19
/Users/skautt/Desktop/cs4090-assignment-4-sk_fork/tests/test_advanced.py:19:
PytestUnknownMarkWarning: Unknown pytest.mark.mock - is this a typo? You can register custom marks
to avoid this warning - for details, see https://docs.pytest.org/en/stable/how-to/mark.html
@pytest.mark.mock

tests/test_advanced.py:33
/Users/skautt/Desktop/cs4090-assignment-4-sk_fork/tests/test_advanced.py:33:
PytestUnknownMarkWarning: Unknown pytest.mark.mock - is this a typo? You can register custom marks
to avoid this warning - for details, see https://docs.pytest.org/en/stable/how-to/mark.html
@pytest.mark.mock

tests/test_advanced.py:43
/Users/skautt/Desktop/cs4090-assignment-4-sk_fork/tests/test_advanced.py:43:
PytestUnknownMarkWarning: Unknown pytest.mark.mock - is this a typo? You can register custom marks
to avoid this warning - for details, see https://docs.pytest.org/en/stable/how-to/mark.html
@pytest.mark.mock

tests/test_advanced.py:58
/Users/skautt/Desktop/cs4090-assignment-4-sk_fork/tests/test_advanced.py:58:
PytestUnknownMarkWarning: Unknown pytest.mark.mock - is this a typo? You can register custom marks
to avoid this warning - for details, see https://docs.pytest.org/en/stable/how-to/mark.html
@pytest.mark.mock

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
- Generated html report: file:///Users/skautt/Desktop/cs4090-assignment-4-sk_fork/report.html -
===== 31 passed, 4 warnings in 0.59s =====
```

## TDD Tests:

```
TDD TESTS

Run TDD Tests

===== test session starts =====
platform darwin -- Python 3.13.2, pytest-8.3.5, pluggy-1.5.0
rootdir: /Users/skautt/Desktop/cs4090-assignment-4-sk_fork
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
collected 3 items

tests/test_tdd.py ... [100%]

===== 3 passed in 0.09s =====
```

## BDD Tests:

```
BDD TESTS

Run BDD Tests

===== test session starts =====
platform darwin -- Python 3.13.2, pytest-8.3.5, pluggy-1.5.0
rootdir: /Users/skautt/Desktop/cs4090-assignment-4-sk_fork
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
collected 5 items

tests/test_bdd.py ..... [100%]

===== 5 passed in 0.12s =====
```

## Property Tests:

```
HYPOTHESIS TESTS

Run Property-Based Tests

===== test session starts =====
platform darwin -- Python 3.13.2, pytest-8.3.5, pluggy-1.5.0
rootdir: /Users/skautt/Desktop/cs4090-assignment-4-sk_fork
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
collected 5 items

tests/test_property.py ..... [100%]
===== Hypothesis Statistics =====

tests/test_property.py::test_generate_unique_id_property:

- during generate phase (0.10 seconds):
- Typical runtimes: < 1ms, of which < 1ms in data generation
- 100 passing examples, 0 failing examples, 9 invalid examples

- Stopped because settings.max_examples=100

tests/test_property.py::test_filter_tasks_by_priority_property:

- during generate phase (0.08 seconds):
- Typical runtimes: < 1ms, of which < 1ms in data generation
- 100 passing examples, 0 failing examples, 4 invalid examples

- Stopped because settings.max_examples=100

tests/test_property.py::test_filter_tasks_by_completion_property:

- during generate phase (0.09 seconds):
- Typical runtimes: ~ 0-1 ms, of which < 1ms in data generation
- 100 passing examples, 0 failing examples, 10 invalid examples

- Stopped because settings.max_examples=100

tests/test_property.py::test_filter_tasks_by_completion_false_property:

- during generate phase (0.09 seconds):
- Typical runtimes: < 1ms, of which < 1ms in data generation
- 100 passing examples, 0 failing examples, 13 invalid examples

- Stopped because settings.max_examples=100

tests/test_property.py::test_task_creation_property:

- during generate phase (0.06 seconds):
- Typical runtimes: < 1ms, of which < 1ms in data generation
- 100 passing examples, 0 failing examples, 0 invalid examples

- Stopped because settings.max_examples=100

===== 5 passed in 0.44s =====
```

## LESSONS LEARNED

The main lesson I learned is that pytest is an easy way to create test cases that can be used to test your application. Many times while programming, I try to manually find bugs and test them. This lets me automate the process ensuring that certain features operate in a manner that I want them to.