

Python Unit Testing (TDD) Assignment

Intern: Sumit Dhar

Date: 17th September, 2025

Objective

The purpose of this assignment was to complete a hands-on unit testing project by:

1. Setting up a Python virtual environment.
 2. Completing TDD-style implementations to pass existing test cases.
 3. Writing new unit tests with mocking.
 4. Completing boto3-related tests using moto.
 5. Generating a code coverage report.
-

Step 1: Forking and Cloning the Repository

- Forked the repository from the given source:

<https://gitlab.com/atulya.pandey/python-unittest-exercise>

- Cloned it to my local machine using:

```
git clone https://gitlab.com/sumiths.0015/python-unittest-exercise.git
```

Step 2: Creating & Activating Virtual Environment and Installing Project Dependencies

- Installed virtual environment using conda:

```
conda create -n <virtual_env_name>
```

- Activates virtual environment:

```
conda activate <virtual_env_name>
```

- Installed all required packages:

```
pip install -r requirements.txt
```

Step 3: Running Initial Tests

- Ran `pytest` before code changes to see failing tests: **pytest**

```
(testing_DG_1) PS F:\Internship\Python\unit_testing\python-unittest-exercise> pytest
Hint: make sure your test modules/packages have valid Python names.
Traceback:
C:\Users\Admin\anaconda3\Lib\importlib\__init__.py:88: in import_module
    return _bootstrap.gcd_import(name[level:], package, level)
    ~~~~~^~~~~~
tests\test_tdd_example.py:1: in <module>
    from src.tdd_example import TDDExample
E   ModuleNotFoundError: No module named 'src'

===== short test summary info =====
ERROR tests/test_dynamodb_example.py
ERROR tests/test_general_example.py
ERROR tests/test_s3_example.py
ERROR tests/test_tdd_example.py
!!!!!!!!!!!!!!!!!!!! Interrupted: 4 errors during collection !!!!!!!!!!!!!!!!!!!!!!!
===== 4 errors in 2.50s =====
```

Step 4: TDD Implementation in `tdd_example.py`

- Implemented all 4 functions to pass the tests in `test_tdd_example.py`.
- Ensured all test cases passed after implementation.

Step 5: Writing Unit Tests for `general_example.py`

- Wrote unit tests for the methods in `general_example.py`.
- Used mocking for `load_employee_rec_from_database()` when testing `fetch_emp_details()` using `unittest.mock`.

```
@patch.object(GeneralExample, 'load_employee_rec_from_database')
def test_fetch_emp_details(mocked_load):

    db_record = ['emp001', 'Sam', '100000']

    mocked_load.return_value = db_record

    expected = {
        'empId': db_record[0],
        'empName': db_record[1],
        'empSalary': db_record[2]
    }

    actual = general_example_instance.fetch_emp_details()
    assert expected == actual
```

Step 6: Completing test_dynamodb_example.py with moto

- Referred to test_s3_example.py for structure, and corrected using mock_aws.
- Used moto to mock AWS DynamoDB services using mock_aws.
- Ensured test methods successfully interacted with mocked resources.

Step 7: Verifying All Tests Pass

- Re-ran all tests to verify correctness using : pytest

```
(testing_DG_1) PS F:\Internship\Python\unit_testing\python-unittest-exercise> python -m pytest
===== test session starts =====
platform win32 -- Python 3.13.5, pytest-6.2.5, py-1.11.0, pluggy-1.5.0
rootdir: F:\Internship\Python\unit_testing\python-unittest-exercise
plugins: anyio-4.7.0, cov-3.0.0
collected 12 items

tests\test_dynamodb_example.py ... [ 25%]
tests\test_general_example.py ... [ 50%]
tests\test_s3_example.py .. [ 66%]
tests\test_tdd_example.py .... [100%]

===== 12 passed in 12.91s =====
(testing_DG_1) PS F:\Internship\Python\unit_testing\python-unittest-exercise> █
```

Step 8: Generating Code Coverage Report

- Generated report using:

```
python -m pytest --cov=src --cov-report=term-missing
```

```
(testing_DG_1) PS F:\Internship\Python\unit_testing\python-unittest-exercise> python
-m pytest --cov=src --cov-report=term-missing
===== test session starts =====
platform win32 -- Python 3.13.5, pytest-8.4.2, pluggy-1.5.0
rootdir: F:\Internship\Python\unit_testing\python-unittest-exercise
plugins: anyio-4.7.0, cov-7.0.0
collected 12 items

tests\test_dynamodb_example.py ... [ 25%]
tests\test_general_example.py ... [ 50%]
tests\test_s3_example.py .. [ 66%]
tests\test_tdd_example.py .... [100%]

===== tests coverage =====
coverage: platform win32, python 3.13.5-final-0
Name                               Stmts  Miss  Cover   Missing
-----
src\__init__.py                      0      0   100%
src\boto3_example.py                 19      0   100%
src\general_example.py               13      0   100%
src\tdd_example.py                   21      4    81%   12, 22, 29, 38
TOTAL                                53      4    92%
===== 12 passed in 13.57s =====
(testing_DG_1) PS F:\Internship\Python\unit_testing\python-unittest-exercise> █
```

Step 8: Analyzing "Miss" and correcting it

```
def reverse_string(self, input_str: str) -> str:
    """
    Reverses order of characters in string input_str.
    """
    if not input_str:
        return ""

    return input_str[::-1]

def find_longest_word(self, sentence: str) -> str:
    """
    Returns the longest word in string sentence.
    In case there are several, return the first.
    """
    if not sentence:
        return ""

    import re

    words = re.findall(r'\b\w+\b', sentence)

    if not words:
        return ""

    return max(words, key=len)
```

```
(testing_DG_1) PS F:\Internship\Python\unit_testing\python-unittest-exercise> python
-m pytest --cov=src --cov-report=term-missing
===== test session starts =====
platform win32 -- Python 3.13.5, pytest-8.4.2, pluggy-1.5.0
rootdir: F:\Internship\Python\unit_testing\python-unittest-exercise
plugins: anyio-4.7.0, cov-7.0.0
collected 12 items

tests\test_dynamodb_example.py ... [ 25%]
tests\test_general_example.py ... [ 50%]
tests\test_s3_example.py .. [ 66%]
tests\test_tdd_example.py .... [100%]

===== tests coverage =====
_____ coverage: platform win32, python 3.13.5-final-0 _____

Name                               Stmts  Miss  Cover   Missing
-----
src\__init__.py                     0      0  100%
src\boto3_example.py                19      0  100%
src\general_example.py              13      0  100%
src\tdd_example.py                  13      0  100%
-----
TOTAL                               45      0  100%
===== 12 passed in 13.68s =====
(testing_DG_1) PS F:\Internship\Python\unit_testing\python-unittest-exercise> |
```

Step 8: Feature branch pushed to GitHub:

<https://gitlab.com/sumiths.0015/python-unittest-exercise/-/tree/feature>

- All tests pass and code coverage report generated successfully.
-

Summary

In this assignment, I worked on a **Python - unit testing** project that followed the principles of **Test-Driven Development (TDD)**.

I began by setting up a virtual environment and installing all required dependencies. I then completed the implementation of test-driven methods to make existing test cases pass, wrote new unit tests using **unittest** and **pytest**, and used mocking for database-related functions.

For AWS-related functionality, I used the **moto** library to mock DynamoDB and completed test cases accordingly.

Finally, I ran all test cases to ensure they passed and generated a code coverage report to measure test completeness. The project was pushed to a remote **GitLab** repository as a **feature** branch.