Coverage Testing Report

GitHub Repository URL: https://github.com/NickBland/2810ICT-milestone-2.git

1. Test Summary

Tested Functions	Test Functions
`initDatabase(filePath)`	`test_initDatabase_with_valid_csv()`
	`test_initDatabase_with_empty_csv()`
	`test_initDatabase_with_invalid_file_path()`
	`test_initDatabase_with_empty_file_path()`
	`test_initDatabase_with_nonexistent_file()`
`searchDatabase(filters, db)`	`test_searchDatabase_with_keyword()`
	`test_searchDatabase_with_nutrient_range()`
	`test_searchDatabase_with_nutrient_level()`
	`test_searchDatabase_with_invalid_minmax()`
	`test_searchDatabase_with_all_filters()`
	`test_searchDatabase_with_low_protein()`
	`test_searchDatabase_with_mid_protein()`
	`test_searchDatabase_with_high_protein()`
	`test_searchDatabase_with_no_filters()`
`addToComparison(selected_food, comparison_list)`	`test_addComparisonNone()`
	`test_addComparisonAddOne()`
	`test_addComparisonAddTwo()`
	`test_addComparisonAddSame()`
`displayResults(results, grid)`	`test_displayResults()`

2. **Statement Coverage Test**

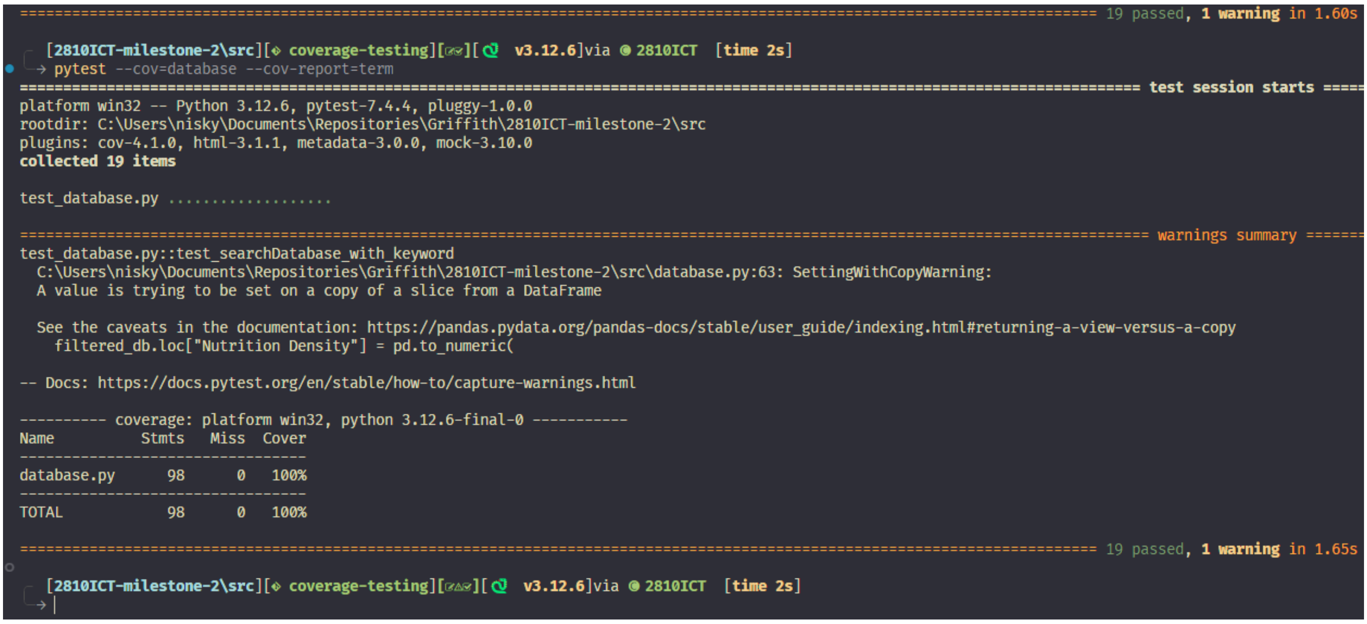
2.1 Description

For each of the functions that were required to be tested, the first test case was usually designed to test the function with valid inputs. This is because the function should be able to handle valid inputs correctly. From there, with the coverage tool, statements were analysed for what was 'missed' by this first test. Normally, at this point, it would be tests that should fail (i.e. invalid inputs). So, a new test that should fail was made, and then re-run with the coverage tool. This process was repeated until all statements were covered.

2.2 Testing Results

The following command was run to check the statement coverage of the tests. A screenshot of the report is included below.

```
pytest --cov=database --cov-report=term
```



```
=============================================================================================================== 19 passed, 1 warning in 1.60s

  [2810ICT-milestone-2\src][◆ coverage-testing][][ v3.12.6]via ● 2810ICT  [time 2s]
 → pytest --cov=database --cov-report=term
===================================================================================================================== test session starts =====
platform win32 -- Python 3.12.6, pytest-7.4.4, pluggy-1.0.0
rootdir: C:\Users\nisky\Documents\Repositories\Griffith\2810ICT-milestone-2\src
plugins: cov-4.1.0, html-3.1.1, metadata-3.0.0, mock-3.10.0
collected 19 items

test_database.py ...................

===================================================================================================================== warnings summary ======
test_database.py::test_searchDatabase_with_keyword
  C:\Users\nisky\Documents\Repositories\Griffith\2810ICT-milestone-2\src\database.py:63: SettingWithCopyWarning:
  A value is trying to be set on a copy of a slice from a DataFrame

  See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
    filtered_db.loc["Nutrition Density"] = pd.to_numeric(

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html

---------- coverage: platform win32, python 3.12.6-final-0 -----------
Name           Stmts   Miss  Cover
-----------------------------------
database.py       98      0   100%
-----------------------------------
TOTAL             98      0   100%

=============================================================================================================== 19 passed, 1 warning in 1.65s

  [2810ICT-milestone-2\src][◆ coverage-testing][][ v3.12.6]via ● 2810ICT  [time 2s]
 → |
```

3. **Branch Coverage Test**

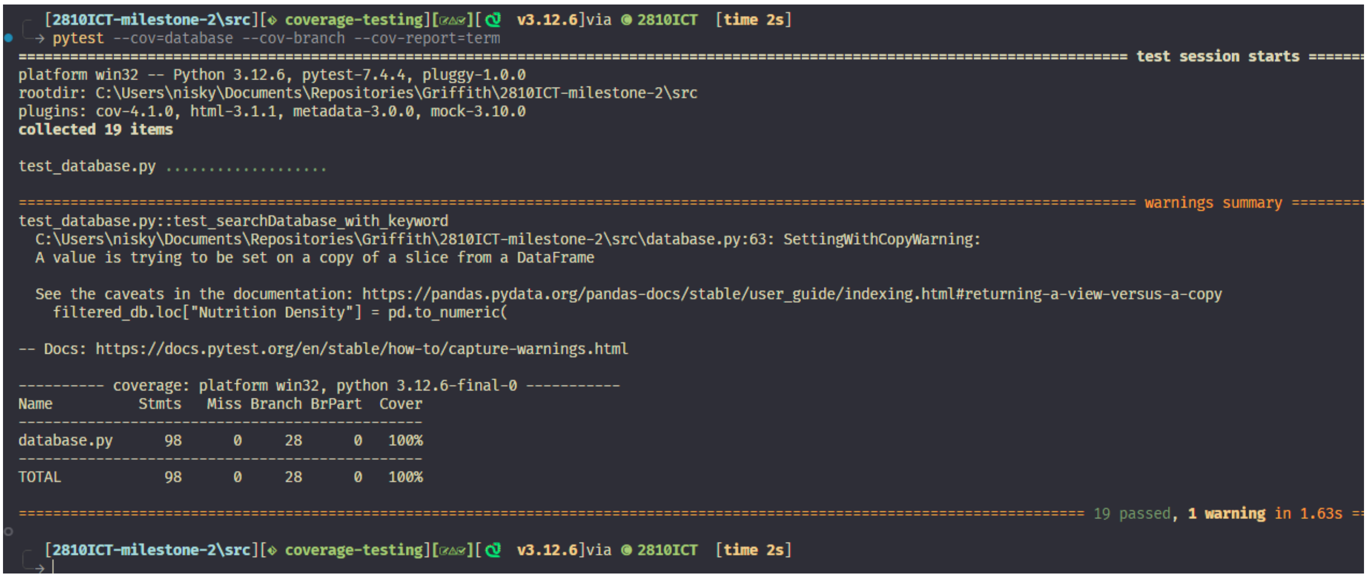
3.1 Description

The process for branch coverage was more-or-less the same. For each test case created, branch coverage was run to check if there were any branch statements that had not been covered by the test. If there were, a new test was created to cover that branch. This process was repeated until all branches were covered.

3.2 Testing Results

The following command was run to check the branch coverage of the tests. A screenshot of the report is included below.

```
pytest --cov=database --cov-branch --cov-report=term
```



```
  [2810ICT-milestone-2\src][◆ coverage-testing][][ v3.12.6]via ● 2810ICT  [time 2s]
 └→ pytest --cov=database --cov-branch --cov-report=term
========================================================================================================================= test session starts =======
platform win32 -- Python 3.12.6, pytest-7.4.4, pluggy-1.0.0
rootdir: C:\Users\nisky\Documents\Repositories\Griffith\2810ICT-milestone-2\src
plugins: cov-4.1.0, html-3.1.1, metadata-3.0.0, mock-3.10.0
collected 19 items

test_database.py ...................

========================================================================================================================== warnings summary ========
test_database.py::test_searchDatabase_with_keyword
  C:\Users\nisky\Documents\Repositories\Griffith\2810ICT-milestone-2\src\database.py:63: SettingWithCopyWarning:
  A value is trying to be set on a copy of a slice from a DataFrame

  See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
    filtered_db.loc["Nutrition Density"] = pd.to_numeric(

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html

---------- coverage: platform win32, python 3.12.6-final-0 -----------
Name          Stmts   Miss Branch BrPart  Cover
-----------------------------------------------
database.py      98      0     28      0   100%
-----------------------------------------------
TOTAL            98      0     28      0   100%

============================================================================================================================== 19 passed, 1 warning in 1.63s ==

  [2810ICT-milestone-2\src][◆ coverage-testing][][ v3.12.6]via ● 2810ICT  [time 2s]
 └→ |
```