Static Testing Tools and White Box Design

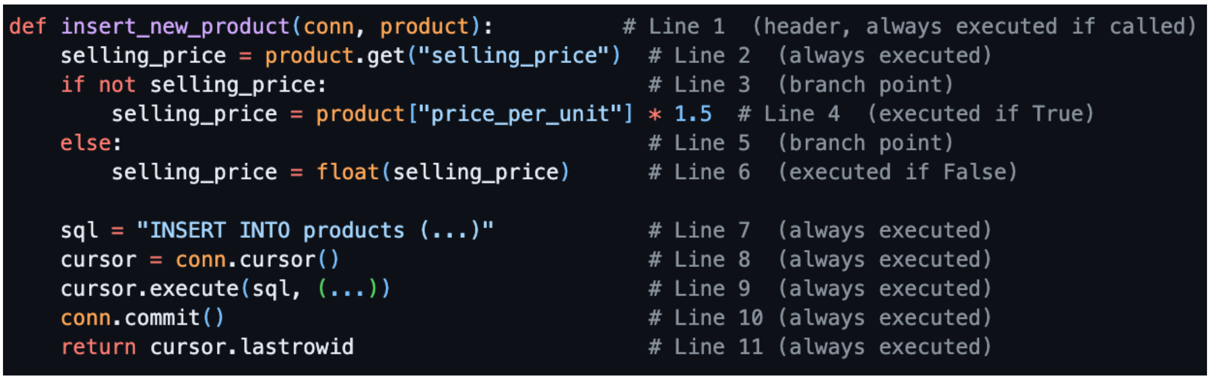
White Box Test Design:

Overview

This document outlines the white-box test design for the GSM (Grocery Store Manager) application, including:

- Statement Coverage (line coverage)
- Branch Coverage (decision coverage)
- Calculations for coverage metrics
- Test techniques used across unit, integration, and API tests

Example statement coverage



```
def insert_new_product(conn, product):          # Line 1  (header, always executed if called)
    selling_price = product.get("selling_price")  # Line 2  (always executed)
    if not selling_price:                         # Line 3  (branch point)
        selling_price = product["price_per_unit"] * 1.5  # Line 4  (executed if True)
    else:                                         # Line 5  (branch point)
        selling_price = float(selling_price)      # Line 6  (executed if False)

    sql = "INSERT INTO products (...)"            # Line 7  (always executed)
    cursor = conn.cursor()                        # Line 8  (always executed)
    cursor.execute(sql, (...))                    # Line 9  (always executed)
    conn.commit()                                 # Line 10 (always executed)
    return cursor.lastrowid                       # Line 11 (always executed)
```

Test Case 1: (selling_price provided)

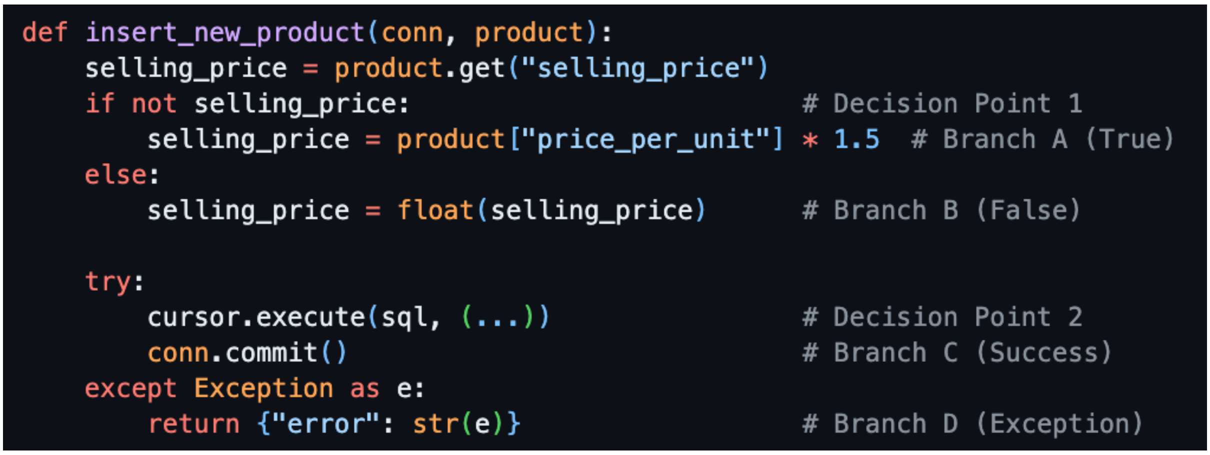
- Executes lines: 1, 2, 3 (False), 6, 7, 8, 9, 10, 11
- Lines NOT executed: 4

Test Case 2: (selling_price NOT provided)

- Executes lines: 1, 2, 3 (True), 4, 7, 8, 9, 10, 11
- Lines NOT executed: 6

Combined Statement Coverage: 11/11

Branch coverage example



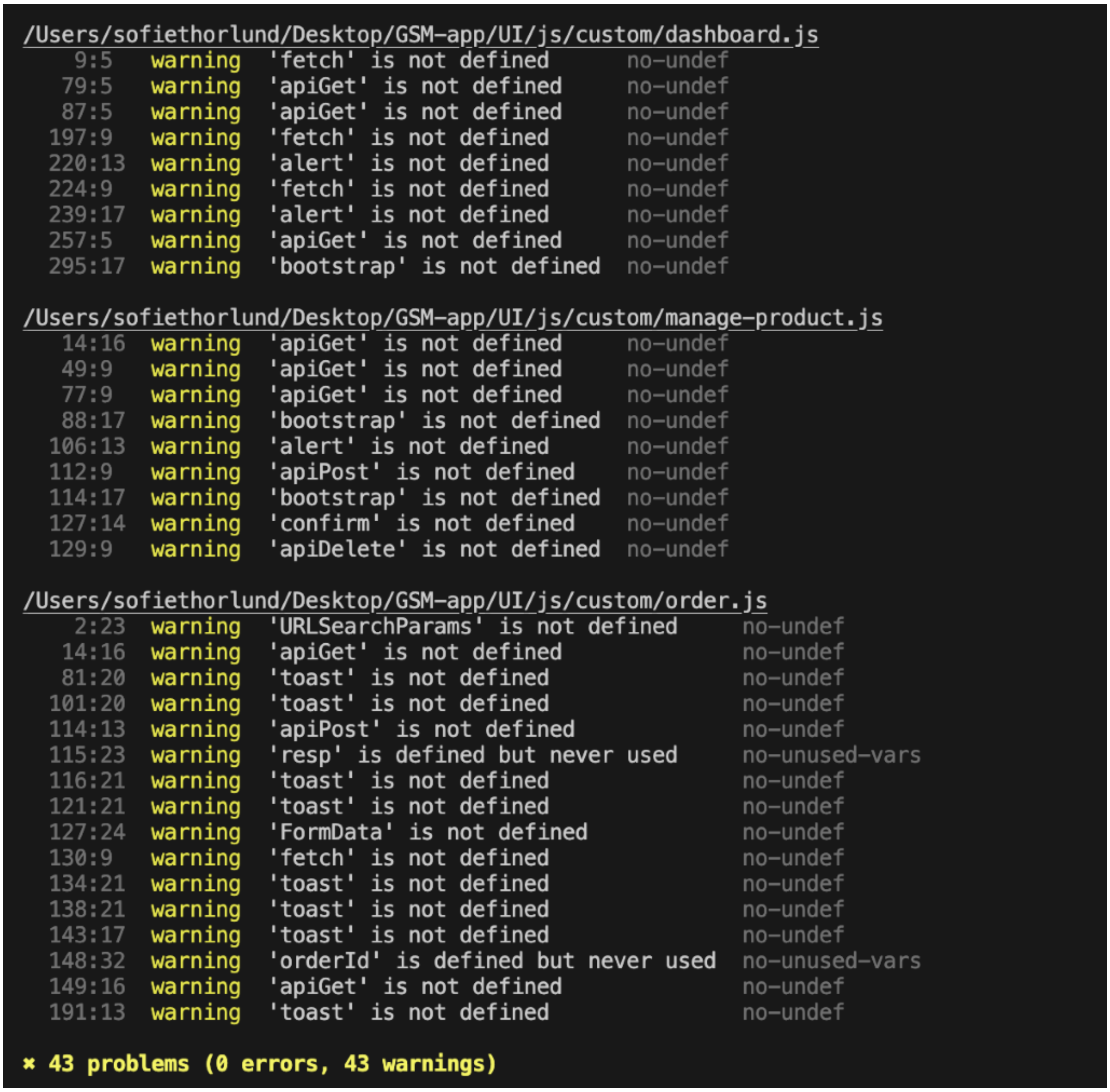
```python
def insert_new_product(conn, product):
    selling_price = product.get("selling_price")
    if not selling_price:                         # Decision Point 1
        selling_price = product["price_per_unit"] * 1.5  # Branch A (True)
    else:
        selling_price = float(selling_price)      # Branch B (False)

    try:
        cursor.execute(sql, (...))                # Decision Point 2
        conn.commit()                             # Branch C (Success)
    except Exception as e:
        return {"error": str(e)}                  # Branch D (Exception)
```

4 branches meaning 3 tests

1. Decision point 1 = A (True), Decision point 2 = C (Success)
2. Decision point 1 = B (False), Decision point 2 = C (Success)
3. Decision point 1 = A (True), Decision point 2 = D (Exception)

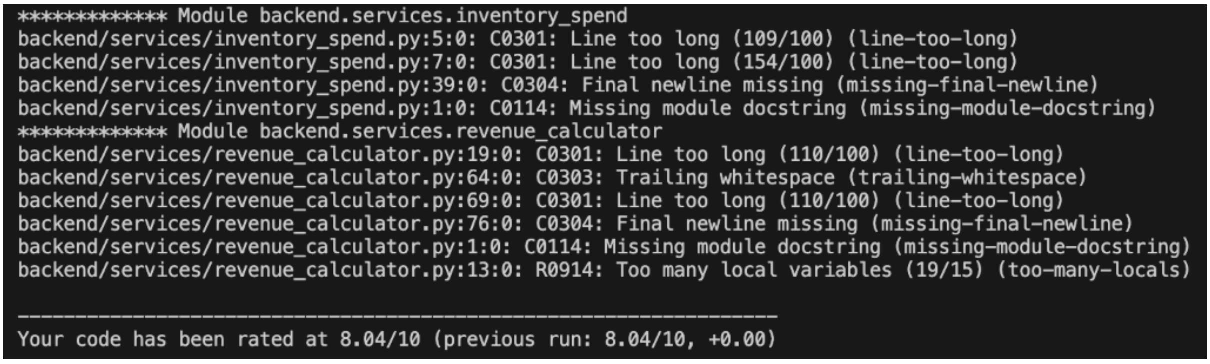
Combined branch coverage = 100%

ESLint:



```
/Users/sofiethorlund/Desktop/GSM-app/UI/js/custom/dashboard.js
    9:5   warning  'fetch' is not defined      no-undef
   79:5   warning  'apiGet' is not defined     no-undef
   87:5   warning  'apiGet' is not defined     no-undef
  197:9   warning  'fetch' is not defined      no-undef
  220:13  warning  'alert' is not defined      no-undef
  224:9   warning  'fetch' is not defined      no-undef
  239:17  warning  'alert' is not defined      no-undef
  257:5   warning  'apiGet' is not defined     no-undef
  295:17  warning  'bootstrap' is not defined  no-undef

/Users/sofiethorlund/Desktop/GSM-app/UI/js/custom/manage-product.js
   14:16  warning  'apiGet' is not defined     no-undef
   49:9   warning  'apiGet' is not defined     no-undef
   77:9   warning  'apiGet' is not defined     no-undef
   88:17  warning  'bootstrap' is not defined  no-undef
  106:13  warning  'alert' is not defined      no-undef
  112:9   warning  'apiPost' is not defined    no-undef
  114:17  warning  'bootstrap' is not defined  no-undef
  127:14  warning  'confirm' is not defined    no-undef
  129:9   warning  'apiDelete' is not defined  no-undef

/Users/sofiethorlund/Desktop/GSM-app/UI/js/custom/order.js
    2:23  warning  'URLSearchParams' is not defined     no-undef
   14:16  warning  'apiGet' is not defined              no-undef
   81:20  warning  'toast' is not defined               no-undef
  101:20  warning  'toast' is not defined               no-undef
  114:13  warning  'apiPost' is not defined             no-undef
  115:23  warning  'resp' is defined but never used     no-unused-vars
  116:21  warning  'toast' is not defined               no-undef
  121:21  warning  'toast' is not defined               no-undef
  127:24  warning  'FormData' is not defined            no-undef
  130:9   warning  'fetch' is not defined               no-undef
  134:21  warning  'toast' is not defined               no-undef
  138:21  warning  'toast' is not defined               no-undef
  143:17  warning  'toast' is not defined               no-undef
  148:32  warning  'orderId' is defined but never used  no-unused-vars
  149:16  warning  'apiGet' is not defined              no-undef
  191:13  warning  'toast' is not defined               no-undef

✖ 43 problems (0 errors, 43 warnings)
```

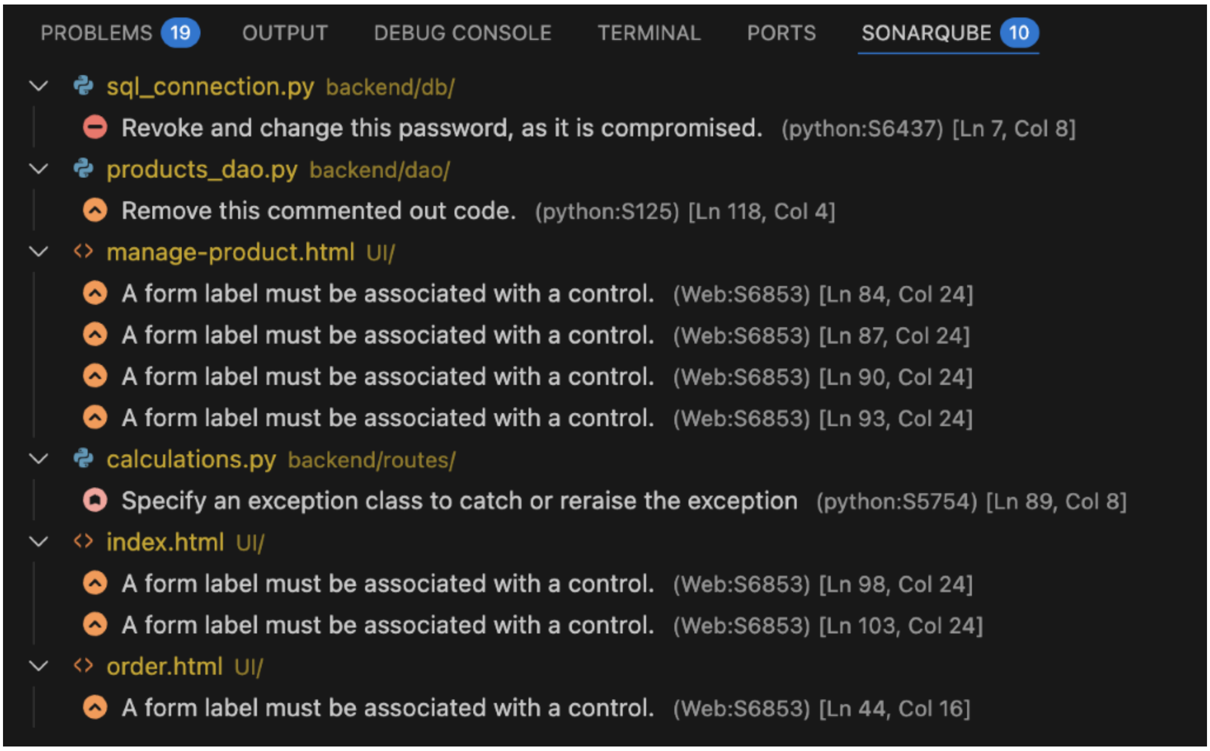
PyLint:



```
************* Module backend.services.inventory_spend
backend/services/inventory_spend.py:5:0: C0301: Line too long (109/100) (line-too-long)
backend/services/inventory_spend.py:7:0: C0301: Line too long (154/100) (line-too-long)
backend/services/inventory_spend.py:39:0: C0304: Final newline missing (missing-final-newline)
backend/services/inventory_spend.py:1:0: C0114: Missing module docstring (missing-module-docstring)
************* Module backend.services.revenue_calculator
backend/services/revenue_calculator.py:19:0: C0301: Line too long (110/100) (line-too-long)
backend/services/revenue_calculator.py:64:0: C0303: Trailing whitespace (trailing-whitespace)
backend/services/revenue_calculator.py:69:0: C0301: Line too long (110/100) (line-too-long)
backend/services/revenue_calculator.py:76:0: C0304: Final newline missing (missing-final-newline)
backend/services/revenue_calculator.py:1:0: C0114: Missing module docstring (missing-module-docstring)
backend/services/revenue_calculator.py:13:0: R0914: Too many local variables (19/15) (too-many-locals)

-------------------------------------------------------------------
Your code has been rated at 8.04/10 (previous run: 8.04/10, +0.00)
```

Sonarcube:

SonarQube was chosen because it performs deeper static analysis than traditional linters. It focuses on maintainability, security, code smells, and technical debt, which are aspects not covered by pylint or ESLint alone.



sql_connection.py backend/db/

Revoke and change this password, as it is compromised. (python:S6437) [Ln 7, Col 8]

products_dao.py backend/dao/

Remove this commented out code. (python:S125) [Ln 118, Col 4]

manage-product.html UI/

A form label must be associated with a control. (Web:S6853) [Ln 84, Col 24]

A form label must be associated with a control. (Web:S6853) [Ln 87, Col 24]

A form label must be associated with a control. (Web:S6853) [Ln 90, Col 24]

A form label must be associated with a control. (Web:S6853) [Ln 93, Col 24]

calculations.py backend/routes/

Specify an exception class to catch or reraise the exception (python:S5754) [Ln 89, Col 8]

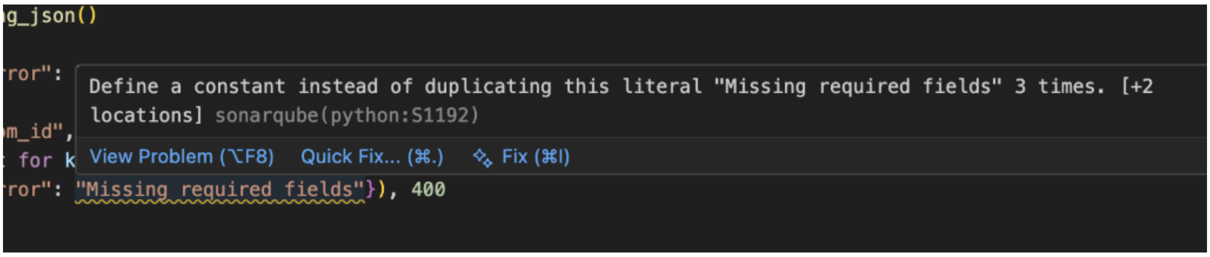
index.html UI/

A form label must be associated with a control. (Web:S6853) [Ln 98, Col 24]

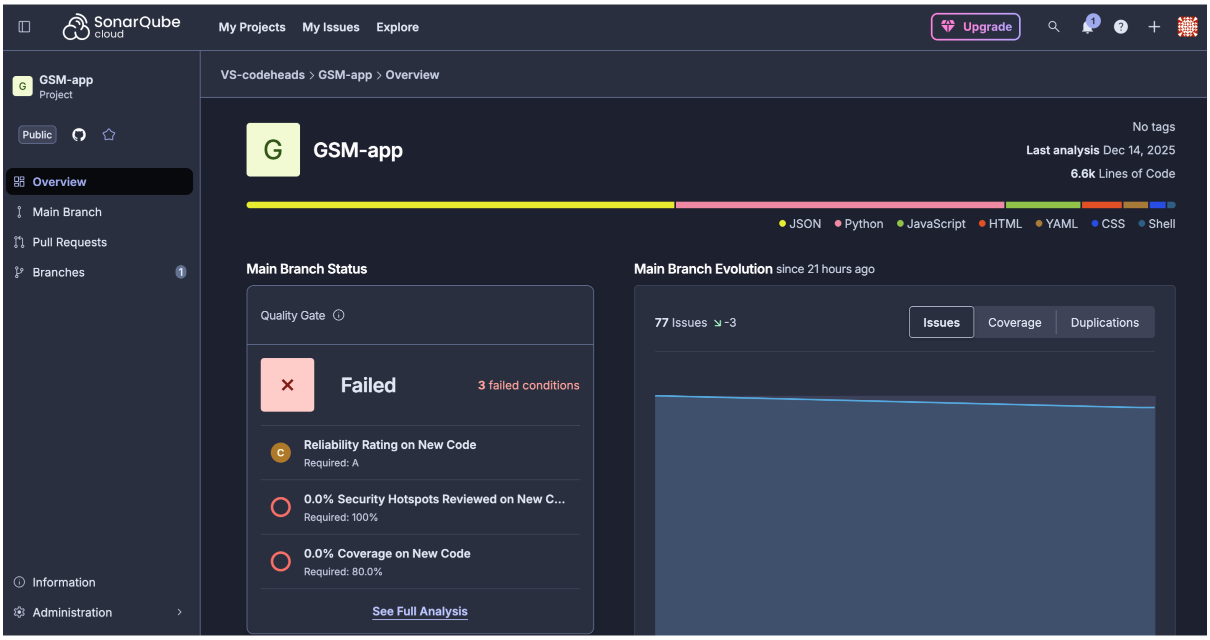
A form label must be associated with a control. (Web:S6853) [Ln 103, Col 24]

order.html UI/

A form label must be associated with a control. (Web:S6853) [Ln 44, Col 16]



```
g_json()

ror":   Define a constant instead of duplicating this literal "Missing required fields" 3 times. [+2
        locations] sonarqube(python:S1192)
m_id",
 for k  View Problem (⌥F8)   Quick Fix... (⌘.)   Fix (⌘I)
ror": "Missing required fields"}), 400
```



VS-codeheads › GSM-app › Overview

G GSM-app

No tags
Last analysis Dec 14, 2025
6.6k Lines of Code

Main Branch Status

Quality Gate

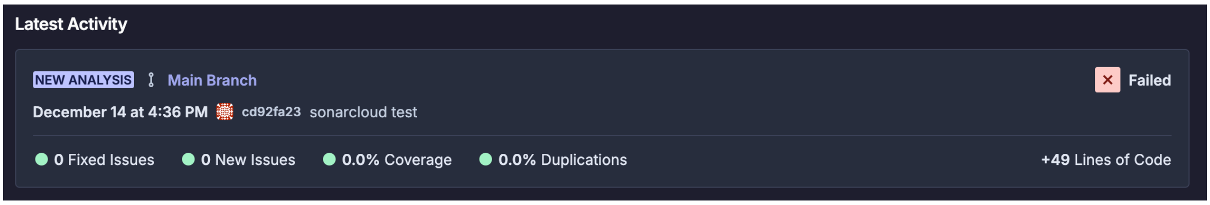
Failed — 3 failed conditions

- Reliability Rating on New Code — Required: A
- 0.0% Security Hotspots Reviewed on New C... — Required: 100%
- 0.0% Coverage on New Code — Required: 80.0%

See Full Analysis

Main Branch Evolution since 21 hours ago

77 Issues ↘ -3



Latest Activity

NEW ANALYSIS Main Branch ✕ Failed

December 14 at 4:36 PM cd92fa23 sonarcloud test

● 0 Fixed Issues ● 0 New Issues ● 0.0% Coverage ● 0.0% Duplications +49 Lines of Code

Here we would find comprehensive feedback of any pushed code in sonarqube cloud.