

Coverage Report

Victor Delattre

January 1, 2024

1 Introduction

This report outlines the coverage analysis of the ‘products.py’ module.

2 Coverage Analysis

The ‘products.py’ module has a total of **37** statements, with **95%** statement coverage from the test suite. This indicates that **35** out of **37** statements were executed.

There are a total of **18** branches within the tested code, with **3** branches only partially covered. This suggests that some decision paths in the code have not been fully tested.

3 Recommendations

To improve the test suite, I recommend:

- Identifying the untested or partially tested branches and adding test cases that cover these paths.
- Ensuring that all functional aspects of the module are covered by unit tests to avoid potential bugs in untested code.

4 Conclusion

The current test suite provides substantial coverage of the ‘products.py’ module, but there is room for improvement, particularly in branch coverage. Enhancing the test suite will increase confidence in the reliability of the code.

Coverage report: 95%

coverage.py v6.5.0, created at 2023-12-15 17:35 +0100

<i>Module</i>	<i>statements</i>	<i>missing</i>	<i>excluded</i>	<i>branches</i>	<i>partial</i>	<i>coverage</i>
products.py	37	0	0	18	3	95%
Total	37	0	0	18	3	95%

coverage.py v6.5.0, created at 2023-12-15 17:35 +0100

Coverage for **products.py**: 95%

37 statements 37 run 0 missing 0 excluded 3 partial

« prev ^ index » next coverage.py v6.5.0, created at 2023-12-15 17:35 +0100

```
1 from login import login
2 import re
3 from checkout_and_payment import checkoutAndPayment
4 import csv
5
6
7 # Display all the products
8 def display_csv_as_table(csv_filename):
9     with open(csv_filename, 'r', newline='') as csvfile:
10         csv_reader = csv.reader(csvfile)
11         header = next(csv_reader)
12         print(header)
13         # Print each row
14         for row in csv_reader:
15             print(row)
16
17
18 # Display products filtered by name
19 def display_filtered_table(csv_filename, search):
20     with open(csv_filename, 'r', newline='') as csvfile:
21         csv_reader = csv.reader(csvfile)
22         header = next(csv_reader)
23         print(header)
24
25         condition_index = header.index("Product")
26         # Print each row
27         for i, row in enumerate(csv_reader):
28             if re.search(row[condition_index], search, re.IGNORECASE):
29                 print(row)
30
31
32 # Search for a product and buy it
33 def searchAndBuyProduct():
34     login_info = None
35     marker = True
36     # Login as a user
37     while marker:
38         login_info = login()
39         if login_info is not None:
40             marker = False
41             break
42     # Search for products then begin to shop
43     while True:
44         search = input("Search for products in inventory (type all for whole inventory):")
45         if search.lower() == "all":
46             display_csv_as_table("products.csv")
47         else:
48             display_filtered_table("products.csv", search)
49         check = input("Ready to shop? (Y/N)")
50         if check.lower() == "y":
51             break
52     checkoutAndPayment(login_info)
```

« prev ^ index » next coverage.py v6.5.0, created at 2023-12-15 17:35 +0100