

Report: Coverage for searchAndBuyProduct()

Using a tool called pytest-cov, we can check how much coverage our software has. After running a coverage test for the whole document to gather data that we need, we get the following:

Coverage report: 87%

coverage.py v7.3.4, created at 2023-12-27 23:58 +0100

Module	statements	missing	excluded	coverage
checkout_and_payment.py	95	5	0	95%
login.py	42	30	0	29%
logout.py	11	1	0	91%
products.py	37	0	0	100%
test_search_and_buy_product.py	93	0	0	100%
Total	278	36	0	87%

coverage.py v7.3.4, created at 2023-12-27 23:58 +0100

Overall the coverage done for the project is 87%.

What we are interested in though is `searchAndBuyProduct()`, which is located in the `products.py` module. The coverage for the module is 100%, which means our testing scripts have a satisfying complete coverage with our testing implementations.

```
28
29 #Search for a product and buy it
30 def searchAndBuyProduct():
31     login_info = None
32     marker = True
33     #Login as a user
34     while marker:
35         login_info = login()
36         if login_info is not None:
37             marker = False
38         break
39     #Search for products then begin to shop
40     while True:
41         search = input("Search for products in inventory (type all for whole inventory):")
42         if search.lower() == "all":
43             display_csv_as_table("products.csv")
44         else:
45             display_filtered_table("products.csv", search)
46         check = input("Ready to shop? (Y/N)")
47         if check.lower() == "y":
48             break
49     checkoutAndPayment(login_info)
```

All branches and lines in the function have been covered in one way or another with different testings, resulting in a total of 17 statements (or lines) being executed from the function.

There are 2 while loops and 3 if conditions, one of the latter being an if-else statement, making 5 different branching points in the code.