**Testing for savedItems.py**

>>> import savedItems as s

Connecting...

Connected

#Done in python terminal by Daniel Austin, July 27

All tests were done in python terminal, and any time where I’ve displayed the database was fetched using the displayData() function which lists all the rows in the database, and corroborated using MySQL Workbench.

Tests were performed sequentially. Unless specifically specified each test was performed directly after the one before it, and the database state carries over from test to test.

**---**

**Test case:** Add an item

**Command:** s.addNewItem("ItemName", "amazon.ca", "image.com", "800")

**Result:** Item is now in database as:

('ItemName', 'amazon.ca', 'image.com', Decimal('800.00'))

**Comment:**

**Passed: Y**

**---**

**Test case:** Add a duplicate URL

**Command:** s.addNewItem("DifferentItemName", "amazon.ca", "diffimg.com", "900")

**Result:** The item is **not** added to the database. The database still looks like:

('ItemName', 'amazon.ca', 'image.com', Decimal('800.00'))

**Comment:** Notice that only the URL must be the same for the add to be denied. If the name is the same the save is also denied, as can be seen in the next test

**Passed: Y**

**---**

**Test case:** Add a duplicate name

**Command:** s.addNewItem("ItemName", "bestbuy.ca", "diffimg.com", "900")

**Result:** The item is **not** added to the database. The database still looks like:

('ItemName', 'amazon.ca', 'image.com', Decimal('800.00'))

**Comment:**

**Passed: Y**

**---**

**Test case:** Adding an item with different name and URL, but the same imageURL and price

**Command:** s.addNewItem("ItemName2", "website.ca", "image.com", "800")

**Result:** The item is added to the database. The database now is:

('ItemName', 'amazon.ca', 'image.com', Decimal('800.00'))

('ItemName2', 'website.ca', 'image.com', Decimal('800.00'))

**Comment:** Notice that if name and URL are different other values can be the same and still be added.

**Passed: Y**

**---**

**Test case: Delete an item which exists**

**Command:** s.removeItem("ItemName")

**Result:** The item has been removed. The database now looks like:

('ItemName2', 'website.ca', 'image.com', Decimal('900.00'))

**Comment:**

**Passed: Y**

**---**

**Test case: Delete an item which does not exist**

**Command:** s.removeItem("NotReal")

**Result:** The item has not been removed because it was not in the database. The database is unaltered and looks like:

('ItemName2', 'website.ca', 'image.com', Decimal('900.00'))

**Comment:**

**Passed: Y**

**---**

**Test Case:** Clean the cart with many objects inside

**Comment:** Before this test I deleted all values from the database and added the following 3 items to the database. It looked like this just before the test was performed

('A', 'A', 'A', Decimal('100.00'))

('B', 'B', 'B', Decimal('200.00'))

('C', 'C', 'C', Decimal('300.00'))

**Command:** s.cleanCart()

**Result:** After running the database is entirely empty.

**Passed: Y**

**---**

**Test Case:** Clean the cart with no objects inside

**Comment:** This test was performed when there are no items in the database. There is no change in database values

**Command:** s.cleanCart()

**Result:** There is no change in database values, it remains empty

**Passed: Y**