Unit Test Documentation

Project Name: Online Shopping Cart  
Version: 1.0  
Testing Phase: Phase 1  
Test Date: December 2, 2024

Function Title: display\_csv\_as\_table  
Test Designed by: [TingtingLyu]  
Test Designed Date: November 26, 2024

Function1: test\_display\_csv\_as\_table

Objective:

The display\_csv\_as\_table function is designed to display the contents of a CSV file, row by row, starting with the header. The function checks whether the input parameter csv\_file\_name is a string, and then reads the specified CSV file and outputs its content.

Dependencies and Preconditions:

Dependencies: The function depends on the get\_csv\_data function.

Preconditions: None

**Test Case Explanations:**

1. **Invalid Input Tests**: These tests check how the function handles various invalid inputs (integer, float, list, None) and ensure that the appropriate exceptions are raised.
2. **File Not Found Test**: This test ensures that if the file does not exist, the function raises a FileNotFoundError.
3. **Empty File Test**: This test checks how the function handles an empty CSV file, ensuring it raises a StopIteration exception.
4. **Valid Input Test**: This test ensures that when a valid CSV file is provided, the function correctly prints the file's content, including the header and data rows.

Function2: display\_filtered\_table(csv\_file\_name)

Objective:

The display\_filtered\_table function is designed to filter and display the contents of a CSV file based on the provided search condition. If no search condition is provided, it displays all the data. If a search condition is given, it filters and shows only matching records. The function first checks whether the csv\_file\_name is a string, and based on the search target, either calls display\_csv\_as\_table or performs data filtering.

Dependencies and Preconditions:

Dependencies: The function depends on the get\_csv\_data function and display\_csv\_as\_table function.

Preconditions: None.

**Test Case Explanations:**

1. **Valid Input - No Search**: When no search condition is provided, the function should display all the product data.
2. **Valid Input - Search Condition**: When a search condition is provided, the function should filter and display the products that match the given search target.
3. **Valid Input - Empty Search**: When an empty search condition is provided, the function should only display the header.
4. **Valid Input - Single Product**: When the CSV contains a single product, the function should display the product data correctly.
5. **Valid Input - Same Product Name**: When multiple products share the same name, the function should display all matching products.
6. **Valid Input - Special Characters**: When product names contain special characters, the function should handle them correctly and display the data.
7. **Invalid Input - No Match**: If no products match the search condition, the function should only display the header, with no product data.
8. **Invalid Input - Case Insensitive**: The search should be case-insensitive, and the function should display products matching the search condition regardless of case.
9. **Invalid Input - Invalid File Name**: When the csv\_file\_name parameter is invalid (such as a float, list, integer, or None), the function should raise a TypeError with the appropriate error message.

Function3: test\_search\_and\_purchase\_product

Objective:

The search\_and\_purchase\_product function allows users to search for products and make a purchase. The user first logs in, then can either choose to view all products or filter products based on specific conditions. Once the user confirms the purchase, the checkout process is initiated.

Dependencies and Preconditions:

Dependencies: The function depends on login, display\_csv\_as\_table, display\_filtered\_table, and checkout\_and\_payment functions.

Preconditions: None.

**Test Case Explanations:**

1. **Valid Input - Query All Products**: When the user chooses to query all products, the system should call display\_csv\_as\_table to display all product data.
2. **Valid Input - Exact Product Search**: When the user inputs an exact product name (e.g., "Phone"), the system should call display\_filtered\_table to display matching products.
3. **Valid Input - Query by Stock Range**: When the user chooses to search by stock range, the system should call display\_filtered\_table to show products that fall within the stock range.
4. **Valid Input - Query by Price Range**: When the user chooses to search by price range, the system should call display\_filtered\_table to show products within the specified price range.
5. **Valid Input - Fuzzy Product Search**: When the user enters part of a product name for a fuzzy search, the system should call display\_filtered\_table to show matching products.
6. **Valid Input - Initial Letter Search**: When the user searches by the initial letter of a product's name, the system should call display\_filtered\_table to show products that start with that letter.
7. **Invalid Input - Invalid Search Query**: When the user inputs an invalid search condition, the system should only display the header with no product data.
8. **Invalid Input - Invalid Login Info**: When the user provides invalid login information, the system should raise an exception and stop the operation.
9. **Invalid Input - Invalid Stock Range**: When the user inputs an invalid stock range (e.g., a negative number), the system should handle the invalid input and not call the filtering function.
10. **Invalid Input - Invalid Price Range**: When the user inputs an invalid price range (e.g., a negative number), the system should handle the invalid input and not call the filtering function.