Unit testing of the task 3.2

Test display\_csv\_as\_table Function

**Invalid Input Tests( This all will lead to error ):**

**File name as integer**  
Input: 123  
Expected: Raise TypeError with message "File name must be a string".

**File name as float**  
Input: 45.67  
Expected: Raise TypeError with message "File name must be a string".

**File name as list**  
Input: ["file1.csv", "file2.csv"]  
Expected: Raise TypeError with message "File name must be a string".

**File name as** None  
Input: None  
Expected: Raise TypeError with message "File name must be a string".

**File Not Found Tests:**

**File does not exist**  
Input: "non\_existent\_file.csv"  
Expected: Raise FileNotFoundError.

**File Content Tests:**

**Empty file**  
Input: Empty file  
Expected: Raise StopIteration.

**Invalid file format**  
Input: "invalid\_file.csv" (assumed to be missing)  
Expected: Raise OSError (or FileNotFoundError).

**Valid Input Tests:**

**File with only header**  
Input: "Product,Price,Quantity\n"  
Expected: Output header: ['Product', 'Price', 'Quantity'].

**Single data row**  
Input: "Apple,1.2,10"  
Expected: Output header and data row:

['Product', 'Price', 'Quantity']

['Apple', '1.2', '10']

**Multiple data rows**  
Input: "Apple,1.2,10\nBanana,0.5,20\nOrange,0.8,15"  
Expected: Output header and all data rows.

**Mixed data types**  
Input: "Apple,1.2,10\nOrange,0.8,15"  
Expected: Output header and data rows with mixed data types.

**Product name with special characters**  
Input: "Apple,1.2,10\nBanana-Special,0.5,20"  
Expected: Output header and data with special characters in product name.

**Product name with spaces**  
Input: "Apple Juice,1.2,10"  
Expected: Output header and product name with spaces.

**Product price as integer**  
Input: "Apple,10,20"  
Expected: Output header and product with integer price.

**Product price as float**  
Input: "Apple,1.50,10"  
Expected: Output header and product with float price.

**Large values in data**  
Input: "BigProduct,10000,5000"  
Expected: Output header and product with large numbers.

**Empty product name**  
Input: ",10,5"  
Expected: Output header and data with empty product name.

**Test display\_filtered\_table Function**

Invalid Input Tests (All should raise errors):

File name as integer  
Input: 123  
Expected: Raise TypeError with message "File name must be a string."

File name as float  
Input: 45.67  
Expected: Raise TypeError with message "File name must be a string."

File name as list  
Input: ["file1.csv", "file2.csv"]  
Expected: Raise TypeError with message "File name must be a string."

File name as None  
Input: None  
Expected: Raise TypeError with message "File name must be a string."

File Not Found Tests:

File does not exist  
Input: "non\_existent\_file.csv"  
Expected: Raise FileNotFoundError.

File Content Tests:

Empty file  
Input: Empty file  
Expected: Raise StopIteration.

Invalid file format  
Input: "invalid\_file.csv"  
Expected: Raise OSError (or FileNotFoundError).

Valid Input Tests:

File with only header  
Input: "Product,Price,Quantity\n"  
Expected: Output header: ['Product', 'Price', 'Quantity'].

Single data row  
Input: "Apple,1.2,10"  
Expected:

['Product', 'Price', 'Quantity']

['Apple', '1.2', '10']

Multiple data rows  
Input: "Apple,1.2,10\nBanana,0.5,20\nOrange,0.8,15"  
Expected:

css

复制代码

['Product', 'Price', 'Quantity']

['Apple', '1.2', '10']

['Banana', '0.5', '20']

['Orange', '0.8', '15']

Mixed data types  
Input: "Apple,1.2,10\nOrange,0.8,15"  
Expected: Output header and rows with mixed data types.

Product name with special characters  
Input: "Apple,1.2,10\nBanana-Special,0.5,20"  
Expected: Output header and data with special characters in product name.

Product name with spaces  
Input: "Apple Juice,1.2,10"  
Expected: Output header and product name with spaces.

Product price as integer  
Input: "Apple,10,20"  
Expected: Output header and product with integer price.

Product price as float  
Input: "Apple,1.50,10"  
Expected: Output header and product with float price.

Large values in data  
Input: "BigProduct,10000,5000"  
Expected: Output header and product with large numbers.

Empty product name  
Input: ",10,5"  
Expected: Output header and data with empty product name.

**Test search\_and\_purchase\_product() Function**

**Test Dependencies:**

Mocked Functions/Objects:

UserInterface.get\_user\_input: Mocks the user input prompts.

product\_data.get\_csv\_data: Mocks the loading of CSV data.

product\_search.display\_csv\_as\_table: Mocks the display of product data as a table.

product\_data.get\_products: Mocks the retrieval of product data.

user\_login.login: Mocks the login process.

user\_data.UserDataManager.load\_users: Mocks the user data management system.

checkout\_process.checkout\_and\_payment: Mocks the checkout process.