**Test Function Document**

|  |  |
| --- | --- |
| Project Name: Online Shopping Cart | |
| Automation Title: test\_  display\_csv\_as\_table(csv\_file\_name), display\_filtered\_table(csv\_file\_name) and search\_and\_purchase\_product(). | Version: 1.0 |
| Testing Phase: Phase 1 | Date of Test: 27 nov 2024 |
| Module Name: Shopping | |

| **S. No1** | **Equivalence Class** | **Test Case Data** | **Expected Results** | **Actual Results** | **Status (Pass/Fail)** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Valid Input** |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Valid Input - Large Numbers | csv\_file\_name="large\_numbers.csv" | Displays products with large numbers correctly | Output as expected | Pass | Tests handling large product numbers |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 2 | Valid Input - Empty Product Name | csv\_file\_name="empty\_product\_name.csv" | Displays empty product names correctly | Output as expected | Pass | Tests handling products with empty names |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | Valid Input - Single Row | csv\_file\_name="single\_row.csv" | Displays product data when only one row is present | Output as expected | Pass | Tests single row CSV handling |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 4 | Valid Input - Header Only | csv\_file\_name="header\_only.csv" | Displays header row when no product data is present | Output as expected | Pass | Tests header-only CSV handling |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 5 | Valid Input - Special Characters | csv\_file\_name="special\_characters.csv" | Displays product data with special characters correctly | Output as expected | Pass | Tests handling special characters in product names |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 6 | Valid Input - Price as Integer | csv\_file\_name="product\_price\_integer.csv" | Displays products with integer price correctly | Output as expected | Pass | Tests handling integer prices |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 7 | Valid Input - Price as Float | csv\_file\_name="product\_price\_float.csv" | Displays products with float price correctly | Output as expected | Pass | Tests handling float prices |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Invalid Input** |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 8 | Invalid Input - Invalid File Name | csv\_file\_name=12.34, search\_target=None | Raises TypeError with message "File name must be a string" | Exception raised as expected | Pass | Tests invalid file name (float type) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9 | Invalid Input - Invalid File Name (List) | csv\_file\_name=['invalid', 'list'], search\_target=None | Raises TypeError with message "File name must be a string" | Exception raised as expected | Pass | Tests invalid file name (list type) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 10 | Invalid Input - Invalid File Name (Integer) | csv\_file\_name=123, search\_target=None | Raises TypeError with message "File name must be a string" | Exception raised as expected | Pass | Tests invalid file name (integer type) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 11 | Invalid Input - Invalid File Name (None) | csv\_file\_name=None, search\_target=None | Raises TypeError with message "File name must be a string" | Exception raised as expected | Pass | Tests invalid file name (None type) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | Invalid Input - Non-Existent File | csv\_file\_name="non\_existent\_file.csv" | Raises FileNotFoundError as the file does not exist | Exception raised as expected | Pass | Tests non-existent file handling |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 13 | Invalid Input - Empty File | csv\_file\_name="empty\_file.csv" | Raises StopIteration or returns empty data | Exception raised as expected | Pass | Tests handling of empty file content |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 14 | Invalid Input - Invalid CSV Format | csv\_file\_name="invalid\_file.csv" | Raises OSError or FileNotFoundError for invalid CSV format | Exception raised as expected | Pass | Tests invalid CSV file handling |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 15 | Invalid Input - Invalid Content | csv\_file\_name="invalid\_content.csv" | Raises OSError or handles malformed CSV correctly | Exception raised as expected | Pass | Tests handling of malformed CSV data |

| **S. No2** | **Equivalence Class** | **Test Case Data** | **Expected Results** | **Actual Results** | **Status (Pass/Fail)** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Valid Input** |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Valid Input - No Search Target | csv\_file\_name="products.csv", search\_target=None | Displays all product data, including header and rows | Output as expected | Pass | Tests displaying all products when no search is provided |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 2 | Valid Input - Exact Product Search | csv\_file\_name="products.csv", search\_target="Banana" | Displays only the row for "Banana" | Output as expected | Pass | Tests displaying the exact product when searched |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | Valid Input - Empty Search | csv\_file\_name="products.csv", search\_target="" | Displays only the header row | Output as expected | Pass | Tests displaying only header when search term is empty |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 4 | Valid Input - Single Product | csv\_file\_name="products.csv", search\_target=None | Displays the product row for "Apple" | Output as expected | Pass | Tests displaying single row when there's only one product |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 5 | Valid Input - Same Product Name | csv\_file\_name="products.csv", search\_target="Apple" | Displays all rows for "Apple" | Output as expected | Pass | Tests displaying multiple products with same name |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 6 | Valid Input - Special Characters | csv\_file\_name="products.csv", search\_target=None | Displays products with special characters in name | Output as expected | Pass | Tests handling special characters in product names |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 7 | Valid Input - Case-Insensitive Search | csv\_file\_name="products.csv", search\_target="apple" | Displays "Apple" product case-insensitively | Output as expected | Pass | Tests case-insensitive search handling |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Invalid Input** |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 8 | Invalid Input - Invalid File Name (Float) | csv\_file\_name=12.34, search\_target=None | Raises TypeError with message "File name must be a string" | Exception raised as expected | Pass | Tests invalid file name (float type) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9 | Invalid Input - Invalid File Name (List) | csv\_file\_name=['invalid', 'list'], search\_target=None | Raises TypeError with message "File name must be a string" | Exception raised as expected | Pass | Tests invalid file name (list type) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 10 | Invalid Input - Invalid File Name (Integer) | csv\_file\_name=123, search\_target=None | Raises TypeError with message "File name must be a string" | Exception raised as expected | Pass | Tests invalid file name (integer type) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 11 | Invalid Input - Invalid File Name (None) | csv\_file\_name=None, search\_target=None | Raises TypeError with message "File name must be a string" | Exception raised as expected | Pass | Tests invalid file name (None type) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | Invalid Input - No Matching Products | csv\_file\_name="products.csv", search\_target="Pineapple" | Displays only the header, no matching product data | Displays only header | Pass | Tests when no products match the search term |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 13 | Invalid Input - Invalid Search Condition | csv\_file\_name="products.csv", search\_target="invalid" | Displays only the header, no matching products | Displays only header | Pass | Tests invalid search condition (non-matching) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 14 | Invalid Input - Negative Stock Range | csv\_file\_name="products.csv", search\_target="stock", stock\_min=-5, stock\_max=10 | Does not call display\_filtered\_table, handles invalid stock range | Does not call display\_filtered\_table | Pass | Tests handling of invalid stock range input (negative value) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 15 | Invalid Input - Invalid Price Range | csv\_file\_name="products.csv", search\_target="price", min\_price=-100, max\_price=500 | Does not call display\_filtered\_table, handles invalid price range | Does not call display\_filtered\_table | Pass | Tests handling of invalid price range input (negative value) |

| **S. No3** | **Equivalence Class** | **Test Case Data** | **Expected Results** | **Actual Results** | **Status (Pass/Fail)** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Valid Input - Query All Products | 'all' | Call display\_csv\_as\_table to show all products data | Calls display\_csv\_as\_table | Pass | Tests querying all products |
| 2 | Valid Input - Exact Product Search | 'Phone' | Call display\_filtered\_table to show "Phone" products | Calls display\_filtered\_table | Pass | Tests exact product search |
| 3 | Valid Input - Query by Stock Range | 'stock', '5', '10' | Call display\_filtered\_table to show products in stock range | Calls display\_filtered\_table | Pass | Tests querying by stock range |
| 4 | Valid Input - Query by Price Range | 'price', '100', '800' | Call display\_filtered\_table to show products in price range | Calls display\_filtered\_table | Pass | Tests querying by price range |
| 5 | Valid Input - Fuzzy Product Search | 'Laptop' | Call display\_filtered\_table to show "Laptop" matching products | Calls display\_filtered\_table | Pass | Tests fuzzy product search |
| 6 | Valid Input - Initial Letter Search | 'L' | Call display\_filtered\_table to show products starting with "L" | Calls display\_filtered\_table | Pass | Tests search by product name initial |
| 7 | Invalid Input - Invalid Search Query | 'invalid' | Only display header, no products | Displays only header | Pass | Tests invalid search condition |
| 8 | Invalid Input - Invalid Login Info | None | Raises exception and stops operation | Raises exception | Pass | Tests invalid login information |
| 9 | Invalid Input - Invalid Stock Range | 'stock', '-5', '10' | Does not call display\_filtered\_table, handles invalid input | Does not call display\_filtered\_table | Pass | Tests invalid stock range |
| 10 | Invalid Input - Invalid Price Range | 'price', '-100', '500' | Does not call display\_filtered\_table, handles invalid input | Does not call display\_filtered\_table | Pass | Tests invalid price range |