School of Computing (SoC)

ST0524

Fundamental of Programming 2

ASSIGNMENT 1

Instructions and Guidelines:

1. CA1 an individual assignment that accounts for 30% of the module.
2. Submit softcopies of CA1 in Brightspace by the given deadlines given below. You are required to:
3. Prepare the Declaration of Academic Integrity (SOC) for Assignment 1. Your Assignment 1 will NOT be marked if these forms are not submitted.

The form is available in Brightspace under Continual Assessments Specification > CA1 > Declaration of Academic Integrity (SOC) form.

1. Rename your JavaScript file using this naming convention: yourAdminNo\_name.js

For example: 123456\_JaneTan.js

1. Zip all files using FILE EXPLORER, rename the zip file using this naming convention: yourAdminNo\_name.zip

For example: 123456\_JaneTan.zip

1. In your submission, include a short header indicating your Name (as in SAS2), Class and Admission number in your program.

|  |
| --- |
| Sample header:  // Name : Jane Tan  // Class: DIT/1B/01  // Admin No: 2301010  //  :  : |

CA1Submission Deadline in BrightSpace

Date: 13th January 2025 (Mon)

Time: 8am (Week 14)

1. You must use JavaScript to develop the application in Visual Studio Code.
2. You may consult your Lecturer for overall design and structure of your assignment. However, you are NOT allowed to ask him/her to debug your program. Students who seek debugging help from Lecturer will only be awarded a maximum grade of C+. It can be lower depending on the quality of work submitted.
3. Please read the SP Policy on use of AI tools for academic submission in your CA1 folder in Brightspace.
4. You are required to do demo/interview of your programs to your Lecturer individually. You may be asked to explain the program logic or make amendments to your codes during the sessions. Please check with your Lecturer for the further details.
5. If you are absent from the demo/interview without a valid approved Leave of Absence, you will be awarded ZERO mark even if you submit your assignments. If you are not well on your day of interview and have a valid approved Leave of Absence (LOA), please INFORM your Lecturer in as soon as possible, apply for LOA in Student portal and submit a copy of the approved Leave of Absence by email to your lecturer. Your Lecturer will arrange for a make-up interview session.
6. 50% of marks to be deducted for submission of assignment within ONE calendar day after the deadline. No marks shall be awarded for assignments submitted more than one day after the deadline.

For example: If the submission deadline is 8 Jan, 8am, any submission after that till 9 Jan, 8am will have 50% marks deducted. Any submission from 9 Jan, 8am onwards will N OT be accepted i.e. given zero.

1. No marks will be awarded, if the work is copied, done by someone or you have allowed others to copy your work. This is a very serious offence of plagiarism committed by all involved (i.e. all the givers & recipients). Please refer the clause in RED below regarding plagiarism.

Warning: Plagiarism means passing off as one's owned the ideas, works, writings, etc., which belong to another person. In accordance with this definition, you are committing plagiarism if you copy the work of another person and turning it in as your own, even if you would have the permission of that person. Plagiarism is a serious offence and disciplinary action will be taken against you. If you are guilty of plagiarism, you may fail all modules in the semester, or even be liable for expulsion.

If you share your assignments with your classmate(s), it is also an academic offence and disciplinary action will be taken against you, although you have done the work yourself.

So, both giver and receiver will be subject to disciplinary hearings/action.

Refer to SP Student Handbook on plagiarism, rules and regulation on assessment for more details.

Assignment 1

Requirements

1. Objective

The objective of this assignment is to apply the various advance programming methodology such as higher order function, data structures, asynchronous and callback function.

1. Overview

You are required to use JavaScript to extract data using the endpoint documents. A command prompt user interface application is required to allow user to interact with the dataset.

1. Requirements

The following is the software design and interface.

Back End Interface

Data set in csv files

Front End Command Prompt User Interface

You are not allowed to modify the back-end code and dataset to meet the front-end command prompt user interface requirements. All data manipulation and preparation are to be done at the front-end before the final output are presented to the user.

Documentation for the dataset endpoint is provided with endpoint document. Instruction on how to consume end point is uploaded in Lesson 5 – Async Functions, Consume Endpoint.

1. Assessment Criteria

Marks are allocated based on the following criteria. You may be asked to explain codes during the demo session.

* Proper declaration of datatype, arrays, scope, and meaningful naming of variable.
* Program produces correct output according to the requirements.
* Appropriate implementations of programming techniques, such as Callback, Async/Await, Promises and data structure.
* Data validation is correct and complete from calls to the back-end API.
* Program executes properly without crashing. The program will indicate the cause of error(s). The application will terminate as per user selection in the user manual (Exit).
* You may add in additional requirements if it help users interpret the data better.

1. Marks Distribution

|  |  |  |
| --- | --- | --- |
| No | Description | Marks |
| 1 | Extract data from backend server | 10 |
| 2 | Proper input interface for each function requirements | 10 |
| 3 | Display data as per stated requirements | 25 |
| 4 | Proper data preparation and calculation as per stated requirements | 25 |
| 5 | Data validation from use of back-end API | 5 |
| 6 | Error handling and proper termination of application | 5 |
| 7 | Use of advanced programming techniques, such as Callback, Async/Await, Promises and data structure | 20 |

Data Set 1

Shopping Trend Data.

Requirement:

1. Display the first 10 records of the Shopping Trend Data.
2. Display the lowest, highest, average and median purchase amount (USD) by Male and Female.
3. Display the category in the Shopping Trend Data.
4. User can query and view the Shopping Trend Data by category.
5. Display the lowest and highest total amount of sales (by Purchase Amount (USD)) by season.

* Query by season, sum all the Purchase Amount (USD)
* Push into an array of object.
* Sort the lowest and highest of total amount of sales (by Purchase Amount (USD)) by season.

6. Display frequency of visit for customer that use promo code and for customer that do not use promo code.

* Filter customer use promo code (Promo Code Used)
* Categorize the frequency of visit (Weekly, Fortnightly, Annually)

Note:

You must use the **most efficient way** to retrieve the required data using the given endpoint.

Sample output:

User Menu for user to select the items to view the Shopping Trend Data.

|  |
| --- |
|  |

**Requirement 1**

Output display format:

**Shopping Trend Data**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Customer ID | Age | Item Purchased | Category | Purchase Amount (USD) | Season |
| **<customerID>** | **<age>** | **<itemPurchased>** | **<catergory>** | **<PurchaseAmountUSD>** | **<season>** |
| **<customerID>** | **<age>** | **<itemPurchased>** | **<catergory>** | **<PurchaseAmountUSD>** | **<season>** |
| **<customerID>** | **<age>** | **<itemPurchased>** | **<catergory>** | **<PurchaseAmountUSD>** | **<season>** |
|  |  |  |  |  |  |

**Requirement 2**

Output display format:

Lowest, highest, average and median purchase amount (USD) by Male and Female.

|  |  |  |
| --- | --- | --- |
|  | Male | Female |
| Lowest | USD **<Lowest Male>** | USD **<Lowest Female>** |
| Highest | USD **<Highest Male>** | USD **<Highest Female>** |
| Average | USD **<Average Male>** | USD **<Average Female>** |
| Median | USD **<Median Male>** | USD **<Median Female>** |

Expected Output:

Lowest, highest, average and median purchase amount (USD) by Male and Female.

|  |  |  |
| --- | --- | --- |
|  | Male | Female |
| Lowest | USD 20 | USD 20 |
| Highest | USD 100 | USD 100 |
| Average | USD 59.54 | USD 60.25 |
| Median | USD 60.00 | USD 60.00 |

**Requirement 3**

Output display format:

Shopping Trend Data by Category

**<category>**

**<category>**

**<category>**

**<category>**

Expected Output:

Shopping Trend Data by Category

Clothing

Outerwear

Accessories

Footwear

Requirement 4

Prompt user to select the category to select the category.

Select the category to display:

1. Clothing

2. Outerwear

3. Accessories

4. Footwear

Output display format:

Display Category – Clothing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Customer ID | Age | Item Purchased | Category | Season | Payment Method |
| **<customerID>** | **<age>** | **<itemPurchased>** | **<catergory>** | **<season>** | **<paymentMethod>** |
| **<customerID>** | **<age>** | **<itemPurchased>** | **<catergory>** | **<season>** | **<paymentMethod>** |
| **<customerID>** | **<age>** | **<itemPurchased>** | **<catergory>** | **<season>** | **<paymentMethod>** |

Requirement 5

Output display format:

Display the sum of purchase amount (USD) group by season and sorted by sum of purchase amount.

|  |  |
| --- | --- |
| Season | Sum of Purchase amount (USD) |
| Summer | **<Sum of Purchase for Summer>** |
| Winter | **<Sum of Purchase for Winter>** |
| Spring | **<Sum of Purchase for Spring>** |
| Fall | **<Sum of Purchase for Fall>** |
| Grand Total | **<Sum of All Purchase>** |

Expected Output:

Display the sum of purchase amount (USD) group by season and sorted by sum of purchase amount.

|  |  |
| --- | --- |
| Season | Sum of Purchase amount (USD) |
| Summer | 55777 |
| Winter | 58607 |
| Spring | 58679 |
| Fall | 60018 |
| Grand Total | 233081 |

**Requirement 6**

Output display format:

Frequency of visit for customer that use promo code and for customer that do not use promo code.

|  |  |  |
| --- | --- | --- |
|  | Promo Code Used | |
| Frequency of Purchases | No | Yes |
| Bi-Weekly | **<Count Bi-Weekly No>** | **<Count Bi-Weekly Yes>** |
| Weekly | **<Count Weekly No>** | **<Count Weekly Yes>** |
| Monthly | **<Count Monthly No>** | **<Count Monthly Yes>** |
| Annually | **<Count Annually No>** | **<Count Annually Yes>** |
| Quarterly | **<Count Quarterly No>** | **<Count Quarterly Yes>** |
| Fortnightly | **<Count Fortnightly No>** | **<Count Fortnightly Yes>** |
| Every 3 Months | **<Count Every 3 Months No>** | **<Count Every 3 Months Yes>** |

Expected Output:

Frequency of visit for customer that use promo code and for customer that do not use promo code.

|  |  |  |
| --- | --- | --- |
|  | Promo Code Used | |
| Frequency of Purchases | No | Yes |
| Bi-Weekly | 321 | 226 |
| Weekly | 305 | 234 |
| Monthly | 316 | 237 |
| Annually | 334 | 238 |
| Quarterly | 322 | 241 |
| Fortnightly | 299 | 243 |
| Every 3 Months | 326 | 258 |

Data Set 2

US Superstore data

Requirement:

1. Display the first 10 records of the US Superstore data.
2. Display the lowest, highest, average and median sales by market segment (Consumer, Corporate, Home Office).
3. Display the category in the US Superstore.
4. User can query and view the US Superstore Data by category.
5. Display the lowest and highest total amount of sales (Sales) by category.
   * Query by category, sum all the sales.
   * Push into an array of object.
   * Sort the lowest and highest of total amount of sales (Sales) by category.

6. Display the most profit and least profit by sales. Display the customer name, Category, Sub-Category, Product Name, Sales, Profit that has highest profit and lowest profit and product they purchase

Note:

You must use the **most efficient way** to retrieve the required data using the given endpoint.

Sample output:

User Menu for user to select the items to view for the Shopping Trend Data.

|  |
| --- |
|  |

**Requirement 1**

Output display format:

US Superstore Data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Customer ID | Customer Name | Segment | State | Product Name | Sales (USD) |
| **<customerID>** | **<customerName>** | **<segment>** | **<state>** | **<productName>** | **<sales>** |
| **<customerID>** | **<customerName>** | **<segment>** | **<state>** | **<productName>** | **<sales>** |
| **<customerID>** | **<customerName>** | **<segment>** | **<state>** | **<productName>** | **<sales>** |

**Requirement 2**

Output display format:

Lowest, highest, average and median sales by market segment.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Consumer | Corporate | Home Office |
| Lowest | $ **<Lowest Consumer>** | $ **<Lowest Corporate>** | $ **<Lowest Home Office>** |
| Highest | $ **<Highest Consumer>** | $ **<Highest Corporate>** | $ **<Highest Home Office>** |
| Average | $ **<Average Consumer>** | $ **<Average Corporate>** | $ **<Average Home Office>** |
| Median | $ **<Median Consumer>** | $ **<Median Corporate>** | $ **<Median Home Office>** |

Expected Output:

Lowest, highest, average and median sales by market segment.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Consumer | Corporate | Home Office |
| Lowest | $ 0.444 | $ 0.556 | $ 0.99 |
| Highest | $ 13999.96 | $ 17499.95 | $ 22638.48 |
| Average | $ 223.73 | $ 233.82 | $ 240.97 |
| Median | $ 53.72 | $ 56.54 | $ 52.44 |

**Requirement 3**

Output display format:

US Superstore Data by Category

**<category>**

**<category>**

**<category>**

Expected Output:

US Superstore Data by Category

Office Supplies

Furniture

Technology

**Requirement 4**

Prompt user to select the category to select the category.

Select the category to display:

1. Office Supplies

2. Furniture

3. Technology

Output display format:

Display Category – Furniture

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Order ID | Order Date | Product ID | Product Name | Category | Sales (USD) |
| **<orderID>** | **<orderDate>** | **<productID>** | **<productName>** | **<category>** | **<sales>** |
| **<orderID>** | **<orderDate>** | **<productID>** | **<productName>** | **<category>** | **<sales>** |
| **<orderID>** | **<orderDate>** | **<productID>** | **<productName>** | **<category>** | **<sales>** |

Requirement 5

Output display format:

Display the sum of sales group by category and sorted by sum of sales.

|  |  |
| --- | --- |
| Category | Sales |
| Office Supplies | $ **<Sum of Office Supplies>** |
| Furniture | $ **<Sum of Furniture>** |
| Technology | $ **<Sum of Technology>** |
| Grand Total | $ **<Sum of Sales>** |

Expected Output:

Display the sum of sales group by category and sorted by sum of sales.

|  |  |
| --- | --- |
| Category | Sales |
| Office Supplies | $ 719047.03 |
| Furniture | $ 741999.80 |
| Technology | $ 836154.03 |
| Grand Total | $ 2297200.86 |

**Requirement 6**

Output display format:

Lowest Sales Profit

Customer Name: **<customerName>**

Category: **<category>**

Sub-Category: **<subCategory>**

Product Name: **<productName>**

Sales: $ **<sales>**

Profit: -$ **<profit>**

Highest Sales Profit

Customer Name: **<customerName>**

Category: **<category>**

Sub-Category: **<subCategory>**

Product Name: **<productName>**

Sales: $ **<sales>**

Profit: -$ **<profit>**

Expected Output:

Lowest Sales Profit

Customer Name: Cindy Stewart

Category: Technology

Sub-Category: Machines

Product Name: Cubify CubeX 3D Printer Double Head Print

Sales: $ 4499.985

Profit: -$ 6599.978

Highest Sales Profit

Customer Name: Tamara Chand

Category: Technology

Sub-Category: Copiers

Product Name: Canon imageCLASS 2200 Advanced Copier

Sales: $ 17499.95

Profit: $ 8399.976