Assignment - Google Sheets & JavaScript Practice

Report by Riina Kikkas

[Link to Google Sheets]

Task 1: The function readProducts() opens a Google Sheet by its URL, selects the first sheet, reads all the data into a 2D array, and prints that data to the execution log.

```
function readProducts() {
  let sheet = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/1xVpNTNOauE--y1HepQhk0K4D0yIS2mRETcCfGGw_Cpc/edit?usp=sharing")
    .getSheets()[0];
  let data = sheet.getDataRange().getValues();
  console.log(data);
}
```

Execution log

```
20:56:19    Notice    Execution started

20:56:23    Info      [ [ 'Product', 'Price', 'Amount' ],
                        [ 'Banana', 2, 112 ],
                        [ 'Apple', 5, 1444 ],
                        [ 'Cucumber', 4, 1225 ],
                        [ 'Tomato', 2, 25 ],
                        [ 'Potato', 1, 23 ],
                        [ 'Orange', 3, 25 ],
                        [ 'Juice', 4, 37 ] ]

20:56:22    Notice    Execution completed
```

Task 2: Adds up all numbers in the “Price” column and prints the total.

```
function sumPrices() {
  let sheet = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/1xVpNTNOauE--y1HepQhk0K4D0yIS2mRETcCfGGw_Cpc/edit?usp=sharing").
  getSheets()[0];
  let data = sheet.getRange(2, 2, sheet.getLastRow() - 1).getValues();
  let total = 0;
  for (let i = 0; i < data.length; i++) total += data[i][0];
  console.log("Total price:", total);
}
```

Execution log

```
21:00:58    Notice    Execution started

21:01:00    Info      Total price: 21

21:00:59    Notice    Execution completed
```

Task 3: Finds most expensive product

```
function mostExpensive() {
  let sheet = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/1xVpNTNOauE--y1HepQhk0K4D0yIS2mRETcCfGGw_Cpc/edit?usp=sharing").
  getSheets()[0];
  let data = sheet.getDataRange().getValues();
  let max = 0, product = "";
  for (let i = 1; i < data.length; i++) {
    if (data[i][1] > max) {
      max = data[i][1];
      product = data[i][0];
    }
  }
  console.log("Most expensive product:", product, "-", max);
}
```

Execution log

21:03:27	Notice	Execution started
21:03:30	Info	Most expensive product: Apple - 5
21:03:29	Notice	Execution completed

Task 4: Calculates and prints the total stock value of each product.

```
function totalValue() {
  let sheet = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/1xVpNTNOauE--y1HepQhk0K4D0yIS2mRETcCfGGw_Cpc/edit?usp=sharing").
  getSheets()[0];
  let data = sheet.getDataRange().getValues();
  for (let i = 1; i < data.length; i++) {
    let name = data[i][0];
    let total = data[i][1] * data[i][2];
    console.log(name + " total value: " + total);
  }
}
```

Execution log

21:05:24	Notice	Execution started
21:05:26	Info	Banana total value: 224
21:05:26	Info	Apple total value: 7220
21:05:26	Info	Cucumber total value: 4900
21:05:26	Info	Tomato total value: 50
21:05:26	Info	Potato total value: 23
21:05:26	Info	Orange total value: 75
21:05:26	Info	Juice total value: 148
21:05:25	Notice	Execution completed

Task 5:

- Adds a new column header called “Total Value” in column D.
- Calculates each product’s total (price × amount).
- Writes the total values in that new column.

```
function addTotalValue() {
  let sheet = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/1xVpNTNOauE--y1HepQhk0K4D0yIS2mRETcCfGGw_Cpc/edit?usp=sharing").
  getSheets()[0];
  let data = sheet.getDataRange().getValues();

  sheet.getRange(1, 4).setValue("Total Value"); // New column header

  for (let i = 1; i < data.length; i++) {
    let total = data[i][1] * data[i][2];
    sheet.getRange(i + 1, 4).setValue(total);
  }
}
```

Product	Price	Amount	Total Value
Banana	2	112	224
Apple	5	1444	7220
Cucumber	4	1225	4900
Tomato	2	25	50
Potato	1	23	23
Orange	3	25	75
Juice	4	37	148

Task 6: Hightligths rows where amount < 50.

```
function highlightLowStock() {
  let sheet = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/1xVpNTNOauE--y1HepQhk0K4D0yIS2mRETcCfGGw_Cpc/edit?usp=sharing").
  getSheets()[0];
  let data = sheet.getDataRange().getValues();
  for (let i = 1; i < data.length; i++) {
    if (data[i][2] < 50) sheet.getRange(i + 1, 1, 1, 3).setBackground("#f8d7da");
  }
}
```

Product	Price	Amount	Total Value
Banana	2	112	224
Apple	5	1444	7220
Cucumber	4	1225	4900
Tomato	2	25	50
Potato	1	23	23
Orange	3	25	75
Juice	4	37	148

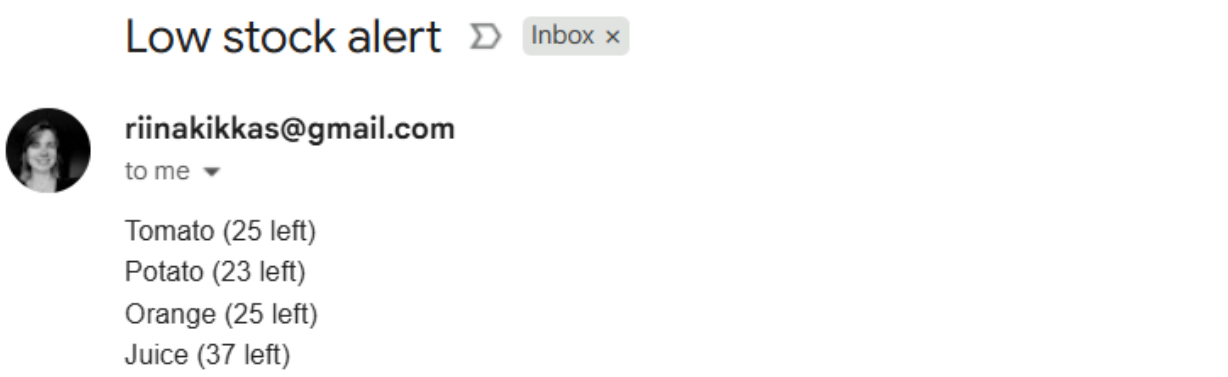
Task 7: Adds dates when data was last updated.

```
function addUpdateDate() {
  let sheet = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/1xVpNTNOauE--y1HepQhk0K4D0yIS2mRETcCfGGw_Cpc/edit?usp=sharing").
  getSheets()[0];
  let date = new Date();
  sheet.getRange("E1").setValue("Last Updated");
  sheet.getRange("E2").setValue(date);
}
```

Product	Price	Amount	Total Value	Last Updated
Banana	2	112	224	16/10/2025
Apple	5	1444	7220	
Cucumber	4	1225	4900	
Tomato	2	25	50	
Potato	1	23	23	
Orange	3	25	75	
Juice	4	37	148	

Task 8: It checks the spreadsheet for all products where the Amount is less than 50. If it finds any, it creates a message listing those items and sends an email alert.

```
function sendLowStockAlert() {
  let sheet = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/1xVpNTNOauE--y1HepQhk0K4D0yIS2mRETcCfGGw_Cpc/edit?usp=sharing").
  getSheets()[0];
  let data = sheet.getDataRange().getValues();
  let message = "";
  for (let i = 1; i < data.length; i++) {
    if (data[i][2] < 50) message += data[i][0] + " (" + data[i][2] + " left)\n";
  }
  if (message) MailApp.sendEmail("riinakikkas@gmail.com", "Low stock alert", message);
}
```



Task 9: Sorts Products by Total Value Desc

```
function addTotalValueAndSort() {
  let sheet = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/1xVpNTNOauE--y1HepQhk0K4D0yIS2mRETcCfGGw_Cpc/edit?usp=sharing").
  getSheets()[0];
  let lastRow = sheet.getLastRow();
  sheet.getRange(1, 4).setValue("Total Value");
  let prices = sheet.getRange(2, 2, lastRow - 1).getValues();
  let amounts = sheet.getRange(2, 3, lastRow - 1).getValues();
  let totals = prices.map((p, i) => [Number(p[0]) * Number(amounts[i][0])]);
  sheet.getRange(2, 4, totals.length, 1).setValues(totals);
  sheet.getRange(2, 1, lastRow - 1, 4).sort([{column:4, ascending:false}]);
}
```

Product	Price	Amount	Total Value	Last Updated
Potato	1	23	7220	16/10/2025
Banana	2	112	4900	
Tomato	2	25	224	
Orange	3	25	148	
Cucumber	4	1225	75	
Juice	4	37	50	
Apple	5	1444	23	

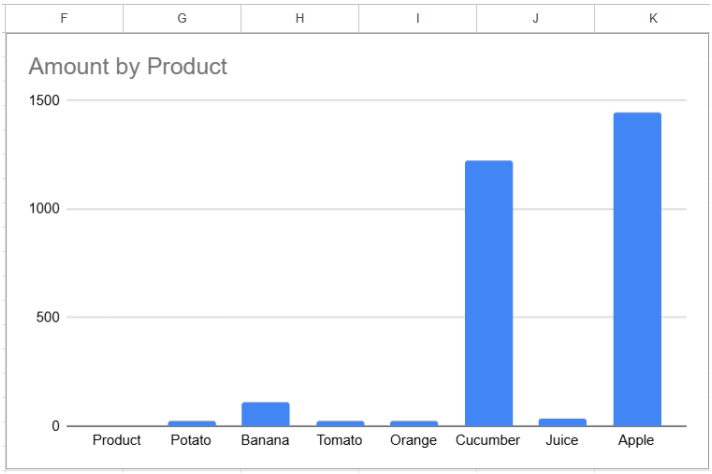
Task 10: Creates or updates a “Summary” sheet showing the total amount, average price, and the product with the highest total value.

```
function buildSummarySheet() {
  let ss = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/1xVpNTNOauE--y1HepQhk0K4D0yIS2mRETcCfGGw_Cpc/edit?usp=sharing");
  let sheet = ss.getSheets()[0];
  let data = sheet.getDataRange().getValues().slice(1);
  let totalAmount = data.reduce((s,r)=>s+Number(r[2]),0);
  let avgPrice = data.reduce((s,r)=>s+Number(r[1]),0)/data.length;
  let totals = data.map(r=>[r[0], Number(r[1])*Number(r[2])]);
  totals.sort((a,b)=>b[1]-a[1]);
  let top = totals[0];
  let summary = ss.getSheetByName("Summary") || ss.insertSheet("Summary");
  summary.clear();
  summary.getRange(1,1,5,2).setValues([
    ["Metric","Value"],
    ["Total items (Amount)", totalAmount],
    ["Average price", avgPrice],
    ["Top by value", top[0]],
    ["Top value", top[1]]
  ]);
}
```

Metric	Value
Total items (Amount)	2891
Average price	3
Top by value	Apple
Top value	7220

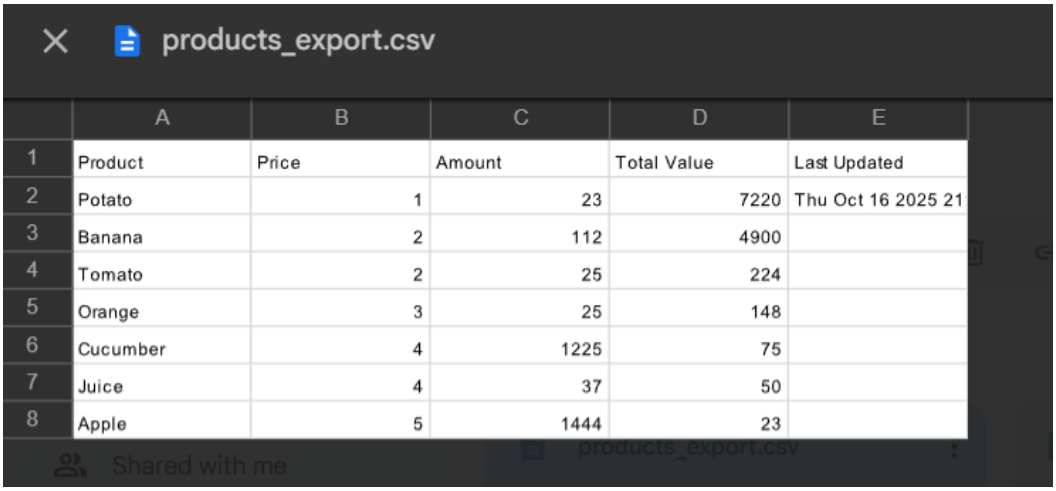
Task 11: Creates a diagram

```
function insertAmountChart() {
  let sheet = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/1xVpNTNOauE--y1HepQhk0K4D0yIS2mRETcCfGGw_Cpc/edit?usp=sharing").
  getSheets()[0];
  let lastRow = sheet.getLastRow();
  let chart = sheet.newChart()
    .asColumnChart()
    .addRange(sheet.getRange(1,1,lastRow,1))
    .addRange(sheet.getRange(1,3,lastRow,1))
    .setPosition(1,6,0,0)
    .setOption("title","Amount by Product")
    .build();
  sheet.insertChart(chart);
}
```



Task 12: Exports a CSV file to Google Drive

```
function exportToCSV() {
  let sheet = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/1xVpNTNOauE--y1HepQhk0K4D0yIS2mRETcCfGGw_Cpc/edit?usp=sharing").
  getSheets()[0];
  let data = sheet.getDataRange().getValues();
  let csv = data.map(r => r.map(v => String(v).includes(",") ? `"${String(v).replace(/"/g,'""')}"` : v).join(",")).join("\n");
  DriveApp.createFile("products_export.csv", csv, MimeType.CSV);
}
```



	A	B	C	D	E
1	Product	Price	Amount	Total Value	Last Updated
2	Potato	1	23	7220	Thu Oct 16 2025 21
3	Banana	2	112	4900	
4	Tomato	2	25	224	
5	Orange	3	25	148	
6	Cucumber	4	1225	75	
7	Juice	4	37	50	
8	Apple	5	1444	23	