Assignment - Apps Script Data Transfer

Report by Riina Kikkas

This project automates data transfer in Google Sheets using Apps Script. It demonstrates reading data from one sheet, filtering it with conditions, and writing the results to a dynamically created report sheet.

Link to spreadsheet

Task 1: This script creates a new sheet named "Report", filters the data from the "Loans" sheet to include only the rows where the Returned column is empty (unreturned books), and writes the filtered results into the new report sheet.

```
/**
 * Creates a "Report" sheet that lists only unreturned loans.
 */
function createUnreturnedReport() {
  const ss = SpreadsheetApp.getActiveSpreadsheet(); // get the active spreadsheet
  const src = ss.getSheetByName("Loans");           // read data from the "Loans" sheet
  const data = src.getDataRange().getValues();      // get all data including the header row
  const header = data[0];                           // first row = header
  const rows = data.slice(1);                       // remaining rows = actual data

  // Filter: keep only rows where the "Returned" column (E = index 4) is empty or null
  const onlyUnreturned = rows.filter(r => r[4] === "" || r[4] === null);

  // Prepare output (header + filtered rows)
  const out = [header, ...onlyUnreturned];

  // Replace old "Report" sheet if it already exists
  const old = ss.getSheetByName("Report");
  if (old) ss.deleteSheet(old);
  const report = ss.insertSheet("Report");

  // Write the filtered data to the new "Report" sheet
  if (out.length > 0) {
    report.getRange(1, 1, out.length, out[0].length).setValues(out);
  }

  // Optional: automatically resize columns for better readability
  report.autoResizeColumns(1, header.length);
}
```

Student	BookID	BorrowDate	DueDate	Returned
Anna	BK-001	9/25/2025	10/5/2025	
Mari	BK-003	9/15/2025	9/28/2025	
Rasmus	BK-005	9/20/2025	9/30/2025	
Liis	BK-006	9/22/2025	9/29/2025	

Task 2: This script creates a new *Report* sheet showing only overdue and unreturned books. It filters out rows where books have been returned or are not yet due, calculates how many days each loan is overdue, adds that as a new column, and writes the final result to a new sheet.

```
/**
 * Creates a "Report" of loans that are overdue AND not returned.
 */
function createOverdueReport() {
  const ss = SpreadsheetApp.getActiveSpreadsheet(); // get the active spreadsheet
  const src = ss.getSheetByName("Loans");           // read data from the "Loans" sheet
  const data = src.getDataRange().getValues();      // get all data including the header row
  const header = data[0];                           // first row = header
  const rows = data.slice(1);                       // remaining rows = actual data

  const today = new Date();                         // get today's date

  // Filter: keep rows where the book is not returned AND the due date is a valid date before today
  const overdue = rows.filter(r => {
    const returned = r[4];                          // Returned column (E)
    const due = r[3];                               // DueDate column (D)
    const isUnreturned = (returned === "" || returned === null); // not returned
    const isDate = due instanceof Date;              // make sure the value is a real date
    return isUnreturned && isDate && due < today;    // keep only overdue rows
  });

  // Add a new column "DaysOverdue" that shows how many days late each loan is
  const withDays = overdue.map(r => {
    const due = r[3];
    const days = Math.floor((today - due) / (1000 * 60 * 60 * 24)); // calculate difference in days
    return [...r, days]; // add DaysOverdue value at the end of each row
  });
```

```
  // Add a header for the new column
  const outHeader = [...header, "DaysOverdue"];
  const out = [outHeader, ...withDays];

  // Delete the old "Report" sheet if it exists, then create a new one
  const old = ss.getSheetByName("Report");
  if (old) ss.deleteSheet(old);
  const report = ss.insertSheet("Report");

  // Write the filtered and calculated data to the new "Report" sheet
  if (out.length > 0) {
    report.getRange(1, 1, out.length, out[0].length).setValues(out);
  }

  // Automatically resize columns for better readability
  report.autoResizeColumns(1, outHeader.length);
}
```

Student	BookID	BorrowDate	DueDate	Returned	DaysOverdue
Anna	BK-001	9/25/2025	10/5/2025		23
Mari	BK-003	9/15/2025	9/28/2025		30
Rasmus	BK-005	9/20/2025	9/30/2025		28
Liis	BK-006	9/22/2025	9/29/2025		29