

Assignment 1: Web Application Mimicking Google Sheets

Objective

Develop a web application that closely mimics the user interface and core functionalities of Google Sheets, focusing on mathematical and data quality functions, data entry, and key UI interactions.

Features and Implementation

1. Spreadsheet Interface

- **Mimic Google Sheets UI:** Implement a UI with a toolbar, formula bar, and a grid layout for cells.
- **Drag Functions:** Enable dragging of cell content, formulas, and selections.
- **Cell Dependencies:** Ensure that formulas update dynamically when related cells change.
- **Basic Cell Formatting:** Allow bold, italics, font size, and color adjustments.
- **Row and Column Management:** Implement adding, deleting, and resizing rows and columns.

2. Mathematical Functions

The following functions will be implemented:

1. **SUM(range)** - Calculates the sum of a range of cells.
2. **AVERAGE(range)** - Computes the average of values in a range.
3. **MAX(range)** - Returns the maximum value in a range.
4. **MIN(range)** - Returns the minimum value in a range.
5. **COUNT(range)** - Counts numerical values in a range.

3. Data Quality Functions

1. **TRIM(cell)** - Removes leading and trailing spaces from a cell.
2. **UPPER(cell)** - Converts text to uppercase.
3. **LOWER(cell)** - Converts text to lowercase.
4. **REMOVE_DUPLICATES(range)** - Removes duplicate rows from a selected range.
5. **FIND_AND_REPLACE(range, findText, replaceText)** - Finds and replaces text within a range.

4. Data Entry and Validation

- Support for different data types (numbers, text, dates).
- Basic validation for numeric cells to ensure only numbers are entered.

5. Testing

- Provide an interactive UI for users to test functions.
 - Display results of function execution clearly.
-

Bonus Features

- Additional mathematical and data quality functions.
 - Support for complex formulas and absolute/relative cell referencing.
 - Save and load spreadsheet functionality.
 - Data visualization with charts and graphs.
-

Evaluation Criteria

- **Fidelity to Google Sheets UI:** Accuracy of interface and user experience.
 - **Feature Completeness:** Full implementation of required functionalities.
 - **Mathematical and Data Quality Accuracy:** Correctness of functions.
 - **Usability:** User-friendly and intuitive interactions.
 - **Code Quality:** Readability, maintainability, and modular design.
 - **Bonus Features:** Implementation of advanced features.
 - **Documentation:** A clear README explaining the tech stack and data structures used.
-

Tech Stack

Frontend

- **HTML/CSS:** For UI design and styling.
 - **JavaScript (React/Angular/Vue):** For dynamic UI interactions and state management.
-

Conclusion

This web application aims to closely resemble Google Sheets while implementing core spreadsheet functionalities with an efficient UI. Future improvements could include collaboration features and advanced data visualization tools.