

# Extended Spreadsheet Program Proposal

Team Members:

Mayank Goel (2023CS10204)

Ahilaan Saxena (2023CS10076)

Mohil Shukla (2023CS10186)

March 23, 2025

## Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Overview</b>                              | <b>2</b> |
| <b>2</b> | <b>Motivation</b>                            | <b>2</b> |
| <b>3</b> | <b>Objectives</b>                            | <b>2</b> |
| <b>4</b> | <b>Features</b>                              | <b>2</b> |
| 4.1      | Core Functionalities . . . . .               | 2        |
| 4.2      | Extended Formula Support . . . . .           | 2        |
| 4.3      | Undo/Redo and Autosave . . . . .             | 3        |
| 4.4      | Cut/Copy/Paste Ranges . . . . .              | 3        |
| 4.5      | Visualization of Dependency Chains . . . . . | 3        |
| 4.6      | Additional Enhancements . . . . .            | 3        |
| <b>5</b> | <b>User Interface (UI)</b>                   | <b>3</b> |
| 5.1      | Web-based Interface . . . . .                | 3        |
| <b>6</b> | <b>API Integration</b>                       | <b>4</b> |
| <b>7</b> | <b>Testing and Validation</b>                | <b>4</b> |
| <b>8</b> | <b>Conclusion</b>                            | <b>4</b> |

# 1 Overview

In this project, we propose to extend our existing spreadsheet program, originally implemented in C, by re-implementing it in Rust and adding a host of new functionalities. In addition to re-creating the core features from the C lab, our extended version will include extensions such as an interactive web-based user interface and a stock exchange API for financial operations.

## 2 Motivation

The main motivation for this project is to enhance our spreadsheet program by leveraging Rust's performance and safety, while also making the tool more accessible and feature-rich. The planned web UI will improve usability and the API integration will open new opportunities for financial data analysis.

## 3 Objectives

- Re-implement the original C lab spreadsheet program in Rust with full compatibility.
- Introduce advanced features such as undo/redo, autosave (every 60 seconds), and cut/copy/paste functionalities.
- Extend cell support to include strings and float values alongside integers.
- Develop a web-based user interface to display and interact with the spreadsheet.
- Integrate a stock exchange API to enable financial operations such as stock comparisons, track record analysis, and portfolio management.

## 4 Features

### 4.1 Core Functionalities

- Basic spreadsheet operations: setting cell values, arithmetic formulas, and dependency-based recalculations.
- Support for multiple data types: integers, floats, and strings.

### 4.2 Extended Formula Support

- New formulas including MEDIAN, MODE, and matrix operations on ranges.
- Ability to type `help` to view a list of all supported formulas.

### 4.3 Undo/Redo and Autosave

- **Undo/Redo:** Enables users to revert or reapply recent changes.
- **Autosave:** Automatically saves the spreadsheet every 60 seconds.

### 4.4 Cut/Copy/Paste Ranges

- Support for selecting ranges and performing cut, copy, and paste operations.

### 4.5 Visualization of Dependency Chains

One of the new features is the ability to visualize the dependency relationships among cells. This can help users understand which cells are impacted by changes in other cells, especially when complex formulas are involved.

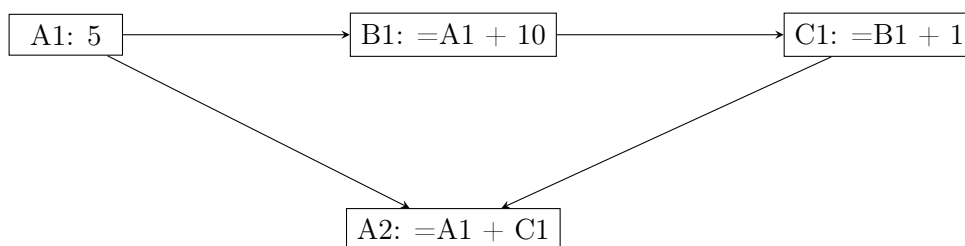


Figure 1: A layout for a cell-dependency graph visualization

### 4.6 Additional Enhancements

- A formula bar to display the formula of the selected cell.
- Extended support for editing and displaying float values.

## 5 User Interface (UI)

### 5.1 Web-based Interface

The UI will be developed as an interactive web application using Rust-based web frameworks. Key elements include:

- **Spreadsheet View:** A dynamic grid displaying the spreadsheet, enabling cell selection and editing in-place.
- **Command Search Box:** A text input area to execute commands (similar to executing from the terminal) or search for functionalities.
- **Graph Visualization:** Ability to open generated dependency graphs in a new browser tab.

## 6 API Integration

The API section will focus on integrating data from a stock exchange. Planned features include:

- **Stock Comparison:** Allow users to compare different stocks based on real-time and historical data.
- **Track Record Analysis:** Provide detailed performance histories of individual stocks.
- **Portfolio Management:** Enable users to manage and analyze their investment portfolios.
- **Additional Operations:** Potentially include basic statistical analyses (e.g., moving averages) on stock data.

It should be noted that while these financial features are ambitious, we will implement them to the best of our ability within the project timeline. If time allows, support might be integrated to simulate buying and selling of stocks from a dynamic portfolio in real-time.

## 7 Testing and Validation

- **Unit Testing:** Create thorough tests for both the backend logic and UI components.
- **Integration Testing:** Verify that the Rust backend, web UI, and stock exchange API work together seamlessly.
- **User Acceptance Testing:** Collect user feedback to ensure the interface is intuitive and all functionalities perform as expected.

## 8 Conclusion

This proposal outlines our plan to re-implement and extend the existing C lab spreadsheet program in Rust. By adding a modern web UI and integrating a stock exchange API, we aim to build a versatile and powerful tool that caters to both general spreadsheet users and those requiring financial data analysis. We recognize the ambitious nature of some of these extensions and are committed to delivering a robust solution within the project timeline.