

Automating Lab Product Stability Tracking: Reducing Task Time by 70% in One Week

Ben Christiansen-Developer & Designer

Built a JavaScript-based app that automated the manual calculation of lab product stability, reducing a 20-minute task to 6 minutes and freeing MLS staff to focus more on patient sample analysis.

Outcome Statement

Developed an app that automatically calculates remaining stability hours for lab products, cutting a 20-minute manual task down to 6 minutes — a 70% faster process. Staff gained time back immediately after launch, boosting overall productivity.

Context

Medical Laboratory Scientists (MLS) are required to quickly analyze and report results for a high volume of patient specimens.

They are responsible for instrument maintenance and quality control tests. These duties are time consuming and tools are needed to make them more efficient.

For example, we discussed the current manual workflow for calculating remaining stability of lab products, taking 20 minutes on average. A way to automate the process would help free up time to focus on patient samples.

My role in the project

Collaborating closely with MLS to understand the features they would find most useful, I applied my HTML, CSS, and JavaScript skills to design and code the calculation app from scratch to automate the calculation of remaining product stability. Upon completion, I collaborated with the lab team to test the app's accuracy and ease of use.

The mandate: 50%+ reduction in task time in 1 week

MLS staff were tasked with maintaining stability of reagents (lab products) on two specimen analyzers, with each analyzer used on alternating days for testing. MLS manually calculates remaining reagent stability hours to prepare for the next day. We needed a solution to reduce the 20-minute manual process, improving efficiency and freeing up time for patient sample analysis.

Note:

AI tools were used to assist with research, debugging, and code explanation. All design decisions, implementation, and final testing were completed by me.

Day 1-2:

I spoke with Anny, an MLS, to understand what features and functions the app would require. We discussed the importance of it being easy to use and 100% accurate.

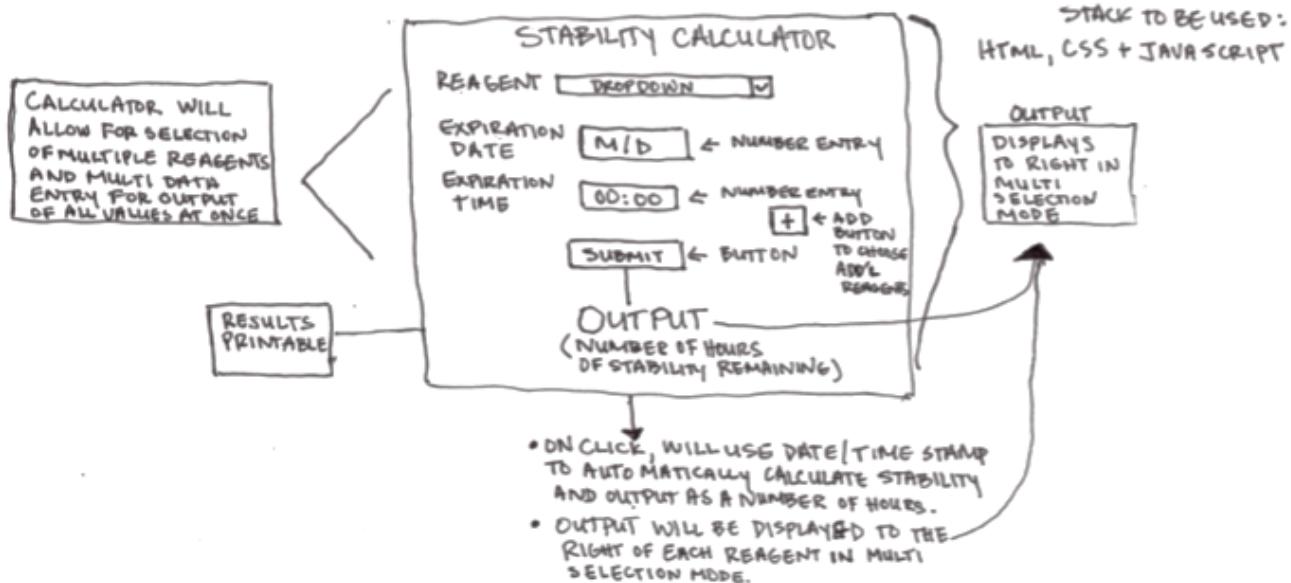
We collaborated on a wireframe to visualize the design and plan the UX/UI components. This included multiple value inputs to calculate and display the remaining stability hours, with an option to print the results.

Project wireframe: used to complete initial HTML coding and CSS styling.

- 2 DIFFERENT ANALYZERS
- SHARED REAGENTS - EACH WITH IT'S OWN STABILITY CRITERIA (EX: ^{SOME} EXPIRE 48 HRS FROM OPEN DATE / TIME, OTHERS > 120 HRS)
- CURRENT PROCESS: MED TECH MANUALLY CALCULATES NUMBER OF HOURS LEFT OF STABILITY AND ENTERS VALUE INTO ANALYZER THE REAGENT IS BEING MOVED TO.

SOLUTION

BUILD A REAGENT EXPIRATION "CALCULATOR" TO OUTPUT NUMBER OF HOURS LEFT OF STABILITY. THIS TOOL WILL BE EASY TO USE AND ENSURE ACCURACY, AVOIDING USE OF EXPIRED PRODUCT AND/OR PREVENTING WASTE BY OF PRODUCT DISCARD TOO EARLY.



Initial prototype:

Remaining Stability Calculator



Select Reagent: Expiration Date: Expiration Time:

Day 3-4:

I shared the app prototype with Anny and we discussed the following updates:

- Submit button to be moved below the inputs.
- A new input for reagent volume.
- A “-” button to delete unintentional entries.
- A “reset” button to clear entries was needed.
- Output rounded down to the nearest whole hour.

Version 2 wireframe:

10/11/25 - MEETING WITH ANNY P. MLS. VERSION 2 CHANGES/ADDITIONS

① MOVE SUBMIT BUTTON.
② ADD INPUT (NUMBER TYPE) FOR VOLUME ENTRY. NO CALCULATION NECESSARY FOR THIS FIELD.
SIMPLY NEEDED FOR PRINTOUT OF REMAINING STABILITY HOURS OUTPUT.

OUTPUT/PRINTOUT
- OUTPUT HOURS SHOULD ROUND DOWN TO NEAREST WHOLE NUMBER.
- PRINT VERSION SHOULD HAVE ALTERNATING GRAY/WHITE ROWS FOR READABILITY.

Remaining Stability Calculator



Select Reagent: Expiration Date: Expiration Time:

* MLS TIMED TASK AND DOING IT MANUALLY, TAKES ABOUT 20 MIN.

Day 3-4 continued:

HTML and CSS were updated to accommodate the layout changes and button additions.

JavaScript code was written using reagent, expiration date and expiration time inputs and comparing the values to a timestamp from the user's computer in ISO 8601 format / UTC time. This calculated the difference in hours and output the remaining stability rounded down to the nearest whole hour.

Version 2 completed: Revised prototype with new buttons and improved usability.

Remaining Stability Calculator

Select Reagent: Neoplastine C1 Plus

Expiration Date: mm/dd/yyyy

Expiration Time: - - -

Volume:

SUBMIT

trash can icon

Day 5:

I added CSS and JavaScript event listeners for the Add, Minus, and Reset buttons. The Add button appends a new row of input fields, the Minus button removes the selected row, and the Reset button clears all input fields in all rows. Clicking the Submit button calculates and displays all outputs.

Version 3 wireframe:

- VERSION 3
- ① ADD A MINUS “-” OR TRASH BUTTON TO ENABLE DELETION OF ENTRIES.
 - ② ADD A RESET BUTTON TO CLEAR ALL ENTRIES/DATA.

Remaining Stability Calculator

Select Reagent: Neoplastine C1 Plus

Expiration Date: mm/dd/yyyy

Expiration Time: - - -

Volume:

+ - ⚡

SUBMIT

trash can icon

Day 5 Continued:

Version 3 completed: Refined design supporting multiple input rows and streamlined workflow.

The screenshot shows a web-based calculator titled "Remaining Stability Calculator". At the top, there is a decorative header with a colorful, abstract background. Below the header, there is a form with the following fields: "Select Reagent:" dropdown menu set to "Neoplastine C1 Plus", "Expiration Date:" date input field showing "mm/dd/yyyy", "Expiration Time:" time input field showing "00:00:00", "Volume:" text input field showing "0.00", and "Hours Remaining:" text input field showing "00:00:00". To the right of these fields are three small buttons: a plus sign, a trash can, and a refresh/circular arrow. Below the form is a green "SUBMIT" button. At the bottom center is a small printer icon.

Day 6-

After further consideration and confirming with Anny, I changed the JavaScript functionality by making the Reset button clear only the input fields and output for the row in which it is clicked. This prevents MLS from having to start over when only one row needs to be deleted. I added an event listener to the Print button to open the printer dialog. I also implemented a print-specific CSS media query to format the report for readability, which included zebra striping on alternate rows and hiding buttons and images to ensure a clean, professional layout when printing.

Report with CSS formatting: Final print layout showing zebra striping and clean formatting.

Remaining Stability Calculator

The screenshot shows the "Remaining Stability Calculator" interface with three input rows. Each row contains the following fields: "Select Reagent:", "Expiration Date:", "Expiration Time:", "Volume:", and "Hours Remaining:". The first row has a light gray background and displays the following data: "Neoplastine C1 Plu", "11/14/2025", "04:12 PM", "2ml", and "61". The second row has a white background and displays the following data: "Fibrinogen", "11/19/2025", "04:15 AM", "2ml", and "170". The third row has a light gray background and displays the following data: "Liatest D-DI Buffer", "11/11/2025", "04:18 AM", "2ml", and "-22". Below the rows is a green "SUBMIT" button.

Day 7:

Finalized the HTML, CSS, and JavaScript files in Visual Studio Code and pushed all project files to a GitHub repository, including a detailed README documenting the project.

MLS and I tested the app to verify all components functioned correctly and outputs were accurate.

Reflection:

This project deepened my understanding of JavaScript and front-end development while highlighting the impact of close collaboration with end users. Working alongside MLS staff showed me how thoughtful design and automation can significantly improve efficiency. Seeing the tool reduce task time by over 70% reinforced my motivation to build solutions that make everyday work easier and more effective.