COLORADO SCHOOL OF MINES
EARTH • ENERGY • ENVIRONMENT

CSCI 370 Final Report

Future Business Innovators (FBI)

Emily Marks
Doug Meany
Kate Callan

Revised June 15, 2025

CSCI 370 Summer 2025

Prof. Kathleen Kelly

Table 1: Revision history

Revision	Date	Comments
New	May 13, 2025	Completed Sections: i. Introduction ii. Functional Requirements iii. Non-functional Requirements iv. Risks v. Definition of Done vi. Team Profile vii. References viii. Appendix A – Key Terms
Rev – 2	May 19, 2025	Completed Sections: i. System Architecture
Rev – 3	May 27, 2025	Completed Sections: i. Software Test and Quality ii. Project Ethical Considerations
Rev – 4	June 4, 2025	Completed Sections: i. Project Completion Status ii. Future Work iii. Lessons Learned iv. Acknowledgements
Rev – 5	June 8, 2025	Completed References; Made revisions
Rev – 6	June 15, 2025	Final Revisions

Table of Contents

I. Introduction

Compton Integrated Analytics plans to invite a group of small to medium oil and gas operators to participate in the MiLo Beta program, a Software as a Service (SaaS) tool for interactive well-placement planning and estimation. During the Beta phase, participants will use MiLo's web interface to create and adjust "benches" and "wells." To turn these user interactions into data-driven insights, we developed an analytics application, added onto MiLo, that records every user's session. Administrators can then download an Excel report to understand usage patterns, prioritize feature work, and ultimately improve the MiLo experience before full launch.

This document outlines the context, goals, and background of the project, as well as key stakeholders and responsibilities related to the development of the MiLo Beta user analytics application.

A. Project goal

The goal of this project is to build a user analytics application for the MiLo Beta program. Metrics such as time spent in the software, key interaction points/times, and exit points should be recorded and exported in a clear format.

B. Client Background

Compton Integrated Analytics is a consulting firm near Denver that specializes in custom software, geologic analysis, and data analytics for the energy industry but mainly focused on oil and gas. A user analytics application is necessary to provide a way to analyze and use the metrics for refining the Beta program.

C. Existing Software Platform

The existing software platform is based on CodeIgniter, a PHP framework used for building web applications. The software was initially developed by a third party, offshore development team hired by Compton Integrated Analytics. Continued development is needed to refine the software based on real user feedback from the Beta program.

D. Source of Data

The data will be sourced from user interactions with the MiLo Beta program software.

E. Hardware Interface

Not applicable.

F. Definitions and Acronyms

SaaS: Software as a Service
MiLo: The name of the Beta program software
KPI: Key Performance Indicator

G. Stakeholders and Users

The primary users of the software will be small to medium-sized oil and well businesses. Stakeholders include business owners and operational managers who will use the software as a tool for well placement planning and calculations.

H. Maintenance Responsibility

Compton Integrated Analytics will be responsible for maintaining the software.

II. Functional Requirements

The following functional requirements define the core capabilities the system must support. Each requirement is critical to the intended operation of the platform:

A. The system must track the amount of time a user spends on each page and log any modifications they make to the initial Wine Rack View.

B. Collected data must be formatted for easy interpretation and stored in a way that allows for download and access by administrators.

C. Upon data download, charts should be generated automatically to help visualize the collected information.

D. Access to the data download and visualization functionality must be restricted to admin-level users and should be located logically within the website's interface.

E. Each user session must be assigned a unique identifier that links all associated data logs.

F. All data collected must be accurate and verifiable.

III. Non-Functional Requirements

The functional requirements outlined above are supported by the following non-functional requirements, which define system-wide constraints related to performance, compatibility, maintainability, and usability:

A. Data collection must not significantly impact MiLo's runtime performance and should support concurrent user sessions.

B. The codebase must be clearly documented to allow for seamless hand-off to Compton Integrated Analytics, and it should follow a modular design to facilitate future updates or bug fixes.

C. Data collection must operate consistently across all supported browsers and be resilient to unexpected session terminations, retaining data even if the user exits abruptly.

D. Exported files and visualizations must be easily accessible and interpretable by admin users.

E. The system must comply with ADA standards to ensure accessibility.

IV. Risks

Risk	Likelihood	Impact	Mitigation Plan
Team lacks web development experience	Very Likely	Moderate	Allocate time for team members to learn necessary web development skills.
Potential to disrupt the existing site during implementation	Unlikely	Moderate	Ensure the client maintains a full backup of the current source code before deployment. Conduct thorough testing in a controlled environment prior to going live.
User privacy concerns with data tracking	Unlikely	Major	Ensure all collected data remains anonymous. No personally identifiable information will be stored.

V. Definition of Done

The following outlines the criteria for determining when the project is complete and ready for client acceptance. This includes minimal viable feature set, testing procedures, and delivery method.

Minimal useful feature set:

- An Excel file is generated and stored in a location easily accessible to site administrators.
- The Excel file must include:
    - The amount of time each user spends on a given page.
    - All button clicks and text box updates performed by the user, each with corresponding timestamps
- An easily accessible method for admin users to delete all collected user analytics.
- Only admin level sessions can access analytics UI or trigger exports

Client Testing Procedures:

- The client will perform the following checks on a staging deployment or local sandbox:
- The client will verify that the Excel file is generated correctly and accurately tracks user interaction data.
- The client will confirm that the existing site functionality remains unaffected.

Product Delivery:

- The final product will be delivered as a zipped folder containing the updated version of the website's source code. Our client also has GitHub access to the project repository.
- A walkthrough tutorial is included with delivery of product so our client is able to test and view changes in a local environment.
- Delivery will occur once all required features have been implemented and verified to function as intended.

VI. System Architecture

Figure 1 provides a comprehensive overview of the user interaction flow on the fracking visualization platform. This website is designed to allow oil industry professionals to model representations of fracking areas using a web interface. After logging in, users are directed to an initial setup page (Page 1), where they input relevant parameters using text boxes. These tools enable users to define and modify sets of wells or infrastructure relevant to their project.

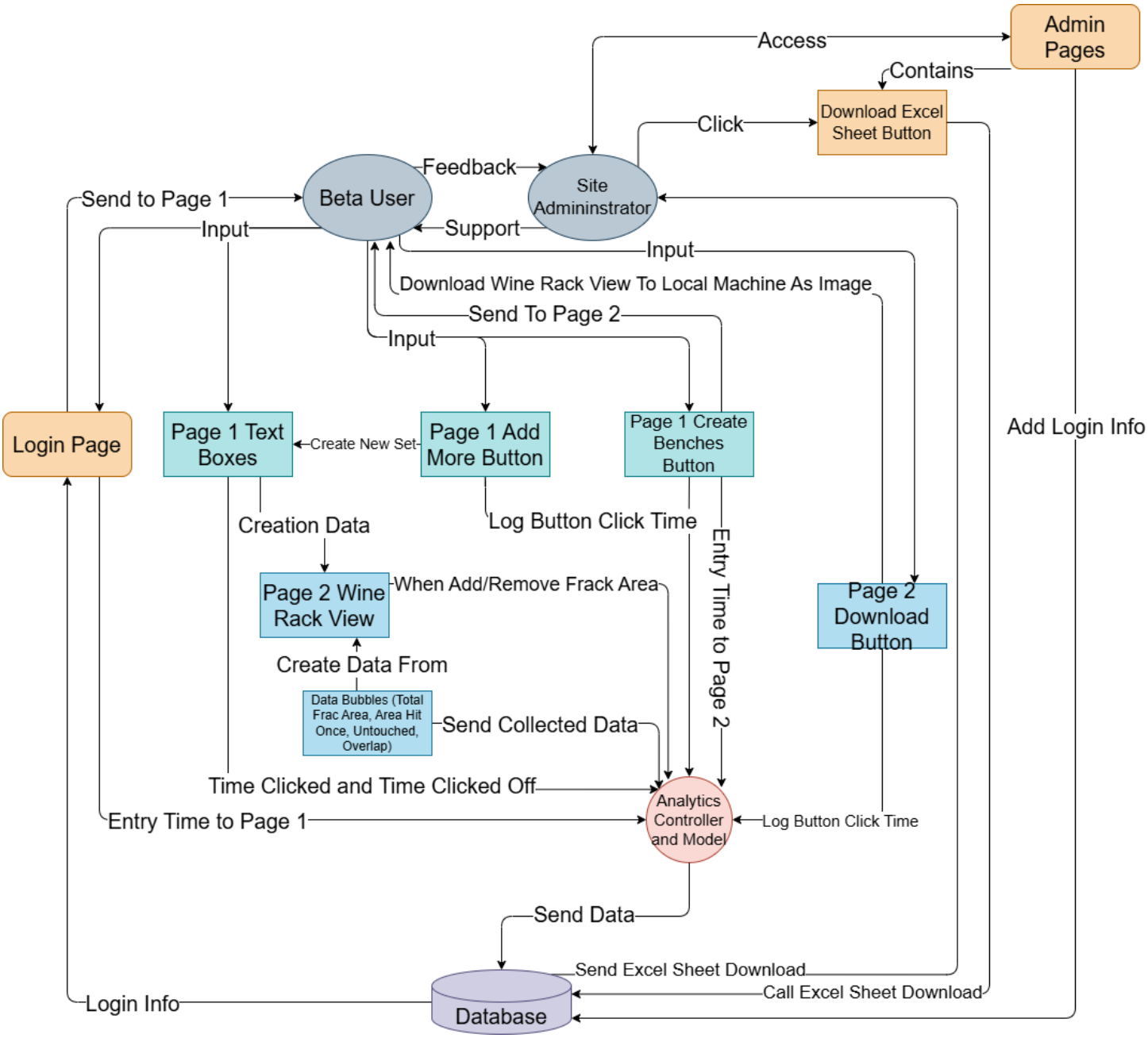


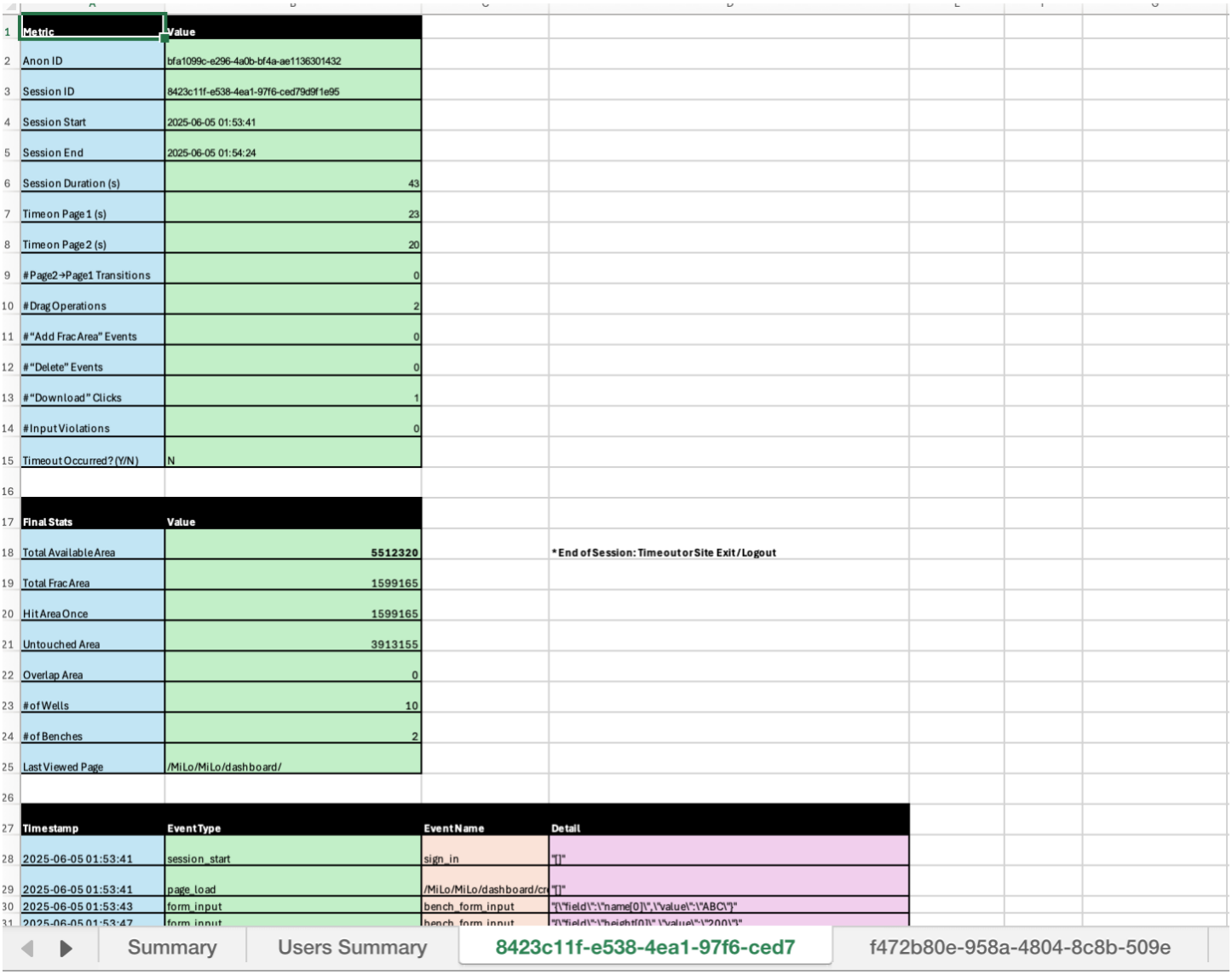
Figure 1. User Interaction Flowchart

Once data has been entered, users proceed to Page 2 with the "Create Benches" button, where they can see the Wine Rack View. The Wine Rack View is a visual representation of the fracking layout. This interface allows users to drag and drop individual well elements into place, as well as add and remove wells as needed. As users interact with the visualization, visual data bubbles update dynamically in real time. These bubbles reflect various percentages of interest, such as total fracking area, area touched once, untouched area, and overlapping areas. The visual feedback enables users to adjust well placement and assess spatial relationships within their model. Users may also download the Wine Rack View as an image, which supports operational documentation and review.

An admin page enables system administrators to manage user accounts, as well as access the excel sheet created to store user data. This ensures proper oversight and maintenance of platform functionality, as well as facilitating feedback loops between users and administrators.

The contribution made by our team to this platform is the implementation of an integrated data collection system, represented in the flowchart as the Analytics Controller and Model. This module serves to log all meaningful user interactions across Page 1 and Page 2. It captures button click events with timestamps, entry times to each interface component, and interactions with the Wine Rack View such as addition, removal, and repositioning of fracking elements. Each of these events is recorded and compiled into a structured Excel spreadsheet. Figures 2, 3, and 4 below show the layout of how the collected analytics are displayed.

This system ensures that user activity is fully traceable, supporting future evaluations of user behavior, interface efficiency, and decision-making patterns. Administrators may retrieve and review this dataset through the admin page, enhancing internal auditing of the platform.



Metric	Value
Anon ID	bfa1099c-e296-4a0b-bf4a-ae1136301432
Session ID	8423c11f-e538-4ea1-97f6-ced79d9f1e95
Session Start	2025-06-05 01:53:41
Session End	2025-06-05 01:54:24
Session Duration (s)	43
Time on Page 1 (s)	23
Time on Page 2 (s)	20
# Page2→Page1 Transitions	0
# Drag Operations	2
# "Add Frac Area" Events	0
# "Delete" Events	0
# "Download" Clicks	1
# Input Violations	0
Timeout Occurred? (Y/N)	N

Final Stats	Value
Total Available Area	5512320
Total Frac Area	1599165
Hit Area Once	1599165
Untouched Area	3913155
Overlap Area	0
# of Wells	10
# of Benches	2
Last Viewed Page	/MiLo/MiLo/dashboard/

*End of Session: Timeout or Site Exit / Logout

Timestamp	Event Type	Event Name	Detail
2025-06-05 01:53:41	session_start	sign_in	"[]"
2025-06-05 01:53:41	page_load	/MiLo/MiLo/dashboard/cr	"[]"
2025-06-05 01:53:43	form_input	bench_form_input	"[\"field\":\"name[0]\",\"value\":\"ABC\"]"
2025-06-05 01:53:47	form_input	bench_form_input	"[\"field\":\"height[0]\",\"value\":\"200\"]"

Summary | Users Summary | 8423c11f-e538-4ea1-97f6-ced7 | f472b80e-958a-4804-8c8b-509e

Figure 2. Per-Session Metrics Excel Sheet with Example Data

Anon ID	# Sessions	Avg. Session Duration	Avg. Time P1	Avg. Time P2	% Sessions Timed Out	Avg. # Page2→Page1 Transitions	Avg. # Drag Operations	Avg. # "Add Frac Area"	Avg. # "Delete"	Avg. # Downloads	# Sessions Ending on Page 1
bfa1099c-e296-4a	4	69	47	22	0%	0.5	1	0.5	0.3	0.8	1
2f6b1ef9-5a30-44c	1	26	11	15	0%	0	3	1	1	0	0

Figure 3. Per-User Metrics Excel Sheet with Example Data

Metric	Value
Total Users	2
Total Sessions	5
Avg. Session Duration (s)	60
Avg. Time on Page 1 (s)	39.4
Avg. Time on Page 2 (s)	20.6
Avg. # Page2→Page1 Transitions	0.4
Avg. # Drag Operations	1.4
Avg. # "Add Frac Area" Events	0.6
Avg. # "Delete" Events	0.4
Avg. # Downloads	0.6
Timeout Rate (% sessions)	0%
Avg. Benches Created per Session	1
Avg. # of Wells per Bench	5
Avg. # Sessions Ending on Page 1	0.2

Summary | Users Summary

Figure 4. Overall Summary Metrics Excel Sheet with Example Data

The exported Excel workbook is organized into a concise, three-tier structure that begins with a high-level "Summary" sheet. On the Summary tab, we have a table of key metrics: total users, total sessions, average session duration, average time on pages, timeout rate, etc. in rows 2–15 beneath a bold header. The second tab, Users Summary, closely resembles the Summary sheet but scoped to each individual's data. Each row begins with the user's anonymous ID, followed by their session count, average session length, page-time averages, timeout percentage, transition counts, drag/Add/Delete event totals, download count, and the number of sessions ending on Page 1. Finally, each user session is represented in its own worksheet, where we combine raw event logs with a compact session summary. At the top of each session sheet, main statistics occupy the fifteen rows. Rows 17-25 contain the final statistics collected at the time of the recorded "session_end" event. Starting on row 27, the detailed event log is laid out in a table.

We maintain a single .xlsx template file in our project root that defines all sheet names, header styles, and column layouts. When an admin clicks the "Download Excel File" button on the User Analytics tab we created (seen in Figure 5 below), the Excel controller (Excel.php) loads this template via PHPSpreadsheet, then invokes functions in the Analytics_model to calculate every value. The controller writes those values into the corresponding cells on the Summary and Users Summary sheets, clones and renames the session worksheet for each session ID, and streams the populated Excel workbook back to the browser.

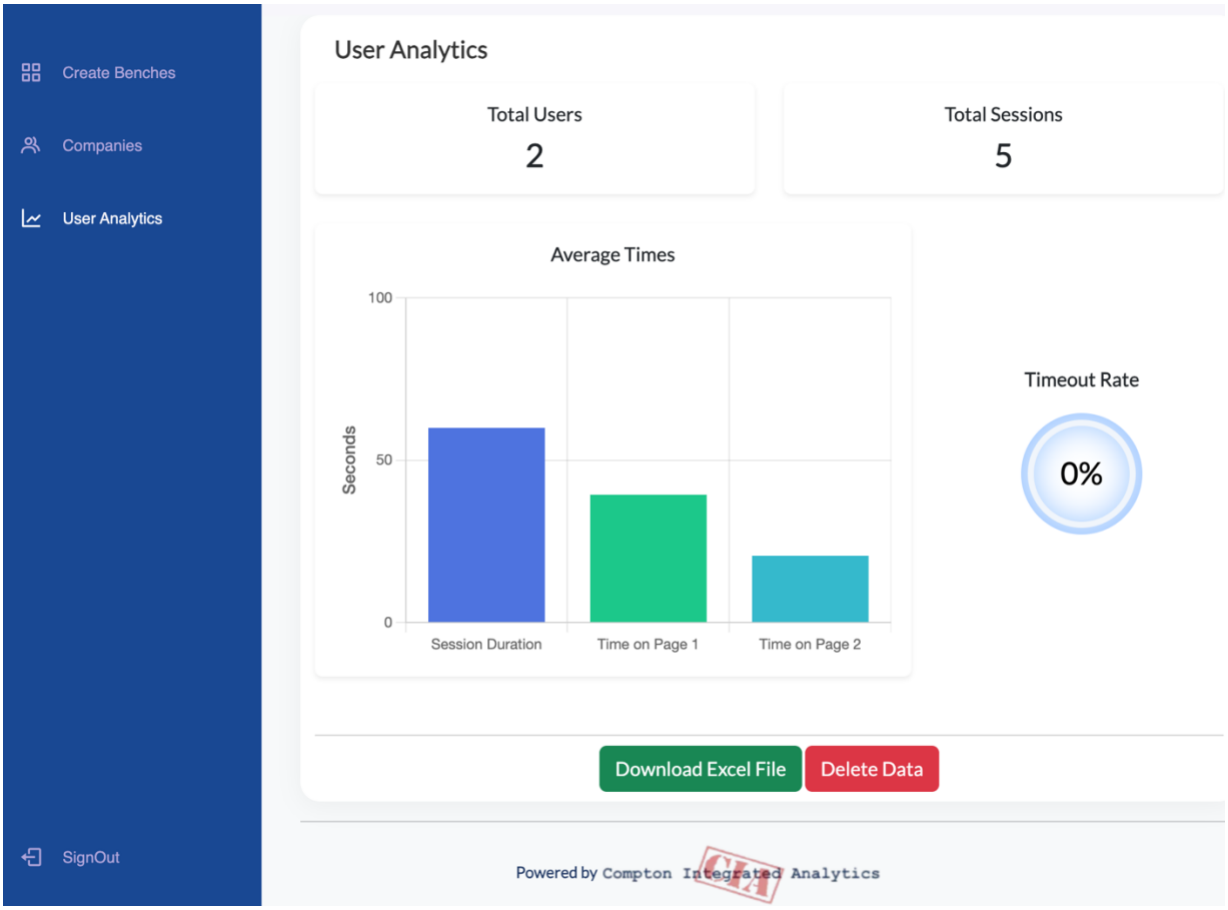


Figure 5. Screenshot of the User Analytics tab for MiLo Beta Web Application

The User Analytics tab, seen in Figure 5 above, is only viewable to admin users. This tab provides real-time metrics, interactive visuals, and a one-click data export in a single screen. In the Useranalytics controller (application/controllers/Useranalytics.php), we load the Analytics_model to compute all required statistics (total users, total sessions, average session duration, per-page times, timeout rate, etc.) and package them into a $metrics array. Those metrics are then passed to the useranalytics view (application/views/useranalytics.php) to be displayed. Below the visuals, are the “Download Excel File” and “Delete All Data” buttons. The “Delete All Data” button, with user confirmation, provides a simple way for the admin user to clear all collected data. Layouts, font sizes, and color contrasts were chosen to ensure the analytics interface is clear and accessible for all administrators.

Technical design issues:

- Missing Database: During initial setup, the repository lacked the SQL dump defining the application's schema and seed data for the “companies” and “users” tables. Without this, it was impossible to instantiate the relational database and use the site locally. To resolve this, we obtained the missing SQL file from the client, imported it into our local MySQL instance, and successfully populated all the required tables.
- Unreliable Session-End Detection: Accurately determining when a user's session terminates was a significant challenge. Relying on browser lifecycle events (*beforeunload/unload*) is not 100% reliable because many browsers throttle or suppress these events in background tabs, during rapid navigation, or when the application is backgrounded on mobile devices. To mitigate this, we use navigator.sendBeacon() to transmit a "session_end" event payload at page unload, although it still cannot fire under all circumstances (sudden network loss, force quit). Due to browser and network variability, no solution we found can guarantee 100% accuracy in tracking these exits.

VII. Software Test and Quality

We launched the Wine Rack page in Chrome (via our local XAMPP/phpMyAdmin environment) by clicking the "Create Benches" button, opened DevTools (Console), then exercised three scenarios: submitting with zero added benches, submitting after adding two benches, and submitting after adding two but removing one. In each case we watched for our console debug log and then queried the user_events table in phpMyAdmin for the latest button_click record. Tools utilized were Chrome DevTools, phpMyAdmin for SQL validation, and manual form interactions.

Functional Requirements:

A. Accurately track the amount of time a user spends on each page and what modifications they make to the initial Wine Rack View.
   a. Purpose: Ensure the time spent on each page and the timestamps of each interaction are logged with 100% accuracy for every session.
   b. Description: Carry out a predetermined list of actions and then look at the DevTools and phpMyAdmin table to see if all the interactions were logged.
   c. Tools utilized/required: Chrome DevTools, phpMyAdmin
   d. Threshold for acceptability: Every log must include an accurate timestamp, session ID, event type, and detail if needed.
   e. Edge Cases: Rapid successive inputs, leaving the page
   f. Results: In all scenarios, the "button_click" events logged correctly; Rapid-fire bench additions still generated distinct timestamps; A browser close without logout did not fire *beforeunload* in Chrome, but was fired upon reopening the site with same browser.

B. Format the data for easy interpretation and store it in a way that can be downloaded/accessed by administrators.
   a. Purpose: Make sure the data is formatted in the way that was discussed with our client
   b. Description: Press the download button on the user Analytics Page and ensure the resulting excel file is formatted correctly
   c. Tools utilized/required: User Analytics Page, Excel
   d. Threshold for acceptability: All data must be in the correct place and no additional data is present where it should not be.
   e. Edge Cases: No user data exists, incomplete data, when the dataset gets very large
   f. Results: Client confirmed layout was ideal and easy to interpret; All data represented correctly in each scenario

C. Charts to help visualize the data should generate automatically when the data is downloaded.
   a. Purpose: Ensure the spreadsheet contains accurate visuals upon every download.
   b. Description: Open the spreadsheet and inspect the embedded visuals to ensure accuracy.
   c. Tools utilized/required: User Analytics Page, Excel
   d. Threshold for acceptability: All visuals must be 100% accurate
   e. Edge Cases: empty/incomplete datasets
   f. Results: Full exports populated all cells correctly

D. Only admin-level users should be able to access this function, and it must be in a logical place on the website.
   a. Purpose: Make sure only the admin level users can access the user analytics for the site.
   b. Description: Log on to the site with both admin and non-admin users and make sure the User Analytics page only appears in the sidebar for admin users.
   c. Tools utilized/required: XAMPP to run the site
   d. Threshold for acceptability: no users without admin designation should be able to access the User Analytics page
   e. Edge Cases: no role assigned
   f. Results: Admin-only access has been confirmed; the User Analytics tab on the sidebar is only visible to admins

E. Each user session should have a unique id associated with all logged data.
   a. Purpose: Ensure each UUID is correctly assigned.
   b. Description: Open the website locally using multiple browsers/incognito tabs and look at the phpMyAdmin table to see the ID assigned to each.
   c. Tools utilized/required: phpMyAdmin, Chrome Developer Tools (DevTools)

d. Threshold for acceptability: All event logs from a single session must have the same UUID, sessions on different browsers must have different UUIDs
e. Edge Cases: multiple tabs open on the same browser, user clears browser data mid-session, the session expires
f. Results: Session IDs remained consistent per tab. Clearing localStorage mid-session generated a new anon_uid and session_id, as designed.

Non-Functional Requirements:

A. The data collection must not significantly impact MiLo's runtime and should support concurrent user sessions.
   a. Purpose: Validate that the logging of the data does not significantly impact MiLo's runtime.
   b. Description: Use the site and have the Chrome DevTools open to observe the live logging of events to see if there is any change in performance
   c. Tools utilized/required: Chrome DevTools
   d. Threshold for acceptability: any delay should be less than 1 second of wait time
   e. Edge Cases: concurrent actions, poor internet connection
   f. Results: No UI freezes or dropped frames observed. Performance of website maintained during testing.

B. The code must be documented clearly for hand-off to Compton Integrated Analytics and follow modular design for easy future updates or bug fixes.
   a. Purpose: Ensure the code is clear and easy to maintain.
   b. Description: Review the code and add documentation wherever it is needed.
   c. Tools utilized/required: VSCode, GitHub
   d. Threshold for acceptability: What each file does should be clear and any complex functions should be accompanied by an explanation in the comments above it.
   e. Edge Cases: incomplete documentation of source code
   f. Results: Added documentation for all newly added code; reviewed code and ensured documentation was clear.

C. Data collection should function in all supported browsers and retain data if the user exits the session abruptly.
   a. Purpose: Ensure data is accurately collected regardless of what browser the user is on.
   b. Description: Test the website locally on multiple browsers and incognito tabs and have phpMyAdmin open to ensure data collection is successful.
   c. Tools utilized/required: multiple browsers, phpMyAdmin
   d. Threshold for acceptability: It must work on all browsers.
   e. Edge Cases: unsupported browsers?
   f. Results: All major browsers logged initial events; Safari suppressed before unload event

D. Admins must be able to locate and interpret exported files and visuals with ease.
   a. Purpose: Make sure our additions to MiLo are easily accessible and usable.
   b. Description: Log into MiLo from an admin account and download and inspect the Excel file.
   c. Tools utilized/required: Excel
   d. Threshold for acceptability: Everything must be clearly labeled and 100% accurate.
   e. Edge Cases: admins that are unfamiliar with Excel
   f. Results: The User Analytics and download button are in an easy place to locate.

E. ADA compliant
   a. Purpose: Ensure the website is ADA compliant
   b. Description: Audit our additions to ensure they are compliant with the ADA.
   c. Tools utilized/required: XAMPP to launch the website locally
   d. Threshold for acceptability: must be compliant
   e. Edge Cases: lack of knowledge on ADA compliance
   f. Results: All buttons and chart elements are clear; color contrast ratios are high enough

VIII. Project Ethical Considerations

ACM 1.1 - Contribute to society and to human well-being, acknowledging that all people are stakeholders in computing.

“In addition to a safe social environment, human well-being requires a safe natural environment. Therefore, computing professionals should promote environmental sustainability both locally and globally.”

ACM 3.1 - Ensure that the public good is the central concern during all professional computing work.

“The public good should always be an explicit consideration when evaluating tasks”

We may be in violation of these parts of the ACM ethical code. Our project supports the development of a tool used for fracking; an activity widely criticized for its negative environmental impact. While the tool is designed to improve operational efficiency and data management, it ultimately contributes to a process associated with groundwater contamination, methane emissions, and long-term ecological damage.

As computing professionals, we have a responsibility to consider how our work affects both society and the environment. Even if our contribution is purely technical, we are still enabling an industry with significant ethical concerns. This raises important questions about the role of developers in shaping the consequences of technological progress, and whether the benefits of the tool outweigh the potential harm it supports.

ACM 2.6 - Perform work only in areas of competence.

IEEE 3.04 - Ensure that they are qualified for any project on which they work or propose to work by an appropriate combination of education and training, and experience.

Our team began this project with no prior experience in the web development technologies required, specifically PHP and JavaScript. While this initially presented a challenge, we have made a consistent effort to build our competence by self-teaching and applying what we’ve learned throughout the development process. However, we have not had access to guidance from experienced developers in these areas. To maintain transparency, we kept our client informed of our experience level and the steps we’ve taken to address this gap.

Although this situation places us outside full compliance with ACM 2.6 and IEEE 3.04, we believe our approach has been ethically sound. By acknowledging our limitations, actively working to overcome them, and keeping our client informed, we have done our best to meet the intent of these standards.

ACM 1.6 - Respect privacy.

IEEE 3.12 - Work to develop software and related documents that respect the privacy of those who will be affected by that software.

To respect user privacy while still collecting meaningful usage data, we implemented anonymous user IDs instead of gathering personally identifiable information. This allows us to track individual behavior without collecting names, emails, IP addresses, or other sensitive information.

While explicit consent would ideally be obtained, we believe this approach provides a responsible balance between functionality and privacy. It aligns with the ethical standards outlined by both the ACM and IEEE by minimizing risk to users and respecting their right to remain anonymous.

IX. Project Completion Status

The analytics feature set for the MiLo Beta program is largely complete and meets the core goals defined at the beginning of the project. We successfully implemented a system that tracks user interactions, logs them as a structured database, and presents the information in a clear and accessible format.

A new table was added to the existing database to store user sessions, each associated with an anonymous identifier. The identifier is currently based on the user's browser, which was the only method we could find that ensured full anonymity. While we originally planned to associate data with user login IDs, we were unable to do so without compromising anonymity. Future work may find a better balance between traceability and privacy.

We developed a dynamic Excel export system that creates a separate tab for each user session and includes detailed metrics such as time spent on each page and all logged interactions. The structure and formatting of this file match the template shown previously.

In addition to the downloadable data, we created a new admin page that displays high-level session statistics. This includes total users, total sessions, and a chart comparing the average time spent on each page with the overall session duration. There is also a bubble indicator showing the percentage of sessions that ended due to timeouts. As requested, we added a button on this page that allows administrators to clear all collected data from the database. This button also requires confirmation to ensure data isn't accidentally deleted.

Some features requested by the client, particularly more advanced analytics such as per-field interaction tracking and refined session timelines, were not implemented due to time constraints. These features are discussed in detail in the Future Work section and could be added with further development.

Overall, the core functionality of the system is complete, stable, and performing as intended. The system collects meaningful user metrics, maintains anonymity, and delivers data in a format that is both usable and extensible.

X. Future Work

Adding More Analytics

There were several analytics features requested by our client that we were not able to implement in this phase of the project. These include tracking the time from page load to a user's first interaction, as well as measuring when a user starts and stops interacting with each input field such as Frac Area Name, Height, Width, and Depth. Additional data points like the number of benches present at each point the "Add More" button is clicked, timestamps for each of those clicks, and the total time between the first bench input and the final download would all provide more detailed insight into user behavior. These features would require further work in both JavaScript and PHP, along with adjustments to how events are logged and stored in the database. Estimated time: 15-20 hours.

Fixing Bugs on the Original Site

Some bugs persist in the base version of the site that should be addressed to ensure consistent user experience. One notable issue involves the Wine Rack View, where certain layout intersections cause the bottom-most bench to be repositioned incorrectly. These bugs will require careful debugging within both the front-end and back-end. Familiarity with the CodeIgniter PHP framework, JavaScript, and CSS layout behavior will be essential. Estimated time: 10-15 hours depending on the depth of investigation required.

Implementing a Save Feature for the Wine Rack View

A save feature would allow users to preserve their bench placement and other changes in the wine rack view between sessions. This would improve usability and allow for more flexible workflows. Implementing this would require storing user state either in the session, a database, or local browse storage, and restoring that state upon page load. It would also require associating saved data with unique user or session IDs. Experience that would help with this enhancement would be experience with persistent storage techniques and session management in PHP and JavaScript. Estimated time: 20-25 hours.

Turning the Excel Document into a Page on the Site

Currently, the analytics are only available via a downloadable Excel file. A more accessible approach would involve turning this data into a browsable page within the admin interface of the website. The file could still be available for download, but the addition of a

view on the site would streamline usability for the client. This would require rendering structured data as an HTML table or embedded chart and possibly integrating a JavaScript charting library for visual representation. Experience with front-end development, table rendering, and use of the PHPSpreadsheet library would be helpful. Estimated time: 15-20 hours.

XI. Lessons Learned

Throughout this project, our team encountered a wide range of technologies, tools, and professional standards that were unfamiliar to us at the beginning of the project. Each one presented a learning curve, and through research, experimentation, and problem-solving, we gained valuable insight into how these pieces fit into a professional software development process. The lessons we learned covered technical implementation, local server environments, and the ethical responsibilities that come with engineering work.

One of the biggest areas of growth was in web development. At the beginning of the project, we had no experience with PHP, but we quickly learned to use it for server-side scripting, session handling, and interaction with databases. Writing PHP code that was both functional and maintainable took time, especially as we worked to integrate it with front-end tracking features. We had to adapt to a PHP-based development environment, including routing and the MVC structure used in the CodeIgniter framework.

A major milestone in the project was learning to use PHPSpreadsheet, a library for generating Excel files through PHP code. This was essential for meeting the client's requirement to export user interaction data in a format that administrators could download and interpret easily. Working with PHPSpreadsheet involves learning how to format tables, write files, and ensure that the generated files contain accurate and readable information.

We also gained a deeper understanding of JavaScript. Beyond basic interactivity, we used JavaScript to capture and timestamp user actions like button clicks and text input. This taught us a lot about browser event handling, the asynchronous nature of JavaScript, and how to pass information cleanly between the client and server sides of the application.

Another key area of learning involved working with a local server environment. We used XAMPP to host the website locally and test all new features before they went live. This required us to configure Apache and MySQL, and it gave us a better understanding of how server infrastructure supports the software we build. In the process of testing and refining the application, we learned how to create and import databases using phpMyAdmin in XAMPP. We also became more comfortable with SQL and the structure of relational databases. This gave us insight into how database design impacts performance, scalability, and accuracy of data collection.

We also had to become familiar with the IEEE and ACM ethical guidelines. Because our software collects user interaction data, we were responsible for ensuring that data collection was done ethically and with consideration for privacy and consent. We learned about anonymizing data, limiting collection to only what is necessary, and avoiding practices that could lead to identifying individual users.

At the start of the project, we underestimated how difficult the entire task would be. We assumed it would be fairly simple to build a data tracking system, but the reality was much more complex. Even defining what we needed to build required more research and planning than we expected. This underestimation is also related to our ethical responsibilities. According to both the IEEE and ACM code of ethics, engineers have a duty to assess the risks, feasibility, and implications of what they are building before making promises or commitments. We now understand that claiming something will be easy without properly researching is a mistake and a failure to meet professional standards.

In summary, this project gave us hands-on experience with new tools, real world software development, and ethical responsibilities that come with building systems for other people. We learned how to navigate technical challenges, ask the right questions, and approach our work with more care and professionalism. These are lessons we will all carry forward into future projects and careers.

XII. Acknowledgments

First and foremost, our team would like to express our sincere gratitude to our client, Sarah Compton. Her enthusiasm, support, and availability throughout the project made her an exceptional client and collaborator. She consistently offered valuable feedback, took time to meet with us regularly, and brought a genuine excitement to every interaction. Working with her has not only been professionally rewarding but also personally enjoyable. We truly could not have asked for a better partner for this project.

Additionally, we would like to thank OpenAI for providing access to ChatGPT. This project required us to work with technologies and code that were entirely new to us, often without any documentation or access to the original developers. In the absence of formal guidance, ChatGPT served as an invaluable resource, helping us understand unfamiliar concepts and troubleshoot issues as they arose. Its support played a key role in allowing us to bring this project to completion.

XIII. Team Profile

Doug Meany is a computer science major with a focus area in data science, originally from Fairfield, Connecticut. He completed a data analysis internship in Tokyo, where he applied statistical tools to real-world problems. Outside of work, Doug enjoys playing Dungeons & Dragons, hitting the trails for a good hike, and getting lost in story-driven video games.

Emily Marks is a Computer Science major at the Colorado School of Mines, specializing in Robotics & Intelligent Systems. She's originally from Vero Beach, Florida, but has lived in Colorado since 2017. Emily has hands-on experience in technical and customer service roles in both manufacturing and retail. Outside of coding, she's a former hula-hoop dance performer and instructor, and hula-hooping remains her favorite hobby.

Kate Callan is a Computer Science major currently on the general CS track, originally from Dallas, Texas. She has worked at the Registrar's Office for the past year, where she handles sensitive student data across multiple databases, updates academic records, and helps maintain the integrity of institutional systems. Outside of school she enjoys crocheting, watching F1 and being in nature any chance she gets.

References

[1] ChatGPT, response to author query. OpenAI [Online]. https://chatgpt.pro/ (accessed May, 2025).
[2] “PHP: Hypertext Preprocessor.” https://www.php.net/manual/en/ (accessed May, 2025).
[3] “DevDocs — JavaScript documentation.” https://devdocs.io/javascript/ (accessed May, 2025).
[4] “Getting Started with PHPUnit 12.” https://phpunit.de/getting-started/phpunit-12.html (accessed May, 2025).

Appendix A – Key Terms

Term	Definition
MiLo	*The name of the beta program software*
Wine Rack	*The bench and frack areas within it, the ovals in the long rectangle resemble a wine rack on our diagram*
Bench	*A horizontal strip of land with a height that’s the diameter of the fracking area a width of one mile on our diagram. There can be multiple benches on the same plot of land They can hold as many fracking areas as needed*
Frac Area	*The elliptical drilling area with the well in the center*