



COLORADO SCHOOL OF MINES
EARTH • ENERGY • ENVIRONMENT

CSCI 370 Final Report

Team CEA - Collaborative Engineers Anonymous

Diana Rogozyan
Jakob Walker
Nina Winslow
Sophia Mulholland

December 6th, 2024



CSCI 370 Fall 2024

Kathleen Kelly

Table 1: Revision history

Revision	Date	Comments
New	8/23/24	Added the first 5 sections.
Rev – 2	09/11/24	System Architecture section added.
Rev – 3	10/16/24	Added Software Quality and Ethical Considerations.
Rev – 4	11/08/24	Added Project Completion Status, Future Work, Lessons Learned sections.
Rev – 5	12/3/24	Updated Paper based on feedback from other teams.

Table of Contents

I. Introduction.....	2
II. Functional Requirements.....	3
III. Non-Functional Requirements.....	3
IV. Risks.....	3
V. Definition of Done.....	4
VI. System Architecture.....	4
VII. Software Test and Quality.....	7
VIII. Project Ethical Considerations.....	11
IX. Project Completion Status.....	15
X. Future Work.....	16
XI. Lessons Learned.....	17
XII. Acknowledgments.....	17
XIII. Team Profile.....	18
References.....	18
Appendix A – Key Terms.....	19

I. Introduction

The Colorado Education Association (CEA) is a teacher’s union dedicated to providing the best public education for every student while ensuring educators are respected and supported [1]. The CEA is creating a website that allows educators to measure how efficiently their time is being used and whether they are being overworked or underpaid. Previous students in past field session teams have created the base website and rudimentary functionality of the survey used to collect data, along with the data visualization. The primary goal of this field session is to update this website to ensure that users can properly input their data and see visualizations of said data compared to other users. This website must be functional for a live demonstration at a teachers conference in November.

All data used on the website is created by users completing surveys. Users of this site will be teachers, educators and administrators. Super users of this site will be employed by CEA to ensure that the website runs smoothly and is maintained.

II. Functional Requirements

Below is listed the technical requirements that the team completed over this semester. These features are implemented to the site where the user privileges range from user to super admin.

- Update live website to pull in and understand most recent code
- Update data comparison page to display information efficiently
- Update super admin tab to pop up on every page on website
- Create system that allows users to reset their password if they have forgotten it
- Add multi user access and different experiences based on user level of access
- Add an overview of all aggregated data for higher level users
- Stretch Goal: Update design of web page for user ease
- Stretch Goal: Create clear data visualization for presentation to users

III. Non-Functional Requirements

Below is listed the non-technical goals the team attained over the semester.

- Finish functional website by November 13th for presentation at the bargaining conference
- Modify the site to be more user friendly, with educators in mind

IV. Risks

Below is listed the project risks that the team evaluated and prepared for.

- The website is not live by the date of the presentation
 - Likelihood: Unlikely
 - Impact: Major
 - Risk Mitigation Plan: In the event that the website is not functioning properly, the team has allotted two days before the presentation to create a working state
- The website is live but contains problematic errors by the date of the presentation
 - Likelihood: Unlikely
 - Impact: Major
 - Risk Mitigation Plan: In the event that the website is not functioning properly, the team has allotted two days before the presentation to create a working state
- User data is not stored properly or securely
 - Likelihood: Unlikely
 - Impact: Moderate
 - Risk Mitigation Plan: Update the already existing database to be more secure in data storage using hashes and other encryption methods.

V. Definition of Done

The project will be considered complete when the live website operates in an error-free state with the most up-to-date code and fulfills the core requirements for functionality. By the end of the project, the primary consumers, including teachers and a “super admin,” will be able to seamlessly access and utilize the website. Specifically, users must be able to create an account, complete a survey, view and print their data, compare their data, and reset their password.

To further clarify, the survey will be created by the CEA and integrated into the website. The survey will include a mix of multiple-choice, rating-scale, and short-answer questions tailored to gather data on user performance and experiences. It should take approximately 10–15 minutes to complete and will likely be completed once per semester or as determined by the CEA’s requirements.

The website will feature data analytics, including visualizations such as bar charts, line graphs, and comparison tables that allow users to analyze trends over time or compare their results with aggregated data from other users. These visuals will provide actionable insights based on the survey data, displayed clearly and interactively.

Lastly, the site’s functionality will be described by the order of the user’s journey. Users will begin by registering for an account and logging in. From there, they will access and complete the survey, view a dashboard summarizing their data, and, if they are a super admin user, compare their results with aggregated data from other users. Administrators and super admins will have access to additional features, such as managing user accounts and generating reports. A feature will be considered done if it functions correctly under the assumption of a smart user.

VI. System Architecture

Description of System Architecture (Figure 1)

The team has encountered several technical design issues during the project setup. Initially, the team struggled with setting up the environment due to outdated packages, particularly with Django. This has caused some delays, and the team anticipates further risks related to the continued availability of certain included libraries. These libraries have already contributed to live website run failures, meaning the team may need to explore alternate library options and potentially rewrite portions of the team's code. Additionally, during the setup process, the team identified interferences between libraries, where certain dependencies override others, leading to bugs. As a result, a thorough review of all included libraries is necessary to ensure stability moving forward. The current system includes foundational interfaces and components inherited from previous iterations, such as basic login and survey management features, but the team plans to expand functionality using new tools like React for frontend enhancements and Django REST Framework for backend APIs.

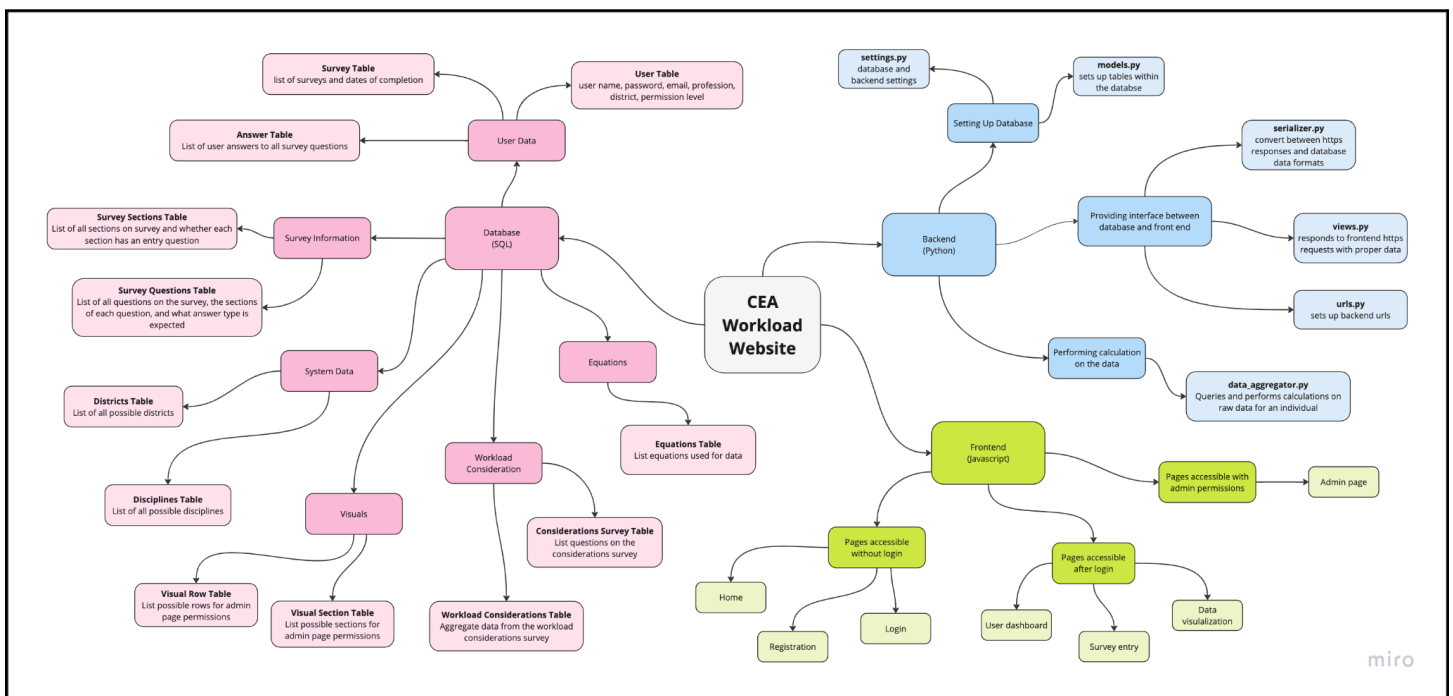
The frontend is written in JavaScript and includes the visual components for each web page. When the user is not logged in, the home page, registration page, and login page are accessible. When the user is logged in, the user dashboard, survey entry, and data visualization are accessible. Users with super admin permissions can

see the super admin page, which allows them to give permissions to others and aggregate data for the whole state, while users with just admin permissions can see a page to aggregate data from a district. Super admin permissions are granted manually by an existing super admin through the backend interface.

The backend is written in Python, providing an interface between the frontend and the database and performing calculations. The backend is set up using `settings.py`, which configures the database and backend settings, and `models.py`, which sets up tables within the database. An interface between the database and frontend is made with `serializer.py`, which converts between HTTP responses and data table formats, `views.py`, which responds to frontend HTTP requests with proper data, and `urls.py`, which sets up backend URLs. Calculations are performed on the data using `data_aggregator.py`, which queries and performs calculations on raw survey data for an individual.

The database uses SQL to hold all user and system information. Specifically, SQLite is used as the database management system for development, chosen for its simplicity and compatibility with the project's scope. The database includes user data tables, survey information tables, system data tables, and workload consideration data tables. A database schema will be provided to visualize these relationships and clarify how data is structured and linked, improving understanding among team members and clients.

Figure 1: Website Architecture Diagram



Description of Entity Relation Diagram (Figure 2)

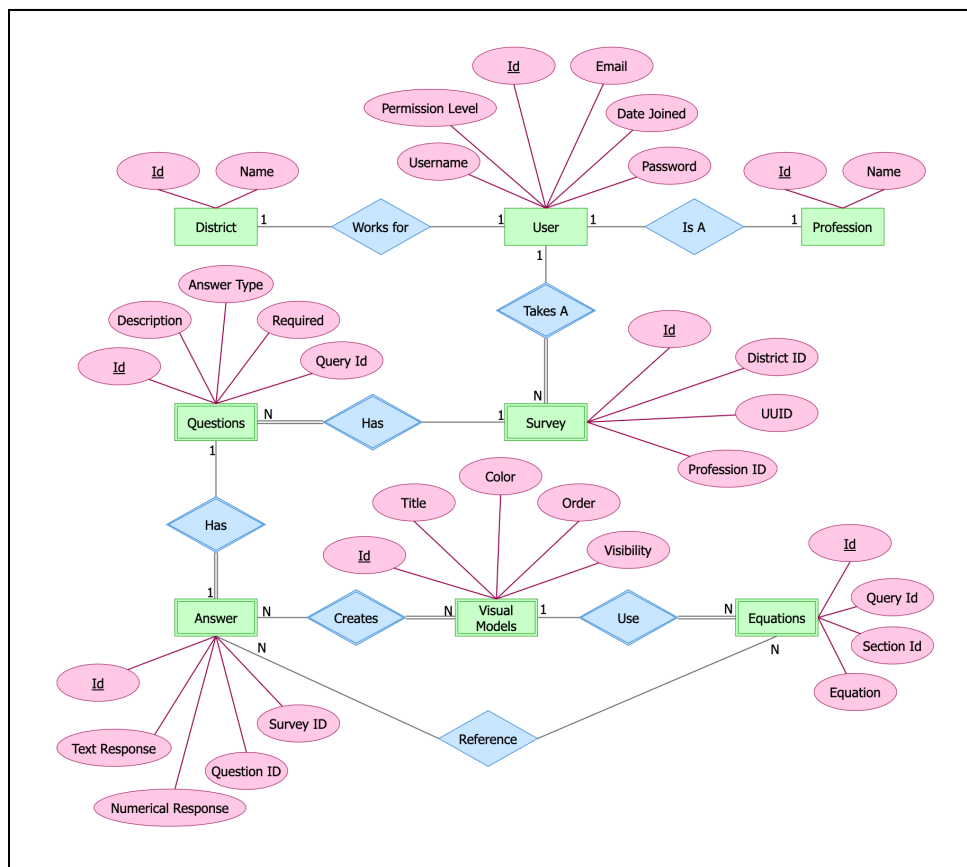
The database for the CEA product includes eight core tables with complex relations. These tables include District, User, Profession, Questions, Survey, Answer, Visual Models, and Equations. Each user created for this

site becomes an entry in the User table, with attributes Username, Permission Level, ID, Email, Date Joined, and Password. Each Password is stored in the table after being hashed and encrypted so the password is unrecognizable. Each User is a Profession, which has attributes ID and Name, and each User works for a District, which has attributes ID and Name.

The User will take a Survey, which has attributes ID, District ID, Unique User ID, and a Profession ID. Each Survey has multiple Questions, with attributes ID, Description, Answer Type, Required, and Query ID. Each Question has multiple Answers, with attributes ID, Text Response, Numerical Response, Question ID, and Survey ID. The Answer entries also create Visual Models, with attributes ID, Title, Color, Order, and Visibility. These Visual Models use Equations, which have attributes ID, Query IDs, Section IDs, and the actual equation associated with the row in the table. Equations may reference the answers the user provides.

The largest table in this database is the Answers table. It contains a large volume of information from other tables, including many row IDs. The responses from the user vary in size, so the table needs significant storage capacity for all stored responses. The entity relation diagram (ERD) is included to provide a visual representation of the database structure. It illustrates the relationships between tables and helps the audience, including team members and stakeholders, better understand the organization and functionality of the database.

Figure 2: Database Entity Relationship Diagram



VII. Software Test and Quality

Update Live Website to Pull In and Understand Most Recent Code Test Plan:

Purpose of Test: The purpose of this test is to ensure that the live website for the team's product is reflecting the most recent code changes.

Description of Test: Using Heroku, the host of the team's live website, the team will ensure that the git commit attached to the website deployment reflects the code that is most recent. The team will do this by cross-referencing the git logs with the heroku logs. The team will also do a user test to make sure that the newest updates are reflected in what the user sees on the site. To provide context, Heroku is a cloud platform used for deploying, managing, and scaling applications, making it essential to ensure the integration between it and the team's repo is seamless.

Tools Utilized / Required for Test: The team will need to use the Heroku CLI to pull and look at the most recent logs on the team's website to make sure there are no errors. The team will also need access to the team's Git repo to ensure that the commits tracked in Heroku are the most recent ones in the team's repo. Lastly, the team will need access to the live website to manually make sure that the team's code changes are visible on the site.

Threshold for Acceptability: The live website is reflecting the most recent changes.

Edge Cases: There is a chance that the heroku app maintenance may change the team's website configurations, which would bring the live website down again. To counteract this, the team plans to stay up to date with Heroku's maintenance schedule and try to change any hardcoded values to automatically change with the newest changes Heroku might make.

Results of Testing: The live website is working with all the newest code changes.

Update Super Admin Tab to Pop Up on Every Page on the Website Test Plan

Purpose of Test: The purpose of this test is to ensure that the Super Admin tab is consistently available on every navigable page for users with Super Admin privileges. This test ensures that the Super Admin tab is visible only to Super Admin users and is accessible across all pages of the site. The test also ensures that the tab remains functional and does not impact site performance or usability.

Description of Test: To ensure this functionality, the team will start by logging in as a Super Admin and navigating through each page on the site to confirm the visibility and functionality of the Super Admin tab. Additionally, the team will log in as users without Super Admin privileges to verify that the tab is not visible for these users. If time permits, feedback from the stakeholders will be gathered to assess the effectiveness and accessibility of the tab placement. This feedback will look like a one on one session with the stakeholder where they test out the product and give pointers on what they like and dislike.

Tools Utilized / Required for Test: The team will need to use a development server with the website and Super Admin privileges enabled and multiple user accounts with varying privilege levels (Super Admin and non-Super

Admin). Additionally, the team will need different web browsers for manual testing, including Chrome, Firefox, and Safari, to ensure cross-browser compatibility, as well as browser developer tools to monitor errors, assess performance, and confirm tab visibility across all navigable pages.

Threshold for Acceptability: The threshold for acceptability is that the Super Admin tab is consistently visible and accessible on every navigable page only for Super Admin users. This feature must not be accessible to users without Super Admin privileges. The tab should function properly across all pages without affecting page load times, layout, or usability. The placement should align with stakeholder requirements and meet the expectation that the Super Admin tab is always visible and intuitive to locate for authorized users.

Edge Cases: A user without Super Admin privileges attempts to access or interact with the Super Admin tab. The system must ensure that the tab is not visible to these users on any page. The Super Admin tab must be available on all supported browsers. The team will test on various browsers to ensure the tab remains visible and functional. Testing will include rapidly switching between pages or navigating to rarely accessed pages (e.g., error pages or archived sections) to confirm that the Super Admin tab remains available and functional.

Results of Testing: The Super Admin tab is only accessible to a user with super admin privileges and is visible from every page.

Update Data Comparison Page to Display Information Efficiently Test Plan:

Purpose of Test: This test is designed to ensure that data is correctly being displayed for the user to read and understand. This includes two sided pages with the numbers of each entry and calculation, and the data visualizations of the entries and calculations.

Description of Test: This test will involve double checking that the data and data visualizations match what should actually be displayed. User responses to the displayed data will also be accounted for to see which areas need to be improved visually.

Tools Utilized / Required of Test: This test needs real users to be able to view and give feedback. This feedback will look like users completing a survey after viewing and using a test version of the site to explain what they could and couldn't understand on the data comparison page. This test also needs to see the correct output for a specific set of data to ensure data is properly being handled and displayed.

Threshold for Acceptability: The Data Comparison page should display and communicate the data in an efficient and effective manner. Data should have the proper units, and should be formatted properly on the page so there is no text overflow. Users should be able to understand how to interpret the data that is shown when the data is being compared.

Edge Cases: Users may input data that is negative or extremely large. This will impact the accuracy of the data shown , and the graphs may not be properly processed.

Results of Testing: The data comparison page correctly displays information.

Create System that Allows Users to Reset Their Password Test Plan:

Purpose of Test: The purpose of this test is to make sure that the “Forgot Password” feature is working as expected. Users should be able to click on a “Forgot Password?” link from the login page, where they can then enter their email address and receive an email with a link to reset their password.

Description of Test: This test will rely heavily on manual user testing to make sure that the flow of the feature is correct and that the wording on the email works. The team will also have additional unit tests and code reviews to make sure that the code itself is readable and working as expected.

Tools Utilized / Required for Test: The main tool in the team's testing is the feedback from the users who test the feature. This feedback will look like answers on a survey the user will take after testing the feature with what they liked and disliked about the feature, along with what did or didn't make sense with the feature. Additionally, the team will use the team's trello board to keep track of code reviews and any comments the team might have on the code changes required for this feature.

Threshold for Acceptability: The user is able to click a “Forgot password” link from the login page, enter their email address, receive an email to reset their password if their email address is connected to an account, and click a link in the email to reset their password. Lastly, the user should be able to login with their new password.

Edge Cases: It is possible that the user forgot their email address or entered a wrong email address when creating their account. In these cases, the team will have to resort to those users contacting the super admin, as the super admin can see all the users in the system and will be able to access the account information needed to reset the password.

Results of Testing: Users are now able to reset their password.

Add Multi-User Access and Different Experiences Based on User Level of Access Test Plan:

Purpose of Test: This test is to ensure that admins and super admins can access different areas of the website and data depending on the user's level of access. This is to ensure that the correct level of access is being granted, as well as the specific experiences that come with the level of access.

Description of Test: Various “dummy” accounts with different levels of access will be used to ensure that each user level is able to access the correct areas of the website. These dummy accounts will also be used to see if the correct data is being displayed depending on the account's level of access. As multi-user access presents a significant challenge in QAQC, the team acknowledges the inherent difficulty in covering all edge cases for a system with such varying permissions.

Tools Utilized / Required for Test: This test requires the creation of three different dummy accounts that will be used by the team. This is to ensure that the normal user, admin user, and the super admin user can only view the necessary areas of the website.

Threshold for Acceptability: This test is acceptable if only the correct access and experiences are granted to the correct level of user. Super admins and admins should be able to access admin-only pages and see an overview of data from regular users.

Edge Cases: An edge case would be if a user is seeing areas of the website that are not granted by permissions of the user's current level of permission. An example would be if a regular user is able to access an overview of each districts' data, which is an experience that should only be granted to admin users.

Results of Testing: Different users correctly have different levels of access, and can only view pages and utilize functionalities according to their permission level.

Add an Overview of all Aggregated Data for Higher Level Users Test Plan:

Purpose of Test: The purpose of this test is to ensure that the aggregated data overview functionality provides higher-level users (e.g., administrators or district managers) with accurate and visually comprehensive summaries of survey entries. The test ensures that the data is aggregated correctly, visualizations are clear and useful, and that users can utilize this data for presentations, union meetings, and informed decision-making on educator information across various districts and regions.

Description of Test: This test will begin by verifying the accuracy and comprehensiveness of the aggregated data. The team will first confirm that the data for all user surveys is collected and accurately aggregated, then displayed through visualizations (charts, graphs, comparisons). Next, the team will check the user interface to ensure that higher-level users can easily access and interpret the data. After that, the test will evaluate the responsiveness and performance of the overview under different loads and across different devices. Stakeholder feedback will be collected, if time permits, to ensure that the visualizations meet user expectations and requirements for presentations and meetings.

Tools Utilized / Required for Test: Development server with a populated database of survey entries. Access to accounts with higher-level user privileges to access aggregated data. Visualization libraries (e.g., Chart.js, D3.js) to confirm the correctness of generated charts and graphs. Web browsers and testing tools to assess responsiveness, load times, and performance across Chrome, Firefox, and Safari. Browser developer tools and monitoring tools to assess data loading times and optimization.

Threshold for Acceptability: The threshold for acceptability is that the aggregated data overview is accurate, easily interpretable, and displays all required information on user surveys. The data visualizations should be clear, useful, and available to higher-level users on demand. This feature must work on all supported browsers and devices, displaying the aggregated data with no noticeable delay. The visualizations must provide meaningful insights and comparisons that align with stakeholder specifications and can be used effectively in presentations, union meetings, and district evaluations.

Edge Cases: Testing with both minimal and maximal sets of survey data entries to confirm that the aggregation and visualization processes remain performant and accurate regardless of data size. Ensuring that visualizations are accessible, with alternative text or descriptions, so that higher-level users with disabilities can still interpret the data. Testing access to the aggregated data overview from both higher-level user accounts and regular accounts to confirm that only authorized users can view this data. Ensuring the aggregated data overview and visualizations are compatible across multiple browsers and devices, with no loss of functionality or usability. Ensuring that any updates to survey data entries are reflected in real-time or near-real-time within the aggregated data overview.

Results of Testing: Admins are able to see aggregated data.

VIII. Project Ethical Considerations

ACM/IEEE Ethical Principles Relevant to the Project

This project involves accessing and managing personal user data, making it critical to adhere to the highest ethical standards outlined by ACM and IEEE. Below, we highlight the relevant principles and explain their importance to the project.

Relevant ACM Principles

1.6 *Respect Privacy*

The team must respect user privacy by ensuring that all user data is handled responsibly and protected against unauthorized access.

1.7 *Honor Confidentiality*

Any sensitive information must remain confidential and only be accessed or shared when necessary and with proper authorization.

2.1 *Strive to achieve high quality in both the processes and products of professional work*

This project should meet high-quality standards, ensuring robust and reliable systems that serve user needs effectively.

2.9 *Design and implement systems that are robustly and usably secure*

Robust security measures must be implemented to prevent unauthorized access and data breaches while maintaining usability.

3.6 *Use care when modifying or retiring systems*

Since this project builds upon work from previous teams, care must be taken to avoid introducing issues or breaking existing functionality.

Relevant IEEE Principles

1.03 *Approve software only if they have a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish quality of life, diminish privacy or harm the environment*

All software updates should prioritize user safety, privacy, and overall public benefit, aligning with specifications and rigorous testing.

2.01 *Provide service in their areas of competence, being honest and forthright about any limitations of their experience and education*

It is essential to communicate the team's knowledge limitations to stakeholders and seek help for areas beyond the team's expertise.

3.10 *Ensure adequate testing, debugging, and review of software and related documents*

Thorough testing, debugging, and reviews are required to deliver a reliable and error-free product.

3.14 *Maintain the integrity of data, being sensitive to outdated or flawed occurrences*

The team must address outdated or inaccurate data to ensure integrity and reliability.

8.01 *Further their knowledge of developments in the analysis, specification, design, development, maintenance, and testing of software*

This project offers an opportunity for the team to grow as software engineers and expand their technical expertise.

Explanation of Importance

Given that this project deals with user data, the team must respect privacy and confidentiality, ensure system security, and maintain data integrity. Building upon an existing project requires careful consideration to avoid disrupting previous functionality while achieving project goals. Additionally, as students, the team is committed to learning and applying these principles to improve their skills and deliver a quality product.

Most Vulnerable Ethical Areas

Certain areas of these principles are more vulnerable to violations:

ACM 1.6 *Respect Privacy*

Protecting user data is a major concern. Improper handling could lead to breaches, exposing sensitive information to malicious actors.

ACM 1.7 *Honor Confidentiality*

Maintaining confidentiality ensures user trust, and any lapses could result in legal and reputational damage.

IEEE 1.03 *Approve software only if it meets specifications and does not diminish privacy*

Any failures in this area could lead to loss of user trust, legal consequences, and diminished reputation.

IEEE 3.14 *Maintain the integrity of data*

Failure to ensure accurate and reliable data can compromise the system's effectiveness and user trust.

Consequences of Violations

Violating these principles could lead to:

- *User Trust Issues*: Breaches of privacy or confidentiality could deter users from engaging with the system.
- *Legal Ramifications*: Mishandling sensitive data could result in lawsuits or regulatory penalties.
- *Reputational Damage*: A flawed or insecure system could tarnish the reputation of both the project and its contributors.
- *Loss of Future Opportunities*: Employers may hesitate to trust individuals associated with a compromised project.

Michael Davis Tests

Harm Test

The team had to make a difficult choice when the team tackled the first bug with the team's product: a live website loading issue. This was the top priority bug that the team's group was tasked to fix. The live website deployment was continuously failing due to a specific library being used in a new admin-user feature, and the team's group spent hours trying to understand what was going on and how to fix it. After several weeks of having no answer, the team had to weigh the benefits and costs when deciding how to move forward with it. The team's last resort was to remove the latest admin-user feature completely, after carefully thinking how this might impact the team's product.

The benefits:

- The team would have time to work on the major bugs on the website in time for a round of beta testing.
- The team would get more experience with software engineering itself, rather than focusing the team's efforts on researching python libraries.
- The client would be able to give more feedback on bugs and features with access to the live website.
- Regular users would not be impacted by the removal of the feature.

The costs:

- This would be a “band aid” fix for the things that might be going wrong underneath the layers of code.
- The team loses an admin-user feature that is important to how data aggregation works in the system.
- The team might not have time to fix this feature or rebuild it later in the semester.

After talking to the team's client, the team came to the conclusion that the team needed to remove the admin feature that was causing the problem with the live website so that the team can shift the team's priorities. To the client, it was more important that the team's group got the website ready for beta testing and live demonstration, rather than fixing difficult bugs in admin-user features that only got used by 1-2 people. Though this wasn't a perfect fix, removing that feature had more benefits than costs.

Mirror Test

Another issue the team's group had to fix was implementing a user verification system. This user verification system would only allow for real users to access the website. The team's group and the team's client felt this was an important issue to fix so that only real users can input their data into the website. When admins and super admins see an overview of the aggregated data, only real people's data should be shown to allow these higher leveled users to understand where some districts may be overworked. The data that is going to be collected by the team's client is important since the Colorado Education Association is a teacher's union, and needs this data to help support teachers.

The mirror test is if the team's group would feel proud of the changes the team made. the team's group does feel that these changes are important to the team's client, and everyone who uses this website. If the team's group did not implement these precautions, the team's client may face large inputs of fake data and may not even be aware of what data is from a real user and a fake user. This means the team's client would be making real decisions off of skewed data. Preventing fake users from accessing this website makes the team's group feel proud that the team helped create a more secure website.

Ethical Considerations

If the software quality plan is not properly implemented or comprehensive, several ethical issues could arise. If the software quality plan is not implemented properly or lacks comprehensiveness, it could lead to serious ethical issues impacting user privacy, trust, and data reliability. Insufficient testing may expose vulnerabilities that put sensitive user data at risk, violate privacy, and potentially lead to unauthorized access to restricted information. Inadequate accessibility and testing can also result in discrimination against users with disabilities, creating unequal access to the tool. Without rigorous data aggregation and visualization checks, there is a risk of misrepresenting information, which could lead to misguided decisions, especially for stakeholders relying on accurate data for presentations and meetings. Additionally, compromised software quality can erode user trust and stakeholder confidence, leading to dissatisfaction and possibly undermining the tool's intended purpose.

Failure to meet data protection regulations or ensure reliable access control may result in legal and compliance issues, which could bring regulatory penalties and damage the organization's credibility. Such issues can also affect the organization's social impact, as poorly functioning software may hinder NGOs and educational organizations in delivering vital services. This, in turn, misuses resources and diverts attention from essential initiatives, ultimately diminishing the positive influence these organizations aim to achieve. Prioritizing a comprehensive software quality plan is essential to uphold ethical standards, protect user data, and maintain the trust and effectiveness required for meaningful organizational impact.

IX. Project Completion Status

The team successfully achieved the objectives they established for the CEA Workload Calculator. The team was able to clean up some of the code and overall look of the website to improve user functionality. The first task the team's group had was making the Workload Calculator website work. When the team was originally given this project, the website was only a blank white screen, but the team found the issue and debugged it to make the website work. The team also cleaned up the look of the website and fixed some minor bugs. The formatting on the menu bar is improved where the menu bar text is now easier to read and does not have a weird shift when the user hovers over the text. The team also fixed a printing issue on the data visualization page where the user can now correctly print the graphs shown on the page. The decimal points on the data visualization was also changed to be formatted better and more uniform. Timestamps were also added to surveys so when a user fills out multiple surveys in a day the user can differentiate between multiple surveys filled out on the same day. Some of the wording on the website was also rewritten to look more professional and relevant to the website.

The next main problem was fixing the survey when a user fills out a survey on the “Complete a Survey” page. When a user inputs an incorrect value, the box will now appear as red and communicate to the user what the correct bounds are. The bottom bar that represents what sections of the survey the user is on is formatted better and correctly shows what section has an incorrect input. The textboxes were also formatted better where the team removed a scrolling feature that would change the user’s answer.

Admin functionality was also improved. The admin pages were also updated to fix bugs that impacted the user experience on this page. When an admin views all admin permission requests the requests do not disappear and the admin is able to completely view all requests. The sorting feature in manage user for admin users is also more clarified and intuitive for the user.

There were two main features that the team was unable to implement. This includes a Captcha system and a data aggregation feature for admins. The Captcha system the team wanted to implement was a Captcha system using Google’s Captcha API. Data aggregation is a feature for admin users, which allows them to view the overall data from users who fill out surveys.

X. Future Work

Website Optimization and Discovered Bugs

The main priority for future work on this website is optimizing its overall performance and structure. On October 11th, an update to the Heroku system hosting the website caused it to stop functioning. The team believes this update impacted the website's memory usage, revealing that the site was not running optimally. Upon further investigation, the team identified several memory leaks and poorly written code.

Due to time constraints, the team was unable to conduct a comprehensive review of the entire codebase, as these issues were discovered midway through the field session. Future developers should prioritize a detailed examination of the code structure and setup to identify and resolve these issues. Addressing this problem requires a basic understanding of web development, including Heroku, JavaScript, and database management. With thorough documentation provided, the team estimates this task could be completed in approximately two weeks, ensuring the website remains functional and stable to support future developments. This optimization is crucial and should take precedence over the addition of new features.

Additional Security Implementations

One key feature the team recommends adding is a Captcha system to verify user authenticity. This security measure would prevent unauthorized access and ensure that only real users interact with the website. After researching potential solutions, the team identified Google's Captcha system as both cost-efficient and reliable. Implementing this feature should take around two weeks and requires a foundational understanding of cybersecurity and web development.

Stretch Goals and Ease-of-Use Improvements

Several stretch goals remain for future teams to explore.

One significant addition could be a pop-up tab on the survey page to assist users in filling out the questionnaire. This tab would provide unit conversions for survey responses, simplifying the process for users unfamiliar with the required units. Implementing this feature would improve the user experience and should take about one week. It primarily involves working on the website's user interface and requires knowledge of web development.

Another improvement is to ensure overall compatibility with different screen sizes. This applies to both mobile and desktop users, and would require extensive testing on several different devices. Due to the number of pages that are accessible on the site, the team estimates that this may take upwards of 3 weeks to complete. It requires knowledge of web development, in particular, CSS file properties.

Conclusion

These outlined tasks—optimizing the website, improving security with a Captcha system, adding data aggregation tools, and implementing user-friendly features—are essential for ensuring the website's functionality and long-term success. Future developers should approach these tasks sequentially, starting with optimization to establish a stable foundation for further enhancements.

XI. Lessons Learned

Throughout working on the Workload Calculator with the Colorado Education Association, the team has learned valuable lessons about how to make project development go smoothly. Since this project was a group project, the first tool the team implemented was a communication system to ensure all team members have the same understanding. The team created a group chat for all team members to communicate when they are free to meet up and to provide updates about the website. This helped the team stay organized and ensure any questions the team had could be answered. A Trello board was also used to divide tasks up for each team member and to make sure everyone understood which tasks had the most priority. Clear and frequent communication with the team's client was also a crucial lesson the team learned about. The team learned that creating a weekly meeting with the team's client was important to help the team's client understand what the team is working on and to ensure the team is working on the correct tasks. When the team was unable to meet with the team's client the team would email them about the team's code updates and questions the team had about the project.

Good documentation about the code and the setup of a project was another important lesson the team learned. When the team began this project, the team was given a project that has been worked on by two previous field session teams. The team had to pick up from a half-way done project and learn about how the website was set up through the documentation provided in the README file. The README file provided the team with a basic understanding of the set up, but the team quickly realized that there was more to the website than what was provided. While working on this project, a large amount of time was spent on learning how to set up the code before any changes on the website could be made.

Looking ahead, the team is optimistic that the knowledge and improvements made during this session will provide a solid foundation for future developers. With thorough documentation and the lessons learned from this project, future teams will be well-prepared to take the Workload Calculator to new heights, delivering even greater value to the Colorado Education Association and its users.

XII. Acknowledgments

The team would like to express the team's gratitude to the CEA for providing the team with an impactful project to work on that will help improve the livelihood of educators in Colorado.

The team would also like to thank Caleb Bartel, the team's advisor throughout the semester, for helping the team gain a deeper understanding of how to work efficiently as a team.

XIII. Team Profile



Sophia Mulholland
Senior
Computer Science
Hometown: Colorado Springs
Clubs & Orgs: Capoeira Club, International Rescue Committee Intern



Diana Rogozyan
Junior
Computer Science / Combined Student with M.S. in STEM Ed
Hometown: Golden
Clubs & Orgs: Education Alliance, DECtech K12 Program



Jakob Walker
Junior
Computer Science / Computer Engineering
Hometown: Colorado Springs
Clubs and Orgs: CASA, Tau Beta Pi



Nina Winslow
Junior
Computer Science
Hometown: Littleton
Clubs and Orgs: Society of Women Engineers, Sigma Kappa

References

[1] "About CEA," CEA, Jun. 06, 2024. <https://coloradoea.org/about/>.

Appendix A – Key Terms

Term	Definition
<i>CEA</i>	<i>Colorado Education Association.</i>
<i>Heroku</i>	<i>An app to deploy the team's code to a live website.</i>