Computer Science 3090 Dashboard

Design Document

Team sd25-14

Client: Simanta Mitra

Adviser: Simanta Mitra

Team Members/Roles:

Bradley Gaines - Component Design, Cross-Component Integration, Testing

Breckin Bartels - Component Design, Cross-Component Integration, Testing

Kat Christofferson - Component Design, Cross-Component Integration, Testing

Ria Patel - Component Design, Cross-Component Integration, Project Manager

Thomas Payton - Component Design, Cross-Component Integration, Testing

Varun Jain - Component Design, Cross-Component Integration, Client Interaction

Team email: sdmay25-14@iastate.edu

Team Website: https://sdmay25-14.sd.ece.iastate.edu

Executive Summary

Dr. Simanta Mitra teaches the course Computer Science 3090 at Iowa State University. Throughout this course, Dr. Mitra uses multiple applications, including Canvas, Gitlab, CATme, and datasheets, for assignments, projects, and feedback. With numerous applications being used continuously, organization and efficiency are hard to maintain throughout the semester. Dr. Mitra proposed a project, given his desire to have a dashboard that integrates the data from all applications into one website. That project is our Computer Science 3090 dashboard! This dashboard is essential for the entire Computer Science 3090 faculty, allowing easier access and organization for all class materials, resources, and feedback. The dashboard is designed to enable primary use by professors, secondary by the head teaching assistant, and tertiary by the teaching assistants.

Requirements for our project were originally very minimal. Dr. Mitra requested that the dashboard be available to all faculty involved with Computer Science 3090, allowing for different views based on what data pertained to the user accessing the dashboard. He requested that all data would be represented throughout the dashboard in an organized and meticulous manner. His final requirement requested was to make sure the dashboard could be reused for following semesters without erasing the semester data but rather storing it for future use. Otherwise, Dr. Mitra left design decisions in our hands, requesting that we continuously update him on our choices for feedback on what would be most beneficial for the users.

Our initial approach on the project was to divide our project into components to take advantage of our team's strengths. We separated the project into three components: frontend, backend, and security. While most team members are contributing to multiple components, the breakdown allows us to understand the scope of our project and the best way to proceed to create a dashboard that we were proud of. After we understood the scope of our project, we began to research on how to best implement the dashboard. We decided to use Node.JS for the frontend, Node.js with TypeScript for the backend, Iowa State University Services for hosting, and an MySQL database for data management. We began by forming our components separately, then worked together to connect frontend to the backend. We implemented security measures once we had a functional base of the dashboard.

The dashboard meets the requirements as detailed by Dr. Mitra. The requirements given to us were focused on creating a basic dashboard, and then progress through continuous updates to include desired features as time permits. We have presented our final design and it has been approved by Dr. Mitra. We have given full rights and control of the dashboard to Dr. Mitra. He will be able to use the dashboard for future semesters of Computer Science 3090. Given the desire, future additions/updates to the dashboard can be made.

Learning Summary

Development Standards & Practices Used

Development Standards:

- IEEE 830-1998: Recommended Practice for Software Requirements Specifications.
- IEEE 1012-2016: Standard for System, Software, and Hardware Verification and Validation.
- ISO/IEC 27001-2022: Information Security, Cyber Security, and Privacy Protection.

Summary of Requirements

Functional Requirements:

- Professors must be able to manage all student progress, assignment grades, attendance records, and teaching assistant feedback/reflections.
- Head teaching assistants must be able to manage all student progress, assignment grades, attendance records, and teaching assistant feedback/reflections.
- Teaching assistants must be able to manage their respective students' progress, assignment grades, attendance records, and professor feedback/reflections.
- Application needs to be able to reprogram for future course sections.

Resource Requirements:

- Development server.
- Development database.
- ISU server.
- ISU RDS.
- The dashboard must represent all data given in an uploaded CSV file.

User Interface Requirements:

- Role-specific views for professors, teaching assistants, and head teaching assistants.
- Friendly and easy to navigate interface for all users.

Aesthetic Requirements:

- Clean and simple visual style.

Physical Requirements:

- Application must be compatible with various screen sizes and resolutions.

Quantitative Requirements:

- System must be capable of handling data for over 500 users.
- Dashboard response time should be less than 5 seconds for any action.

Applicable Courses from Iowa State University Curriculum

- Computer Science 3090 – Project Development. This course helped us understand how to connect a frontend and backend.
- Computer Science 3190 – Introduction to JavaScript and Web development. We used Typescript as our programming language, which is a variant of JavaScript, taking this course helped us understand how to use JavaScript effectively to create a web app.
- Computer Science 3630 – Database management. This course helped us understand how to effectively put a database in place. ER diagrams along with some other practice from this course were applied to our project.

New Skills/Knowledge Acquired

Senior design has provided us with the opportunity to explore a diverse range of tools and technologies that extend beyond the scope of the ISU curriculum. For instance, our team has gained proficiency in TypeScript and Node.js for developing our project. Additionally, we have learned to integrate Google Authentication with Node.js and implement React Router for seamless navigation within the frontend. On the backend, we have acquired knowledge of using JWT tokens to enhance the security of our

application. These skills have broadened our technical expertise and enabled us to build a robust and secure solution.

TABLE OF CONTENTS

FIGURES, TABLES, SYMBOLS & DEFINITIONS

FIGURES:

- Figure 21 – Postman Testing

DEFINITIONS:

ER Diagram: Entity Relationship Diagram

API (Application Programming Interface): A set of rules and protocols that allows different software applications to communicate with each other.

Backend: The part of a system or application that is not directly accessed by the user, normally it is responsible for the manipulation and storing of data.

CATME: A web application used in Computer Science 309 for managing team formations and peer evaluations.

Frontend: The part of a system or application in which the user directly interacts with.

FERPA: The Family Educational Rights and Privacy Act, a federal law that protects the privacy of student education records.

JWT (JSON Web Token): A secure way of transmitting information between parties as a JSON object.

LMS (Learning Management System): A software application for administering, documenting, tracking, reporting, and delivering educational courses.

REST API: An architectural style for designing networked applications that uses HTTP requests to access and manipulate data.

SQL (Structured Query Language): A standardized programming language used for managing and manipulating databases.

TypeScript: A programming language developed and maintained by Microsoft, it is a strict syntactical superset of JavaScript and adds optional static typing to the language.

UI/UX: User Interface/User Experience - the design and interaction elements of an application that create the user experience.

Waterfall: A linear project management approach where each phase of the project must be completed before moving on to the next.

1 Introduction

1.1 Problem Statement

Dr. Simanta Mitra, professor of Computer Science at Iowa State University, has been proposing a project for numerous capstone groups to master. He desires a web application that would help facilitate the online workings of his Computer Science 3090 course. This course uses multiple platforms, such as CatMe, Canvas, Gitlab, and datasheets, which makes organization and accessibility complicated for Dr. Mitra. To solve this issue, we created a dashboard that combines all applications used in this course into an easy-to-navigate program with all resources and class information in one place. Since other capstone groups have attempted this application previously, we used their code and research to help navigate through our initial attempt. This application provides easy access for professors and teaching assistants to review team statuses, track attendance, and monitor at-risk students. This dashboard is very important for the course, ensuring that professors and teaching assistants have easier access to all aspects of the course. The dashboard is focused on critical information for the course (attendance, assignments, grades, and projects) but was also updated with highly desirable features (uploading documents, at-risk student awareness, and a centralized comment system) as well.

1.2 Intended Users

Professors, teaching assistants, and the head teaching assistant will benefit greatly from this application. Professors need an efficient way to manage multiple aspects of Computer Science 3090, including student progress, assignments, attendance, and teaching assistants' feedback. Teaching assistants need a streamlined way to provide and manage feedback for student projects. They also need to efficiently manage the groups assigned to them. The head teaching assistants needs an overview of feedback from all teaching assistants to ensure consistency and quality, alongside access to review and manage all student teams. This application satisfies all needs for these groups, and more, making all course aspects easily accessible. Given that most of the needs are centralized for all groups, the application focuses on assignments, attendance, and feedback, but we modified the application in correlation to course changes throughout the semester.

2 Requirements, Constraints & Standards

2.1 Requirements & Constraints

Functional Requirements:

- Professors must be able to manage all student progress, assignment grades, attendance records, and teaching assistant feedback/reflections.
- Head teaching assistants must be able to manage all student progress, assignment grades, attendance records, and teaching assistant feedback/reflections.
- Teaching assistants must be able to manage their respective students' progress, assignment grades, attendance records, and professor feedback/reflections (constraint)
- Application needs to be able to reprogram for future course sections.

Resource Requirements:

- Development server.
- Development database.
- ISU server.
- ISU RDS.
- The dashboard must represent all data given in an uploaded CSV file.

User Interface Requirements:

- Role-specific views for professors, teaching assistants, and head teaching assistants.
- Friendly and easy to navigate interface for all users.

Aesthetic Requirements:

- Clean and simple visual style.

Physical Requirements:

- Application must be compatible with various screen sizes and resolutions.

Quantitative Requirements:

- System must be capable of handling data for over 500 users.
- Dashboard response time should be less than 5 seconds for any action.

2.2 Engineering Standards

Engineering standards are very important in everyday life. They ensure that products, technology, and services are able to work together easily and safely. Standards help maintain quality and reliability, making everyday life safer. Engineers must follow standards to help create solutions that continue to be compatible with existing systems, allowing for a more efficient, safe world.

In relation to our Computer Science 3090 Dashboard, the following standards were most relevant:

- IEEE 830-1998: Recommended Practice for Software Requirements Specifications: This standard provided guidelines for creating software requirements specifications, which were useful in documenting the project's requirements.
- IEEE 1012-2016: Standard for System, Software, and Hardware Verification and Validation: This standard was applicable for ensuring the quality and reliability of the dashboard system, which was important for our client relationship throughout the project.
- ISO/IEC 27001-2022: Information Security, Cyber Security, and Privacy Protection: This standard specified requirements for establishing, implementing, and maintaining information security, which was necessary for data encryption in our project.

The standards were very relevant to our Computer Science 3090 Dashboard project. IEEE 830-1998 provided great insight on how we could maintain recommended practices throughout the dashboard. IEEE 1012-2016 provided guidance on how to make a valuable and strong dashboard, strengthening our reputation with our client. ISO/IEC 27001-2022 provided great insight on security standards, allowing us to make sure the dashboard respects the privacy of our users. Although there are other standards that were applicable to our project, these three were the most important.

Our team agreed that these standards were most important for our project. IEEE 830-1998 was very relevant for the frontend, backend, and security aspects of our project. IEEE 1012-2016 was extremely important for all members of our team, as we take pride in the work we complete, and wanted to provide the best possible dashboard that we are capable of for our client Dr. Mitra. ISO/IEC 27001-2022 was

particularly important for the security implementation throughout our project, ensuring that we treated all users with the utmost respect for their personal and academic information.

We kept all standards in consideration while completing this project. Standards are very important to our team as we are entry level engineers, learning how to adapt to the industry. Although standards provide the necessary requirements for such topics, they also inspire us to learn from others' experiences.

3 Project Plan

3.1 Project Management/Tracking Procedures

The Computer Science 3090 Dashboard project embodied a waterfall and agile combination management style. The waterfall style allowed a scheduled order of tasks for the beginning of our project including designing the frontend, designing the backend, implementing the frontend, implementing the backend, and implementing security. This allowed us to create a functional, but simple, dashboard by the end of the first semester of our capstone. The agile style enabled feedback from our original implementation to be implemented and adjusted in the dashboard during the later parts of our project, primarily during the second semester of our capstone. We tracked progress through diagrams (primarily during capstone semester 1) and the dashboard application (primarily during capstone semester 2). Diagrams were created and updated through Figma to represent our up-to-date feature design. The dashboard shows our up-to-date feature implementations as well.

3.2 Task Decomposition

Below is our task decomposition chart. This chart elucidates the major milestones for each aspect of our project including design, frontend, backend, and security. As shown below, design milestones focused on interpreting the requests of our client, and brainstorming how we planned to implement features into the dashboard. Frontend and backend followed the same milestone format of creating a high-level schematic, implementing the details of the schematic, connecting with the other end, and then continuously updating the end to adjust to new feature implementations. Security had fewer milestones, all of which focused on gaining authorized access to data, encrypting the data, and continuously updating our security measures to adapt to new feature implementations.

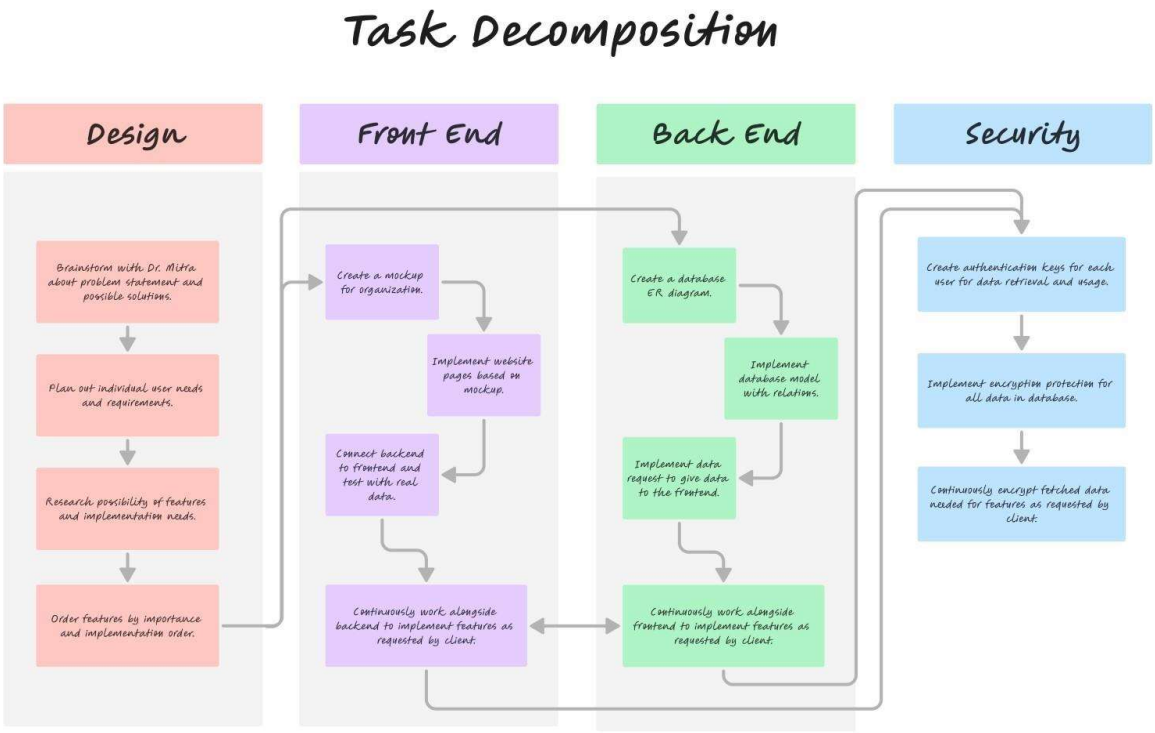


Figure 1 – Task Decomposition Diagram

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Key milestones were broken down into frontend, backend, security, and overall. Most milestones included design ideas through diagrams, and some were actual implementation milestones. Progress was measured in different ways for all milestones since some were continuously updated to fit the timeline that Dr. Mitra had requested. Due to the layout of our project, milestones were not quantitative but rather qualitative.

Frontend milestones included a Figma mockup, screen flow diagram, and implementation of pages. The Figma mockup was continuously updated for new features that Dr. Mitra requested during the capstone semester 1. The mockup shows how each feature is represented in the dashboard. The screen flow diagram follows the same milestone format. The diagram was continuously updated to show the connections between the pages that are requested for representing different features during the capstone semester 1. Capstone semester 2 milestones for frontend development were the actual implementation of pages in the dashboard, showing that said features were correctly embedded into the dashboard application.

Backend milestones included an ER diagram, implementation of tables, and the connection of the frontend to the backend. The ER diagram shows all tables necessary to analyze the fetched data from numerous websites. The diagram shows the breakdown of each table grouping, and how they are connected to other tables. Implementation of tables was another milestone for the backend. This ensured that all fetched data was organized and ready to be pulled. Capstone semester 2 milestones for the backend development included connecting the frontend to the backend. This created a full, functioning dashboard that is ready for use.

Security milestones included data encryption, accessibility keys, and login capabilities. Data encryption ensured that we protected the privacy of our users when fetching data from the academic resources. Accessibility keys allowed us to grab user data from multiple applications used in the course, including CatMe, Canvas, Gitlab, and datasheets. Login capabilities made the dashboard a private application, only to be accessed by individuals with an Iowa State email address. This ensured that any user data is safe and kept private. This also allowed for individual users to only access the data that they are permitted to review. Professors have access to all data, while teaching assistants have access to their student groups' data.

3.4 Project Timeline/Schedule

The Gantt charts below represent our project timeline from the start of capstone semester 1, to through capstone semester 2. On the semester 1 Gantt chart, each color represents different aspects of the dashboard: pink for design, purple for frontend, green for backend, and blue for security. On the semester 2 Gantt chart, each color represents collaborative work groups during the agile management phase of your project. Design was the primary focus for the first few weeks of our project. Since our project embodied more of an agile management style, we focused on frontend, backend, and security for the rest of the project timeline. Frontend, backend, and security all started by creating mockups/schematics for ideal design, but most of our time was spent in a trial-and-error phase by updating the dashboard to fit Dr. Mitra's requests as they came. Once we created a functional dashboard, the rest of our time was flexible to ensure that both the front and back end worked together to implement new features. We continued this simultaneous work style through to the end of our project while ensuring our security measures were also protective.

Gantt Chart

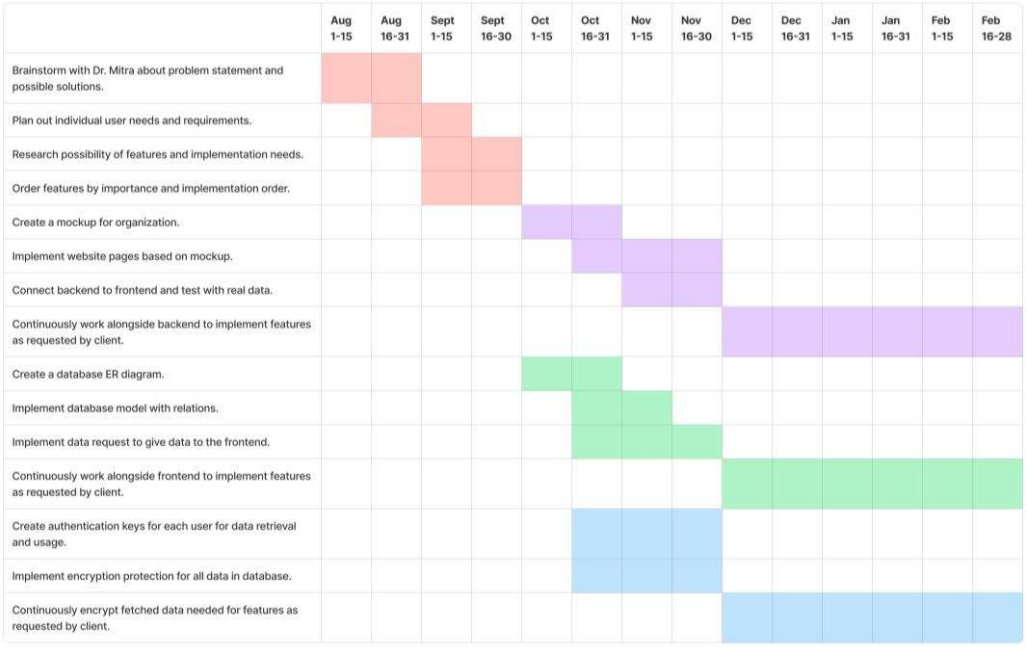


Figure 2 – Semester 1 Gantt Chart

Gantt Chart

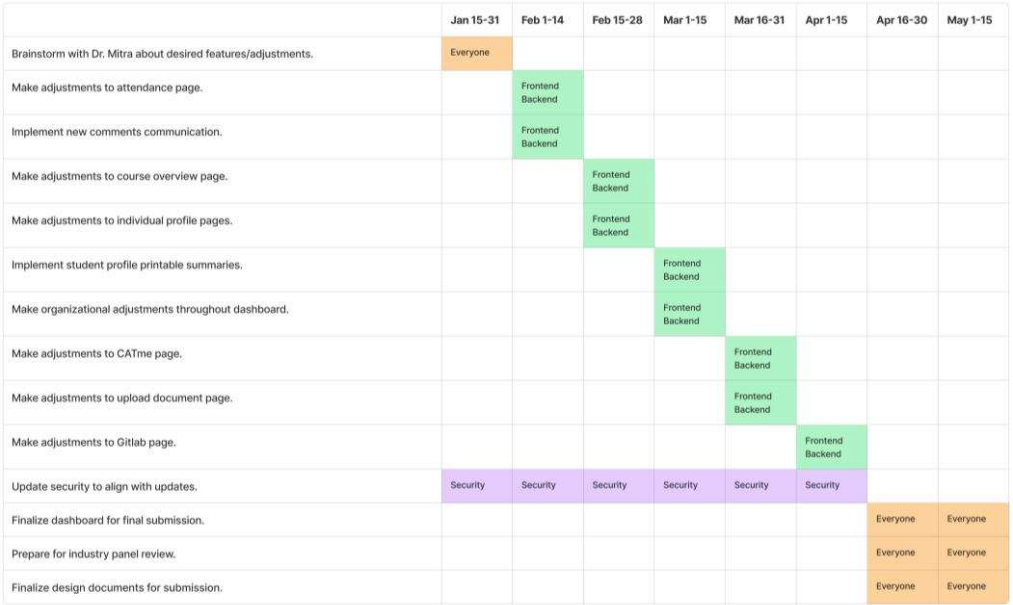


Figure 3 – Semester 2 Gantt Chart

3.5 Risks and Risk Management/Mitigation

Risks for our project were specified for either frontend, backend, security, and a few overall. All of our risks were not quantifiable in terms of a risk factor. None of our risks could have been eliminated by purchasing parts, tools, or technology and implementing them. A few of our risks could have been eliminated by using a different approach but the features related to these risks are desired as an add on function, not a requirement for our project.

Risks for our frontend included privacy and easy usability. Since the dashboard displays students' academic information, we must be careful how we represent the data. For instance, large text fonts and sensitive academic data could not be visible on the home screen. Easy usability was another risk for the dashboard. Making sure that every user was able to understand the dashboard easily and effectively use its features was one of our primary concerns. The dashboard was essential in making it easier for users to access data and resources for the course.

Risks for our backend included accurate tables and data analysis. Tables helped organize the data that we pull from CSV files. Making sure that we kept the data organized allowed us to determine which users have access to which tables. This went hand in hand with data analysis. We originally planned to use direct connections to Canvas, Gitlab, CATme, and datasheets, but we were unable to be granted access to these applications because of student privacy restrictions. This is where we shifted our original plan from direct connections to CSV file uploads. Since we were informed early on that we would not be granted accessibility keys to these applications, our risk was minimized and we shifted our project plan accordingly.

Risks for security included data encryption and academic data fetching. Since we were dealing with sensitive and private data, we needed to make sure that all of the dashboard's data was encrypted. This made sure that the dashboard was safe from outside users trying to access our data. Since we were fetching academic data, it is a big risk to make sure that we were accessing the right data for each user. We had to be very careful in pulling data that only applies to the individual user's determined access.

Overall risks that we faced were having different views for different users and ensuring the dashboard was usable for multiple course sections. Our design allowed different views based on the login information. For example, when the professor logs in, they have access to all of the data but when a teaching assistant logs in, they only have access to their students' data. Since different users have access to different data, the number of pages and the representation of the data looks different. Since this application will be used for multiple semesters of the Computer Science 3090 course, we made sure, the dashboard can be reused fresh. Although the dashboard will be refreshed for new semesters, we kept all past semester data accessible for the professor.

All risks were important for us team members as we worked on this project. We were dealing with very sensitive data, which means that our users privacy came first. Most of our risks revolved around fetching, organizing, and representing academic data. We resolved these issues with encryption, login features, and private displays of information. These risks were not as impactful on our project as predicted since we kept these risks in mind while constructing the dashboard. We realize we were lucky to not be affected by many risks but we do believe this was the case since we thought out all aspects of our project in detail prior to starting our work.

3.6 Personnel Effort Requirements

Task Requirements	Estimated Person-Hours to Complete	Actual Person-Hours to Complete
Design: Brainstorm with Dr. Mitra about problem statement and possible solutions.	10	20
Design: Plan out individual user needs and requirements.	10	10
Design: Research possibility of features and implementation needs.	20	20
Design: Order features by importance and implementation order.	8	10
Frontend: Create a mockup for organization.	8	10
Frontend: Create a screen flow diagram for page connections.	6	10
Frontend: Implement website pages based on mockup.	14	80
Frontend: Connect backend to frontend and test with real data.	18	80
Frontend: Continuously work alongside backend to implement features as requested by client.	30	80
Backend: Create a database ER diagram.	10	10
Backend: Implement database model with relations.	18	80
Backend: Implement data request to give data to the frontend.	12	80

Backend: Continuously work alongside frontend to implement features as requested by client.	30	80
Security: Create authentication keys for each user for data retrieval and usage.	20	30
Security: Implement encryption protection for all data in database.	20	30
Security: Continuously encrypt fetched data needed for features as requested by client.	20	30

All estimated person hours above were a baseline based estimated on our initial design. As the design of our project changed, the hours did as well. All actual person hours above are rounded to the nearest hours based on calculations of weekly time commitments. Implementation and connection tasks, for both frontend and backend, required a large number of hours over what was expected. We spent majority of our two semesters continuously updating the dashboard to fit the client's needs and desires, which correlates to the immense number of hours spent implementing features and connecting the frontend to the backend.

3.7 Other Resource Requirements

Other resources required for the project, outside of team effort, were the ISU database and server. We were able to acquire both from the university. This project did not require any other resources for functional use, but the project did require data from applications that we represented throughout the dashboard, including GitLab, Canvas, CATme, and datasheets. The data was not directly pulled from these applications, but rather made into a CSV file that we parsed through to represent the data throughout the dashboard.

4 Design

4.1 Design Context

4.1.1 Broader Context

While our primary focus was Computer Science 3090, the dashboard could potentially be adjusted to help the organization and efficiency of other courses that follow similar application use. We designed the dashboard around the needs and desires of Dr. Mitra, as well as his teaching assistants. The dashboard addresses societal needs for a home page for academic courses that use multiple applications throughout the semester. Our primary goal for the dashboard was for it to provide a central website that allows faculty to have access to all course materials, resources, and feedback in one place, maximizing efficiency and organization.

	Beneficence	Nonmaleficence	Respect for Autonomy	Justice
Public Health, Safety, and Welfare	Reduces stress of finding class resources.	Does not require users to adjust.	Users can choose their involvement level.	All users have availability to use.
Global, Cultural, and Social	Designed for Computer Science 3090 community.	Users are not required to use.	Users can choose to use.	All users have availability to use.
Environmental	Minimal resources used.	No download required.	Users can access simply with a web address.	Causes no harm to the environment.
Economic	Minimal resource costs.	No purchase required.	Free for all users.	Easily accessible for Computer Science 3090 community.

4.1.2 Prior Work/Solutions

https://seniord.cs.iastate.edu/2022-May-07/files/inline-files/402%20Final%20Presentation.pdf

https://seniord.cs.iastate.edu/2023-Jan-09/files/inline-files/402C%20Demo%201_0.pdf

Before our team started to work on the dashboard, there were 2 Computer Science teams that attempted this project. However, their attempts were unsuccessful to the requirements. One of the teams used Node.js, and MongoDB to develop their project. They were able to access the multi-page application

though a navigation bar, and view all the projects and their statuses. For the backend, they were able to store data in the database, and able to query data from different semester and years. Some of the problems they faced included integration with GitLab and designing the structure of database. Another problem they faced was an inability to deploy the frontend to a server. The second team used React.js and Node.js to develop their project. Due to their poor documentation, we were unsure what they had completed. The issues they face included: no API calls from CatMe, higher authority needed for canvas, and logging authentication issues.

Literature used throughout:

1. Shadcn/ui, "Introduction," shadcn/ui, https://ui.shadcn.com/docs (accessed Dec. 5, 2024).
    1.1. This resource has been used by the frontend as a method for gathering documentation on features that are being used.
        1.1.1. For example, the Frontend is using Shadcn/ui buttons. In order to implement these buttons properly, we utilized the documentation to effectively implement these buttons.
2. "Rest api," GitLab, https://docs.gitlab.com/ee/api/rest/ (accessed Dec. 5, 2024).
    2.1. This resource was used by the frontend to gain knowledge on various API request to Gitlab.
        2.1.1. These API request will be used to gather Gitlab commit history information for each team.
3. "typeorm," GitBook, https://orkhan.gitbook.io/typeorm (accessed Dec. 5, 2024).
    3.1. This resource was used by the backend to figure out how to access the database efficiently through the code.

Similar products/solutions in the market:

1. Canvas
    1.1. Pros: Comprehensive learning management software, widely adopted.

1.2. Cons: Not customizable for specific course needs (like Computer Science3090), difficult to connect and explore data from other resources.

2. Previous Senior Design Projects

2.1. Pros: Rudimentary integration with GitLab and Canvas, simple interface.

2.2. Cons: Limited functionality, security concerns, minimal integration with external services.

Our solution advantages:

- Custom-built for Computer Science 3090 specific needs.
- Data representation for multiple platforms.
- Role-based access control.
- Simple, intuitive interface.
- Secure data handling.

Our solution disadvantages:

- Requires maintenance/work for each semester.
- Dependent on external APIs which could break in the future.
- Limited to single course use.
- Requires basic training/explanations for features.

4.1.3 Technical Complexity

The dashboard in its entirety is a professional product, using real world professional tools and services. We have effectively, from the ground up, built a more narrow and custom Canvas-like tool for Dr.

Mitra and Computer Science 3090. This involves all the moving parts of database, backend and frontend like any corporate application and also includes the handling and management of data.

The dashboard's frontend consists of a Node.JS web app utilizing a modern CSS library, shadcn/ui. Next.JS is a modern web framework based on React.JS that has built-in routing based on the directory hierarchy. We've implemented role-based access control directly in the frontend routing system, ensuring users only see content relevant to their permissions level. The user interface is built using reusable components from shadcn/ui, which we've customized to match our specific needs while maintaining a consistent design system throughout the dashboard. The frontend also includes form validation and file uploads all while maintaining responsive design that ensures the dashboard works across different device sizes and browsers.

The dashboard's backend consists of a Node.js server using TypeScript for type safety and code reliability. We've implemented a RESTful API architecture that handles complex data operations. The backend implements JWT (JSON Web Token) authentication to secure API endpoints and manage user sessions. We've created middleware functions to handle role-based access control, request validation, and error handling.

The database consists of a relational MySQL structure hosted on a server provided by Iowa State, designed to efficiently store and manage course-related data. Our database schema includes multiple interconnected tables that handle user information, course data, attendance records, and team formations. We've implemented foreign key constraints to maintain data integrity and optimize query performance. The database design allows for semester-based data archival while maintaining historical records for future reference. The semester archival system and reset must be completed manually by the professor, details on instructions are provided in the user manual.

4.2 Design Exploration

4.2.1 Design Decisions

The dashboard involves numerous decisions on design since we were given the opportunity to create an open-ended solution. Our most important decisions related to design included how to distribute the dashboard, how to represent information for different groups, and how to safely store and secure data. All design decisions were made to be in the best interest of our users, primarily the professor, but also the teaching assistants.

The first key design decision we made was how we chose to distribute the dashboard for professor and teaching assistant use. We decided to create a website so that everyone has easy access to our application. This was an important decision because choosing an application could restrict users from use based on their device (iOS, Android, etc.). Our project is available to faculty involved with Computer Science 3090, so a website was the most universal option.

The second key design decision we made was how to represent information for different groups. Since professors and teaching assistants will have access to the dashboard, we wanted to make sure each individual was only able to see data that was appropriate for them. Professors would be able to see all data, teaching assistants would be able to see data based on their groups, and the head teaching assistants would be able to overview all data (besides admin restricted pages). This was a very important decision for us because we must make sure we are not violating class information by allowing individuals to see data that is not relevant or permissible to their situation. We solved this issue by having different pages viewable based on the individual that is signed into the website. All pages were designed in the same process: the client mentions an issue/desire, our team brainstorms the best way to solve the issue, we create a page on the frontend to best accommodate the request, backend adjusts the database to direct data appropriately, we present the page to the client, the client provides adjustments, we complete the adjustments, we present the final page to the client.

The third key design decision was how to implement security to safely store and secure data. It was very important that we follow FERPA and other academic safety standards throughout the dashboard. In order to prevent outside infiltration of our data, we encoded all data coming in and out of the dashboard. To help this, we also fetched academic data as requested, instead of storing it within the dashboard. As for within the dashboard, we are made sure our page designs respect the privacy of our users by not including grades on the home page or in large fonts.

All design decisions that we made while working on the project built on our success for the final dashboard. Key decisions were made as a group, allowing all decisions to be well-rounded and well-thought-out. If given the chance to make changes to our original design decisions, none would be adjusted.

4.2.2 IDEATION

One key design decision that we continuously worked on updating was how we represented data for our main user, the professor. This included our website pages, as well as how we designed those pages to represent data efficiently and legibly. Our goal was to make data more accessible for the professor, so we were continuously updating the design of our pages to provide easy navigation. All pages were designed in

the same process: the client mentions an issue/desire, our team brainstorms the best way to solve the issue, we create a page on the frontend to best accommodate the request, backend adjusts the database to direct data appropriately, we present the page to the client, the client provides adjustments, we complete the adjustments, we present the final page to the client.

In the view of the professor, we originally had all information represented on the home page. We decided to have a home page and multiple drill down pages, to make information easier to find. Over time, we decided that the home page will include the most relevant information in snippets including overall team statuses, students who have continuously not been in attendance for lecture, project groups doing poor, at-risk students, and new comments from teaching assistants. In order to see more detail, the professor can drill down into sections to see overall data for that topic. We continuously added more drill down pages to improve the efficiency of finding direct information.

We added a page to represent project demo status, even grouping them by which teaching assistant is the head of the group. We originally decided to show all demo status by color coordination immediately, but after further review we decided to include the most recent demo status and overall status then allow drill down to show all project demos.

We implemented another drill down page that will show comments/feedback from teaching assistants. Our original idea was to show all comments in sequential order. After further review, we made a status system for comments including action taken or action not taken. This allowed the professor to know the status of the comment, and make sure he acted on said comment. The comment system also allows users to put a level of priority on the comment, allowing comments to be ordered in importance. The comment page has a link to the course's discord chat to output messages to students as necessary.

As for the student attendance drill down page, we originally were going to store all attendance records by lecture. In order to see how frequently and when students were missing lecture, we only showed overall individual attendance records then drill down to every attendance status, in order to find patterns in missing attendance if necessary.

All design decisions were made with Dr. Mitra as the primary user, then adjusted for teaching assistant views. Other design decisions were necessary correlating to every feature addition, all of which followed the same ideation process.

4.2.3 Decision-Making and Trade-Off

Since our project decisions were mostly updating and making things more efficient rather than choosing another design option, we created a pro-con table for the original design versus the updated design. The table shows the pro-con for both design approaches. While our designs were continuously evolving, our pro-con table was as well. The table primarily focuses on our drill down pages stemming from our home page. Most drill down pages have their own drill down pages.

Feature	Original Design		Updated Design	
	Pros	Cons	Pros	Cons
Home Page	Simple access	Too much information in one place	Important information visible	1 extra step for more information
Resources	Simple access	Resources not available on dashboard	Resources available on dashboard	1 extra step for more information
Demo Status	Simple access	Too much information in one place	Most recent demo status shown immediately	1 extra step for more information
Comments/ Feedback	Simple access	Too much information in one place	Filtering by status to show importance	1 extra step for more information
Attendance	Simple access	Too much information in one place	Overall attendance record shown immediately	1 extra step for more information

4.3 Final Design

4.3.1 Overview

Our capstone project, Computer Science 3090 Dashboard, aimed to provide easier access for the professor and teaching assistants, to all materials and resources used throughout the course. Materials and resources include course resources, assignment information, project information, comments and feedback, and attendance records by analyzing and representing data from CSV files uploaded by the professor. We created a website dashboard that will not only represent all of the data that would have been represented on Canvas, Gitlab, Catme, and datasheets, but organize it for easier navigation.

Our screen flow sequence is designed for the dashboard to greet the professor with a login page. Once login is authenticated, the class overview page shows. This page allows the user to see snippets of the

most important information including: the number of current at-risk students, the number of students with poor attendance, team status overviews (most recent demo and overall), a link to the upload documents page, a link to the attendance management page, and a link to the at-risk students page. Although these links are provided, clicking on the data snippet will also redirect the user to the appropriate drill down page.

The professor has access to the entire dashboard, including editing capabilities. Teaching assistants have access to all pages stemming from individual team overview for their student groups. Head teaching assistants have access to the entire dashboard, except editing capabilities.

4.3.2 Detailed Design and Visual(s)

The dashboard was primarily broken into three components: frontend, backend, and security. Our team split in order to address all components based on our degree specialties, while still informing each member about advances made in each component. Frontend focused on the website visuals of the dashboard. Backend focused on the connections from analyzing data to their respective location on the frontend. Security focused on making sure all data and the dashboard itself is secure and follows standard guidelines.

Frontend was primarily organized based on a Screen-Flow diagram. The diagram shown below captures the page-to-page connections throughout the dashboard. The arrows show which pages are accessible based on the page that the user is currently on.

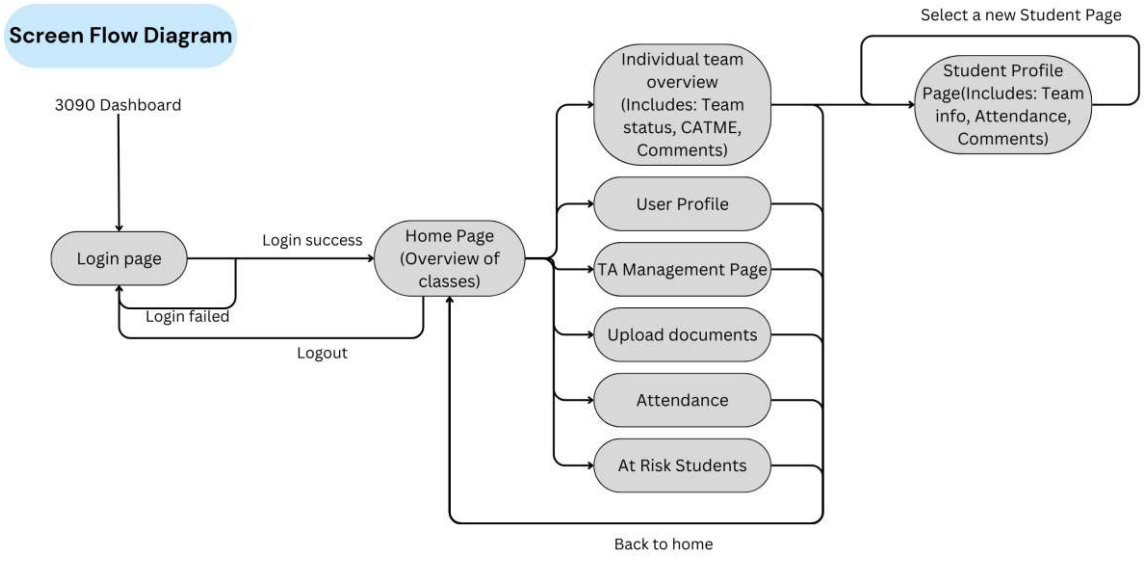


Figure 4 – Screen Flow Diagram

When first accessing the dashboard, the user will come across the landing page. As shown below, the user is given an overview of what features, benefits, and users are included in the dashboard. From this page, the user can launch the dashboard.

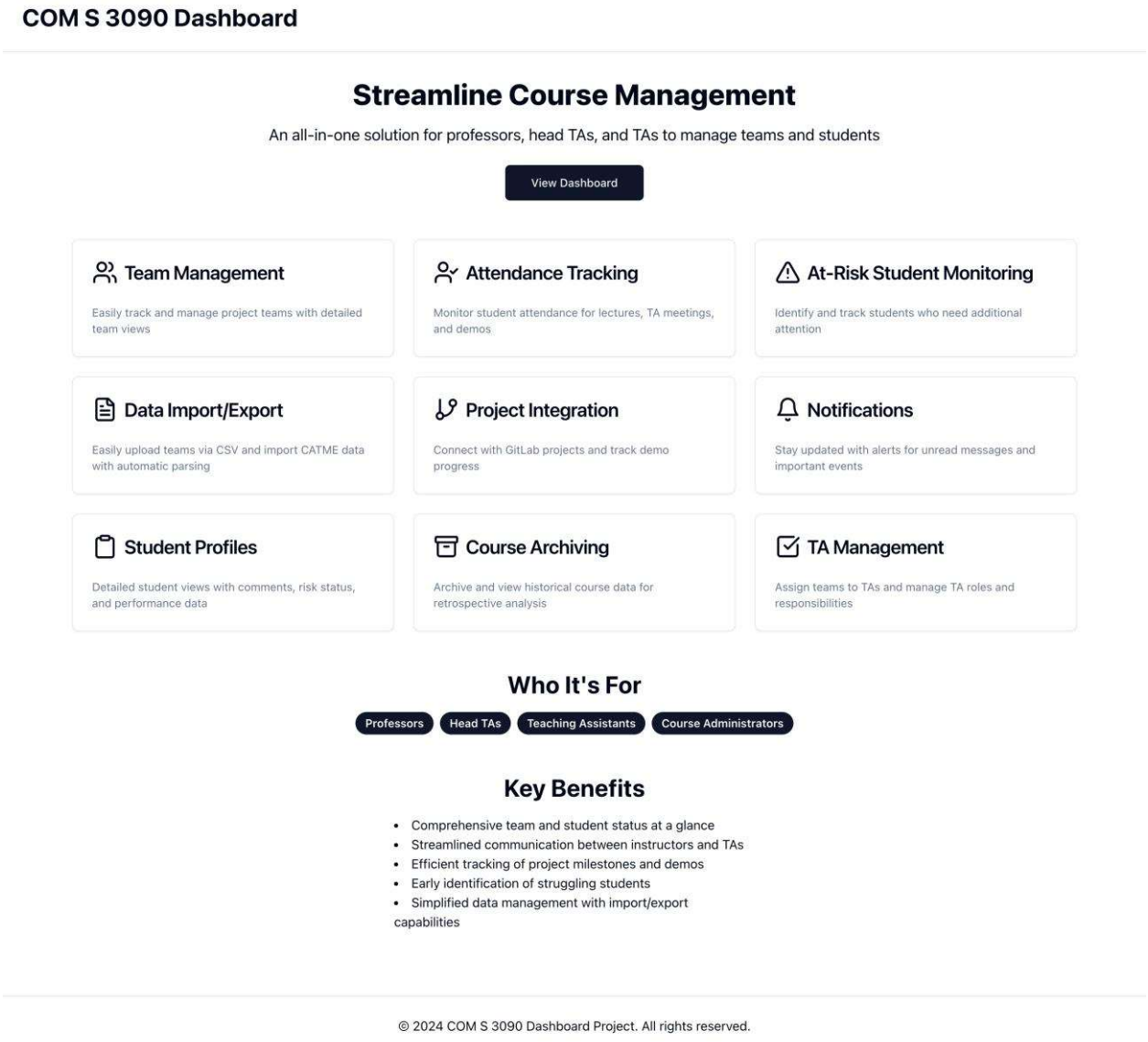


Figure 6 – Landing Page

The login page uses Google Authentication to verify the user and their accessibility. Once authenticated, the home page shows. The home page, shown below, shows snippets of the most important data including: number of current at-risk students, the most common issue amongst the at-risk students, the current semester, and team status overviews with 2 bubbles representing the most recent coding and teamwork statuses. A key for the bubble colors is located at the bottom of the page. The home page also includes links to the upload document page, attendance management page, and at-risk students page, but clicking on the data snippets will also redirect the user to the appropriate drill down page.

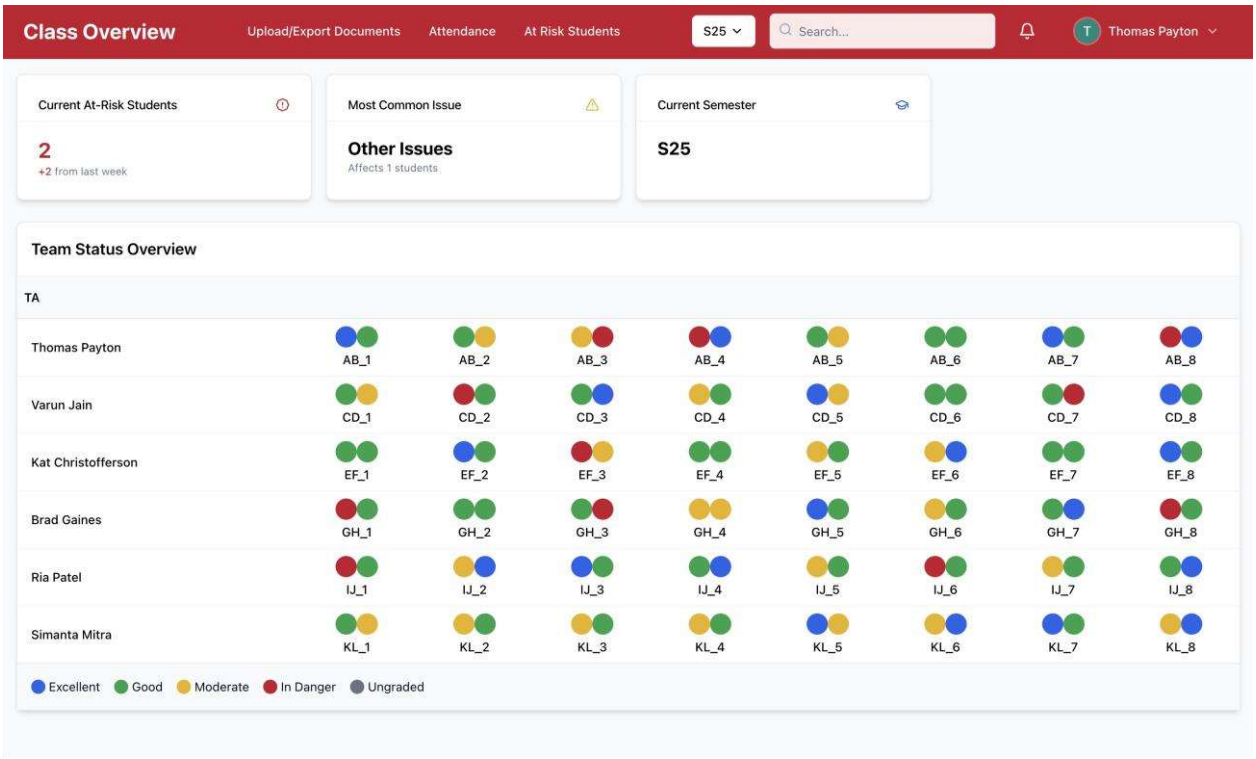


Figure 7 – Home Page

The upload/export documents page, shown below, allows the user to input a CSV file by providing the data type: teams and students or CATme data. This page allows the user to input the data that will be represented throughout the dashboard. This page also allows for exports of class data.

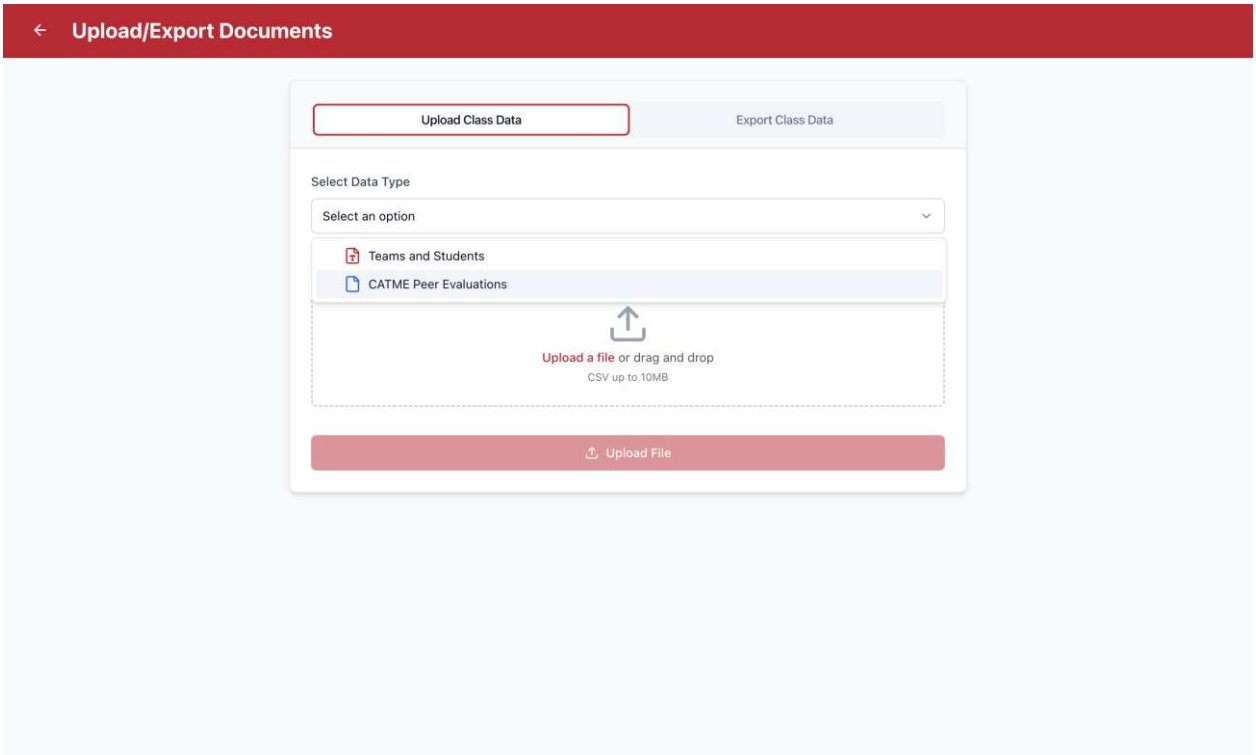


Figure 8 – Upload/Export Documents Page

The attendance management page, shown below, allows the user to input attendance by date and meeting type (lecture, team meeting, TA meeting, etc.). The input is filtered by unexcused absences and excused absences. Full attendance records are also visible on the attendance record popup, to allow the user to change an absence from unexcused to excused or vis-versa if necessary.

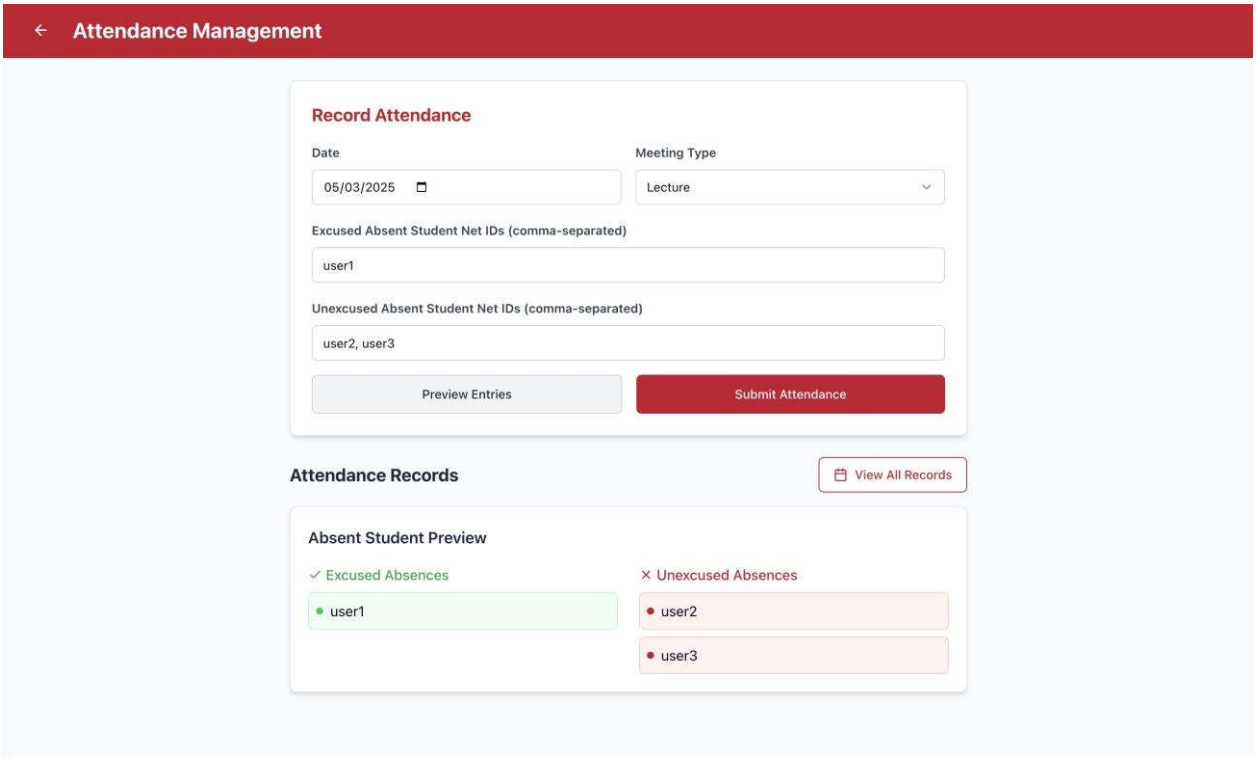


Figure 9 – Attendance Management Page

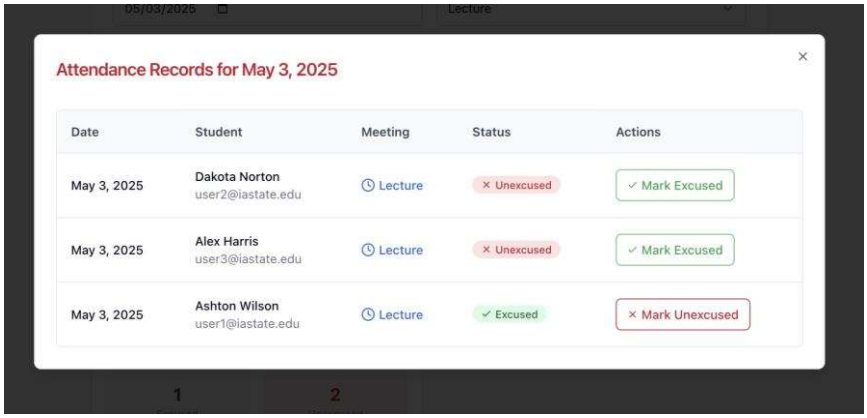


Figure 10 – Attendance Record Popup

The at-risk students page, shown below, shows the current number of at-risk students, the week that it is in the course, and a list of at-risk students that can be filtered using the search function. The user can input the at-risk student information on this page, using the popup, by including: student net-id, week,

semester, reason, and notes. The user can filter out at-risk students to their desire and send out an email to those students using the generated list that the dashboard provides of university emails.

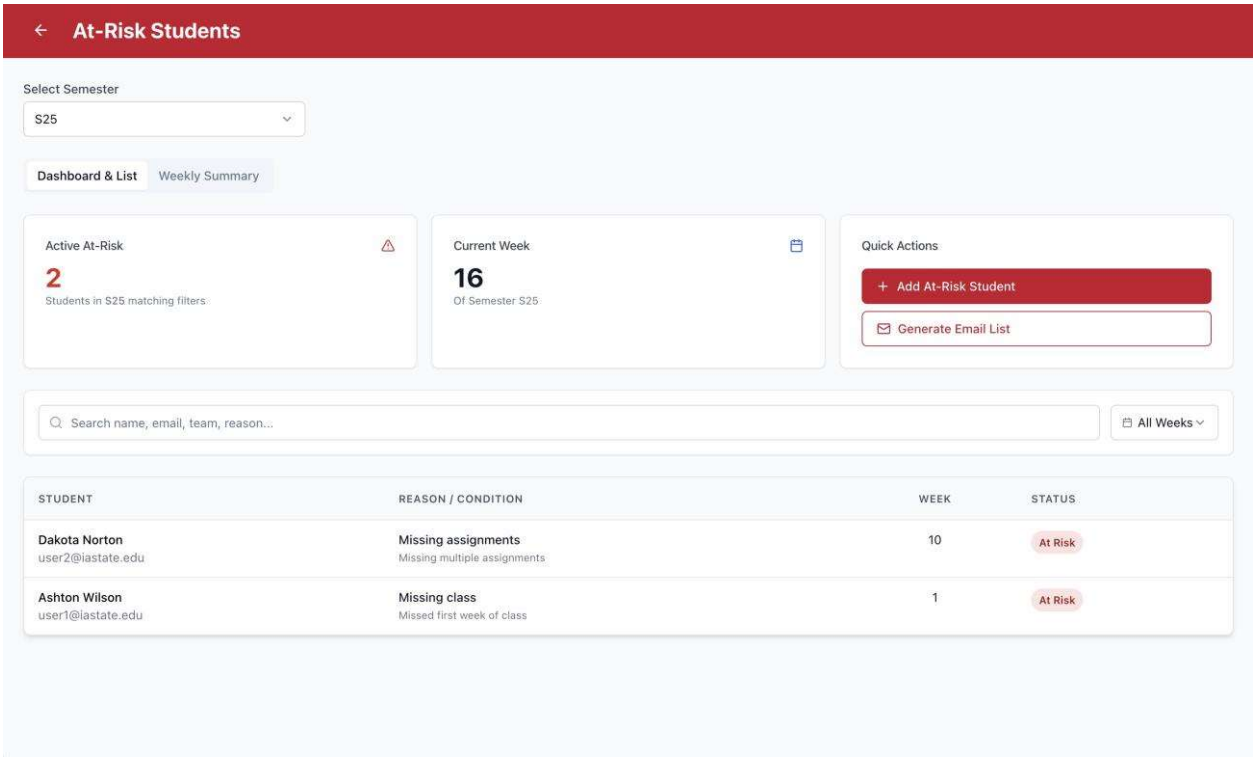


Figure 11 – At-Risk Students Page

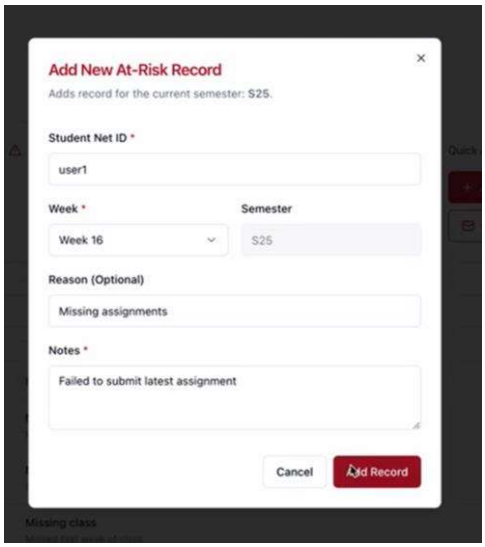


Figure 12 – Entering At-Risk Students Popup

The user profile page, shown below, gives the user's name, email, account information, and access to the account settings if permissible.

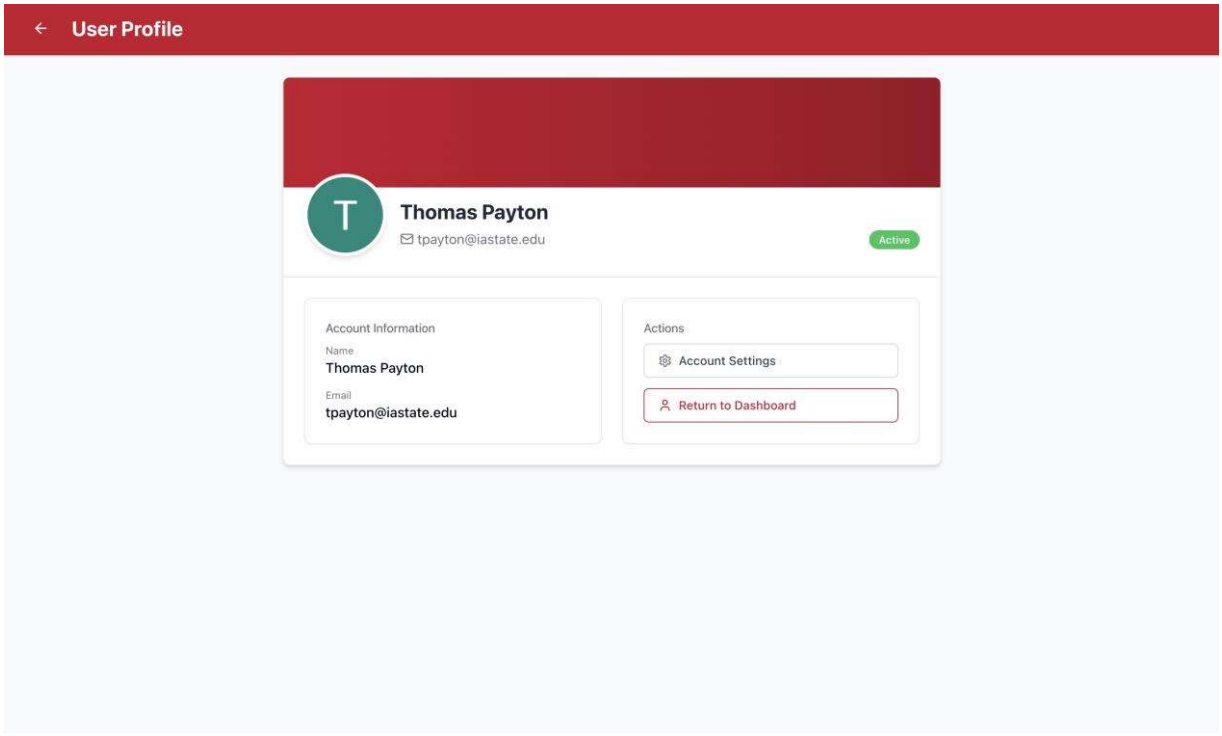


Figure 13 – User Profile Page

The TA Management page, shown below, allows the user to view all active teaching assistants and heading teaching assistants. The page provides the name, email, role, office hours, and assigned teams of all teaching assistants. Actions to add/remove a TA from a team can be made using the actions pulldown. Switching to the Head TA tab allows the head teaching assistant to view all team statuses, even allowing for assigning teams to teaching assistants. Teaching assistants can also be added, using the popup, by providing first name, last name, and email address.

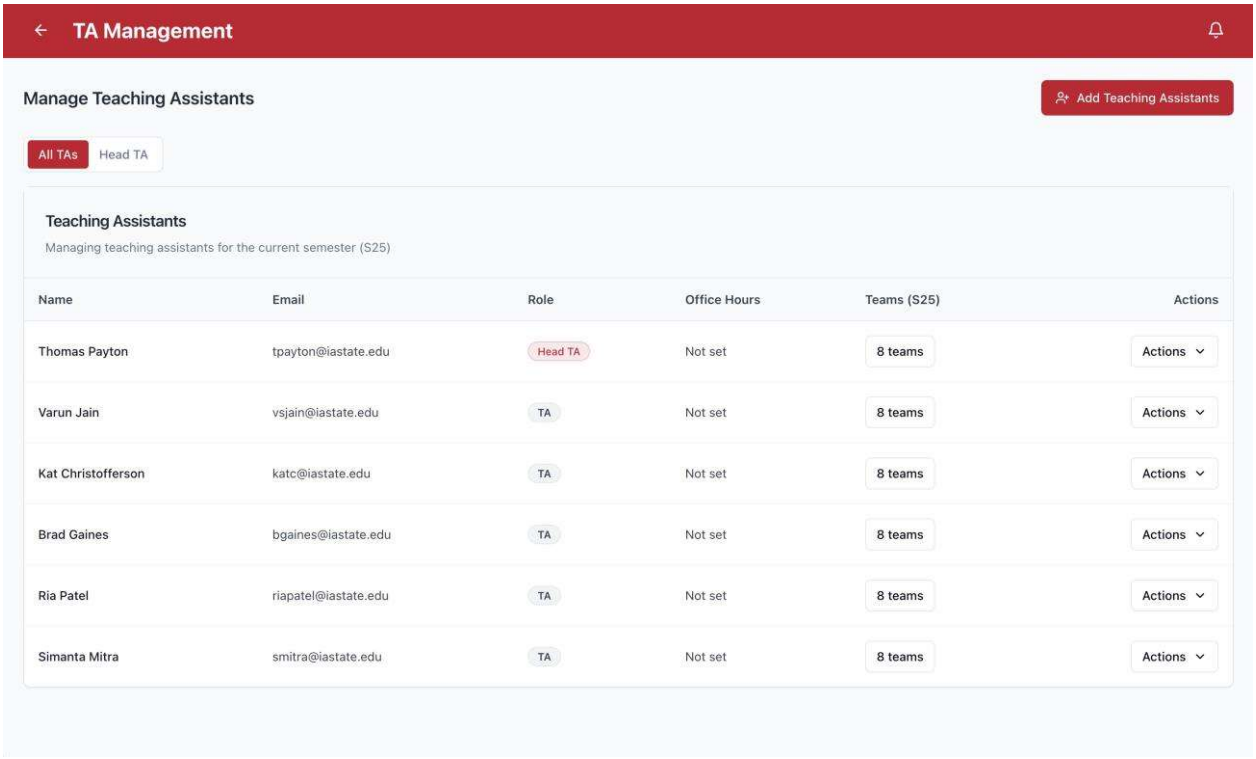


Figure 14 – TA Management Page

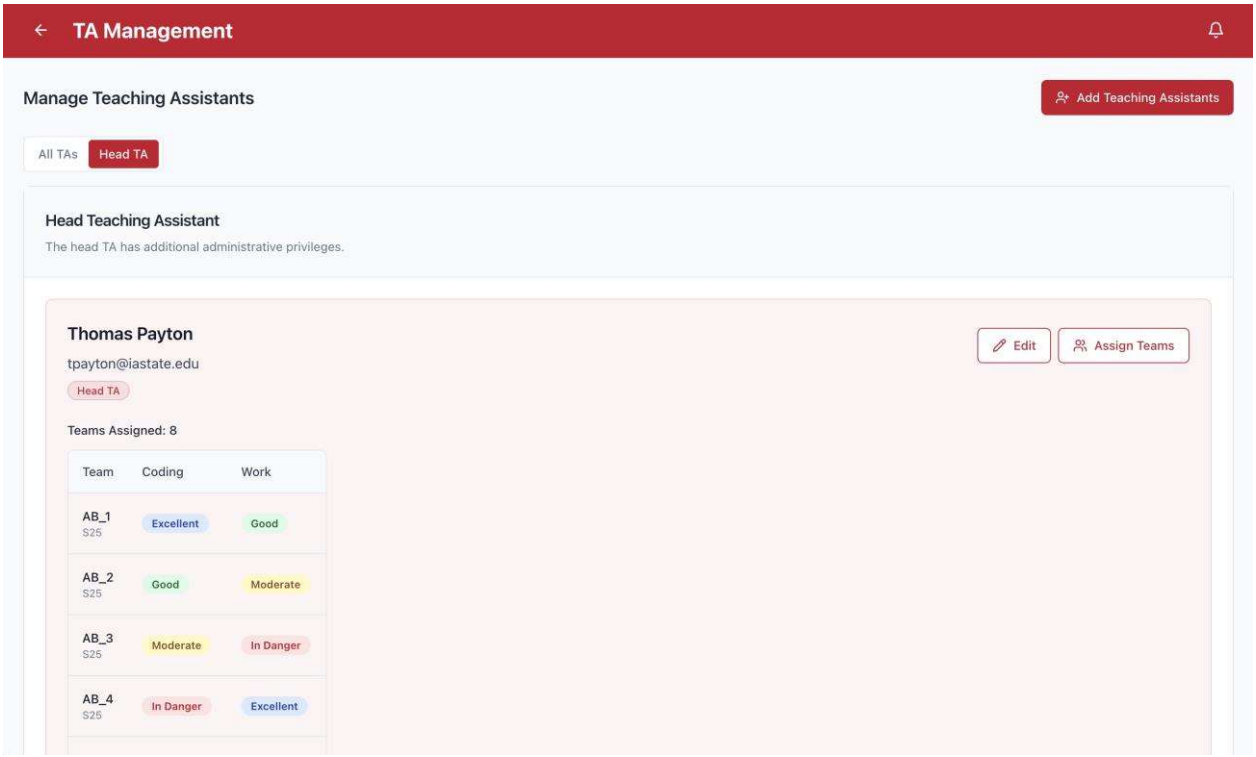


Figure 15 – TA Management Breakdown Page

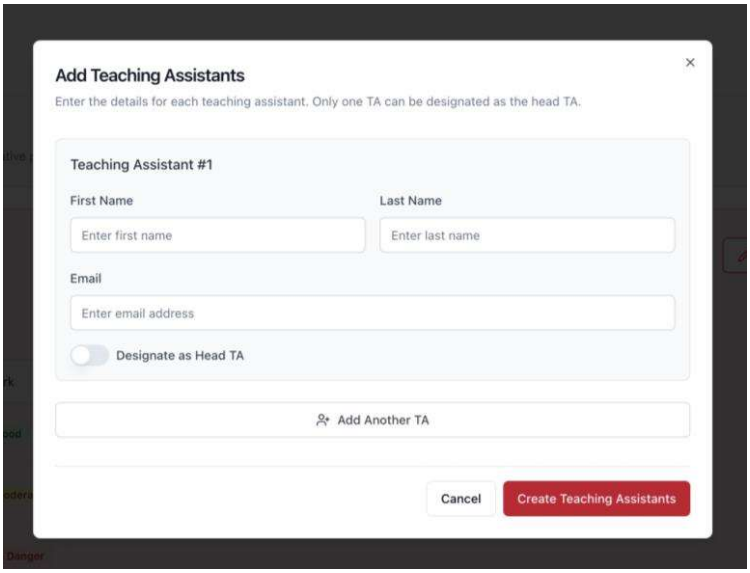


Figure 16 – Entering TA Page

The team details page, shown below, shows the team's name, the students in the team, team information (teaching assistant, and git repository), team demo progress information, CATme results, and team comments. Clicking on the individual students will lead to the individual student's page which includes the student's name, email, team, teaching assistant, at-risk status, attendance record, and comment section. The comment sections allow professors, head teaching assistants, and teaching assistants to internally converse about the students/teams. The comment allows for statuses of action taken or action not taken. Information on team details and student details pages can be edited by the appropriate user.

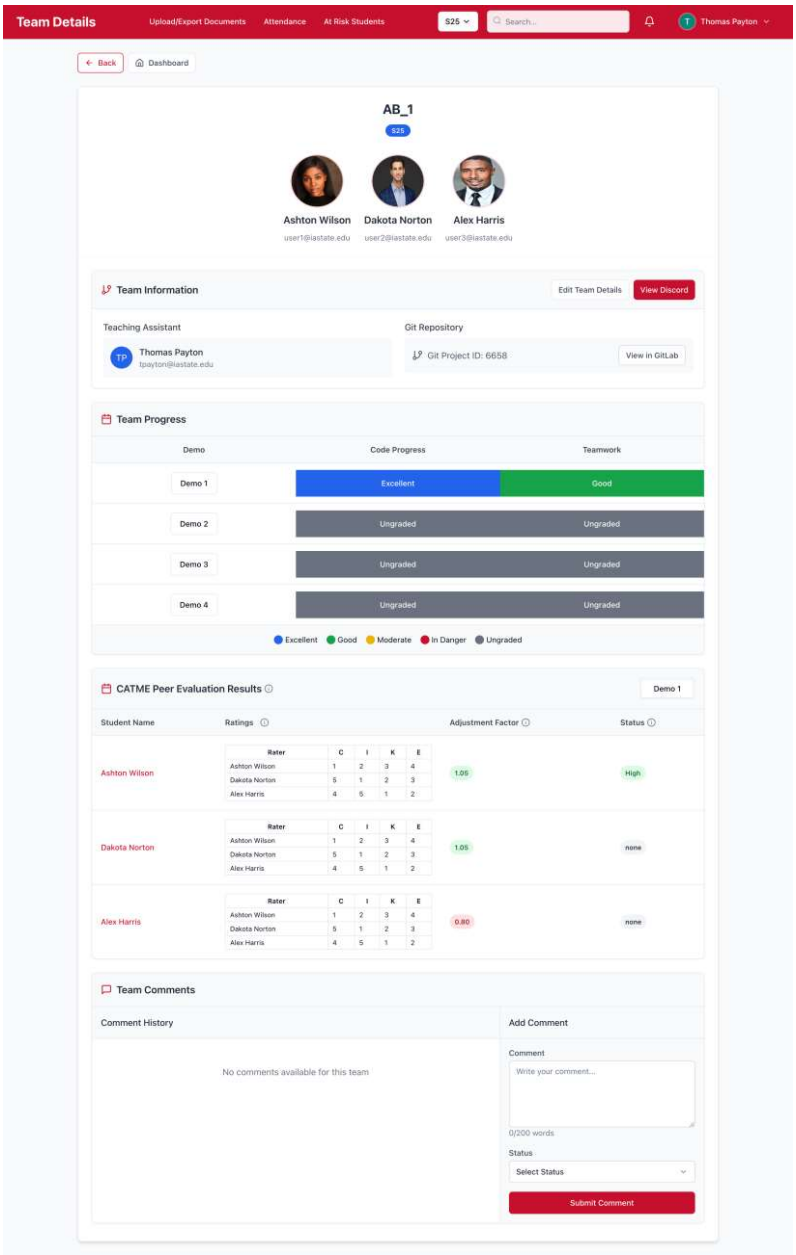


Figure 17 – Team Details Page

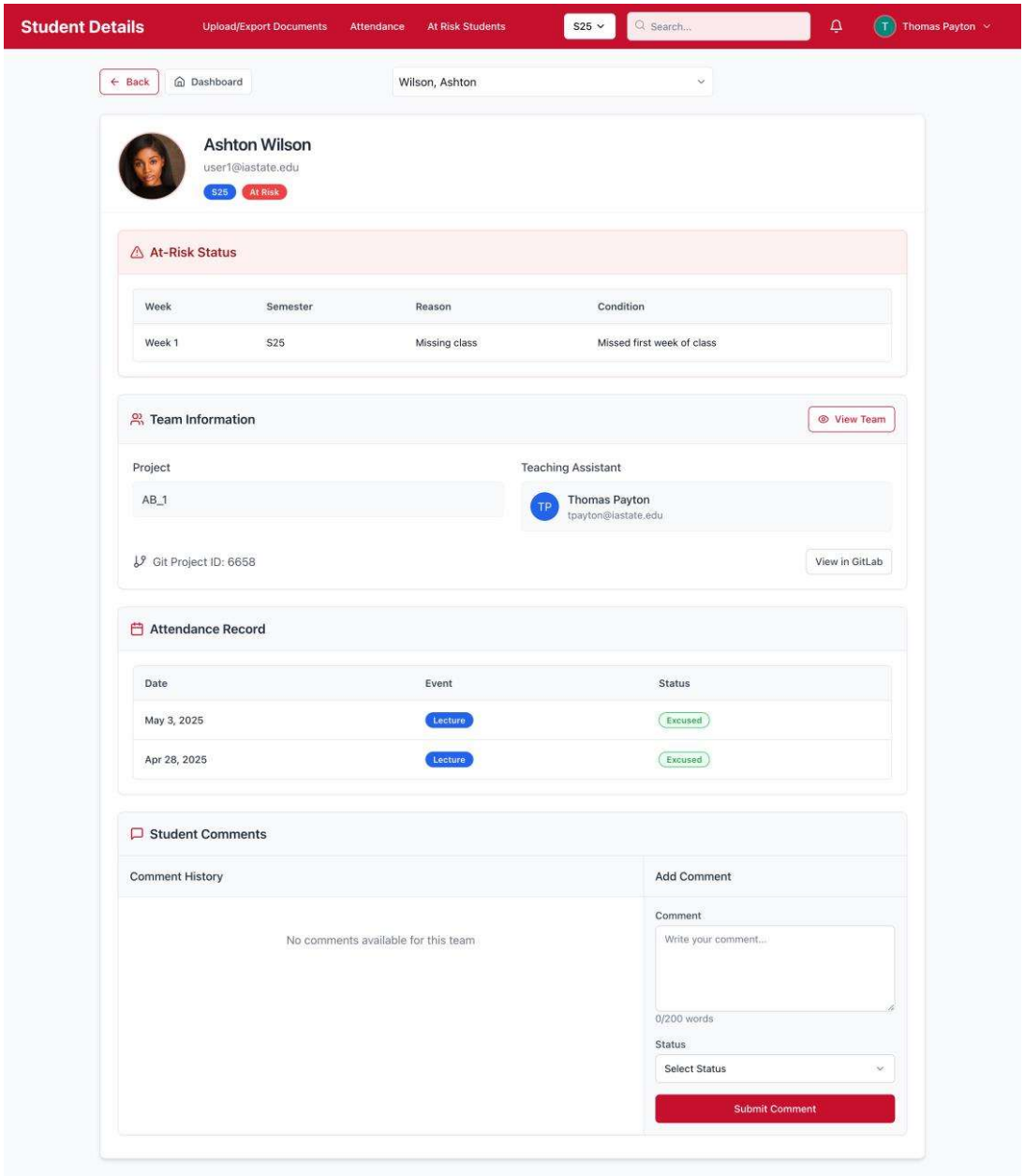


Figure 18 – Student Details Page

Below is an image of what the home page would look like after archiving the dashboard semester data.

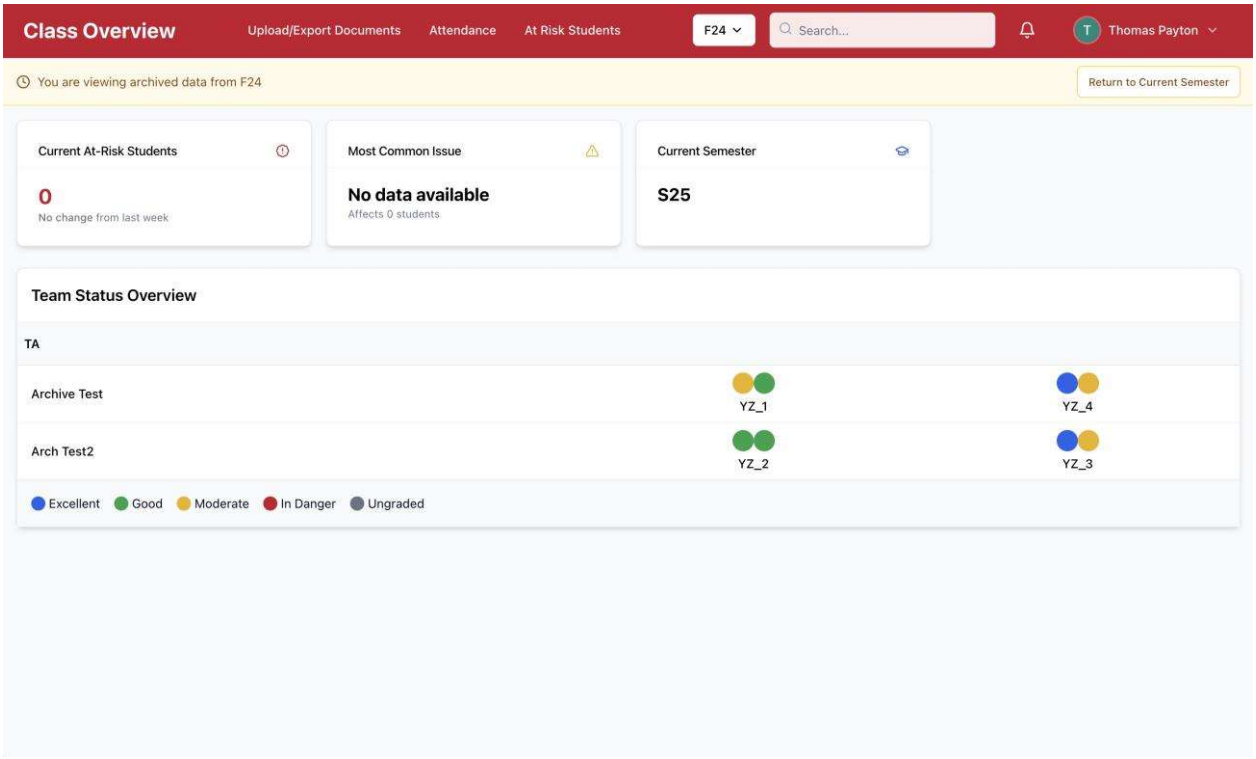


Figure 19 – Archived Example of Home Page

Below is an image of the settings page. The professor is the only user with access to this page as it allows access to archive the dashboard.

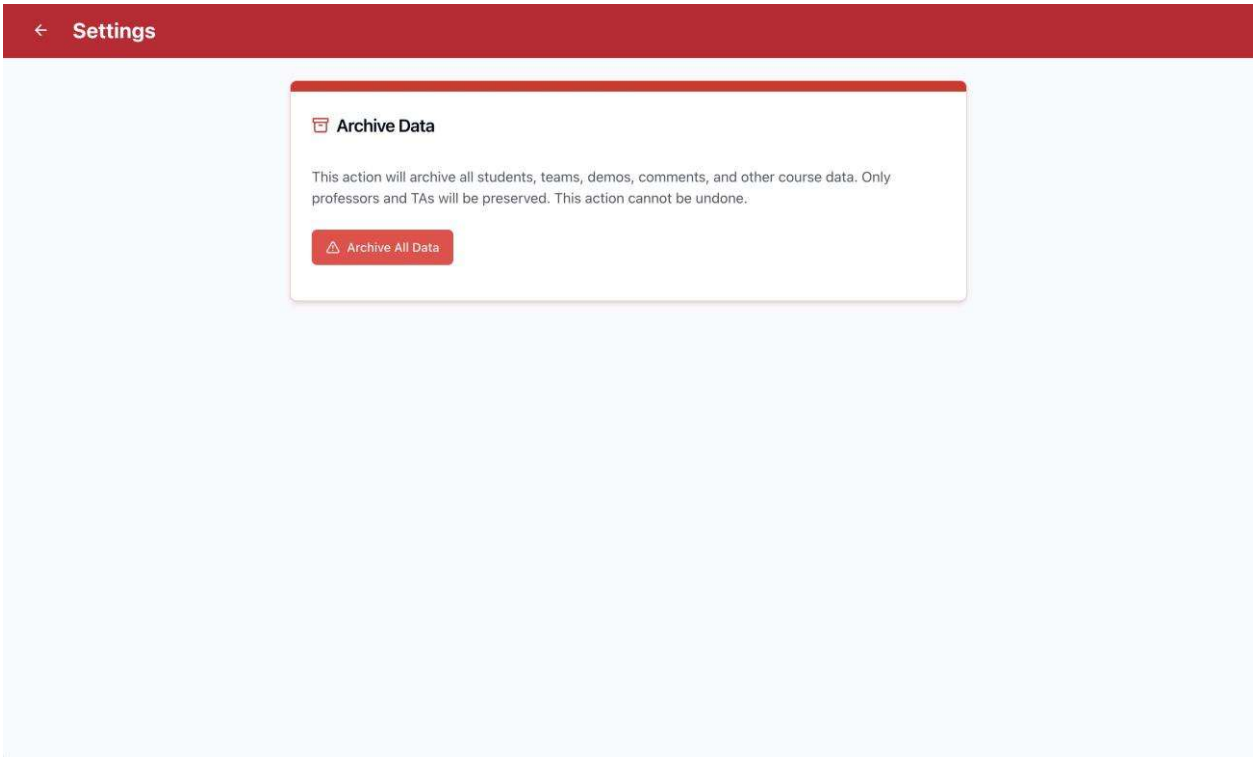


Figure 20 – Settings Page

The backend was primarily organized using an ER diagram. The diagram shown below shows the different tables needed to organize the data that we fetched from CSV files. Each table is organized by the topic and followed by the information that is accessible if granted access to that table. Connections are shown through lines, enabling data to work together to make sure the data represented on the dashboard is fully encompassing the available data.

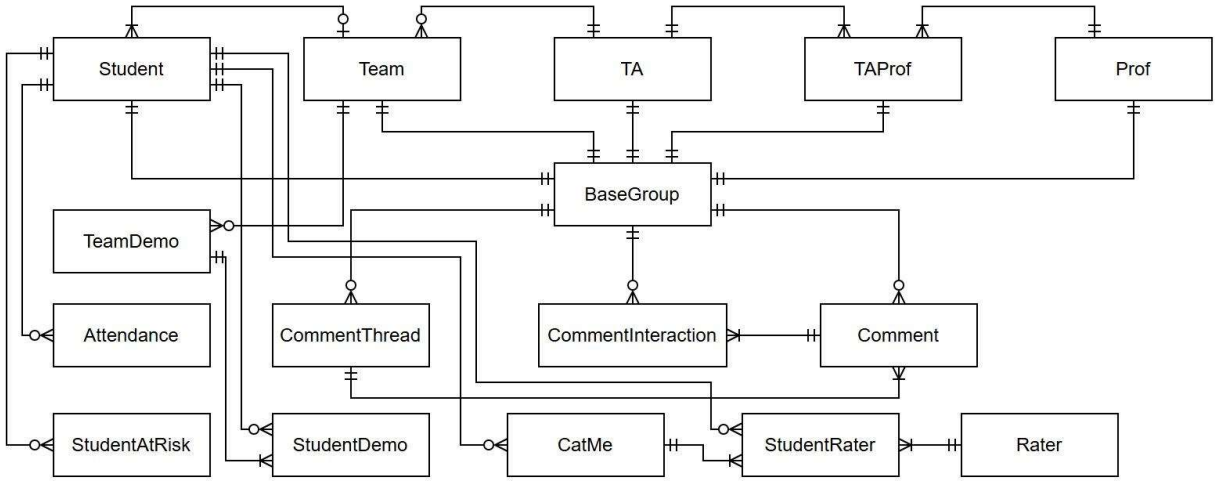


Figure 5 – ER Diagram

The database has been designed to separate unrelated data as much as possible. The ER diagram above shows the relationship between tables: one (one line), many (three-line prong), one and only one (two lines), zero or one (circle and line), one or many (one line and three-line prong), and zero or many (circle and three-line prong). We used multiple tables to help with organization and efficiency. As seem above, tables include Students, Team, TA, TAProf, Prof, BaseGroup, TeamDemo, Attendance, StudentAtRisk, CommentThread, StudentDemo, CommentInteraction, Comment, CatMe, StudentRater, and Rater.

As for security, we encrypted all data flowing in and out of the dashboard. It was very important that we follow FERPA and other academic safety standards throughout the dashboard. In order to prevent outside infiltration of our data, we encoded all data coming in and out of the dashboard. Once a data file is uploaded to the dashboard, the data is secured within the application. As for within the dashboard, we made sure our page designs respected the privacy of our users by not including grades on the home page or in large fonts.

4.3.3 FUNCTIONALITY

The dashboard was intended to be used by professors and teaching assistants of Computer Science 3090. Everyone has access to the dashboard. We gave the professor access first, so to set up the dashboard with assignments and student profiles. From there, teaching assistants had access to make any changes to their groups if necessary. Our dashboard is designed to work for multiple semesters, meaning that it has a reset function in which the professor will need to input student/group information prior to the start of each

semester. All data collected throughout the semester will be stored in the backend, for future use if needed. We also have a function where the professor can upload a file of student names and their groups, in which the dashboard will assign each student a profile within their group.

All page data is populated from CSV files uploaded by the professor. The dashboard gives guidance as to what form of data needs to be given in order for the data to be accurately parsed and represented in the dashboard. Further details on uploading data is given in the User Manual (A5.1), as it mostly pertains the professor.

Once the dashboard was completed, the professor had full control over the dashboard. This allows the professor to make any changes to the dashboard as desired. The professor must input student information at the beginning of the semester, and wrap up the semester dashboard, in order for the dashboard to work to its best capabilities.

4.3.4 Areas of Challenge

The design and implementation of our project satisfies all users greatly. The professor is also our client, which made it easy for us to adjust the dashboard to fit his needs. Most of our group has taken the course prior to this semester, and the feedback from their experience allowed us to meet all student needs as well. If the design does not fit the needs of the users in the future, the professor has full access to make adjustments as needed.

Our primary area of challenge was the constant adjustment of pages. With biweekly meeting with our client, the desires of our client changed quite frequently. Our goal was to create the most efficient dashboard for our client, which meant that any changes requested were implemented. Our pages were being updated every week. While the changes were not drastic, the connection with backend created a small obstacle. Any change made in the frontend needed to be changed in the backend database as well. The constant changes on the pages were time-consuming for our group. In the end, we were able to present a completed dashboard that satisfied all requests made by the client.

4.4 Technology Considerations

The primary technologies we chose to use include Next.JS for the frontend, Node.js with TypeScript for the backend, Iowa State University Server (ISUS) for hosting, and an MySQL database for data management. For the frontend framework, we selected Next.JS because it is a more modern evolution of

React. Additionally, Next.JS provides several advantages including: simplified routing capabilities, more efficient page loading through server-side rendering, and improved development experience through built-in optimizations. These features allowed us to create a more responsive and user-friendly dashboard while maintaining clean and maintainable code.

The backend utilizes Node.js with TypeScript, a choice that created an ideal development environment since both frontend and backend use JavaScript-based technologies. TypeScript adds strong typing to JavaScript, which helped prevent runtime errors and improve code maintainability. This combination allowed our team to work more efficiently as we could share code and knowledge across both frontend and backend development.

For hosting, we used an Iowa State University server in production and a server provided by Iowa State University for development and testing. The university servers provided a controlled testing environment that closely mirrored the real-world setting.

Our database choice of MySQL over NoSQL was driven by the relational nature of our data. The dashboard primarily deals with structured data with clear relationships between entities (students, groups, assignments, etc.). MySQL databases are much better at handling such relationships and maintaining data integrity through features like foreign keys and transactions. This made it easier to maintain consistency across different aspects of the dashboard, such as ensuring student data correctly relates to their respective groups and assignments.

These technology choices involved certain trade-offs. While Next.JS offers many modern features, it has a steeper learning curve compared to traditional React. The use of TypeScript adds development overhead but provides better code safety. The MySQL database choice might have made some types of data changes more complex compared to NoSQL solutions, but the benefits of data integrity and relationship management were worth it for our application.

5 TESTING

Testing for our project was uniquely designed. Since our project took on a waterfall and agile management style, we had to adjust how we categorized tests. Our original prototype testing included Figma mockups of the frontend design and schematics of the backend organization. The best testing style for our project at the agile stage included adding features to the dashboard and then verifying that they worked correctly without causing harm to already implemented features. Additionally, after we'd verified the new features don't impact others, we presented the features to our client to ensure they were worth including or if changes should've been made. Therefore, testing was done continuously throughout our development stages using Postman for testing assistance.

Our testing plan was unique because we did not follow the traditional prototypes, testing stages, and adjustments. Instead, we had an original prototype mockup that we followed when setting up our initial website. Once the website base was completed, we started testing by adding features on the frontend and adjusting our backend to help ensure data was correctly organized for the frontend. Manual testing was completed after every additional feature was added. This ensured that an additional feature being implemented did not interfere with any previously implemented features. If errors occurred in either testing stage, we focused on resolving the issue before continuing development of the feature. Further unit and site reliability testing was completed once the dashboard was developed to a sufficient state that the client was happy with. This was to avoid spending too much time on tests that could have become obsolete depending on what features we chose to include in the final form of the dashboard. Initial security testing was completed when adding login authorization and verification to the dashboard. Further security testing related to data security was tested once the dashboard was near completion.

Due to the nature of our project being a web application, other testing such as environmental testing, safety testing, and field testing, were not applicable. We took all necessary precautions when it came to functional, performance, reliability, and security testing. Final testing stages included demoing the dashboard to teaching assistants of Computer Science 3090 to ensure the dashboard provided all assistance possible to their needs.

5.1 UNIT TESTING

Unit testing was not heavily applicable to the dashboard. Our only source of units were students. We populated the dashboard with mock data in order to confirm that the dashboard was able to withstand the number of students' data that would match a typical semester. We ran through all features of the

dashboard with this database, ensuring that all features were accurate and sustainable. No other unit testing was necessary for the dashboard.

5.2 Interface Testing

Interface testing was heavily applicable to the dashboard. Since the dashboard's medium was a user interface, we tested heavily throughout, ensuring that the interface was accurate to our design. Interface testing was done in sprints, all following a similar format: design a page, implement the page, adapt the page to fit all needs of the client, test the page with mock data, ensure no other pages were harmed. With the addition of any new page, this cycle was repeated. No external tools were used nor needed for interface testing.

5.3 Integration Testing

Integration testing was heavily applicable to the dashboard. Since all pages relied on the database organization in the backend, integration testing was done continuously in sprints. These sprints followed a similar format to interface testing, including a cycle of: full completion of a page, communication between frontend and backend about how the data needs to be organized, update the backend database to be organized efficiently and effectively, manual testing to ensure the frontend page was accurate, ensure no other pages were harmed. With the addition of any new page, this cycle was repeated.

Backend testing was mainly done using Postman. Postman is an all-in-one API platform that allows developers to build and test APIs. For our project, we used Postman to manually test all of our APIs, including GET, POST, and PUT requests. Postman was very useful and fairly easy to use. As you can see from the image below, to test an API call, you need to provide the method, URL, and any other parameters you want to send, such as a JSON body or a file. After the request is sent, Postman returns an HTTP status code and the response body. This was very helpful, as it informed us whether the request was made successfully and correctly. To test other API calls, we simply changed the method, URL, and parameters as needed. Using Postman was very efficient, as the backend team does not need to rely on the frontend to verify if the requests are made correctly or successfully.

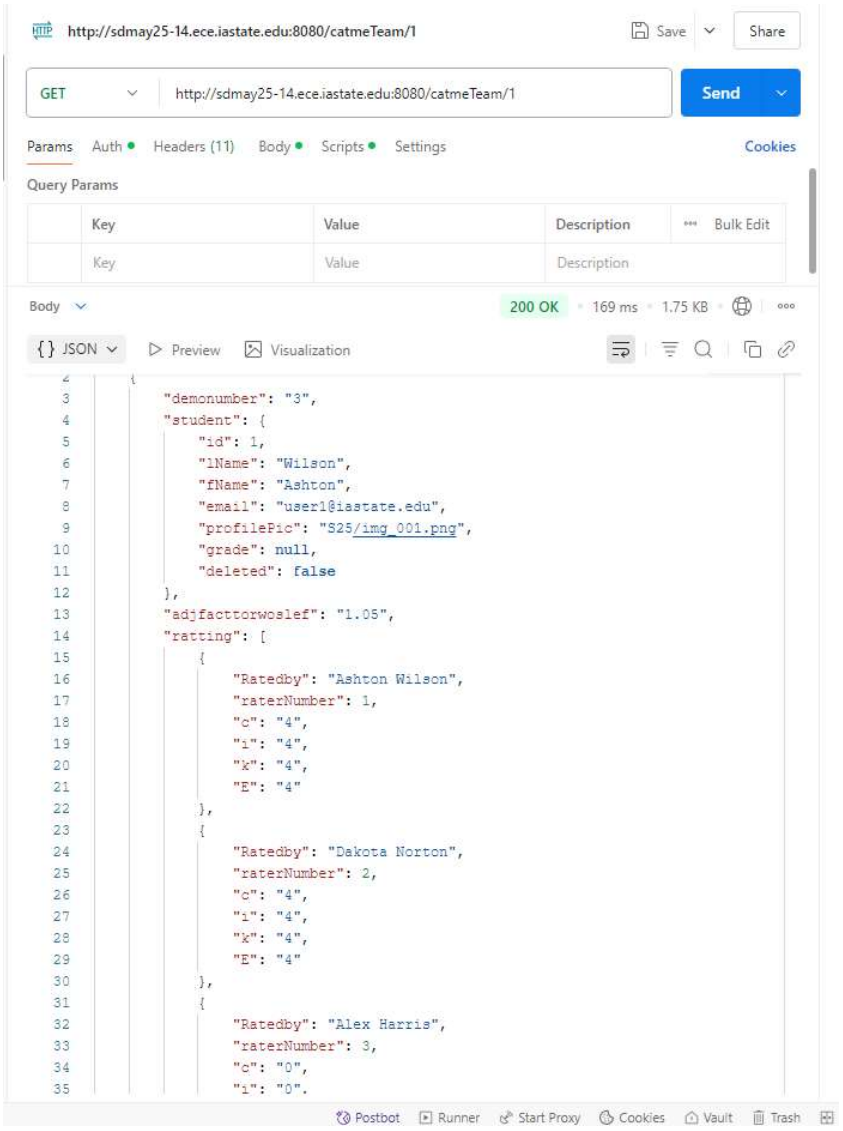


Figure 21 – Postman Testing

5.4 System Testing

System testing was not heavily applicable to the dashboard. Due to the style of interface and integration sprint tests, the dashboard was tested in its entirety after the addition of every new page. We completed final system testing at the end of our project to ensure that the archival system worked as designed. This allowed for us to also test populating the dashboard as would be done for a new semester. No external tools were used nor needed for interface testing.

5.5 Regression Testing

Regression testing was heavily applicable to the dashboard. With the new addition of any page, we tested both the frontend interface and the backend database to make sure no past pages were harmed. Since we completed system testing after every new addition of a page, regression testing issues were minimal. No external tools were used nor needed for interface testing.

5.6 Acceptance Testing

Acceptance testing was heavily applicable to the dashboard. Since the dashboard requirements were very flexible, we needed to make sure all features were created in the best interest of the client. We met with the client every week to present the dashboard status, converse about desired adaptations, and converse about the next desired features to be implemented. The weekly meeting provided us with ease to manage our time efficiently due to the instant feedback we received. Final acceptance testing was completed when we demonstrated the final dashboard to the client.

5.7 User Testing

User testing was heavily applicable to the dashboard. Our primary user was the professor, secondary was the head teaching assistant, and tertiary was the other teaching assistants. Our focus was on the needs of the primary user, the professor. As a result, we met with the professor every week to ensure all needs were addressed. Once we received acceptance of the dashboard from the professor, we moved onto the head teaching assistant's needs. Due to this transition occurring late in the project timeline, we were unable to meet all needs of the head teaching assistant. All needs/desires of the head teaching assistant that were unable to be implemented due to timing constraints are listed in 8.3 Next Steps section. Our primary user was very satisfied with the status of the dashboard and was excited to start implementing it into the course Computer Science 3090.

5.8 Results

The results of our testing were excellent. We ran into minimal issues with unit, interface, integration, system, regression, and acceptance testing. User testing was satisfactorily completed for our primary user, but not for the secondary and tertiary. Our primary goal was to address the needs of the

primary user, so our project and testing was a success. We are extremely satisfied with the status of the dashboard and are excited to see how helpful our project is to the Computer Science 3090 faculty. With minimal issues in testing, we were able to create the dashboard to function perfectly with the features that were implemented during our project timeline.

6 Implementation

Our project was built in its entirety, satisfying all requirements requested by our client, Dr. Mitra. The dashboard was implemented to match our final design perfectly. All pages are working, with the accurate database being filled by CSV files uploaded by the professor. Our final design includes a dashboard that allows for easier navigation and organization for the course Computer Science 3090 at Iowa State University. The final implementation includes pages for: login authentication, home page, upload documents page, attendance page, at-risk students page, user profile pages, TA management page, team breakdown pages, and student breakdown pages. There were no features, functions, or sub-systems that were included in our final design that were unable to be implemented.

6.1 Design Analysis

Our implemented design works great. All requests were satisfied, leading to a dashboard that will help with navigation and organization for the course Computer Science 3090. We are most proud of the organization within the dashboard, providing easy use for all users. With the help of breakdown pages, we provided a dashboard that gives the users quick information if needed, or the ability to access all data necessary to review students and their teams' progress. We have shown the client the dashboard in its entirety, to which he was very pleased. Our evidence to show how well the dashboard works was measured by the satisfaction of the client, which was very high.

At this time, there is nothing about the dashboard that does not work well or as expected. We started implementation of the dashboard very early on, allowing us to have maximum time to complete the dashboard to fit all of the client's needs and desires. Given the chance to start fresh, the only thing we wished we approached differently was meeting with the head teaching assistant earlier on. But, this was not feasible for our situation due to our goal to satisfy the needs of our primary user prior to addressing the secondary and tertiary users. We are thrilled that we were able to provide a dashboard that will truly help the faculty of Computer Science 3090.

7 Ethics and Professional Responsibility

Our team approached engineering ethics and professional responsibility with a stakeholder-centered ideology, with a particular focus on data privacy and community benefit. We defined engineering ethics in our project as the principles that guided our development decisions, especially regarding data handling and user privacy. Professional responsibility included our commitment to delivering a reliable, secure, and practical dashboard for Dr. Mitra, teaching assistants, and future Computer Science 309 students.

To uphold said principles, we had taken the following steps. First, we maintained strict data handling protocols, included role-based access control, ensured that sensitive information was accessible only to authorized users. Second, we actively worked with the users of the dashboard during development through regular meetings and iterative development, which allowed us to continually improve the dashboard and align it with actual user needs and expectations. Third, we focused on efficient development practices by optimizing our server infrastructure and resources used and implementing practical data storage solutions that balanced functionality with security. These steps ensured that our project not only met technical requirements but also adhered to ethical standards.

No ethical and professional responsibility perspectives were changed after our capstone semester 1. Since we worked on the implementation and construction of our project beginning at the end of capstone semester 1, we were able to maintain the same perspectives throughout the capstone semester 2.

7.1 Areas of Professional Responsibility/Code of Ethics

Area of Responsibility	Definition	Relevance to IEEE Code of Ethics
Work Competence	Reliable, timeliness, high quality, and professional work.	“to uphold the highest standards of integrity, responsible behavior, and ethical conduct in professional activities.”
Financial Responsibility	Economically efficient resources and economically reasonable products.	“to avoid unlawful conduct in professional activities, and to reject bribery in all its forms.”
Communication Honesty	Truthful and understandable work.	“to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest and realistic in

		stating claims or estimates based on available data, and to credit properly the contributions of others."
Health, Safety, Well-Being	Minimal safety, health, and well-being risks.	"to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment."
Property Ownership	Respect for property, ideas, and client information.	"to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest and realistic in stating claims or estimates based on available data, and to credit properly the contributions of others."
Sustainability	Protect the environment and nature.	"to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment."
Social Responsibility	Beneficial products and services for society.	"to improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems."

Our team performed very well in social responsibility. Social responsibility was very relevant to our project because the dashboard was centered around helping individuals associated with the course Computer Science 3090. Our approach focused on providing the best solution for the Computer Science

3090 community, putting their needs first. Our approach upheld ethical and professional responsibilities by being respectful and private with the community's data.

Our team was performing not so well in sustainability. Sustainability was not very relevant to our project besides ensuring that we used the least number of resources (servers) as possible. Our approach focused on using one server while calling data from the other applications instead of storing said data in order to minimize storage needs. While we couldn't change this approach for the better, we were committed to continuing to evaluate how to build our database off of one server with minimal storage capacity.

7.2 Four Principles

	Beneficence	Nonmaleficence	Respect for Autonomy	Justice
Public Health, Safety, and Welfare	Reduces stress of finding class resources.	Does not require users to adjust.	Users can choose their involvement level.	All users have availability to use.
Global, Cultural, and Social	Designed for Computer Science 3090 community.	Users are not required to use.	Users can choose to use.	All users have availability to use.
Environmental	Minimal resources used.	No download required.	Users can access simply with a web address.	Causes no harm to the environment.
Economic	Minimal resource costs.	No purchase required.	Free for all users.	Easily accessible for Computer Science 3090 community.

The broader context-principal pair that was most important to our project was beneficence in relation to global, cultural, and social. Our primary goal of the dashboard was to create an application that would benefit all individuals involved in Computer Science 3090 by providing easier access to organized data, materials, and resources. In order to achieve this, we ensured that the dashboard did not cause any harm to our users, but rather made access to their information easier. We created this dashboard with the primary focus of making their lives easier in relation to the course.

The broader context-principal pair that was lacking in our project was beneficence in relation to economics for the school. We added a new tool to the Iowa States systems just for one class and this added more management, complexity, and room for something to go wrong. Rather than using previously

purchased and currently implemented tools we built something new that is based off some of those tools. Of course, the dashboard was the deliverable and primary focus of this project, but it inherently lacked in this context-principle. We made up and justified this deficit with the benefits found in global, cultural, social, and public safety and welfare. This tool greatly improved the lives of all people in Computer Science 3090 from professor down to students, by making a free easily accessible and improved tool to manage the class.

7.3 VIRTUES

Three virtues that were extremely important to our team were honesty, justice, and compassion. We found that these virtues related most to the dashboard as its main goal was to help individuals involved with Computer Science 3090. We had to act in the best interest of our client and his teaching assistants in order to provide a dashboard that benefitted the Computer Science 3090 community. Acting in consideration to honesty, courage, and compassion could be seen throughout the dashboard in every decision especially our privacy and security initiatives.

Bradley had demonstrated curiosity throughout this project. He continuously jumped into any role needed throughout this project and helped out on both the front end and back end. Curiosity was important to Bradley, and most of all our group, because he was able to learn about both ends of the project and help his teammates when needed. One virtue that Bradley had not demonstrated greatly throughout this project was creativity. While Bradley was very knowledgeable, he completed his parts to what the group decided without adding a unique perspective. While this is typically a bad characteristic, since we were working collaboratively on this project, it had worked in favor of our group.

Breckin had demonstrated respect throughout this project. He was always considering our team members and the members of the Computer Science 3090 community. Respect was important to Breckin as he prides himself in being a reliable and understanding team member. One virtue that Breckin had not demonstrated greatly throughout this project was trust. Breckin was very confident in his own abilities to the point that he did not willingly allow other team members to work parallel with him. Thankfully, he had opened his trust up to a few team members so we could all be collaborative on the project.

Kat had demonstrated commitment throughout this project. She was always willing to work on the project even parts outside of her scope. She had not feared learning new things in order to make sure our project was pristine. Commitment was important to Kat because she takes pride in the work she completes. One virtue that Kat had not demonstrated greatly throughout this project was contentment. Kat was never

satisfied with the work she completed and constantly strived to improve it. While this was typically a strong characteristic, it had caused slight timing issues for our group deadlines.

Ria had demonstrated service throughout this project. She was always willing to take on the documentation and presentation assignments. Service was important to Ria because she wanted to make sure anyone outside of our team was able to understand the scope of our project, and hopefully learn from what we completed. One virtue that Ria had not demonstrated greatly throughout this project was helpfulness. Ria does not have a vast amount of previous experience with the knowledge needed for this project. While she had been very willing to learn, she does not bring easy answers to the table.

Thomas had demonstrated determination throughout this project. He was always working on the front end and updating the team of what he had completed. Determination was important to Thomas because he always wanted to make sure he was completing his work to the best of his ability. One virtue that Thomas had not demonstrated greatly throughout this project was cleanliness. Thomas was very intelligent, so his code was strong, but it was hard for another member to understand his logic because he does not comment throughout his code. Although this could have been catastrophic, Thomas had begun to comment throughout his code so that when we handed over the project to the client, updates will be easier to navigate.

Varun had demonstrated gentleness throughout this project. He was always helping other team members on their parts since he possesses prior experience with back-end development. Gentleness was very important to Varun because he wanted to make sure he was helping his team members succeed whenever possible. One virtue that Varun had not demonstrated greatly throughout this project was patience. While Varun was not angered by his impatience, if group members had not completed their parts, even if prior to the deadlines, Varun would step in and complete it instead. While this was not very helpful for our group since we had never been behind on deadlines, we appreciated Varun's tenacity to get work done.

Curiosity, respect, commitment, service, determination, and gentleness had all been greatly influential in the project that we had completed together. Individual creativity, trust, contentment, helpfulness, cleanliness, and patience were all works in progress throughout our team. Although each member stated a virtue that they felt was lacking throughout our project, our group strongly believes that all of these virtues were not harmful to our group dynamic or project success. We had all adjusted our work styles in order to create a solid team that benefitted off of our individual strength.

8 Closing Material

8.1 Summary of Progress

Our team had made significant progress in developing the Computer Science 3090 dashboard. We had successfully established the foundational architecture, including the frontend interface with Next.JS, backend infrastructure with Node.js and TypeScript, and security precautions. We completed implementation of all pages/features requested by the client. Our primary goals remained focused on creating a secure platform that integrated data from multiple sources while maintaining user privacy and providing role-specific features for professors and teaching assistants. The dashboard was completed to the satisfaction of our client. As a result, we have handed over all rights and control of our project to our client, Dr. Mitra. We wish him the best using the dashboard for future semesters of Computer Science 3090.

8.2 Value Provided

The dashboard perfectly addresses the needs of our client and users. Since we worked parallel with our client and users, we were able to ensure that the dashboard fits all of their needs and desires. We were able to address and create solutions throughout the dashboard for major and minor problems that our client had. We were even able to address some cosmetic desires for the dashboard to ensure efficiency for the client. The dashboard was designed for Computer Science 3090 but could potentially be adjusted to be used in other courses that are experiencing similar organizational and efficiency issues.

8.3 Next Steps

While our work on this project may be coming to a close, we truly enjoyed working on this project. We would love to see future groups make additions and adjustments to our project. There are numerous additions that could be beneficial to the dashboard's success including, but not limited to:

- Users' ability to edit student information or remove students from teams, allowing updated even after the student database is created at the beginning of the semester. Changes would be made if students drop out or when modifications occur.
- Users' ability to upload team proposal files on their team page.

- Users' should be able to generate a report file from team status to share publicly with students. The report could be filtered by section, TA, or group.
- Users' ability to filter and view at-risk students by week, demos, TAs, or sections.
- Users' ability to upload CATme highlights per team on their team's profile page for each demo separately.
- A page for team flow charts including coding and teamwork statuses during demos or weekly meetings. Charts should be available for download to share with teams.
- Ability to tag a comment to a student, or multiple students, within the team details page to allow for more flexible feedback.
- Integration with Microsoft for connections to Iowa State University's authentication system.
- Further security for the dashboard to ensure information is confidential and inaccessible to others outside of the Computer Science 3090 community.
- Users' ability to complete weekly meeting forms through this dashboard.
- Instructor and head teaching assistant ability to define and assign tasks to teaching assistants in the form of title, description, deadline, view status, completion status, and the teaching assistants assigned to the task.

We worked really hard on the dashboard to ensure that all of our client's needs would be addressed, and we were successful in that. Any issues that we ran into were solved with alternative solutions that were approved by our client, therefore not needing more attention. Most of the future additions were vocalized by the current head teaching assistant, Mahdi BanisharifDehkordi. Due to our meeting with him being late in the semester, we were able to list features that would be beneficial for teaching assistants. The dashboard was primarily focused on the professor, in which it meets all needs for, but future additions would be in the best interest of the teaching assistants.

9 REFERENCES

[1]"TYPEORM - AMAZING ORM FOR TYPESCRIPT AND JAVASCRIPT (ES7, ES6, ES5). SUPPORTS MYSQL, POSTGRESQL, MARIADB, SQLITE, MS SQL SERVER, ORACLE, WEBSQL DATABASES. WORKS IN NODEJS, BROWSER, IONIC, CORDOVA AND ELECTRON PLATFORMS.," TYPEORM.IO. HTTPS://TYPEORM.IO/

[2] Shadcn/ui, "Introduction," shadcn/ui, https://ui.shadcn.com/docs

[3] jwt, "JWT.IO - JSON Web Tokens Introduction," jwt.io, 2024. https://jwt.io/introduction

[4] "NextAuth.js," next-auth.js.org. https://next-auth.js.org/

[5] A. Deshpande, A. Kulkarni, D. Stone, R. Cherian, and N. Eltoum, "Group 9 -309-Dashboard." Accessed: Dec. 05, 2024. [Online]. Available: https://seniord.cs.iastate.edu/2023-Jan-09/files/inline-files/402C%20Demo2.pdf

[6] "Rest api," GitLab, https://docs.gitlab.com/ee/api/rest/ (accessed Dec. 5, 2024).

[7] A. Deshpande, A. Kulkarni, D. Stone, R. Cherian, and N. Eltoum, "309 Dashboard -Team 9." Accessed: Dec. 05, 2024. [Online]. Available: https://seniord.cs.iastate.edu/2023-Jan-09/files/inline-files/402C%20Demo%201_0.pdf

[8] Vercel, "Next.js by Vercel - The React Framework," nextjs.org, 2024. https://nextjs.org

[9] A. Hanrahan, E. Peterson, N. Hoss, E. Danforth, and L. Thunga, "309 Dashboard." Accessed: Dec. 05, 2024. [Online]. Available: https://seniord.cs.iastate.edu/2022-May-07/files/inline-files/402%20Final%20Presentation.pdf

10 APPENDICES

APPENDIX 1 – OPERATION MANUAL

A1.1 MYSQL WORKBENCH DATABASE SETUP

- Click the "+" icon next to "MySQL Connections" on the home screen.
- Create a name for your database e.g 309 Dashboard.
- Hostname IP: 127.0.0.1.
- Port: 3306.
- Set *mysql database password.*

A1.2 ENTERING CREDENTIAL IN DATABASE

- In **base_group** table, add ID of 1 and set deleted to 0. Apply the change.
- In **professor** table add you first and last name, email, base group id which is 1. Set deleted to 0. After applying this, you will be able to login and access the dashboard.

A1.3 GIT CLONE

- Git clone https://git.ece.iastate.edu/sd/sdmay25-14.git.

A1.4 .ENV SETUP

- File Directory Structure (\home\username):
    - .env
    - sdmay25-14
    - Other files and folders
- JWT secret generation in terminal all in one line.

    node –e "console.log(require('crypto').randomBytes(32).toString('hex'))"
- .env file structure:

    JWT_SECRET= *generated jwt secret*

    DB_HOST=127.0.0.1

    DB_PORT=3306

    DB_USERNAME= *mysql database username*

    DB_PASSWORD= *mysql database password*

DB_NAME= *mysql database schema name*

PORT=8081

A1.5 DEPLOYMENT

I. Pre-run setup:
- cd sdmay25-14
- git checkout deployment
- npm install
- cd backend
- npm install
- cd ../frontend
- npm install

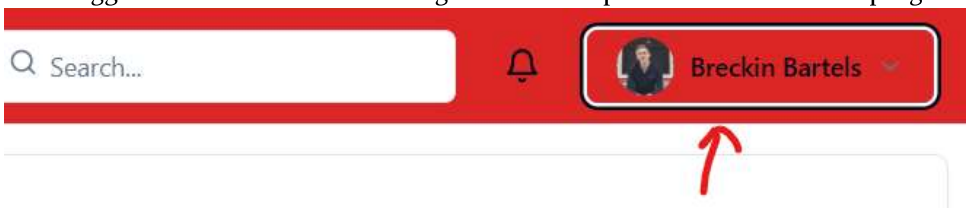
II. Running the backend (port 8081):
- cd backend/src
- screen
- npx ts-node index.ts
- ctrl-AD

III. Running the frontend (port 3000):
- cd frontend
- screen
- npm run dev
- ctrl-AD

IV. Updating the server:
- cd sdmay25-14
- git checkout deployment
- git pull
- fuser –k 8081/tcp
- fuser –k 3000/tcp
- Restart the backend using “Running the backend” instructions
- Restart the frontend using “Running the frontend” instructions

A1.6 How to populate Teaching Assistants, Teams, and Students

I. Once you are logged in to the dashboard navigate to the dropdown menu on the top right.



II. Select TA management:
   a. Select the Add Teaching Assistants button in the top right.
   b. Add all teaching assistants.

III. Navigate back to dashboard->Upload Document.

IV. Select "Teams and Students" as an option.

Teams and Students File Structure:

```
uploadData.zip
    ├── images/
    │   ├── img_001.png
    │   ├── img_002.png
    │   ├── img_003.png
    │   ├── ...
    ├── student_teams.csv
```

V. Select the file you would like to upload (file structure shown above, and contents of files shown below).

VI. Once complete, select the "Go to TA management page" button on the dialog popup.

VII. Assign teams to TAs by pushing the Actions button for each TA.

A1.7 Student and Team CSV File and Profile Pic Format

	A	B	C	D	E
1	First	Last	email	team	ID
2	Ashton	Wilson	user1@exa	HD_1	1
3	Dakota	Norton	user2@exa	HD_1	2
4	Alex	Harris	user3@exa	HD_1	3
5	Sawyer	Garcia	user4@exa	HD_1	4

- This will be in the image folder in the zip file:

img_001.png	PNG File	5 KB	No
img_002.png	PNG File	5 KB	No
img_003.png	PNG File	8 KB	No
img_004.png	PNG File	6 KB	No
img_005.png	PNG File	5 KB	No
img_006.png	PNG File	5 KB	No
img_007.png	PNG File	7 KB	No

APPENDIX 2 – ALTERNATIVE/INITIAL VERSION OF DESIGN

We do not have versions of our project considered before learning more about the project. When we were all assigned to this project at the beginning of capstone semester 1, we worked together to create a baseline of the dashboard. While the dashboard was elaborated since then, our foundation remains the same. Since we did extensive research on our own, and with the client, prior to starting the dashboard, we were able to create a strong foundation, saving us time and energy. This worked greatly in our favor and will continue to attack projects in a similar manner in the future.

We do not have versions of our project considered before client's specifications had changed. We started this dashboard by consulting with our client to find out all of Dr. Mitra's needs and desires. From that, we constructed a list based on priority of implementations. We were able to create a strong foundational dashboard with all of Dr. Mitra's needs during the capstone semester 1. During capstone semester 2, we were able to work on implementing desired features to the dashboard. Since our project took on both agile and waterfall project management styles, we did not have typical drafts of our project, but rather sprints. Our sprint schedule and feature priorities can be seen throughout this design document, specifically in our Gantt chart diagrams. We appreciate Dr. Mitra's flexibility when it came to revising features based on the scope of our resources, allowing us to alter features slightly to still satisfy the needs of our client without having to restart the dashboard.

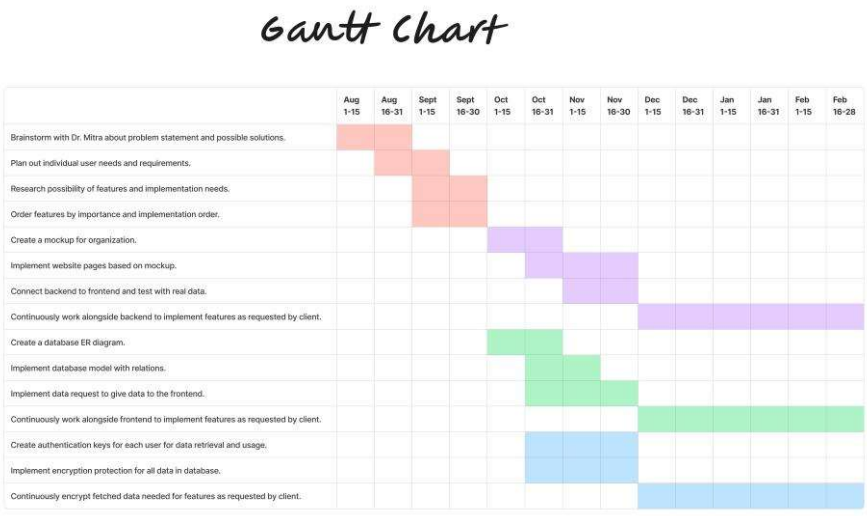
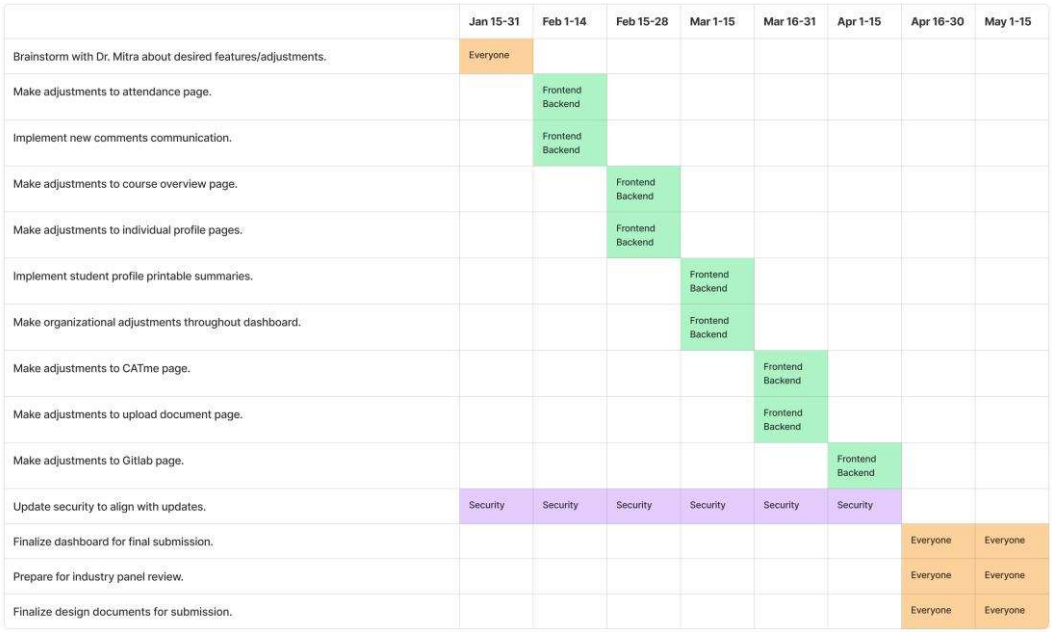


Figure 2 – Semester 1 Gantt Chart

Gantt Chart



	Jan 15-31	Feb 1-14	Feb 15-28	Mar 1-15	Mar 16-31	Apr 1-15	Apr 16-30	May 1-15
Brainstorm with Dr. Mitra about desired features/adjustments.	Everyone							
Make adjustments to attendance page.		Frontend Backend						
Implement new comments communication.		Frontend Backend						
Make adjustments to course overview page.			Frontend Backend					
Make adjustments to individual profile pages.			Frontend Backend					
Implement student profile printable summaries.				Frontend Backend				
Make organizational adjustments throughout dashboard.				Frontend Backend				
Make adjustments to CATme page.					Frontend Backend			
Make adjustments to upload document page.					Frontend Backend			
Make adjustments to Gitlab page.						Frontend Backend		
Update security to align with updates.	Security	Security	Security	Security	Security	Security		
Finalize dashboard for final submission.							Everyone	Everyone
Prepare for industry panel review.							Everyone	Everyone
Finalize design documents for submission.							Everyone	Everyone

Figure 3 – Semester 2 Gantt Chart

We do not have versions of our project that resulted in failure to achieve specifications. As mentioned throughout the design document, we did encounter some challenges with desired features. All challenges were able to be resolved using work-around methods, as elucidated throughout this design document. We did not have any instances where we implemented a feature that did not work, due to the importance we placed on brainstorming and researching, we were able to converse with the client and other campus resources to make sure desired features were achievable before taking the time to design and implement. Due to the importance, we placed on research, we were able to come up with non-typical ways to solve solutions from the start.

While we did not have different versions of our project to show progress, we do have sprint schedules to show the additions we had made and in what order. This can be seen in our Gantt charts, shown above. Our project did not have typical drafts but rather a continuous draft that was constantly being updated and made more efficient and effective. Our final design meets all the requirements that our client requested, and even satisfies all the desired features mentioned by our client, Dr. Mitra.

Appendix 3 – Other Considerations

We want to take a moment to thank our client and advisor, Dr. Mitra, for the opportunity to work on this project. We learned a tremendous amount during the past two semesters. We are grateful for the time, consideration, and flexibility granted to us by Dr. Mitra, that allowed us to create a dashboard that would be truly beneficial for the Computer Science 3090 course. We are extremely proud of ourselves for the final dashboard and hope that we have created a resource for future academia.

Appendix 4 – Code

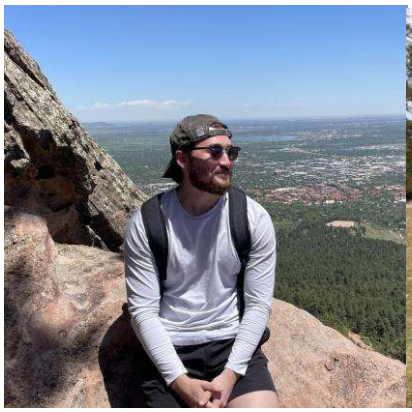
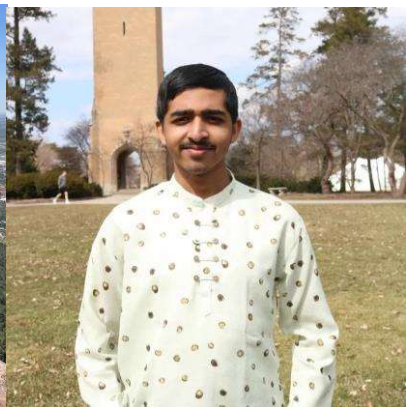
Our git repository can be accessed using this link: https://git.ece.iastate.edu/sd/sdmay25-14.git.

Appendix 5 – Team Contract

A5.1 Team Members



Bradley Gaines	Breckin Bartels	Kat Christofferson
Computer Engineering	*Software Engineering*	*Cyber-Security Engineering*

Ria Patel	Thomas Payton	Varun Jain
Electrical Engineering *(emphasis in Computer Engineering)*	*Software Engineering* *(minor in Cyber-Security Engineering)*	*Computer Engineering*

A5.2 Required Skill Sets for Our Project

1. Frontend Development
    1.1. React.js for web development
    1.2. React Native for mobile development
    1.3. HTML/CSS and UI/UX knowledge for responsive design
2. Backend Development
    2.1. TypeScript for backend development
    2.2. Database design knowledge
    2.3. ISU service knowledge
3. Database Management
    3.1. SQL/MySQL for data storage and retrieval
4. Project Management
    4.1. Agile knowledge
    4.2. Timeline management
    4.3. Communication skills
5. Testing
    5.1. Unit, integration, performance, and user acceptance testing

A5.3 Skill Sets Covered by the Team

- Frontend Development
    - React.js for web development – **Breckin Bartels, Thomas Payton**
    - React Native for mobile development – **Thomas Payton**

    - HTML/CSS and UI/UX knowledge for responsive design – **Breckin Bartels, Thomas Payton**

- Backend Development
    - TypeScript for backend development – **Bradley Gaines, Kat Christofferson, Varun Jain**
    - Database design knowledge – **Varun Jain**
    - ISU service knowledge – **Bradley Gaines, Kat Christofferson**
- Database Management
    - SQL/MySQL for data storage and retrieval – **Bradley Gaines, Kat Christofferson, Varun Jain**
- Project Management
    - Agile knowledge – **Ria Patel**
    - Timeline management – **Ria Patel**
    - Communication skills – **Whole Team**
- Testing
    - Unit, integration, performance, and user acceptance testing – **Whole Team**

A5.4 Project Management Style Adopted by the Team

The Computer Science 3090 Dashboard project embodies a waterfall and agile combination management style. The waterfall style allows a scheduled order of tasks for the beginning of our project including designing the frontend, designing the backend, implementing the frontend, implementing the backend, and implementing security. This allows us to create a functional, but simple, dashboard by the end of the first semester of our capstone. The agile style enables feedback from our original implementation to be implemented and adjusted in the dashboard during the later parts of our project, primary during the second semester of our capstone. We track progress through diagrams and the dashboard application. Diagrams are created and updated through Figma to represent our up-to-date feature design. The dashboard shows our up-to-date feature implementations as well.

A5.5 Initial Project Management Roles

Team leadership roles:

- Bradley Gaines: Backend co-lead
- Breckin Bartels: Frontend co-lead
- Kat Christofferson: Security lead
- Ria Patel: Project manager
- Thomas Payton: Frontend co-lead
- Varun Jain: Backend co-lead

A5.6 Team Contract

- Team members: Bradley Gaines, Breckin Bradley, Kat Christofferson, Ria Patel, Thomas Payton & Varun Jain
- Team Procedures:
    - Whole Team Meeting:
        - Type: In person
        - Day: Thursday
        - Time: 2pm
        - Location: Iowa State University Library
        - Frequency: Weekly
    - Frontend Meeting:
        - Type: Virtual
        - Day: No set day
        - Time: No set time

        - Location: No set location
        - Frequency: Whenever necessary
    - Backend Meeting:
        - Type: In person
        - Day: Monday
        - Time: 1pm
        - Location: Iowa State University Library
        - Frequency: Weekly
    - Client/Advisor Meeting:
        - Type: In person
        - Day: Thursday
        - Time: 3pm
        - Location: Dr. Mitra's office
        - Frequency: Biweekly
- Preferred method of communication: Discord.
- Decision making policy: consensus.
- Record keeping procedures: files in CyBox and dashboard project in website.
- Participation Expectations:
    - Everyone is expected to attend the weekly meetings and meetings with our advisor/client unless notified otherwise.
    - Everyone is expected to complete their tasks assigned during sprints. If someone expects a delay, needs additional time, or needs assistance, they should reach out to the rest of the team.

- Everyone is expected to communicated updates, delays, and/or progress on tasks. Everyone is expected to notify the team if they will not be in attendance for a meeting.
- Everyone should be made aware of and have a say in team decisions before they are made. Everyone should be present, agree upon, and be informed on tasks before they are assigned.

- Leadership:
  - Roles for each team member:
    - Bradley Gaines: Backend component design, cross component integration, testing.
    - Breckin Bartels: Frontend component design, cross component integration, testing.
    - Kat Christofferson: Backend component design, cross component integration, testing.
    - Ria Patel: Backend component design, cross component integration, project manager.
    - Thomas Payton: Frontend component design, cross component integration, testing.
    - Varun Jain: Backend component design, cross component integration, client interaction.
  - Strategies for supporting and guiding the work of all team members include talking all issues through with team members, making sure that everyone knows what resources to use and what the weekly/long-term expectations are.
  - Strategies for recognizing the contributions of all team members include having separate roles for each team member with different responsibilities, and ensuring that everyone has accomplished what they agreed to complete.

- Collaboration and inclusion:
  - Skills, expertise, and unique perspectives each team member brings to the team:
    - Bradley Gaines: frontend, backend

        - Breckin Bartels: UI/UX, frontend, backend
        - Kat Christofferson: security, backend
        - Ria Patel: frontend, backend
        - Thomas Payton: UI/UX, frontend
        - Varun Jain: frontend, backend, security
    - Strategies for encouraging and supporting contributions and ideas from all team members include all team members helping provide a comfortable group setting while still encouraging others to participate to their best ability. All team members bring different strengths, and we understand that the best ideas come from listening to each other.
    - Procedures for identifying and resolving collaboration or inclusion issues include allowing uncomfortable team members to address issues in individual conversations with the other team member helping to resolve issues.
- Goal setting, planning, and execution:
    - Team goals for the semester focus on having a working web application with a sufficient number of features as requested by Dr. Mitra.
    - Strategies for planning and assigning individual and team work include assigning roles and then establishing sprint goals as well as stories to accomplish within each sprint. We will continually check-in with Dr. Mitra to ensure we're on track and working on the correct tasks.
    - Strategies for keeping on task include having stand up meetings weekly. We will go over the tasks that everyone is working on, any issues that are blocking a task from being complete, and what we will be working on in the future.
- Consequences for not adhering to the team contract:
    - We will handle infractions of any of the obligations of this team contract by discussing the infraction with the team and come up with a plan to resolve the issue depending on the severity. If the issue causes a disruption in the project timeline, come up with a plan to get ourselves back on track. Depending on how frequent issues arise, we will talk with advisors/instructors if needed.

    - If infractions continue, we will address the issue with our advisors and instructor to come up with a solution.

- *I participated in formulating the standards, roles, and procedures as stated in this contract. I understand that I am obligated to abide by these terms and conditions, I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.*
    - Bradley Gaines, 5/4/25
    - Breckin Bartels, 5/4/25
    - Kat Christofferson, 5/4/25
    - Ria Patel, 5/4/25
    - Thomas Payton, 5/4/25
    - Varun Jain, 5/4/25