Computer Science 3090 Dashboard

sdmay25-14: Bradley Gaines, Breckin Bartels, Kat Christofferson, Ria Patel, Thomas Payton & Varun Jain

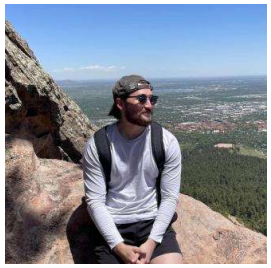
SDMAY25-14

- Bradley Gaines
  - Computer Engineering

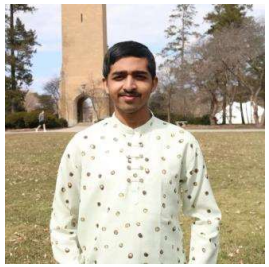
- Breckin Bartels
  - Software Engineering

- Kat Christofferson
  - Cyber-Security Engineering

- Ria Patel
  - Electrical Engineering
    - Emphasis in Computer Engineering



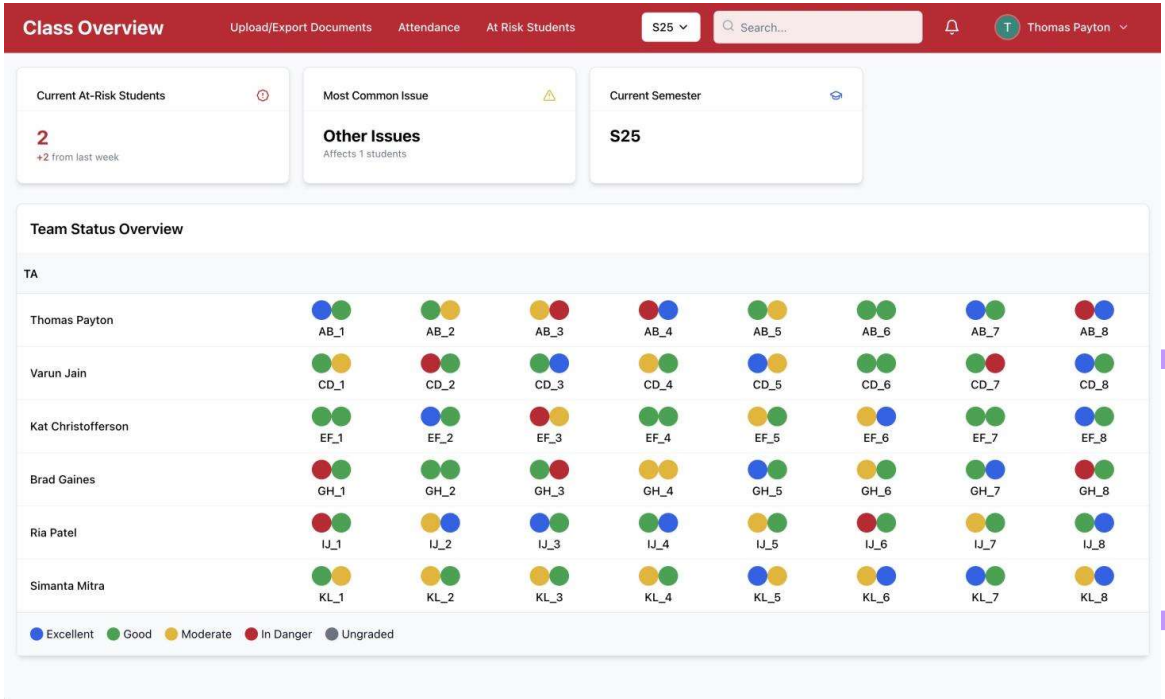
- Thomas Payton
  - Software Engineering
    - Minor in Cyber-Security Engineering



- Varun Jain
  - Computer Engineering

Project Overview

- Project: Computer Science 3090 Dashboard
- Client/Advisor: Dr. Simanta Mitra
- Goal: Create a platform that allows faculty to review students/groups grades, track attendance, review CATme results, review Gitlab activity, and track at-risk students.



Home Page

Product Research

Product Research

- Our research for an application that has all of the necessary aspects fell short.
- In order to solve Dr. Mitra's problem, we created this application/dashboard ourselves.
  - Two groups had previously attempted this project, we used their progress to jumpstart our foundation.
- By creating this platform ourselves, we were able to take feedback from Dr. Mitra and his teaching assistants in order to find out what aspects of the program were desired.
- Dr. Mitra (or other senior design groups) will be able to add functions in the future if necessary!

Users and Needs

Users and Needs

Users	Needs
Professors & Head Teaching Assistants	• **Students enrolled** (names, pictures, etc.) • Attendance tracking • At-Risk students • Grades • Performance data (Gitlab, CATme) • Reviews and reflections
Teaching Assistants	• **Students in groups** (names, pictures, etc.) • Attendance tracking • At-Risk students • Grades • Performance data (Gitlab, CATme) • Reviews and reflections

Project Requirements

Functional 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/User Interface Requirements

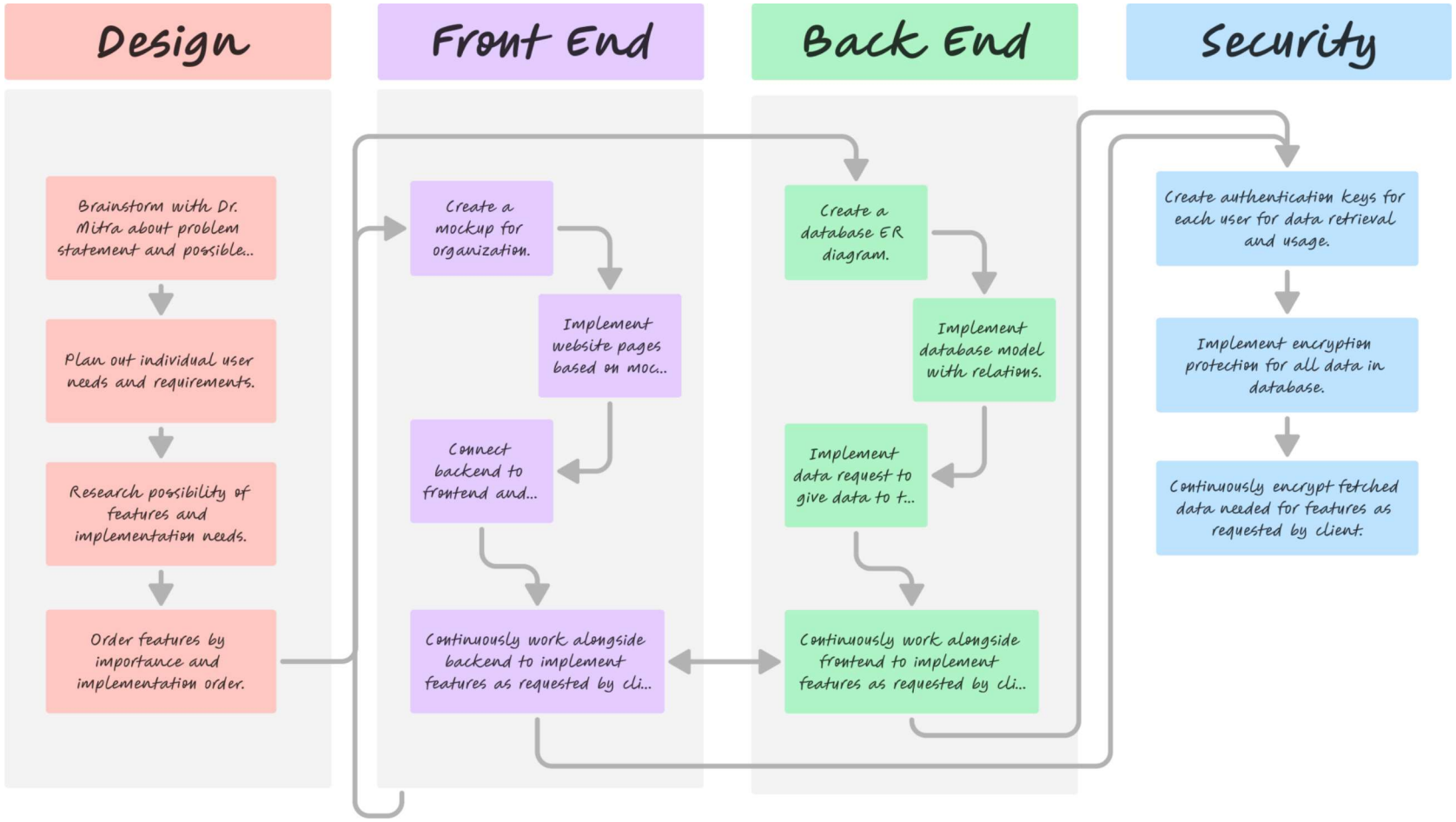
- Resource Requirements:
  - Development server.
  - Development database.
  - ISU server.
  - ISU RDS.
  - The dashboard must represent all data given in an uploaded CSV file.
- User Interface Requirements:
  - Role-specific views for professors, teaching assistants, and head teaching assistants.
  - Friendly and easy to navigate interface for all users.

Aesthetics/Physical and Quantitative Requirements

- 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.

Task Decomposition

Task Decomposition

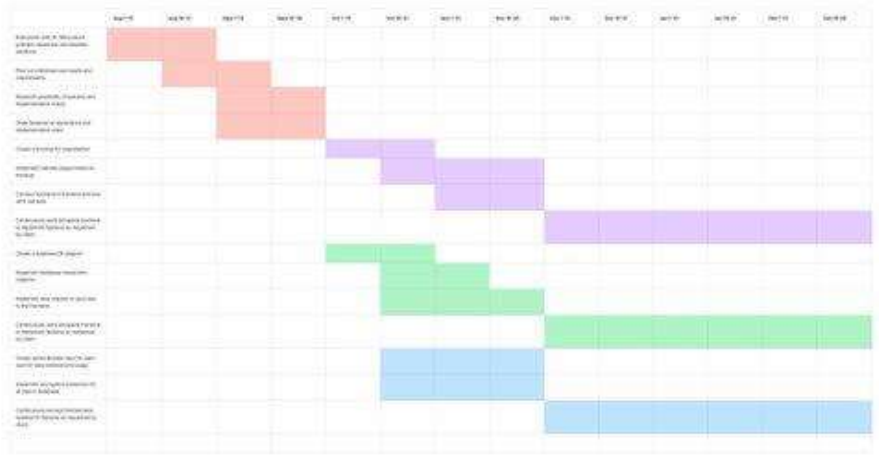


Project Management

Project Management

- Hybrid management style:
  - Waterfall:
    - Scheduled order of tasks to start i.e. design frontend, design backend, implement frontend, implement backend, implement security.
  - Agile:
    - Regular feedback from original implementation.
    - One semester of test run with continuous improvements.

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

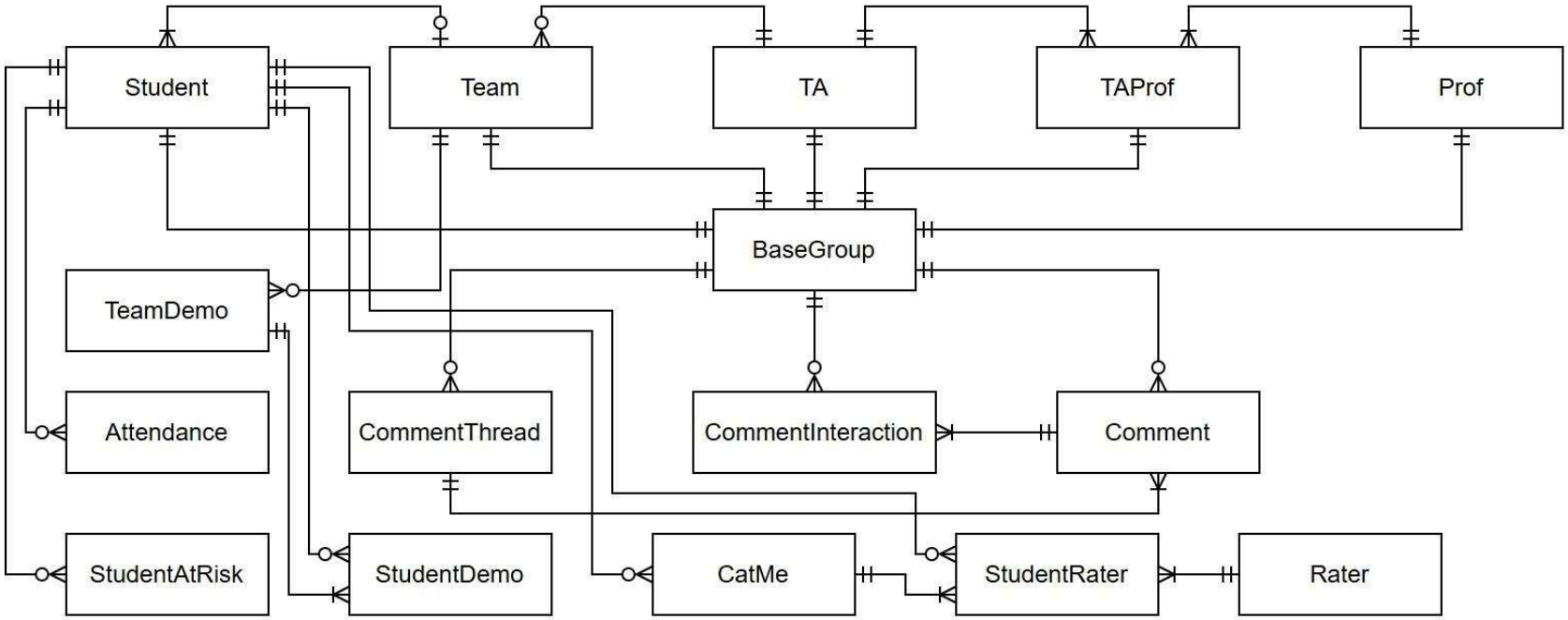
Key Risks

Key Risks

Section	Risk
Frontend	• Privacy. • Easy usability.
Backend	• Tables. • Data analysis.
Security	• Data encryption. • Academic data storage.
Overall	• Different views. • Future adaptations.

Backend Overview

ER Diagram


Student
Team
TA
TAProf
Prof
BaseGroup
TeamDemo
Attendance
CommentThread
CommentInteraction
Comment
StudentAtRisk
StudentDemo
CatMe
StudentRater
Rater

Backend Technical Decisions

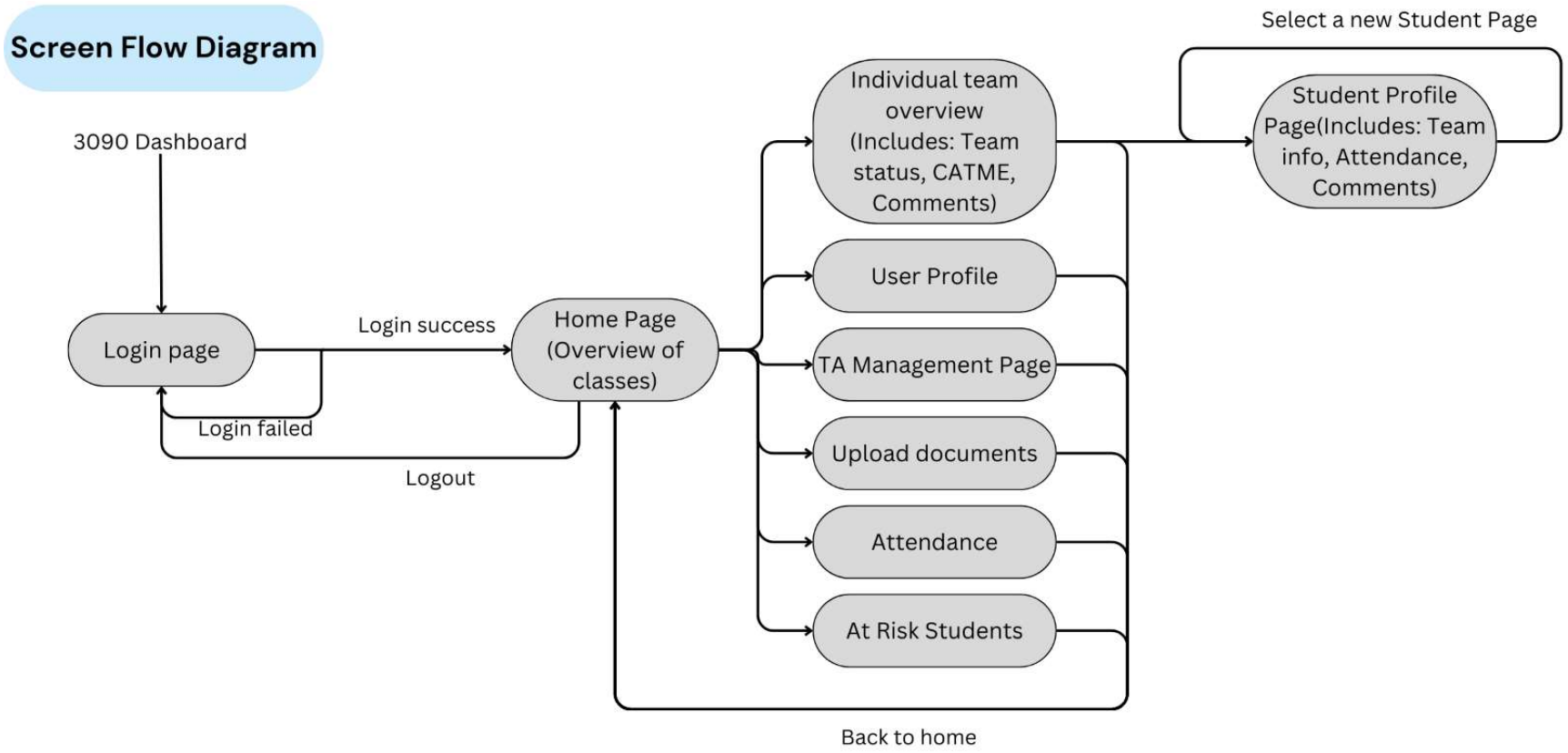
- Node.js.
- TypeScript.
- TypeORM.
- Hosting: Iowa State University Services.
- Data Management: MySQL database.

Frontend Overview

Page Design Cycle

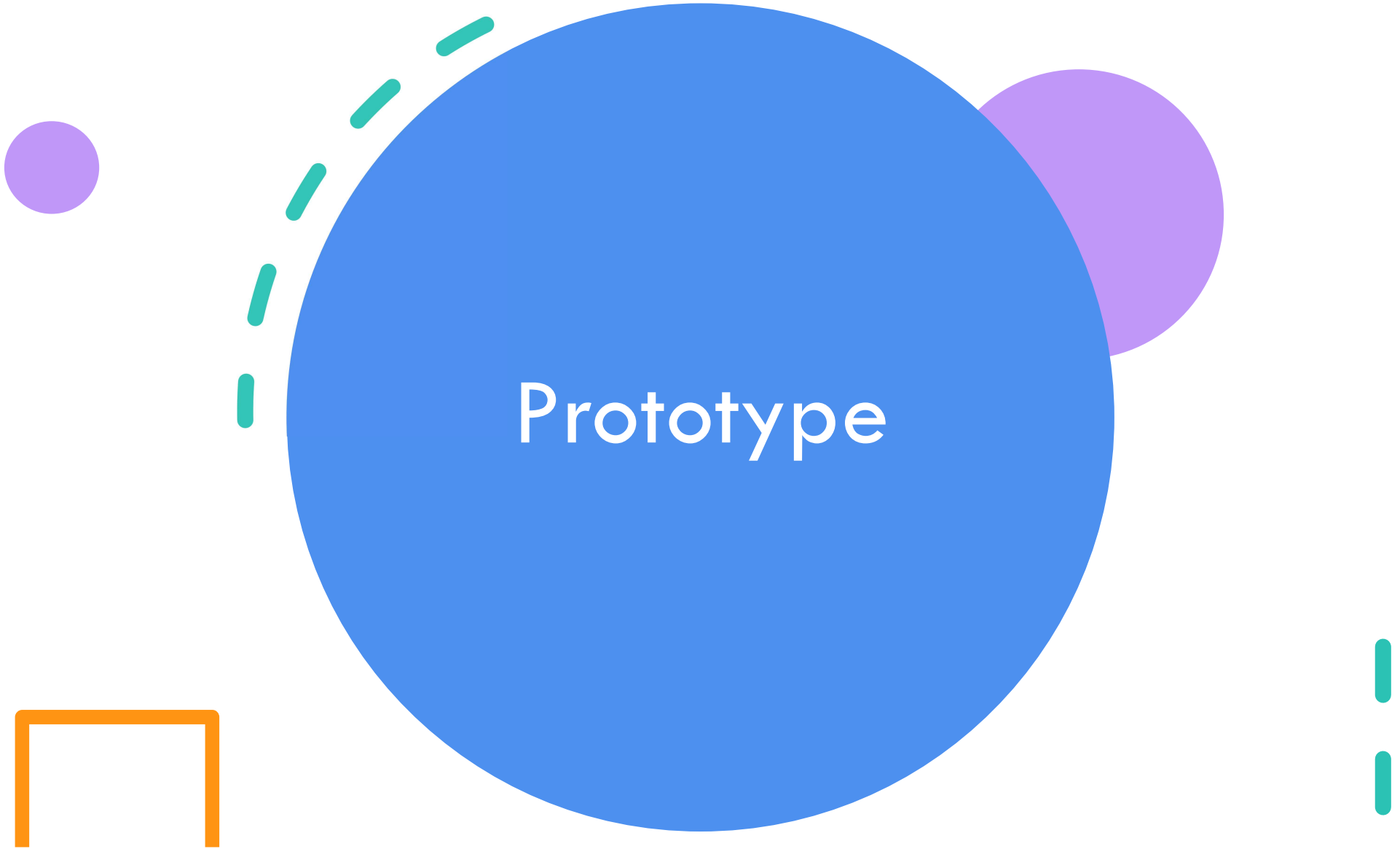
- Client expresses an issue.
- We come up with a page solution.
- Client gives feedback on the page.
- We make final adjustments to the page.
- Finalize connection with the backend.
- Ensure no other pages were harmed in the process.
- Repeat!

Screen Flow Diagram

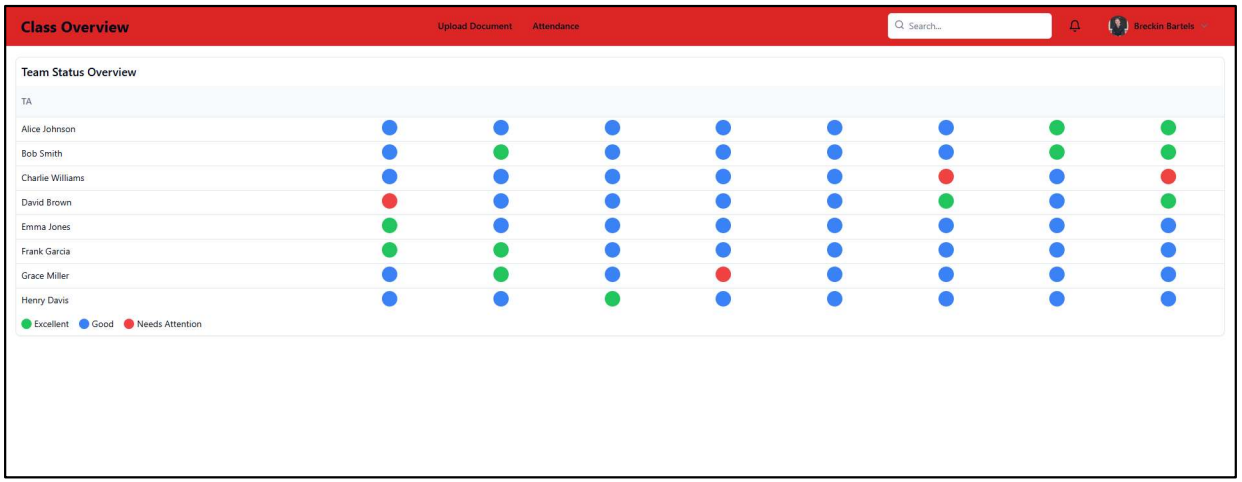


Frontend Technical Decisions

- Next.JS.
- React.
- TypeScript.


Prototype

Prototype – Figma Mockups



Home Page

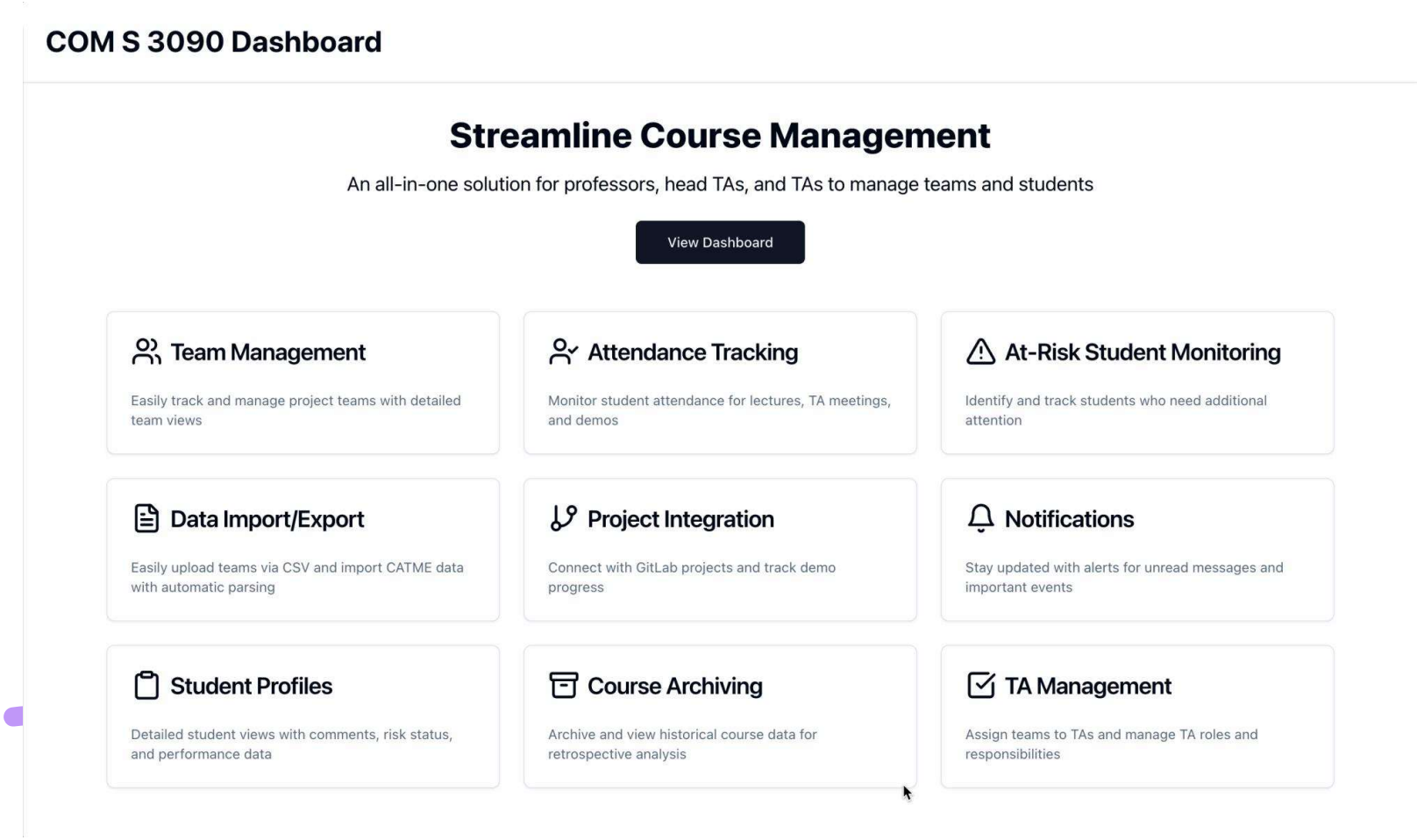
Team Breakdown Page

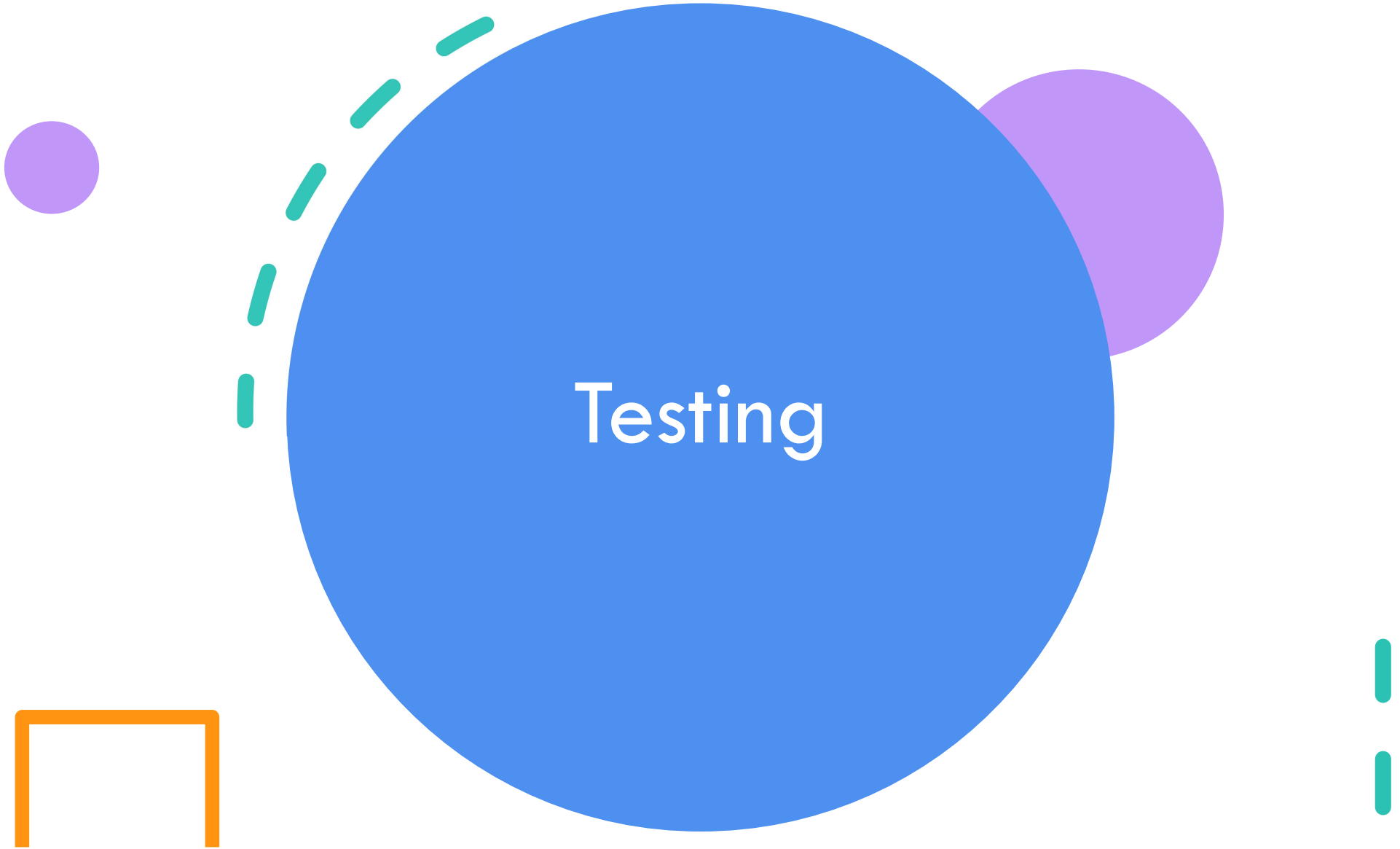
Prototype Reflection

- Our prototype followed an iterative approach
  - Our initial prototype was our Figma mockups for pages on the dashboard.
  - Instead of creating multiple prototypes, our project showed advancement by additions to our website design.
- We learned based on what didn't work:
  - Multiple pages were better for neater organization.
  - Important details should be represented first, then users should have the ability to drill down pages for more information.
  - Color codes allow for quick and easy understanding of the data being conveyed.

Final Project

Demo


COM S 3090 Dashboard
Streamline Course Management
An all-in-one solution for professors, head TAs, and TAs to manage teams and students
View Dashboard
Team Management
Easily track and manage project teams with detailed team views
Attendance Tracking
Monitor student attendance for lectures, TA meetings, and demos
At-Risk Student Monitoring
Identify and track students who need additional attention
Data Import/Export
Easily upload teams via CSV and import CATME data with automatic parsing
Project Integration
Connect with GitLab projects and track demo progress
Notifications
Stay updated with alerts for unread messages and important events
Student Profiles
Detailed student views with comments, risk status, and performance data
Course Archiving
Archive and view historical course data for retrospective analysis
TA Management
Assign teams to TAs and manage TA roles and responsibilities


Testing

Test Plans & Results

- Unit Testing:
  - Mock database.
- Interface Testing:
  - Cycle: design, implement, adapt to feedback, manual test.
- Integration & System Testing:
  - Cycle: page completion, connection from front to back, update back, manual test.
  - Backend testing assisted with Postman.
- Acceptance Testing:
  - Constant communication and approval of client.
- User Testing:
  - Future implementation into course organization.

Implications and Next Steps

- Next steps:
  - Future senior design groups could add features for:
    - Editing student/group information throughout the semester.
    - Uploading team proposal files to team pages.
    - Enhancing filtering capabilities throughout.
    - Uploading CATme highlights per team demo.
    - Tagging comments to a student or multiple students.
    - Integration with Microsoft login.
    - Assigning tasks to teaching assistants from the professor or head teaching assistant.
  - Dr. Mitra could:
    - Make changes to the dashboard in coordination to changes in his courses.
    - Alter the dashboard for use in other courses.
- All future implementation ideas are fully documented in the design documents!

Conclusions & Lessons Learned

- Users include Professors and Teaching Assistants.
- All student information will be available in one location.
- Stronger organization of course materials for storage.

- The project was fun and we all learned an extreme amount.
- Communication with the client is very important.
- The client is always right.

Questions?

Sources

- [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

Sources

- [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