IINSPIRE STEM Survey Visualization Tool

Design Document

Team 35
Client: Dr.Rover/University of Iowa
Adviser: Dr. Rover
Kaitlin Hansen
Jimmy Driskell
Nathan Frank
Lydia McCleary
Thomas Nunez
Abe Scheideman
sdmay24-35@iastate.edu
https://sdmay24-35.sd.ece.iastate.edu/

Revised: December 2023/Version 1.0

Executive Summary

Development Standards & Practices Used

IEEE Standards

- IEEE 26515-2018 - Agile Development Cycle
    - We plan to use this style of development for our project to ensure efficient development that allows for continual communication between team and client in regards to the deliverables.
- IEEE 829 - Software Test Documentation

    ○ Following this standard for test documentation allows our team to easily document our tests to show what each test does and its expected results.

- IEEE Computer Society Code of Ethics:
    - We work in a professional and ethical manner to further the advancement of software engineering.

Summary of Requirements

- A web-based tool that can be accessed by client users
- Create charts based on survey data
- Generate explanations for the created charts
- Provide the user with options to save or print the created charts
- Store statistical survey results in a database to be accessed by the web tool
- A web-based tool that can be accessed by client users

Applicable Courses from Iowa State University Curriculum

COM S 309, COM S 319, COM S 327, COM S 363, COM S 409, SE 317, SE 329, SE 339, SE 417, SE 421

New Skills/Knowledge acquired that was not taught in courses

- JavaScript programming language
- Front-End Framework such as React
- Data Visualization Library (AnyChart)
- Data Processing and Analysis tools

- Data Security
- UX design practices
- Amazon Web Services: EC2 and RDS

Table of Contents

1 Team

1.1 Team Members

Jimmy Driskell

Nathan Frank

Kaitlin Hansen

Lydia McCleary

Thomas Nunez

Abe Scheideman

1.2 Required Skill Sets for Your Project

- Front-end development
    - HTML
    - CSS
    - Javascript
    - Any related frameworks (e.g. React)
- Back-end development
    - SQL - for database design
    - JavaScript
    - Node.js

1.3 Skill Sets covered by the Team

Jimmy: Front-end development, mostly in HTML and Javascript (a bit of experience with React.js)

Katie: Front-end development, specifically HTML/CSS and JavaScript.

Lydia: back-end development (Java, familiarity with JDBC), SQL, and front-end development skills (more limited experience than back-end).

Nathan: Front-end development with React, machine learning, data visualization, APIs, HTML, CSS, JavaScript, and Python

Thomas: back-end development, created the SQL database and tables for a website in a previous internship; also knowledgeable in Java.

Abe: back-end Development, Middleware Frameworks, React, using APIs and some scripting, Project Management

1.4 Project Management Style Adopted by the team

Agile Development with weekly sprints

1.5 Initial Project Management Roles

Back End Development: Lydia McCleary, Thomas Nunez, Abe Scheideman

Front End Development: Jimmy Driskell, Nathan Frank, Kaitlin Hansen

Advisor/Client Communication Lead: Jimmy Driskell

Grad Student Communication Lead: Kaitlin Hansen

University of Iowa Students Communication: Jimmy Driskell

Webmaster: Lydia McCleary

2 Problem Statement, Requirements, and Engineering Standards

2.1 Problem Statement

Students in the IINSPIRE STEM program take a pre and post-participation survey that is analyzed in a very time-consuming process where another person must interpret the results and create data visualizations using a drag-and-drop process using Microsoft Office products. Our project aims to make this process more efficient by creating a web tool that will administer the surveys and programmatically generate graphs and charts related to the results.

2.2 Requirements & Constraints

Functional requirements:

- Our web tool must be able to administer pre and post-program surveys to each program participant (student user)
- It must be possible for the survey to display pre-determined measures/questions
- It must be possible for student users to take surveys for multiple programs
- It must be possible to save the survey data from each participant to the tool's back-end database
- It must be possible for users to export/save the created graphs and visualizations
- It must be possible to generate explanations for the created graphs and visualizations
- It must be possible for users to create an account that links their data to a user profile
- It must be possible for student users to be able to see their previous survey responses
- It must be possible for an admin/instructor user to see the responses of all participants
- It must be possible for an admin/instructor to see the generated results graphs and explanations for all participants
- It must be able to run on all common web browsers (Google Chrome, Firefox, Microsoft Edge, Safari, etc.)
- It must be accessible for both Windows and Mac Operating Systems.

Nonfunctional requirements:

- Our web tool should be easily accessible via clear and understandable visuals
- Our code shall be stored on Iowa State's Gitlab to be maintainable by future teams *Constraint*
- Our web design/front end should create an intuitive user experience
- Our web tool should respond to user interactions in a reasonable amount of time
- The survey results page should create visuals based on the format decided on by the client ***Constraint***
- Our web tool should follow IINSPIRE-LSAMP colorways ***Constraint***
- The front-end should use React/Node.js libraries ***Constraint***
- The back-end should use AWS technologies ***Constraint***

Testing requirements:

- Functional Testing:
    - Survey Functionality: Ensure that students can complete surveys before and after the program.
    - Data Entry Validation: Validate and sanitize user input to prevent errors and security vulnerabilities.
    - Graph Generation: Verify that the system accurately generates graphs based on survey data.
    - Data Visualization: Check the quality and accuracy of data visualization techniques used.
    - User Authentication: Test user authentication and authorization to ensure that only authorized users can access and submit surveys.
    - Export and Save: Confirm that users can export or save survey results and graphs for later reference.
- Usability Testing:
    - User Interface (UI) Usability: Evaluate the user-friendliness of the web page design.
    - Mobile Responsiveness: Ensure that the web page works well on various devices and screen sizes.
- Navigation Testing: Verify that users can easily navigate through the surveys and graphs.
- Security Testing
    - Data protection: Ensure that user data is securely stored and transmitted (use HTTPS)
    - Authentication and Authorization: Verify that only authorized users can access and modify data.
    - Protection against common attacks: Test for common web application vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

2.3 Engineering Standards

IEEE Standards

IEEE 26515-2018 - Agile Development Cycle

- We plan to use this style of development for our project to ensure efficient development that allows for continual communication between team and client in regard to the deliverables.

IEEE 829 - Software Test Documentation

- Following this standard for test documentation allows our team to easily document our tests to show what each test does and its expected results.

IEEE Computer Society Code of Ethics

- We work in a professional and ethical manner to further the advancement of software engineering.

ISO Standards

ISO 29119 - Software Testing

- Like all programs, testing your software is crucial, and this standard consists of five different parts of testing.

ISO 27001 - Information Security

- Security for users is vital for any website, especially ours, because our website will save private user information.

2.4 Intended Users and Uses

There will be two types of users for our system: admin/instructors and students.

Admins/Instructors will be able to:

- Create an account
- Create a survey
- View the results of a survey for all students as graphs
- Export/save graphs
- View the explanations that are generated with each graph

Students will be able to:

- Create an account
- Complete pre and post-program surveys for multiple programs
- Request and view graphs related to their survey results
- View generated explanations of their results graph
- Save/Export their results
- View their results from previous programs if they have participated in multiple

The admin/instructors will be using the survey data to gain insights into how beneficial the program is for participants (from the participant's perspective), what improvements could be made, and how to change the program to enhance the experience for future students. The participant demographic is largely students in STEM or who have an interest in STEM and come from underrepresented backgrounds; therefore, the admins/instructors find their feedback and survey results very valuable for understanding the program's impact on its participants. Additionally, the admins/instructors are performing research related to this program, so it is important for them to have access to all of the student users' results as part of their research activities. It is important to consider who the student users are when making the design for this product because it is important to ensure they are unhindered by the design when it comes to their ability to take the surveys. If the design is easy to navigate and well thought out, participants will be more inclined to not only take the surveys but also be more intentional about their answers and what they share in the surveys.

3 Project Plan

3.1 Task Decomposition

Front-End

- Create the skeleton React App
    - Configure team's computers to use React and related technologies
    - Create a React App
    - Upload it to the team git repository

The next set of Front-End Tasks relate to the creation of the other pages within the application.

- Set up user access screens
    - Login Screen
        - Allow existing users to log in with their username and password
        - Verify the given input matches an existing user otherwise display an error message and direct users to the registration page
    - Registering
        - Allow new users to create an account with a distinct email and password that meets minimum requirements of either admin or student type
        - Require Email address confirmation
        - Send new user data to back-end for storage
- Home Screen
    - Only allow access for logged in users
    - Display Relevant User Data
    - Show a Tool Summary and provide information on IINSPIRE
    - Include Navigation Links
        - To the login page

        - To the survey page
        - To the visualization page
- Survey Page
    - Create the survey using React components
    - Display survey questions directly on our webpage
    - Send the user's survey data to the back-end for storage
- Results Page
    - Retrieve user's survey data
    - Use the AnyCharts library to create graphics that visualize the survey results
    - Enable the user to download the created graphics
- Admin Page
    - Allow admin users to edit surveys
    - Display all student survey responses and generated graphics
    - Enable the user to download the created graphics

Back-end

- Create an AWS account
    - Configure the AWS Amplify instance
    - Configure EC2 instance
    - Configure RDS (Relational Database Service) for hosting
    - Implement limitations on resource usage to avoid unnecessary or excessive costs
- User information storage
    - Able to send email for account retrieval
    - Login credential storage
    - Credential authentication
- Survey data storage
    - Connecting data to visualization so it's visible to the user
    - Ability to anonymize the data – retrieve results without showing the researchers who it is from

3.2 Project Management/Tracking Procedures

We are going to use the agile project management style. Agile will be a good approach to this project because we have a client with whom we would like to be in frequent contact. Using the agile methodology allows us to adapt to the changing needs of our clients as we collaborate to understand the project better. This approach allows us to make changes quickly and without delaying progress. In addition, agile is one of the most popular methodologies in the tech industry, so using it for our project will enable us to be better prepared for our careers after graduation.

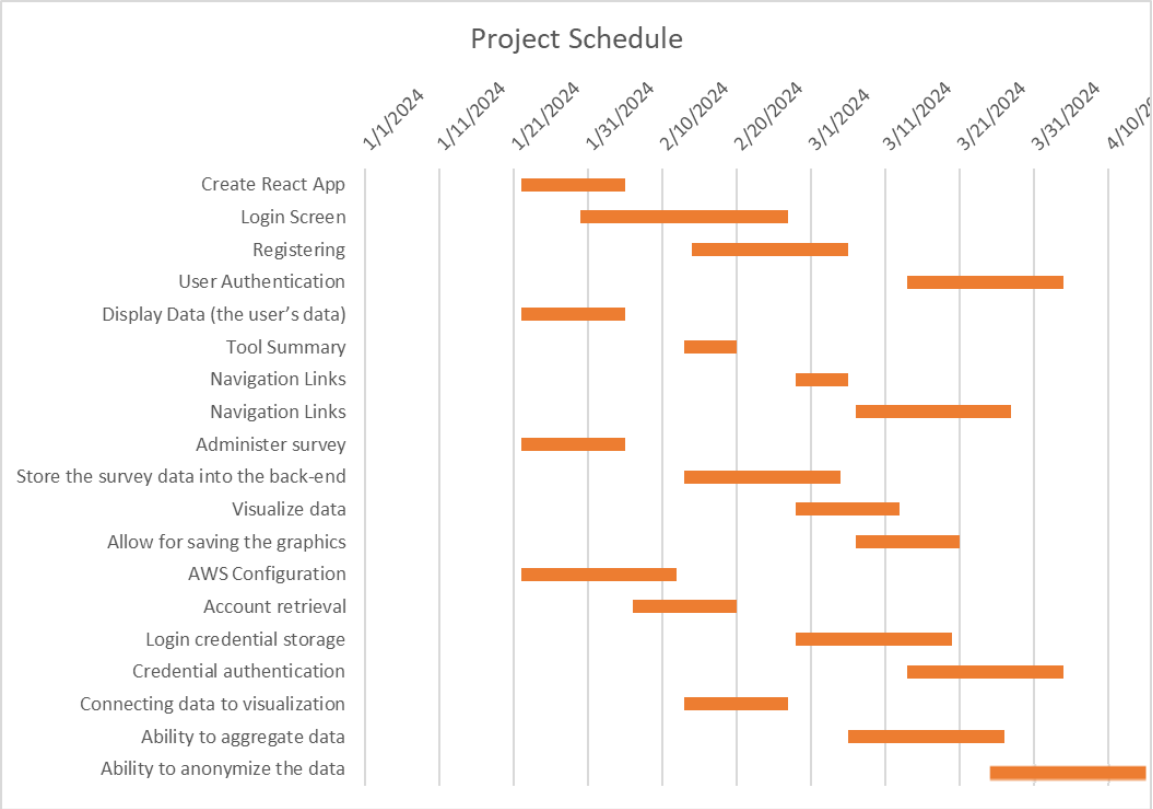
We will use GitLab issues, pull requests, and milestones to organize our development and repositories. We will utilize Discord for communication with the University of Iowa students, researchers, and instructors involved in the project. We will also utilize it for our team's mode of communication.

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Our milestones will be testable, and it will be easy to say "yes: or "no" to see if the project has met that milestone.

- First Semester Milestones
    - Produce a requirements-gathering process document
    - Create screen visualizations via Figma to share with the client
    - Complete all Design Documentation
    - Deliver a completed final presentation
- Miscellaneous back-end Milestones:
    - The back-end can send aggregated survey data to the front end
    - The back-end can process and store user data dynamically – e.g., upon account creation, password creation, etc.
    - Document SQL schema table mappings
- Log In/User Milestones:
    - The user can create an account and later log back in with those credentials
    - User authentication/login works with 100% accuracy
    - Create Administrator and Student User roles and capabilities
        - Administrators can view many Student's survey results and data visualizations
        - Students can complete the Survey and view their personal data visualizations
- Home Screen Milestones:
    - Navigation to Results Page, Account Settings Page, and the Survey Page
    - Display information about the class and the different benchmarks the Survey is measuring.
    - If time allows, we will attempt to incorporate a video produced by the UI research time into the website to explain to students the purpose of the Survey.
- Results page
    - Multiple formats of data visualization are available
    - Results visualizations can be downloaded as a PDF
    - Administrators can view different Students' data
- Survey page
    - IINSPIRE LSAMP survey is embedded in the survey web page
    - Student results are saved to the database upon completing the Survey
    - Student data is saved to the database as they complete the Survey
    - The survey page remembers and navigates the Students to where they left off

3.4 Project Timeline/Schedule



3.5 Risks And Risk Management/Mitigation

Risk	Mitigation	Risk Probability
User data is insecure and gets leaked	Implement code with security in mind and encrypt the database.	10%
Data gathered from the survey is not able to be visualized.	Offer multiple ways to download and generate the visualizations. Cover all possible scenarios in regard to data visualization.	5%
The clients hate our user interface and don't find it easily navigable.	Research how other surveys (ex: Google Forms, Qualtrics) set up their UI.	15%
The user finds the survey too long or irrelevant and spams through it.	Create a pre-survey presentation showing off previous results and how if done correctly can help the user.	30%

Server downtime or technical issues may disrupt the survey delivery.	Host the site on a reliable and scalable infrastructure. Implement redundancy, monitoring and disaster recovery plans to minimize downtime.	33%
Violation of laws or regulations related to data collection and privacy.	Stay informed about relevant laws and regulations, and obtain any necessary approvals or permissions.	5%

3.6 Personnel Effort Requirements

Task	**Average Time Per Person (hours per week)**
User Login	3
Home Screen	2
Survey Page	3
Results Page	4
User Information Storage	4

3.7 Other Resource Requirements

Considering the project involves building a webtool, there are no physical, tangible resources required to finish it. However, there are important resources needed, including:

- A server/domain to host the website on
- IDEs and Programs
    - Visual Studio Code, IntelliJ, or other IDE of choice
    - MySQL to create our database schemas that will be hosted on AWS RDS.
    - JavaScript as the programming language, and Node.js as the runtime environment to create our front and back-end logic.
    - React as the framework for our front-end framework.
    - Jest as a testing tool for the front-end
    - AnyChart to power the visualizations of our Survey Results
    - Postman to test our REST API endpoints to test our front-end and back-end communication.
    - Docker for testing of the individual back-end components
- A Virtual Machine: Amazon EC2, which can be scaled up or down depending on how many users access the website. We can use Amazon CloudTrail to track user activity on our EC2 instances.
- Project Management Tool: GitLab will be our source of project management. We will create issues, milestones, and repositories for our team's development.

4 Design

4.1 Design Content

The project is a web application that will host surveys and have the capability to visualize the survey data. This will consist of a user interface and an AWS-hosted database to store data.

4.2 Design Complexity

1. Considering that there are several different possible graphics to use when visualizing data, we will need to design several options for users to choose between when it comes to making graphics.

2. Since our project is a web tool, we must ensure it can function without any installations needed.

3. Not everyone in our team knows how to use each engineering tool, so being willing to learn is a priority.

4.3 Modern Engineering Tools

- Amazon Web Services - cloud hosting used to host both front-end and back-end
- React - front-end framework
- SQL - back-end queries to the databases
- Relational Databases - to store data for users and surveys
- REST API - communicate between the front-end and back-end
- Git - Version control
- npm (node package manager) - automate the installations, upgrades, configurations, and deletion of computer programs for the front-end
- Jest - testing framework
- JWT authentication and authorization - security for user data

4.4 Design Context

Area	**Description**	**Examples**
Public health, safety, and welfare	The IINSPIRE project will assist in introducing students to STEM and showing where their strengths and growth opportunities are.	This program can assist students in deciding what they want to do, furthering their education, and counseling them on their results.
Global, cultural, and social	The IINSPIRE project aims to assist students in underserved communities and introduce them to STEM.	Hopefully, we can spread STEM education further through the IINSPIRE program and the revamped survey system.
Environmental	The revamped survey system may better equip researchers to advance their students' STEM knowledge. Through this, these students may go on to create new environmental advancements.	A student may go on to assist or create a new invention that can help us become more eco-friendly.

Economic	Our revamped system can help better equip the IINSPIRE students with STEM skills that, in the future, may help them break into the STEM industry.	A student may move on from the IINSPIRE program and continue to pursue STEM, which may provide them with more lucrative job opportunities.

4.5 Prior Work/Solutions

Currently, there is no product that fulfills our client's wishes. Currently, the IINSPIRE LSAMP research project administers surveys to its participants using Qualtrics. The results are then aggregated and analyzed by hand by a research team member. When making graphics of the survey data, a research team member uses Microsoft Excel to create the graphics, but they are limited in the type of graphics they can produce, and it is a very time-consuming process. Some tools can administer surveys and provide graphics related to the results, such as Google Forms, but they are very limited in scope and functionality and do not provide the in-depth and customizable analytics capabilities that our client needs. Furthermore, since the results of these surveys will be used in published research, the information needs to be decoupled from any names or other identifying information while still being able to compare a participant's pre- and post-program survey results, and current solutions do not easily allow for that.

4.6 Design Decisions

1. Building a standalone web application: Our decision to build a standalone web application gives us the necessary framework to start looking into what technologies we can use to build our product. We will be using Node.js, JavaScript, and React to build our front end and connect to our back end.
2. Visualize survey data: The requirement to visualize the survey data collected is the driving feature of our application. Choosing and implementing the necessary visualization tools will contribute to our success or slow us down if we choose incorrectly. Our plan includes using Amazon RDS with MySQL to store our data with AnyChart, a JavaScript library, to visualize our data.

4.7 Proposed Design

4.7.1 Design 0 (Initial Design)

Design Visual and Description

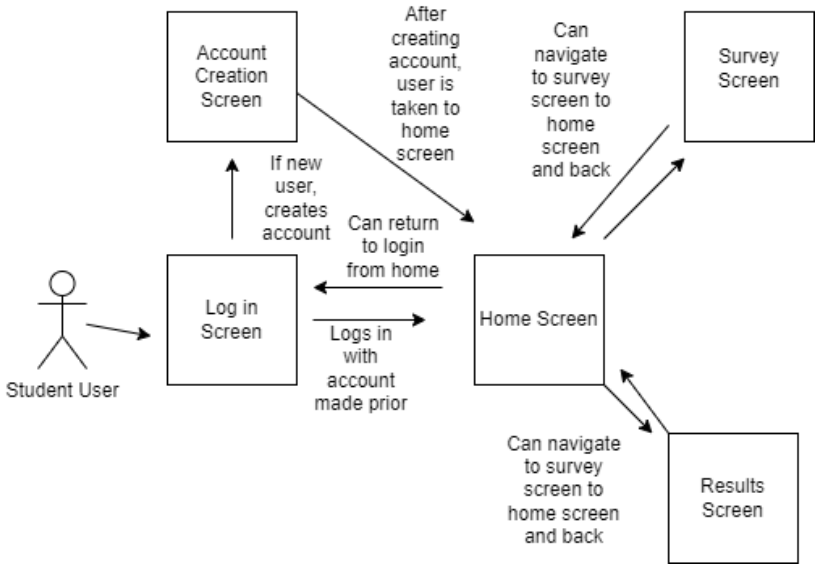


Figure 1: User interface mapping

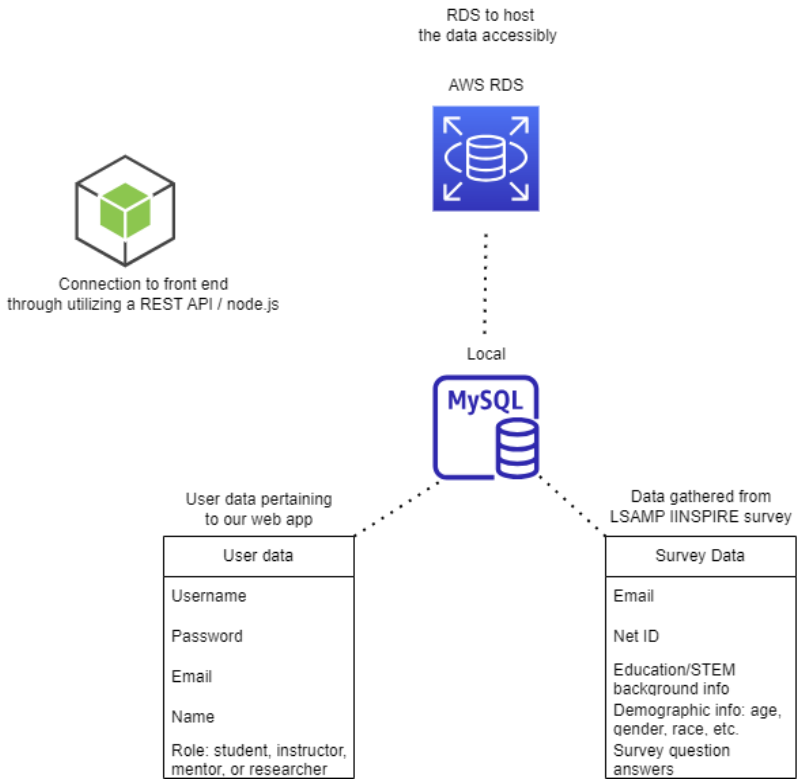


Figure 2: Back-end design with DB schema

Functionality

Figure 1 shows the initial set of user interface screens/pages within the web tool and how users would navigate from screen to screen. The user enters the application at the login screen. They can log in and move to the home screen if they already have an account. Otherwise, they can create an account and be taken to the home screen. From the home screen, the user can navigate to the survey or results screens. They can also return to the home screen from each of these screens. On the survey screen, the IINSPIRE LSAMP participant pre and post-program survey will be administered. Data visualizations on the survey results can be requested and displayed on the results screen. These graphics will help the users better understand their survey results and how their participation in the IINSPIRE LSAMP program changed their views based on pre-defined research measures.

This current design satisfies some of the function requirements but is tailored for the student participant user. It enables students to create an account, view their basic information, take program surveys, and view data related to their survey results. In future iterations, we should expand our design to encapsulate requirements for other users, such as the researchers or program mentors.

4.7.2 Design 1 (Design Iteration)

For this second iteration, there were large changes needed in the back-end that were implemented. In Design 0, the back-end was lacking largely in detail. During the Design 0 iteration, the back-end team wasn't exactly sure what AWS resources were needed. In this iteration, we had more time to

research, collaborate, and make decisions in regards to what we want to use. Due to the fact that we are leveraging AWS to host our application, we needed to conduct research to compare and contrast the tradeoffs between different services and their associated costs. We created a more fleshed out design that includes more back-end detail.

Our current model consists of having a local MySQL database to collect the survey and user data. That database will be hosted using AWS RDS because it integrates well with MySQL. The Node.js framework will be utilized to connect the local database to the cloud hosted one. An AWS EC2 instance will be utilized to administer the database and aid in the transfer of data to and from the front-end of the application (aka from the user's input). API Gateway combined with node.js will be integrated to fulfill API calls between the front-end and back-end of the application. This will enable functionality for survey results pulled from the back-end to be visualized for the user. It will also enable account credentials for a user to be stored and retrieved.

Design Visual and Description

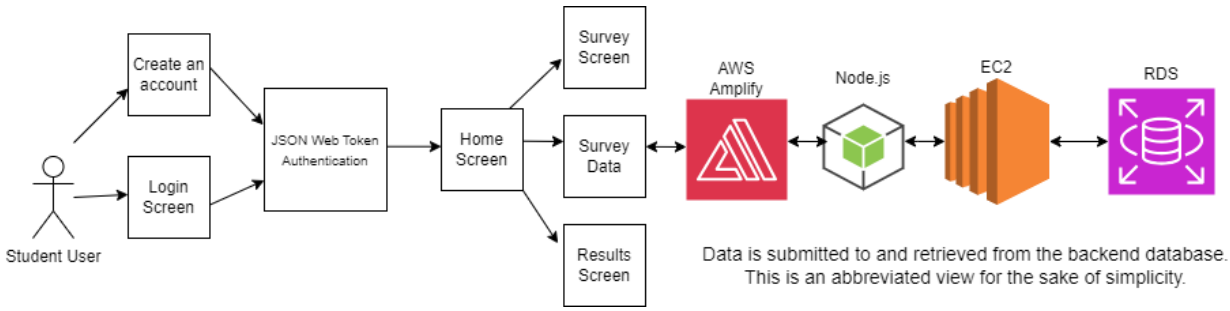


Figure 3: High Level overview, front-end and back-end interaction

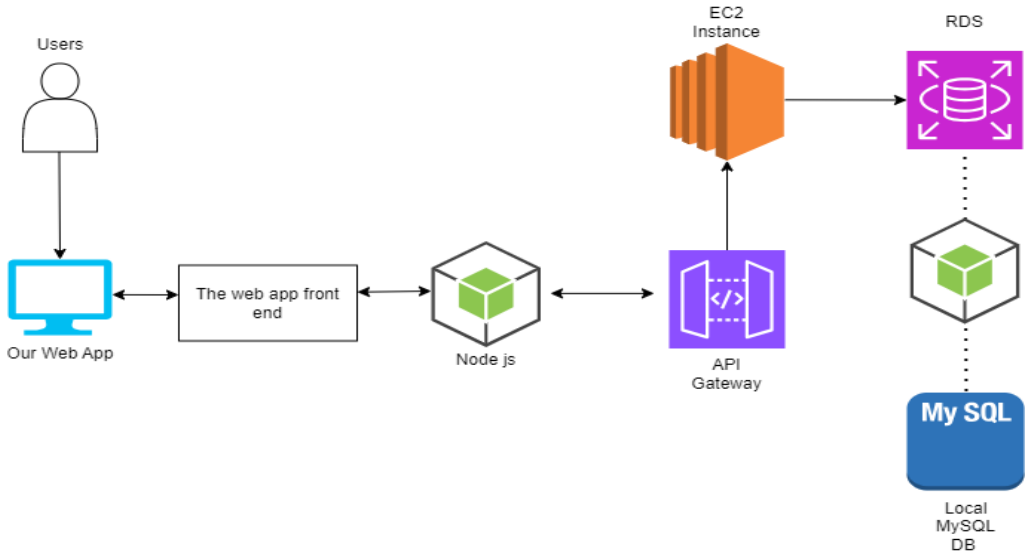


Figure 4: Back-end architecture (revised)

4.8 Technology Considerations

On the front end, we consider frameworks such as Angular and React. We chose React because more of the team had experience with the framework. We will use Node.js for our server environment over others to be consistent with the back-end team because it was taught in previous coursework. After viewing a few different visualization libraries, AnyChart will be used because of its ease of use and the most examples of bubble charts available to reference for examples. AWS Amplify will be used to host the service over S3 because it will make authentication, authorization, and password retrieval easier to implement.

On the back-end, we researched a few different technologies to integrate our database by utilizing JavaScript and Node.js. We ultimately decided on Node.js because it was already part of our plan to use it to connect the front end and back end. There will be a learning curve for everyone involved, but it makes sense to learn one new technology instead of two (if we used Python). Node.js is more efficient than Java and integrates nicely with MySQL using a driver.

Regarding hosting our application, we debated AWS Lambda and AWS EC2. We ultimately decided on EC2 because one of our team members is familiar with the technology already, and EC2 is more cost-efficient.

We will analyze the strengths and weaknesses of using Amazon QuickSight and Tableau.

4.9 Design Analysis

Our proposed design will work because we have experience using the tools we have chosen to use, and we know that the needs of our application can be met with the technologies. On the front end, the website will be easy to navigate because of the simplicity of the design. The back-end will work because all of the data that will be stored can be done with SQL efficiently and securely.

We know that unexpected errors won't occur because the web tool will be constantly tested in each phase of the building process. We have been and will be researching more potential problems that can occur and will come up with solutions accordingly. Also, as mentioned before, the web tool has been well planned out both on the front-end and back-end side, so any possible issues will be recognized and mitigated.

Regarding observations and thoughts about our design, the design itself is very clean and understandable. Each page is easy to navigate, and the interface is simple yet effective. With a big web tool comes its fair share of bugs and general issues, so a lot of the modifying will go into fixing the web tool to be as functional as possible. Other modifications will most likely include adding more features to make the web tool more customizable and user-friendly. What these additions are will depend on user feedback in the future.

Frontend

One of the most complex and important parts of the frontend will be creating data visualizations for students to view when they complete a survey. To ensure we found a library that could create the data visualizations we needed, we created some preliminary code for the data visualizations that could be used in the final product.

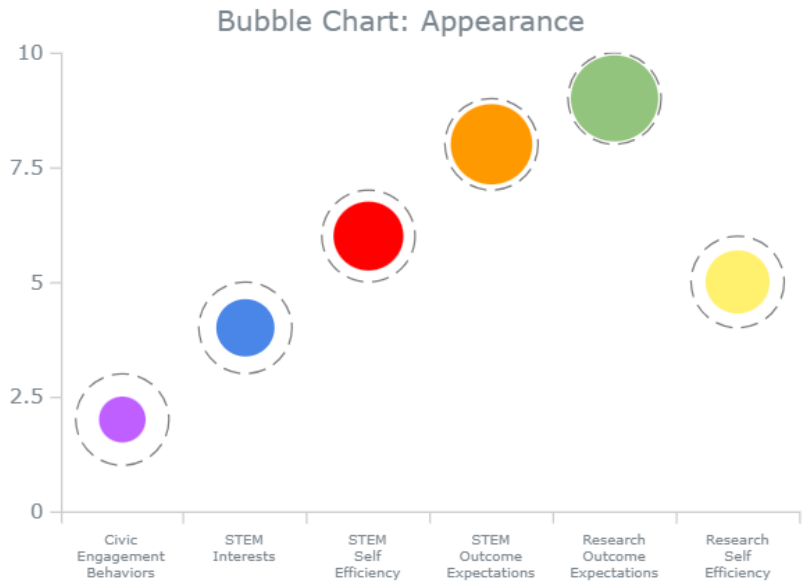


Figure 5: Sample visualization for the survey results

```
anychart.onDocumentReady(function () {

  // create data
  var data1 = [···

  var data2 = [···

  var chart = anychart.cartesian();
  chart.maxBubbleSize("10%");
  chart.interactivity().hoverMode("by-x");
  var series1 = chart.bubble(data1);
  series1.normal().fill(null);
  series1.normal().stroke("#808080", 1, "10 5", "round");

  chart.bubble(data2.map(point => ({ x: point.x, y: point.y,
    size: point.size, fill: point.color, stroke: null })));
  chart.title("Bubble Chart: Appearance");
  chart.xAxis().title("\n\n");
  var xLabels = chart.xAxis().labels();
  xLabels.wordWrap("break-word");
  xLabels.wordBreak("break-all");
  xLabels.hAlign("center");
  xLabels.width(60);
  xLabels.fontSize(8);
  chart.interactivity().hoverMode('none');
  chart.interactivity().selectionMode('none');
  chart.interactivity().zoomOnMouseWheel(false);
  chart.tooltip(false);
  chart.container("container");
  chart.draw();
});
```

Figure 6: AnyChart code to generate visualizations (as seen in Figure 5)

Backend

As a preliminary start for the backend we created some basic tables with mock data in Excel to plan out our database schema. We decided to create three main tables to begin with a table for users, a table for survey responses, and a table for survey questions.

Users Table								
User ID	**Email**	**NetID**	**Age**	**Gender**	**Ethnicity**	**Credits**	**STEM Interest**	**Institution**
1	user1@email.com	user1NetID	19	Male	Hispanic	120	Computer Science	Iowa State
2	user2@email.com	user2NetID	20	Female	Asian	50	Electrical Engineering	University of Iowa
3	user3@email.com	user3NetID	22	Other	Caucasian	37	Biology	DMACC

SurveyResponses Table			
ResponseID	**UserID**	**QuestionID**	**Response**
1	1	1	0
2	1	2	0
3	2	1	1
4	2	2	5
5	3	1	4
6	3	2	3

SurveyQuestions Table	
Question ID	**Question Text**
1	I feel responsible for my community.
2	I believe I should make a difference in my community.
3	I am committed to serve in my community.

Figure 7: Excel tables for drafting the DB schema

From here we started working through creating the actual tables in a local MySQL database so our group could start working with queries to insert data into tables and create tables. There are three tables: a survey questions table, survey responses table, and a users table.

The survey questions table has two main elements: a question's ID number and the question text. The survey responses table is related to this as it contains a user's response (identified by the user's ID number and response ID number) and the user's ID to map users to their survey answers. The users table contains information about each individual user such as their email, NetID, and institution. Each user is uniquely identified with an identification number.

```
1 •  SELECT * FROM db.surveyquestions;
```

Result Grid | Filter Rows: | Edit:

Question_ID	Question_Text
1	I feel responsible for my community.
2	I believe I should make a difference in my community.
3	I am committed to serve in my community.
NULL	NULL

Figure 8: Survey questions table

```
1 •  SELECT * FROM db.surveyresponses;
```

Result Grid | Filter Rows: | Edit:

ResponseID	UserID	QuestionID	Response
1	1	1	0
2	1	2	0
3	2	1	1
4	2	2	5
5	3	1	4
6	3	2	3
NULL	NULL	NULL	NULL

Figure 9: Survey responses table

```
1 •  SELECT * FROM db.users_table;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

User_ID	Email	NetID	Age	Gender	Ethnicity	Credits	STEM_Interest	Institution
1	user1@email.com	user1NetID	19	Male	Hispanic	120	Computer Science	Iowa State
2	user2@email.com	user2NetID	20	Female	Asian	50	Electrical Engineering	University of Iowa
3	user3@email.com	user3NetID	22	Other	Caucasian	37	Biology	DMACC
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 10: Users table

5 Testing

5.1 Unit Testing

For the front end created with React, the Jest Testing Library can write unit tests for React components, simulating user interactions and verifying their behavior. Jest can also be used to test functions and simulate HTTP requests. For the back-end, Node.js code can be tested using Jest to assess the correctness of API endpoints and the back-end logic. Additionally, AWS services can be incorporated into the testing process by setting up isolated environments for testing using Docker containers, enabling the emulation of real AWS resources. In this way, unit testing in this web project integrates seamlessly with the AWS infrastructure, React front-end, and Node.js back-end, helping to identify and rectify issues early in the development cycle to prevent problems from arising.

5.2 Interface Testing

There are multiple interfaces to be tested for our design. Users who access our application will be prompted to enter login credentials or create an account. These credentials will need to be verified and stored. We plan to use JSON Web Tokens to authenticate user access. A capability that will be tested is retrieving survey data from the back-end and displaying it to the user's profile on the front end. While the users take the survey, the information they enter and submit will be transmitted to the back-end to be stored.

5.3 Integration Testing

There are two critical integration paths we will need to test in our design:

1. Our Amazon EC2 instance and Amazon RDS will both need testing to ensure our web application can be built and run on our server with access to get and post data to our database. We will need network stress testing to ensure our web app is up and running, dependency testing to ensure the web application can be built, and testing over API calls and activity on the EC2. Some tools we may use are Amazon CloudTrail and Datadog.
2. Front-end tests over React and JavaScript. We plan on using Jest to test whether our React components are working properly. If a component were to regress by someone accidentally changing or adding to the UI, our tests can catch these changes.

5.4 System Testing

Considering that our project is entirely software, the "system" that we are testing is the entire web tool. Testing the entire system, in our case, involves ensuring an overall positive user experience. This includes what the user sees and how they can tell if they receive correct data. This can be tested using unit testing with Jest, as mentioned earlier, and it should be tested on parts of the web tool that heavily involve front-end and back-end communication. Examples of where this is present include when a user tries to log in and when a user imports data, expecting a result in graphics.

5.5 Regression Testing

To ensure that new additions do not break any previous functionality, we will revise a checklist of all previous functionality after implementing new features and test that it is still working as desired. The critical features we need to ensure we don't break include:

- The login/authentication functionality.

- The survey administration.
- The survey results/data visualization display page.

Since these are the most critical features, we will ensure that we never move forward with implementing other features that break or turn off these three features.

5.6 Acceptance Testing

Our acceptance testing will feature two main approaches. The first will be a team where we will demonstrate our project and cross off each function and non-functional requirement as our demo addresses each requirement. At the end of the demonstration, we will review the list to ensure all requirements are met. The second part of the acceptance testing will involve demonstrating the product to the client for review. This will allow us to receive their feedback and ensure we do not misunderstand their requirements. Since we are taking an agile approach, we hope to conduct this sort of client feedback testing regularly, which will hopefully ensure that by the product's completion, it will be accepted by the client.

5.7 Security Testing

Types of Testing:

1. **Penetration Testing**

- **Objective:** Identify vulnerabilities and security weaknesses in the system.
- **Process:**
    - Planning:
        - Define the scope of the penetration test, including the systems and the components that need to be tested.
    - Reconnaissance:
        - Gather information about the system as if we are on the outside looking in.
    - Scanning: Utilize automated tools to scan for open ports, services, and vulnerabilities.
    - Exploitation:
        - Attempt to exploit vulnerabilities to gain access and/or perform malicious activities.
    - Post-Exploitation:
        - Assess the impact of successful exploits and document the findings.
    - Reporting:
        - Provide an outline of the vulnerabilities, their severity, and recommended remediation.

2. **Authentication and Authorization Testing**

- **Objective:** Verify that only authorized users can access specific functionalities.
- **Process:**
    - User Authentication Testing:
        - Test various mechanisms, including user/pass, MFA, and social logins.
        - Verify the strength of password policies and the effectiveness of account lockout mechanisms.
    - Authorization Testing:
        - Check if users have appropriate permissions based on their roles.
        - Test for privilege escalation vulnerabilities.
        - Ensure that users cannot access unauthorized data or perform restricted actions.

3. **Data Encryption and Transmission Testing**

- **Objective:** Ensure data security during storage and transfer.
- **Process:**
    - Data Storage Encryption Testing:
        - Verify that sensitive data, such as passwords and personally identifiable information is stored using strong encryption algorithms.
        - Test the key management processes for data at rest.
    - Data Transmission Encryption Testing:
        - Ensure that data transmitted over networks is encrypted using secure protocols.
        - Verify the correct implementation of encryption algorithms and key exchange mechanisms.
    - SSL/TLS Testing:
        - Check for vulnerabilities related to SSL/TLS configurations.
        - Verify the use of secure ciphers and protocols.

Tools:

- **Burp Suite** - Used to perform security testing on web applications
- **Nessus** - Used to scan for vulnerabilities within the web application

5.8 Results

Testing Summary:

- Provide an overview of the testing process, this will include the objectives, methods used, and scope of the testing.

Showing Results:

- Create test reports and documentation that detail the outcomes of each type of testing. Reports will highlight any issues, or defects found during testing.

Functional Requirements:

- Utilize traceability matrices that map each requirement to the tests designed to verify them. If a test case passes, it demonstrates that the corresponding functional requirement is met.

Validation:

- Ensure that the testing process includes a validation step to confirm that the project meets the requirements set by the stakeholders and aligns with the design specifications.

6 Implementation

Front-end

The front-end team has developed baseline code with the AnyChart library to produce visualizations that match what the model the client has provided. The front-end has also developed a mockup of the user interface. Over the course of next semester the front-end team will create and design the web application using React and will work with the client to gain feedback about its design and accessibility to the user base.

Back-end

The back-end team has designed the database schema for storing user and survey information. A mock database in MySQL has been developed to test the relationships between tables. The AWS technologies that will be utilized have been decided upon. Next semester, the backend team will start by setting up the AWS account, configuring the services, and setting up the pipeline between the local database, the cloud hosted database, and the front-end side. The connection between front-end and back-end will largely be focused on data transmission such as a user's account information or the survey results being pulled to the front-end for visualization.

7 Professionalism

This discussion is with respect to the paper titled "Contextualizing Professionalism in Capstone Projects Using the IDEALS Professional Responsibility Assessment", *International Journal of Engineering Education* Vol. 28, No. 2, pp. 416–424, 2012

7.1 Areas of Responsibility

Area of Responsibility	Definition	NSPE Canon	Our Definition & IEEE Code of Ethics Definition
Work Competence	Perform work of high quality, integrity, timeliness, and professional competence.	Perform services only in areas of their competence; Avoid deceptive acts.	Our definition: Do not undertake work that you are not qualified to do, unless you are able to obtain that competence. Be honest in communicating your limitations IEEE Code of Ethics Principle #6: To maintain and improve our technical

			competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations
Financial Responsibility	Deliver products ands services of realizable value and at reasonable costs.	Act for each employer or client as faithful agents or trustees.	Our Definition: To create products that bring financial value while still having reasonable costs. IEEE Code of Ethics Principle: N/A
Communication Honesty	Report work truthfully, without deception, and understandable to stakeholders.	Issue public statements in an objective and truthful manner; Avoid deceptive acts.	Our Definition: Hold yourself to a high standard of truthful communication that is technically appropriate for your audience. IEEE Code of Ethics Principle #5: 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	Minimize risks to safety, health, and well-being of stakeholders.	Hold paramount the safety, health, and welfare of the public.	Our Definition: Respect the health, safety, and well being of all stakeholders, make decisions that put these things at risk. IEEE Code of Ethics Principle #1. 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 property, ideas, and information of clients and others.	Act for each employer or client as faithful agents or trustees.	Our Definition: Respect the ideas of others, and give credit where credit is due. IEEE Code of Ethics Principle #5. 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 environment and natural resources locally and globally.	N/A	Our Definition: Respect the environment and its natural resources: do not needlessly waste resources or contribute to environmental destruction. IEEE Code of Ethics Principle #1. 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	Produce products and services that benefit society and communities.	Conduct themselves honorably, responsibly, ethically, and lawfully as to enhance the honor, reputation, and usefulness of the profession.	Our definition: Strive to benefit society through our products 2. to improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems;

7.2 Project Specific Professional Responsibility Areas

High Areas: Work Competence, Communication Honesty, Social Responsibility. We believe we are high performing in these areas. For Work Competence, as a team we try to perform to the best of our abilities to fulfill the requests of our client while learning more about how to act in an ethical professional manner. For Communication Honesty, as a team we strive to keep open channels of communication within the team, with our advisor, and with our client. We also work hard to ensure the level of technicality we use within our communications is appropriate for who we are talking with. For Social Responsibility, we believe we are high performers in this area because our product has a direct social impact. We are striving to design our product to be as great as possible, so it can have the biggest impact on the students who will interact with it.

Medium: Financial Responsibility, Property Ownership. We believe we are medium in these categories because we are still developing our skills in regards to these topics. Since we are using AWS, we know we want to find the tools that will be the best fit for our project, without incurring large costs on our client, but we have not completed that process entirely. For Property Ownership, we are also medium because we know this will be an important area of responsibility when we begin implementing our design, since we are working with user's private data along with pre-determined measures from the research team, however we still have more to learn in this area.

Low: Health, Safety, Well-Being. We selected this area as low because our product does not pose any major threats to healthy, safety, or well-being. However, we still want our users to feel well when interacting with our product, so it is not completely non applicable to whom we have been approaching our design.

N/A: Sustainability. We do not feel as though our project poses any risks to sustainability or towards the environment.

7.3 Most Applicable Professional Responsibility Area

We chose Social Responsibility as our most applicable area of professional responsibility. We intend for our product to directly interact with users, both students and researchers, so we want to ensure that they are having a positive experience. Additionally, we want to ensure that our product is benefiting our society, specifically through helping students become more engaged in STEM, and through helping the researcher's that work with our client, IINSPIRE LSAMP. We believe our client is an organization that benefits society, therefore we want our product to be able to help them continue achieving their goals and amplifying their reach.

8 Closing Material

8.1 Discussion

The project is still in a preliminary stage and therefore most of the requirements have not been implemented. We currently have a visual mockup of the web application using Figma. We will begin implementation starting next semester. We are confident that we are on the right track and will be able to complete the project next semester.

8.2 Conclusion

This semester we have gathered requirements from our client, drafted a high level, end-to-end design of our product, and begun designing the user interface. One of our major constraints was receiving feedback and information from our clients at the University of Iowa. We are working with an interdisciplinary team at Iowa and it is often difficult to find time to meet with those team members let alone with all of them together. Therefore, it took longer to gain all of the knowledge that we needed for the requirements of this project because each member has their own unique knowledge and understanding of the project and what is needed.

Next semester, we are planning to ramp up our development and begin implementing the project. We held off on implementation this semester because we wanted to focus on fully understanding the product we are making, our client's expectations, and our users.

8.3 References

N/A

8.4 Appendices

N/A

8.4.1 Team Contract

Team Members:

1) _Jimmy Driskell_____ 2) _Kaitlin Hansen_____

3) _Thomas Nunez_____ 4) _Nathan Frank_____
5) _Abe Scheideman_____ 6) _Lydia McCleary_____

Team Procedures

Team Meetings: Monday, 10:00am, Virtual (Discord)
TA Meetings: Wednesday, 7:30pm, Virtual (Webex)
Client/Advisor Meetings: Thursday, 2:30pm, Face-to-Face (Durham 3rd Floor)

Our preferred method of communication is Discord between team members. For our client/advisor and TA, we will use email.

When making team decisions we will use Majority Vote.

One Google document will keep track of all our meeting minutes. For each meeting, we will decide who is responsible for taking meeting minutes.

Participation Expectations

TA Meetings
We expect attendance from everyone at the TA meetings. Complications may occur between people's schedules and weekly meetings, so if someone can't make it to a meeting, it's their responsibility to catch up via reading the meeting minutes. Members who can't make it to a meeting must give a heads up that's at least two hours prior to a meeting.

Contributions and Deadlines
Individual contributions to team assignments should be completed 24 hours prior to the respective assignment deadline. Assignments and responsibilities given to each other should be completed before a predetermined deadline.

Communication
If team members need help, make sure to ask via the Discord group chat. It's expected of team members to check the Discord group chat at least once a day. It's also expected of team members to stay in touch with our client/advisor via email in a timely manner.

Decision Making
It is expected that all team members actively contribute to team decision making. It is also expected that all team members do a fair share of the team tasks, and complete the tasks that have been distributed to them.

Leadership

Team Member Roles

Jimmy: I will be responsible with keeping in touch with our client/advisor through email and potentially Discord. In terms of my role on the project itself, I will be working on the front-end side of development.

Katie: I am flexible regarding what leadership roles I assume during our project. I enjoy organization of tasks and distribution of work among team members. I also can also lead the design of certain components of our end product.
Lydia: I am flexible to lead in any capacity necessary. I am skilled at organizing tasks and delegating work across a team.
Nathan: I will be responsible for the testing of the project and making sure the application we create has good test coverage. I will share this responsibility as needed and work with the team to create suitable test cases from the requirements.
Thomas: I will take on the responsibility of assisting in setting up the back-end, I have experience connecting a back-end to a site from a previous internship and feel I can apply this knowledge directly to the task at hand. Along with this as the only Cyber Security Engineering major on the team, I will take on the responsibility of ensuring our site is up to current standards of security and has no major vulnerabilities.
Abe: I am flexible to lead in whatever capacity is necessary. Organizing issue creation and boards on GitLab is an area I can lead in. I enjoy trying to make sure work is delegated well so everyone can make progress towards the team's goals.

Back End Development: Lydia McCleary, Thomas Nunez,Abe Scheideman
Front End Development: Jimmy Driskell, Nathan Frank, Kaitlin Hansen
Advisor/Client Communication Lead: Jimmy Driskell

Strategy for Guiding Work

Utilizing GitHub issues and agile development practices.

Strategy for Recognition
To ensure that we are recognizing the contributions of all team members we will start each meeting with everyone going around and stating what they've accomplished that week, what they plan to do next, and if they need any assistance with anything.

Collaboration and Inclusion

Team Member Expertise

Jimmy: I've worked in numerous programming projects with friends and classmates primarily doing front-end development and have a good understanding in web design. I have experience with working in groups not only on a communication level, but also on a technical level when using software such as GitLab and Trello. I definitely prefer working on the front-end side of the website with the user interface and using languages such as JavaScript and HTML with WebGL.

Katie: I've had software engineering internships focusing on both front-end and back-end development and have experience in web design. However, I prefer working with front-end web development, and languages such as typescript and javascript. I also have strong technical

communication skills and can help convey software or technical concepts to a nontechnical audience.

Lydia: I've had internships that focused primarily on back-end development. This past summer I worked for Oracle Cloud Infrastructure on their Object Storage team. Before that I interned at Principal Financial Group where I utilized AWS to migrate data from on-prem to the cloud. I also have experience with front end development and would love to expand that skill set through this project.

Nathan: I have interned with Principal Financial Group, Co-op Financial Services, and Walmart. In all of these roles, I did application development. I learned about hosting web applications, APIs, and React. I am familiar with AWS, Azure, Python, javascript, HTML, and CSS. I have had the opportunity to work with a few different teams with client-facing applications that I will be able to draw from.

Thomas: I've previously interned with an aerospace manufacturer and an Ag-Tech company. In these internships I have taken on a wide variety of projects involving numerous programming languages. My most recent project was setting up a security alert testing tool using HTML, PHP, AWS, and Python.

Abe: As a senior in software engineering, I have built all the skills necessary to complete programming tasks efficiently. I have also done a couple of software engineering internships. This has given me experience working in a thriving team environment. I have experience working with GitLab, doing code reviews, and doing general development. I have not done much with building interactive websites. I'm more comfortable with back-end development.

Strategy for Supporting Ideas & Contributions

As a team we will aim to assist each other where needed, and let everyone voice their opinion and point of view where applicable.

Identifying & Resolving Collaboration Issues

Be open about potential issues during our weekly meetings. If we are unable to resolve issues within our team during weekly standups, we will bring it up to the TA, and if that doesn't work, bring it up to the advisor.

Goal-Setting, Planning, and Execution

Fall Semester Goals

Become familiar with the IINSPIRE STEM survey, define the problem and requirements for creating an IINSPIRE STEM survey visualization tool, lay out an initial potential protoype, create an initial dataset, decide the features that the team will focus on to develop the tool.

Planning & Delegating Work

Discussing plans thoroughly in weekly meetings and using frequent communication via Discord. Determining each member's strengths and weaknesses, and using those to distribute tasks.

Staying on Track

Making assignments clear in terms of what to do and when it's due at our weekly meetings.

Consequences for Not Adhering to Team Contract

Handling Infractions

If there is an infraction we will bring it up with the team either in meetings or on Discord. We can also reach out to the individual in question in a more private setting, and then bring it up again in the whole group setting if the infractions continue.

Continuing Infractions

If we are unable to handle the infractions within the team we will advise our TA or advisor via email for additional help.

****

a) I participated in formulating the standards, roles, and procedures as stated in this contract.

b) I understand that I am obligated to abide by these terms and conditions.

c) I understand that if I do not abide by these terms and conditions, I will suffer the

consequences as stated in this contract.

1) __Nathan Frank_____ DATE ___9/07/23_____

2) __Katilin Hansen_____ DATE ___9/07/23_____

3) __Thomas Nunez_____ DATE ___9/07/23_____

4) __Jimmy Driskell_____ DATE ___9/10/23_____

5) __Lydia McCleary_____ DATE __9/10/23_____

6) __Abe Scheideman_____ _____ DATE __9/10/23_____