IINSPIRE STEM Survey Visualization Tool

Final Design Document

Team 24

Team Overview

Client

Dr. Diane Rover/University of Iowa

Advisors

Dr. Diane Rover

Team Members & Roles

Alex Moeller - Project Manager

Nick Pinnello - Backend

Charlie Moreland - Backend

Max Strater - Frontend

Isabelle Raghavan - Frontend

Matthew Bennett - Cloud

Landon Gulotta - Cloud/Security

Team Email: sdmay25-24@iastate.edu

Team Website: https://sdmay25-24.sd.ece.iastate.edu/#

Executive Summary

The **LSAMP IINSPIRE STEM Survey Visualization Tool** is an online web application designed to enhance engagement and data accessibility for its users. The platform enables student users to take surveys and visualize their personal survey results while allowing admin users to create and manage surveys, analyze comprehensive results, and extract statistical data for research. This project builds upon the work of a previous Senior Design team, with this year's primary focus on modernizing the tool's user interface (UI), user experience (UX), and functionality to ensure scalability and long-term usability.

Problem Statement and Importance
explain user-oriented results vs. Qualtrics. Ex: admins view the results of all students, but with IINSPIRE students view and grow from their own personal results to achieve their own personal goals.

Key Design Requirements
Our project prioritizes usability, scalability, and maintainability. Our application should provide valuable survey experiences for students, and dashboards/overviews for admin users. Additionally, the architecture should support future changes with minimal effort.

The existing IINSPIRE Survey Tool lacked a modern UI/UX, limiting its usability and appeal to target audiences. Additionally, its non-cohesive architecture posed challenges to maintainability and scalability, limiting its usefulness for future Senior Design groups. Addressing these issues is critical to enhancing user satisfaction and maintaining the program's value as a research and engagement tool.

Design Summary
We have adopted a modern technology stack to meet these goals:

- **Frontend:** React with the Chakra UI component library and TypeScript
- **Backend:** Prisma and MySQL, supported by NextAuth
- **Cloud Infrastructure:** AWS ECS, Secrets Manager, and Amazon RDS

Our design strategy emphasizes modularity and reusability, ensuring ease of use for both current and future developers.

Progress and Achievements

Significant progress was made during the semester. We conducted a focus group with IINSPIRE students to gather feedback on the previous tool, providing valuable insight into all areas of our project. Based on this input, our team developed new wireframe designs using Figma, showcasing a more intuitive and visually appealing interface. The Cloud Team collaborated with Iowa State's ETG to provision an AWS account, allowing us to keep the old version online until our new application is ready for deployment. Additionally, the Backend Team established a MySQL database and designed some initial queries based on the previous team's project.

Next Steps

In the upcoming semester, we aim to implement the wireframe designs into the application, iteratively refine the tool based on user, client, and advisor feedback, and ensure our architecture can support future Senior Design teams. This iterative approach will ensure that the final product is both functional and user-friendly.

Learning 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 professionally and ethically to further the advancement of software engineering.

Summary of Requirements

- A web-based tool that client users can access

- 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 client users can access

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
- CI/CD Testing
- Infrastructure as Code (IAC)
- 3rd party Authentication

Table of Contents

1. Introduction

1.1. Problem Statement

The IINSPIRE STEM program supports underrepresented minority students, helping them engage in STEM fields. Researchers need a tool to administer surveys and visualize results to improve program support more efficiently. The current survey website, developed by the University of Iowa, is fundamental in design and needs essential functionality.

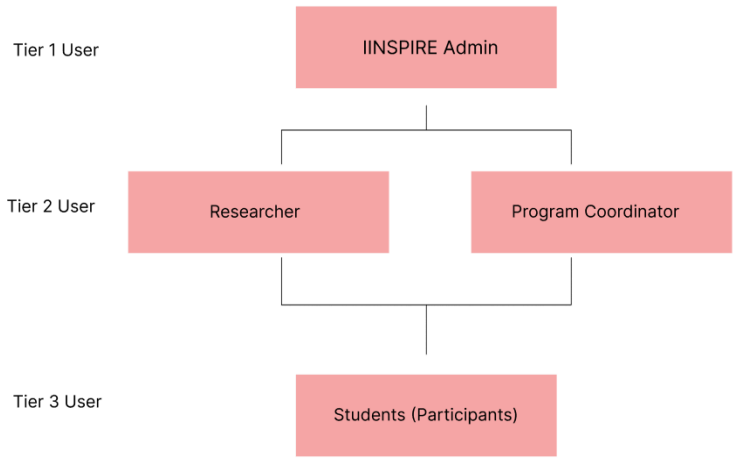
The problem will be solved by redesigning the frontend and upgrading the infrastructure to improve the web app's speed and performance. This will make the survey-taking process faster for participants and allow researchers to create and manage different surveys easily. Continuous feedback from users will guide these improvements and the addition of new features.

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 automate the generation of graphs and charts of results. Current data collection methods take time and effort. Researchers need this data to show the positive benefits of their STEM program. Data will showcase an increase in students' confidence levels towards STEM subjects.

1.2. Intended Users

Our system will have four user types: IINSPIRE Admin, Program Coordinator, Researcher, and Students (Participants). Each user group has specific needs and interactions with the survey system, focusing on the level of information they can view and access.



INSPIRE Admin - Can create/edit surveys, invite/remove users, update/patch the app, allow data access/download.

Program Coordinator - Can create/edit/delete surveys, store/view data results, share/download data, methods to visualize the survey.

Researchers - Can view/download survey results, submit request for new survey questions, request to edit current questions, can assign students to surveys.

STEM Students - sign-in/out, take assigned surveys, view survey results.

INSIGHT Admin (Tier 1):
Global Administrators manage the system and need access to high-level system functions, including the ability to:

- Create/Edit/Delete a survey
- Invite/Remove users
- Update/Patch the app
- View the results of a survey for all students
- Export/Save graphs
- Allow data access/downloads

Administrators must access all participant responses, as this data is essential for their evaluations and research. Their interface should allow for easy,simple access to all needed data and functions, allowing them to manage users, surveys, and the system without unnecessary complexity.

Researchers (Tier 2):
Researchers are responsible for analyzing survey data to assess the STEM program's impact. Their primary actions include:

- Create an account
- View/Download survey results
- Submit requests for new survey questions
- Request to edit current questions
- Can assign students to surveys

Researchers use this data to assess the impact of the STEM program on participants, identifying areas of improvement to adapt the program for future students. They rely on the survey data to shape the program and provide insights into its effectiveness. They need easy access to survey results and the ability to modify surveys quickly.

INSIGHT Staff (Tier 2):
INSIGHT Staff help manage the surveys and need functionality to:

- Create an account
- Create/Edit/Delete surveys
- Store/View survey results
- Share/Download Data
- Methods to visualize survey results

Like researchers, INSIGHT Staff use the survey data to evaluate the program's performance and create different surveys to give to STEM students. Their role is more focused on operational tasks, such as creating and maintaining surveys and organizing results. The App should make it easy to manage and share the data, allowing for easy collaboration with researchers.

Participants (Tier 3):
Participants are students, often from underrepresented backgrounds in STEM, who provide key feedback through surveys. Their interactions with the system include:

- Create an account
- Sign in/out of account
- Take assigned surveys
- View personal survey results

Participants need an intuitive, easily accessible web app to access complete surveys from any device. The system should be accessible and allow for surveys to be easy and quick to take. They must be able to view assigned surveys, track progress, and review personal results.

2. Requirements, Constraints, And Standards

2.1. Requirements & Constraints

- **Functional requirements**

a. Survey Participants should be able to see their result at the end of each survey, and view results from past surveys

b. Participants, INSPIRE Admins, Program Coordinator, and Researchers should be able to login/signup on the website using SSO or through their email

c. INSPIRE Admins should be able to create and edit surveys, invite or remove users, send out surveys, and monitor results.

d. INSPIRE Admins should be able to grant specific permissions to Participants or Researchers.

e. Researchers are able to view/download survey results, submit requests for new survey questions, request to edit current questions and assign students to surveys.

f. Have INSPIRE staff be able to create/edit/delete surveys, store/view data results, share/download data, and methods to visualize the survey

- **Resource requirements**

Frontend: React.js, Next.js, Chakra UI, Formik

Backend: Node.js, Express Framework, Next Auth, MySQL, PGAdmin,Jest

Cloud: AWS ECS, AWS Secrets Manager, S3 buckets, Github Actions, Github, AWS RDS, Cloudwatch

- **UI requirements**

a. Need a different dashboard for every user type

b. Login page

c. Register page

d. Logo for the home page

e. Color theme

f. navbar with home, survey(which will have different options based on the user type) and about

g. revamp the survey itself:

- make buttons and text larger and easier to read
- have a progress bar
- "save" button so participants don't have to do the survey all at once
- "Previous" button so participants can edit/view their answers from the previous page

- break up the survey into "like" sections and describe each section

2.2. Engineering Standards

IEEE 26515-2018 - Agile Development Cycle

This standard is about ensuring flexibility in the development process. It focuses on using Agile methods to allow teams to adapt quickly to changes in project requirements based on stakeholders providing feedback during the project. The main goal is to keep the development process responsive and collaborative, allowing for continuous improvement over time rather than waiting until the end to make changes. This is particularly useful when working on projects where user needs may evolve over time, ensuring that the final product meets expectations.

IEEE 829-1998 - Software Test Documentation

This standard is all about organizing and documenting the software testing process. Its purpose is to ensure that every part of the system is tested properly, and that the results are clearly tracked. By following this standard, teams can easily see what has been tested, what the expected outcomes are, and what still needs to be done. It helps ensure the system behaves as intended, making it easier to catch issues early and maintain high-quality standards throughout development.

IEEE 1448a-1996 - Standard for Information Technology: Software Life Cycle Processes

This standard provides a structured framework for managing software throughout its life cycle, from acquisition to maintenance, based on ISO/IEC 12207. It defines clear processes and terminology for developing, operating, and maintaining software products, including firmware. IEEE 1448a adds compliance methods and clarifications to make the life cycle processes more adaptable to business practices, offering guidance on things like development strategies and software reuse. Its goal is to ensure consistency, quality, and flexibility in software projects.

We saw a lot of relevance to our project, especially for the Agile Development Cycle Standards. Our project is heavily focused on user experience and we will have constant iterations of Agile Cycles to continuously improve on our original design. We plan on conducting numerous interviews with our clients to gain feedback on our design. This aligns with the Agile Standards of stakeholder feedback. Our design will be responsive and collaborative as well. IEEE 829-1998 - Software Test Documentation has a lot of relevance to our project as well. We have many modular parts of our software and each element needs to be tested individually as well as testing of the entire system as a whole. With so many different test cases documentation is vital to the organization and efficiency of our testing efforts. We want to follow all standards for testing documentation to ensure full testing coverage of our Web application.

Another standard we chose was the IEEE 1621-2006 (IEEE Standard for User Interface Design and Management). We chose this standard because it provides guidelines for creating usable, accessible and consistent user interfaces. This is especially relevant to our project since a large portion of our work involves redesigning the UI of the currently existing web application in order to improve overall user experience.

To align with the IEEE 26516-2018 (Agile Development Cycle), our team will use Kanban boards to report their work and their previous work. This will allow us to ensure that our project can be adaptable and meet any user needs. Each week we will have a meeting to discuss what

we did the previous week and what we will be doing the next week. This way our team can all be on the same page throughout the entire project. During these meetings we will also have time for feedback so that each member of the team can have the chance to give feedback to the other group members. This system also allows the advisors to be able to give us feedback on our work week by week.

For IEEE 829-1998 (Software Test Documentation), our team will implement a comprehensive testing framework. With this we will have detailed test documentation which will include, test plans, test cases, and test logs for each project component. Our team will also have automated testing tools during development to ensure consistency in its functionality. All of this will be documented and reviewed by our team members each week and will be discussed during our weekly meeting.

3 Project Plan

3.1 Project Management/Tracking Procedures

Our team is a hybrid (waterfall+agile) approach for project management.

Approach:

Waterfall for High-Level Planning: Each team (Cloud, Backend, Frontend) has a set list of base requirements and deadlines that are defined upfront.

Agile for Execution: Teams will break down the larger requirements into smaller, manageable tasks and work in agile cycles (sprints). Agile allows for flexibility in addressing unforeseen challenges and adapting to changes in requirements during development. Frequent iterations help to prioritize critical features and continuously deliver incremental value.

Coordination and Accountability: One team member will oversee progress, ensure deadlines are met, and assist individual teams with managing their stories.This role bridges the structured planning of Waterfall with the iterative adaptability of Agile, ensuring smooth coordination and delivery.

Justification:

Clear Deadlines: Waterfall approach ensures key deadlines are established and adhered to, supporting project-wide consistency.

Flexibility: Agile cycles allow teams to adapt to changing requirements without derailing overall project timelines.

Collaboration and Ownership: Individual teams retain autonomy over their agile cycles, fostering ownership and accountability within their scope.

Risk Mitigation: Frequent reviews and iterative progress help identify and address risks early, reducing the likelihood of significant roadblocks.

Tools:

Jira: Manage tasks, sprints, and progress using Kanban boards and story tracking. Enables clear accountability and visibility for all teams.

Discord: Primary communication platform for the team and professor, with dedicated channels for team discussions, stand-ups, and file sharing.

Git and GitHub: Version control and collaboration via GitHub repositories, Pull requests ensure code review before integration.

Automated AWS Deployment: GitHub Actions will automatically deploy changes to AWS servers and run tests, allowing for a stable live environment.

3.2 Task Decomposition

Cloud:

- **AWS ECS:**
  - Set up an ECS cluster and define a service for the Node.js server.
  - Configure the ECS task to run the Node.js server.
  - Expose necessary ports on ECS for public access.
  - Integrate ECS with GitHub Actions for automated deployments.
  - Configure auto-scaling and load balancing for high availability.
- **AWS RDS (MySQL):**
  - Provision and configure an RDS instance with MySQL.
  - Set up security groups to allow connections from ECS and specific IPs.
  - Define IAM roles for ECS to securely access RDS via Prisma.
  - Configure and enforce SSL for API encryption.
  - Finalize the connection between RDS and ECS for seamless communication.
- **GitHub/GitHub Actions:**
  - Set up a GitHub repository for CI/CD workflows.
  - Lock down the main branch for stricter code integration policies.
  - Define workflows in GitHub Actions to automate ECS deployment for the Node.js server.
  - Securely define secrets and environment variables for workflows.
  - Configure workflows to build, test, and deploy upon pushes to the main branch.

Frontend:

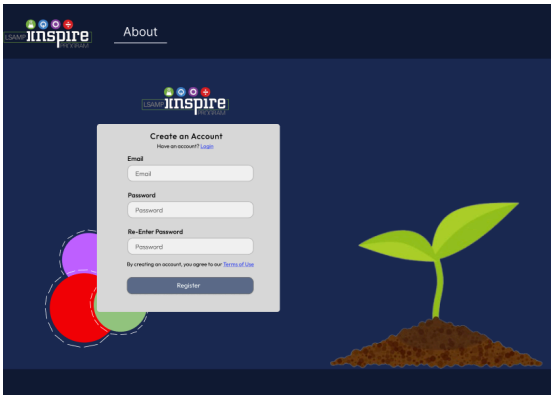
- **Improve user experience**
    - Import frontend code from the previous team as the skeleton for this website.
    - **Refactor the aesthetics of the website**:
        - Create a Figma design as a blueprint for the website.
        - Use Chakra UI to implement the design created in Figma.
    - **API Calls and Forms Refactoring:**
        - Refactor API calls and logic for improved efficiency.
        - Use Formik to manage form states and validation.
        - Learn and implement Next.js for efficient API calls and server-side rendering.
        - Refactor API calls and logic for Next.js server-side rendering.
        - Redesign and implement forms using Formik for enhanced form state control.

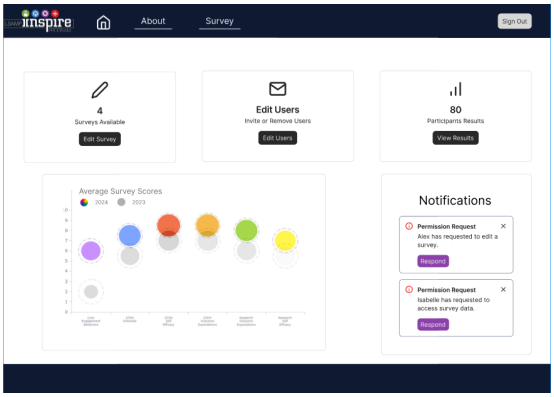
Backend:

- **Metrics for Data Results:**
    - Create detailed metrics such as averages, response rates, medians, and other analytics for admins/researchers.
- **Test Coverage:**
    - Increase test coverage for backend functions.
    - Evaluate current testing strategies (e.g., automated Postman scripts) and explore Jest for backend testing.
    - Integrate API tests into the CI/CD pipeline for continuous validation.
- **Database Usage:**
    - Use pgAdmin to manage user information and data storage in SQL.
    - Retain MySQL for consistency with the current architecture but consider non-relational databases for specific use cases if needed.

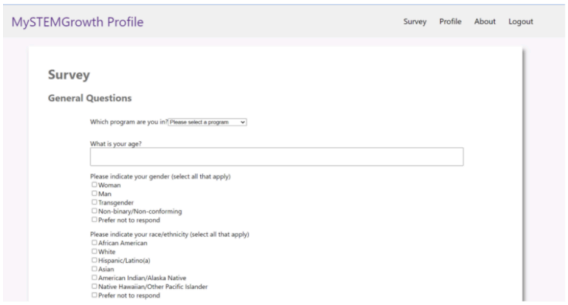
3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

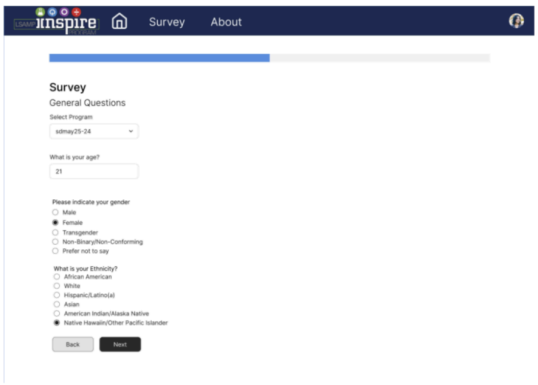
- Full UI redesign that is more appealing for users



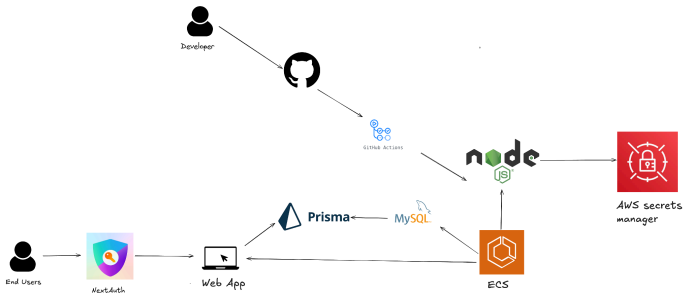


- Modular survey system

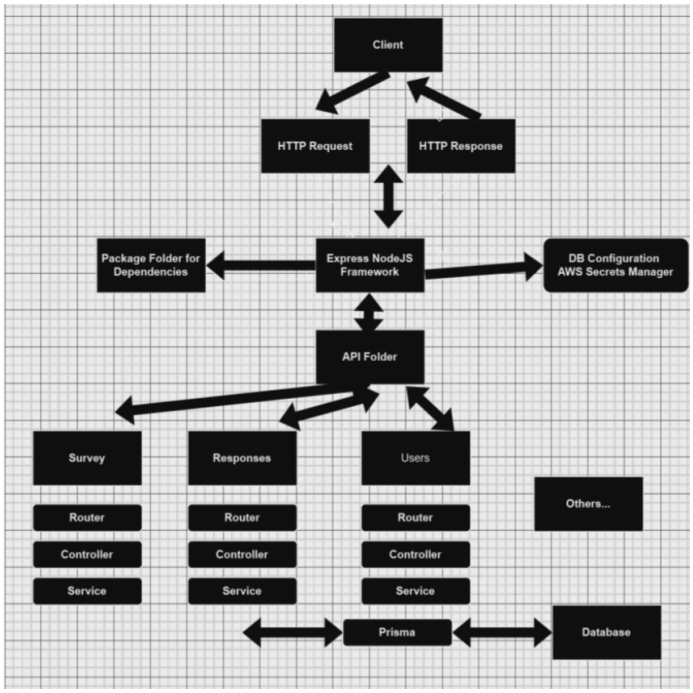




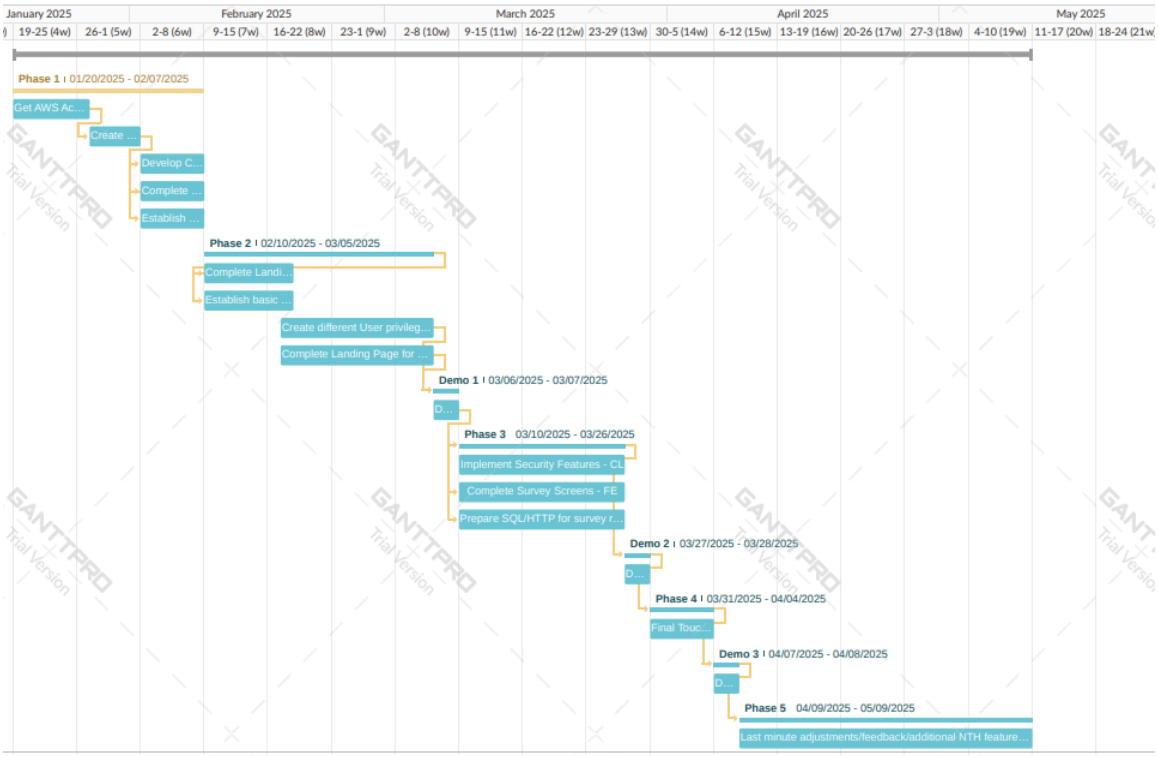
- Cloud Security System



- Backend Structure Rework


Client
HTTP Request
HTTP Response
Package Folder for Dependencies
Express NodeJS Framework
DB Configuration AWS Secrets Manager
API Folder
Survey
Responses
Users
Router
Controller
Service
Router
Controller
Service
Router
Controller
Service
Others...
Prisma
Database

3.4 Project Timeline/Schedule



We have broken down Spring Semester into 5 distinct phases, accompanied by 3 major demo targets. Each phase will approximately correlate to one month (4 weeks) of the semester, with a major demo occurring after phases 2, 3, and 4. Of course, there will be a final presentation, which could be considered demo 4.

Below, there is an easier to read breakdown of all the different tasks and phases, as well as who is responsible for each phase denoted by CL for Cloud team, FE for Frontend Team, and BE for Backend Team.

Proper agile Software Engineering planning and organization would only plan 1-2 iterations out, which means our project will follow a hybrid agile/waterfall style. Therefore, tasks after Phase 2 are technically not in scope yet, and are considered temporary planning items only.

Task Overview

1	⊟	**Phase 1**
1.1		Get AWS Account with ECS up and running - CL
1.2		Create shared Github Repo and attach to AWS - CL
1.3		Develop CI/CD Pipelines - BE
1.4		Complete Home Page - FE
1.5		Establish Basic HTTP Requests
		⊕ Add a task | Add a milestone
2	⊟	**Phase 2**
2.1		Complete Landing Page for Student - FE
2.2		Establish basic SQL Queries - BE
2.3		Create different User privileges - CL/BE
2.4		Complete Landing Page for Program Coordinator - FE
		⊕ Add a task | Add a milestone
3	⊟	**Demo 1**
3.1		Demo current project - All
		⊕ Add a task | Add a milestone

4	⊟	**Phase 3**
4.1		Implement Security Features - CL
4.2		Complete Survey Screens - FE
4.3		Prepare SQL/HTTP for survey results - BE
		⊕ Add a task | Add a milestone
5	⊟	**Demo 2**
5.1		Demo Survey Screens - All
		⊕ Add a task | Add a milestone
6	⊟	**Phase 4**
6.1		Final Touches and Testing
		⊕ Add a task | Add a milestone
7	⊟	**Demo 3**
7.1		Demo Final Project
		⊕ Add a task | Add a milestone
8	⊟	**Phase 5**
8.1		Last minute adjustments/feedback/additional NTH features - All

3.5 Risks and Risk Management/Mitigation

Scope Creep(.6): requirements from clients are not set in stone, and additional features i.e, custom surveys have not been planned out in stone

Plan: Focus on building the core application first, i.e., remaking the first iteration to its completion point, then move on to extra features. Also, we will be able to switch over the cloud developers to any needed role after the core infrastructure is set up.

Project-based Dependencies(.4): different sections of the project rely on other parts to be completed before being able to be started. For example, backend development needs the RDS database to be set up before starting to work with Prisma, and the RDS database needs to communicate with the ECS to test in production.

Timeline slippage(.8): significant project goals i.e. completed figma design, next js app foundation, cannot be fully time blocked as we have such a large team it is impossible to set concrete dates for large features. Also delays can happen in the development/testing phases.

Plan: Break significant goals into smaller goals for each member individually and set concrete dates. Use project management software (and our project manager) to track progress against milestones. Add extra time for delays and have tasks available for members who get done quickly.

Unexpected Costs/ Costs outside Budget(.2) : with ECS being available to scale, costs may randomly shoot up or exceed expected budget during times of extensive use like, for example, testing the application or large groups of survey participants.

3.6 Personnel Effort Requirements

Task	**Average Time Per Person (Hours per week)**
User Login	2
Participant Home Screen	2
Admin Dashboard Screen	2
Participant Survey Page	3
Admin Edit Survey Page	4
Participant Results Page	4
Admin Results Page	4
User Information Storage	4

3.7 Other Resource Requirements

This project involves building a survey system web application, there are no physical parts or materials that are needed to complete the project. Other than physical tools, there are essential resources that are needed, which include:

- AWS Tools
    - ECS to define service and run the Node.js server as well as integrate with GitHub actions
    - RDS for configuring and provisioning MySQL Database as well as API encryption and IAM roles
    - SecretsManager for securely storing, managing, and retrieving user data
    - Cloudwatch for monitoring and logging

- IDEs and Programs
    - Intellij, Visual Studio Code, or any other IDE
    - MySQL database to create schemas that will be hosted on our AWS applications
    - Typescript for our programming language
    - Node.js and the runtime environment to connect our frontend and backend logic
    - GitHub repository for CI/CD and to have a set main branch
    - GitHub Actions to set up CI/CD workflows to automate ECS deployment to the Node.js server
    - Postman for testing our API endpoints for frontend and backend communication
    - NextAuth for username/password security using OAuth
    - Prisma toolkit for easy access when working with our databases

- Project Management Tool: Jira will be our team's source of project management. This allows us to create epics, stories, issues, milestones, and comments for our team’s project development.

4 Design

4.1 Design Context

4.1.1 Broader Context

Describe the broader context in which your design problem is situated. What communities are you designing for? What communities are affected by your design? What societal needs does your project address?

List relevant considerations related to your project in each of the following areas:

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.

4.1.2 Prior Work/Solutions

Universal Design Principles for Web Surveys

Accessibility is a critical component of web design. Crawford et al. discuss the implementation of universal design standards to ensure inclusivity, such as support for screen readers and compliance with W3C accessibility guidelines.

Challenges in Research-Based Web Design

Developing guidelines for survey websites involves navigating browser compatibility and dynamic layouts. IEEE provides insights into these challenges, highlighting the trade-offs between aesthetics and functionality in research-driven web applications

Market Survey and Comparison

Several survey platforms exist, including Google Forms, SurveyMonkey, and Typeform, each with unique features:

1. **Google Forms**
    - **Pros**: Free, easy integration with Google Drive, intuitive interface.
    - **Cons**: Limited customization and advanced analytics.
2. **SurveyMonkey**
    - **Pros**: Rich analytics tools, pre-designed templates, and third-party integrations.
    - **Cons**: Expensive premium plans, steep learning curve for complex surveys.

4.1.3 Technical Complexity

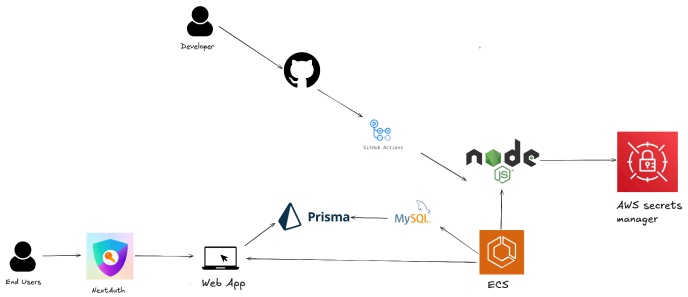
Provide evidence that your project is of sufficient technical complexity. Use the following metric or argue for one of your own. Justify your statements (e.g., list the components/subsystems and describe the applicable scientific, mathematical, or engineering principles)

1. The design consists of multiple components/subsystems that each utilize distinct scientific, mathematical, or engineering principles –AND–
2. The problem scope contains multiple challenging requirements that match or exceed current solutions or industry standards.

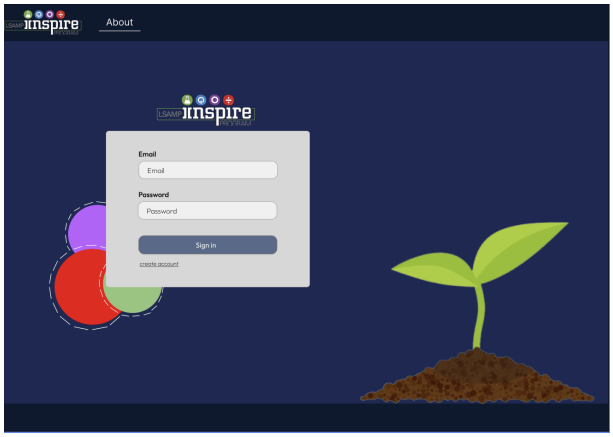
4.2 Design Exploration

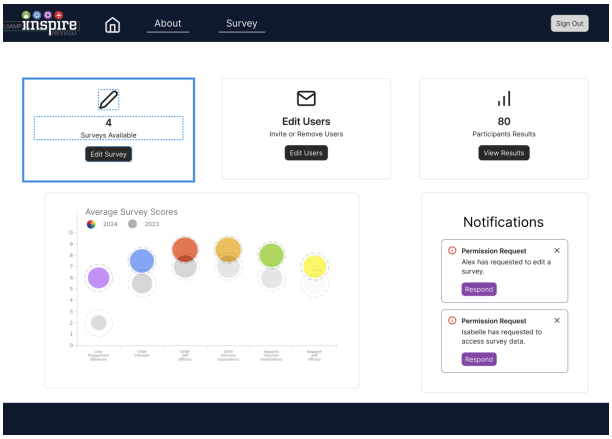
4.2.1 Design Decisions

Cloud Server Rework:

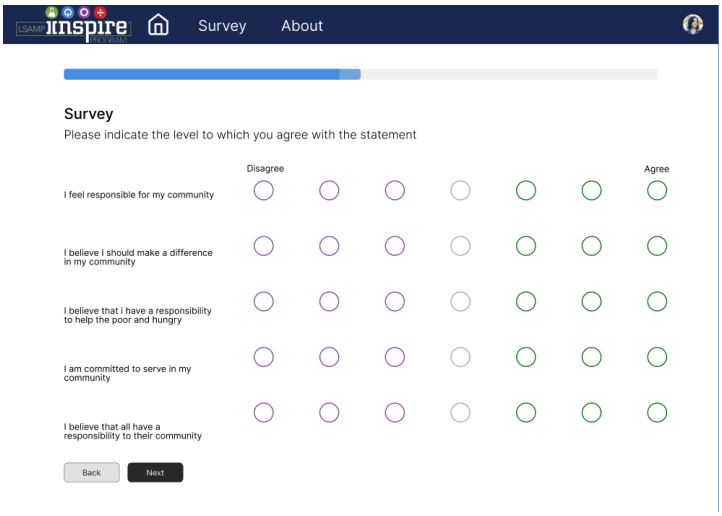


Reworked Landing Pages:





New Survey Markers:



4.2.2 Ideation

We considered many designs for the survey questions page to make it more visually appealing and user-friendly. We had an idea to add sliders and dropdown options for some of the questions to change the style and keep users engaged. Another idea we had was to add color to the survey bubbles themselves as shown in the picture. The color gives the user a better sense of range when deciding if they strongly agree or disagree on a question/ topic.

4.2.3 Decision-Making and Trade-Off

One apparent pros of the design choices we considered is that the page is more visually appealing, which was our goal. Users will be more engaged with these questions compared to the previous design of gray circles. One of these design choices will be the implementation of the various questions. The more we incorporate, the harder it will be to keep track of different question types, which may cause an error.

4.3 Proposed Design

4.3.1 Overview

The IINSPIRE STEM program supports underrepresented minority students, helping them engage in STEM fields. Researchers need a tool to administer surveys and visualize results to improve program support more efficiently. The current survey website, developed by the University of Iowa, is fundamental in design and needs essential functionality. The problem will be solved by redesigning the frontend and upgrading the infrastructure to improve the web app's speed and performance. This will make the survey-taking process faster for participants and allow researchers to create and manage different surveys easily. Continuous feedback from users will guide these improvements and the addition of new features.

4.3.2 Detailed Design and Visual(s)

Our system will be evenly distributed between our three teams of cloud, backend, and frontend. Because we are choosing to redo most of the previous design, each group has significant work to accomplish.

Cloud, Backend, and Frontend Breakdowns

Cloud: All changes have been made to improve simplicity and reduce our systems' complexity. ECS has been integrated to allow for automatic scaling of services, CI/CD allows for changes to be automatically deployed, and Prisma/NextAuth reduces the need for other developers to create similar services

Backend: The team's lack of experience with new technologies like SSO integration, Prisma, and other security features poses a potential setback to project progress as developers get over the learning curve and implement different technologies. We are incorporating many industry standards to the design which is a great experience for all team members to gain new experience with commonly used web technologies.

Frontend: Providing revamped Web and Mobile interfaces for users. Significant visual changes include layout, functionality, style, colors, etc. None of us have significant experience in wireframe design, so we have to learn and experiment.

4.3.3 Functionality

The functionality of our web application can be divided among our four user types: admin, program coordinator, researcher and student.

For our admin, their goals when using our application are to be able to create and edit surveys, invite/remove users, update/patch the app and approve accessing/downloading survey data. Since many of their roles involve responding to requests, we decided to have a notification panel so they can easily approve/disapprove requests from lower tier users. In addition to overseeing requests, admins will also want to get a broad overview of how participants interact with the survey, which is why we display a chart showing the average of survey results over time and an option to view individual participants' survey results.

For the program coordinator their main responsibilities are to create/modify the surveys, view survey data results, share and download survey data, and modify survey visualization methods. Since their main concern is with the survey results itself , their dashboard is designed to allow different types of survey updates and modifications.

Researchers will also focus on the surveys, but with an emphasis on the results and collecting data. Considering this, we plan to allow survey results to be downloaded as csv files to allow for efficient data analysis. We will also give researchers the ability to submit requests for new survey questions and to request to edit current questions. Since they will need to be able to submit requests, we will have a "message" feature to allow them to send requests to admin for approval.

Finally, for students, their intended use of the application is to be able to take the survey and view survey results. After conducting a user feedback session, one common request from students was to have a page dedicated to resources at the University to help with their academic growth and community involvement. Considering this, we decided to add a page for resources divided into academics (such as tutoring and SI), clubs (such as WISE and Solar Car), and events like career fairs and company visits.

4.3.4 Areas of Concern and Development

Our current design heavily focuses on meeting the user's need for the same system functionality with an updated user interface. We have created a figma prototype of the revamped web interface and are iteratively updating this design through user interviews and user experience resource feedback. We are confident that this iterative process will leave us with an upgraded web design that meets user requirements. Our backend design focuses on recreating the previous team's solution using better technologies. We have captured all user functionality requirements from the previous team's design and will gather more information about backend requirements through user interviews with the University of Iowa research team.

The primary concern of our team's current design is the integration of new backend technologies. These technologies are new to the team and could pose a

significant learning curve to getting a working version. However, these technologies will help create a more secure web application and have become the industry standard. Our team will actively monitor the timelines of implementing these technologies, and if some aspects of the backend design cannot be reasonably built our team has prioritized the most important of the backend features to be implemented. Another concern is fixating our design on meeting the student user group's need for an updated Web Interface and not spending enough time meeting the researcher user group's requirements for data collection and data exportation. Our client clarified to our team to prioritize the student user group first over the researcher user group. Our design can provide more functionality depending on the timeframe of our first prototype and the client's satisfaction with our design.

The team has several immediate plans to address these different concerns over our design. The team has scheduled a meeting with a Iowa State's Design Department member to receive feedback on our initial figma prototype. We hope to gain insight about the user experience of our web screen mockup as well as design thoughts regarding the default layout of each screen. The team has also scheduled a meeting with the University of Iowa researcher team. This meeting will help answer concerns over any research user group requirements and how to prioritize implementing them. This meeting will also offer insight into the standardization of data collection and exportation from our web application for the researchers to use. The immediate questions we have for our advisor is due to the slow process of working with ETG to get a new AWS account set up. How will this setback the original project timeline? We are wondering what the nature of the survey portability feature would look like for the University of Iowa team? How different would a different university program survey look in comparison and what infrastructure would we need to support the addition of programs to IINSPIRE.

4.4 Technology Considerations

Front-end:

Next.js: Next.js is a java framework built on React. This is what we will be using to create our web application.

Pros:

- file based routing: for developers, file based routing is easy to visualize and understand because you can organize your route structure via files and directories instead of through code
- built in cache: this allows for faster speed and enhanced performance

Cons:

- potential learning curve for more advanced features

Chakra: Chakra is a component library that is easily integrated with React frameworks, such as Next.js which is what we are using to develop our application

Pros:

- easy to learn
- components are built to work well on different screen sizes, which is important for our project since we will be hosting our web app on desktop and mobile devices

Cons:

- some computational overhead which can impact performance on large-scale projects (although this shouldn't be a concern for our project specifically)

Node.js: While Node.js is primarily a backend technology, the frontend will implement a Node.js package manager to manage dependencies.

Pros:

- npm: manage dependencies and allow the use of third party libraries in our project
- it can be used for server-side rendering with Next.js, which can improve performance

Cons:

- Node.js is a newer platform, so not all libraries are available yet

Front-end design alternatives: One of the drawbacks of Next.js is its steeper learning curve. An alternative platform is Angular since it is more beginner-friendly. However, since the previous version of the application was built in React, this should simplify the migration to Next.js.

Back-end:

NodeJS: The existing project is already using NodeJS for the backend. We looked into other backend frameworks but given that the frontend will be done in React it makes the most sense to continue to use NodeJS. Tons of different libraries already exist.

Pros:

- Fast Execution with V8 Engine: Node.js is powered by Google's V8 engine, which compiles JavaScript into machine code, making execution fast. This speed is beneficial for handling high I/O-bound operations and real-time applications.
- Asynchronous and Event-Driven: Node.js follows an event-driven, non-blocking I/O model, meaning it can handle multiple requests concurrently without blocking. This makes it suitable for highly concurrent applications, like chat servers or live streaming.

Postman: Can continue to use automated scripts in Postman

Pros:

- User-Friendly Interface: Intuitive GUI that makes creating and managing API requests easy.
- Built-in Testing: Allows you to write tests in JavaScript directly within the application.

Cons:

- Limited to API Testing: Primarily focused on testing APIs, so it's not suitable for broader backend application testing.
- Dependency on GUI: While there are command-line options (Newman), the primary functionality is GUI-based, which can be less efficient for some users.

Jest:

Pros:

- Rich Feature Set: Provides a complete solution for unit and integration testing with built-in assertion libraries.
- Fast and Simple: Runs tests in parallel and has a straightforward API, making it easy to start.
- Snapshot Testing: A unique snapshot feature allows you to track changes in API responses or components over time.

Cons:

- JavaScript Focused: Primarily designed for JavaScript and TypeScript applications, so not suitable for other programming languages.
- Learning Curve: While simple for basic tests, more complex features (like mocks) can have a learning curve.

Cloud:

GitHub and GitHub Actions

- Pros: Automated CI/CD; integrated with GitHub; customizable workflows.
- Cons: Limited free usage; complex setup for advanced workflows; potential vendor lock-in.

Node.js Backend

- Pros: JavaScript across the stack; scalable; extensive package ecosystem.
- Cons: Single-threaded limitations; higher memory usage; potential for callback hell.

AWS Secrets Manager

- Pros: Secure credential management; automated secret rotation; easy AWS integration.
- Cons: Additional costs; slight latency; added complexity for small apps.

Prisma ORM

- Pros: Type-safe queries; simplifies database access; schema management.
- Cons: Learning curve; potential performance overhead; limited flexibility for complex queries.

Amazon RDS (MySQL)

- Pros: Managed service with automated backups; scalable.
- Cons: Costs; limited customization; potential latency.

NextAuth (Authentication)

- Pros: Supports multiple providers; secure by default; integrates well with Next.js.
- Cons: Learning curve; mainly for Next.js; limited customization for complex flows.

Amazon ECS

- Pros: Scalable; managed container orchestration; AWS integration.
- Cons: Complex setup; costs increase with scale; AWS-specific, limiting portability.

4.5 Design Analysis

The backend team has developed a system to support the functionality of Program user's editing and modifying the program's survey. This entailed creating a CSV file with all the data needed to create the default version of the survey. We then created a python script to parse the CSV file, allowing us to export an updated version of the survey to our database, which will then be displayed on our web application. Getting this functionality to work has allowed us to disregard any notion of redesigning screens for the admin to edit questions. Instead, this design allows us to provide the functionality of updating the survey without having to take the time to create a complex user interface for admins and instead focus front-end resources on the student experience. This feature of the design will impact the design of our AWS server. Knowing that we will have to execute a Python script means that we will need a Python interpreter and several Python libraries on the server on top of all of the node packages.

Cloud Team Progress

Redeveloped Cloud Design: The cloud team worked on redesigning the cloud infrastructure with a focus on our key goals: increasing security, setting up for long-term development, simplfing the development process, and reducing costs.

- Prisma/NextAuth Integration: simplifies database queries and allows greater code reuse and Next Auth adds layer of authentication in login
- Transitioned from an EC2-based infrastructure to ECS for containerized deployment, improving scalability and reducing operational overhead.
- Decided to consolidate AWS accounts, retaining the original account and tearing down the previous infrastructure after Dr. Rover's conference to preserve data and reduce costs.

Key Updates and Challenges:

- Original plan to self-host the code on GitHub was modified; Iowa State ETG will now host the repository to simplify access for future teams.
- Progress on cloud infrastructure setup has been delayed, as we are waiting for Iowa State ETG to provision resources, including the GitHub repository and AWS account configurations.
- Plans to establish the initial cloud infrastructure were postponed due to provisioning delays and the conference requirement to showcase the existing app.

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 veried 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 prole 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 disrupt existing functionality, we have implemented a two-layer approach to regression testing:

Automated Testing: automated tests are executed whenever a merge request is made to main. These tests make sure that new features do not interfere with existing critical functionality

Code Review Process: All code changes required human review before being integrated. Smaller Features will require at least one team member to review and approve the changes. Larger Features: will require approval from two team members to ensure code quality, maintainability, and compliance with project standards.

Acceptance Testing

Requirement Validation:

> Team will conduct a walkthrough of the project, addressing each requirement. A comprehensive checklist will be used during this process to track progress and confirm that all larger specifications have been met. This step ensures internal confidence in the product's quality before presenting it to the client.

Client Presentation and Feedback:

> The completed project will be presented to the client for review. This presentation will confirm that the project aligns with their expectations and to collect feedback. Following agile practices, we will hold regular feedback sessions throughout development. This

approach ensures the final product is ready for approval upon completion.

5.6 Security Testing (if applicable)

Objective: Identify vulnerabilities and security weaknesses in the system.

Process

- **Planning**
  Define the scope of the penetration test, including the systems and components to be tested.
- **Reconnaissance**
  Gather information about the system from an external perspective.
- **Scanning**
  Use automated tools to identify open ports, active services, and potential vulnerabilities.
- **Exploitation**
  Attempt to exploit identified vulnerabilities to gain unauthorized access or perform malicious activities.
- **Post-Exploitation**
  Assess the impact of successful exploits and document findings.
- **Reporting**
  Provide a detailed report outlining the vulnerabilities, their severity levels, and recommended remediation actions.

5.7 Results

Overview of the Testing Process

- **Objectives**: Define testing goals, including ensuring functionality, reliability, and performance.
- **Methods Used**: Describe the testing techniques employed, such as unit testing, integration testing, and user acceptance testing.
- **Scope**: Specify the features, components, and aspects of the project covered in the testing phase.

Showing Results

- **Test Reports and Documentation**: Create detailed reports summarizing the outcomes of each testing type.
- **Issue Identification**: Highlight any defects or problems encountered during testing for resolution.

Functional Requirements

- **Traceability Matrices**: Map each functional requirement to the corresponding test cases designed to verify them.
- **Verification**: Confirm that passing test cases indicate the fulfillment of related functional requirements.

Validation

- **Stakeholder Requirements**: Include a validation step to confirm that the project meets stakeholder expectations and aligns with the design specifications.

6 Implementation

Describe any (preliminary) implementations of your design thus far. Support any general, descriptive text with relevant images. If your project has inseparable activities between design and implementation, you can list them either in the Design section or this section.

Frontend:

Throughout the semester, the frontend team met with individuals from IINSPIRE, the University of Iowa, and CSSM (Center for Survey Statistics) at Iowa State to get feedback on the previous team's design and what could be improved in the design of not only the survey design but the whole infrastructure of the website. Using these notes, the frontend team developed Figma designs as mock-ups for every single view on the website (e.g. taking the survey, logging in, creating an account, dashboard, etc...).

Cloud Infrastructure:

Cloud team met with the ETG to discuss and improve our infrastructure plans. We gained full access to the previous senior design team's AWS accounts and coordinated with ETG to set up our AWS accounts. After finalizing our design, we conducted a cost estimation to ensure it stayed within budget. Finally, we arranged for hosting costs to be billed directly to our senior design team.

Backend:

App.js Main file to start the server. Creates a Node.js Server using the Express Framework

-Imports dependencies

-Imports routers from the API folder

Middleware setup to parse Json request bodies and parse cookies from incoming requests.
Package.json is a configuration file

Lists information on the node.js server and allows you to list Dependencies for the project.

7 Ethics and Professional Responsibility

Building Trust: Honest communication creates a foundation of trust among group members.

Enhancing Collaboration: Honest communication fosters an environment where group members feel comfortable sharing ideas and giving constructive feedback.

7.1 Areas of Professional Responsibility/Codes of Ethics

High Areas: Work Competence, Communication Honesty, Social Responsibility. 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 our client's requests while learning more about how to act ethically and professionally. For Communication Honesty, we strive to keep open communication channels 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 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 are medium in these categories because we are still developing our skills in regard to these topics. Since we are using AWS, we know we want to find the tools that will be the best for our project, without incurring large costs on our client, but we have yet to complete that process entirely. We are also medium for Property Ownership because we know this will be an important area of responsibility when we begin implementing our design. Since we are working with the user's private data and pre-determined measures from the research team, 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 entirely nonapplicable 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 to the environment.

7.2 Four Principles

Area	Beneficence	Nonmaleficence	Respect for autonomy	Justice
Public health, safety and welfare	The purpose of this tool is to help students get the resources and education they need to grow academically but also personally.	Students will benefit from understanding what areas they excel in and what areas they need to improve in. This will help them in their careers and with their journey in school, as they will know what areas to spend more time improving.	Allowing students to understand their strengths and weaknesses will help them make an informed decision about their future without forcing them to choose a certain path.	Helping students get resources to improve in areas they may be weaker in because of a lack of exposure to STEM related courses or concepts will help level the playing field and give them a fairer chance at success.

Global, cultural and social	Helping underrepresented communities highlights our project's focus on humanity. The biggest goal of our project is to help those who traditionally have been excluded from STEM fields get opportunities they might not have otherwise.	Introducing communities who are underrepresented in STEM to our program will benefit those communities by giving them more information and resources. This complies with the idea of avoiding causing harm.	Our project aims to expose underserved communities to STEM majors and careers. We will disclose pertinent information about these career paths to help students decide if this is the right choice for them.	A major goal of our project revolves around giving equal opportunities to those in cultural, social and economic communities that are underrepresented in STEM fields.
Environmental	While our project itself doesn't directly impact the environment, If the students who participate in the program lead a career in environmental sciences or something related, they could contribute positively to protecting our environment.	Our project does not deplete or harm our environment of its resources.	Introducing students to STEM majors involved in protecting our environment can equip students with the resources and knowledge to contribute towards this effort if it is something that interests them.	By involving individuals from underrepresented communities in environmental fields, we can help create opportunities for them to address environmental challenges that directly impact their communities. This not only benefits these communities but also contributes valuable insights and solutions to broader environmental issues.
Economic	Students who do a career in a STEM field will likely see a positive impact on their financial situation.	A career in a STEM field typically does not harm a person financially, and in fact tends to have the opposite impact.	Exposing students to STEM careers may present opportunities to better their financial situation through high paying careers. Introducing	Helping all students have more equal opportunities to STEM careers also gives more equal opportunities to financial wellbeing.

			these opportunities gives students autonomy over their finances.	

Our project emphasizes the connection between global, cultural and social contexts and justice. We strive to bridge the gap in STEM representation by providing underrepresented students with access to opportunities, resources to succeed academically, and a supportive community that extends through college and beyond. While the project does not directly address environmental issues, it holds the potential to contribute indirectly. Participants in the IINSPIRE program may pursue STEM careers that drive advancements in environmental protection and sustainability.

7.3 Virtues

Compassion: Combining empathy and action to alleviate the plight of others

Justice: Ensuring fairness by refraining from causing harm or disadvantage to others and actively working to restore justice when needed

Integrity: Aligning our conduct with the way we view the world by backing up our words and beliefs with our actions.

Our team has shown compassion by actively working to provide underrepresented students with resources and opportunities in STEM. This effort demonstrates a commitment to alleviating the disparities they face, combining empathy for their struggles with actionable solutions such as mentorship, academic support and access to STEM programs.

By focusing on bridging the gap for students traditionally underrepresented in STEM fields, our team promotes justice. We are making an effort to create equitable access to education, offering opportunities to address systemic inequalities, and building a supportive community, which reflects our commitment to restoring fairness and providing everyone with their due chance to succeed.

Our team has upheld integrity by aligning our actions with our mission to empower underserved communities in STEM. We have demonstrated this virtue by contributing towards the IINSPIRE program and ensuring all team members are held accountable for supporting the project's goals.

Alex:

Virtue I have demonstrated: Adaptability

Why it's important: Adaptability is important because the needs of our client and the structure of our team is dynamic, so flexibility is crucial

How I have demonstrated it: I stepped into a leadership role to address the evolving needs of our team and to act as a liaison between our group and our client.

Virtue I have not demonstrated: Patience

Why it's important: It's important to maintain a composed demeanor during challenging situations or when facing setbacks

How I can demonstrate it: I can demonstrate patience by taking a step back to assess the situation calmly and provide thoughtful responses to feedback

Charlie:

Virtue I have demonstrated: Dedication

Why it's important: Dedication ensures sustained focus toward achieving long term goals

How I have demonstrated it: I have demonstrated dedication by consistently preparing and organizing the necessary resources to ensure our project is fully equipped for implementation next semester

Virtue I have not demonstrated: Assertiveness

Why it's important: Assertiveness is important for effectively communicating ideas and opinions

How I can demonstrate it: I can demonstrate assertiveness by confidently expressing my thoughts during meetings and respectively challenging ideas when necessary

Isabelle:

Virtue I have demonstrated: Humility

Why it's important: Humility is important for recognizing areas for growth and embracing feedback to improve our project

How I have demonstrated it: I have demonstrated humility by actively seeking input from students in the program and experts in Human Computer Interaction and website design

Virtue I have not demonstrated: Perseverance

Why it's important: It's essential for pushing through challenges and staying committed to the project's goals even when progress feels slow or obstacles arise

How I can demonstrate it: I can demonstrate perseverance by continuing to work through technical difficulties or unexpected challenges

Landon:

Virtue I have demonstrated: Compassion

Why it's important: Compassion allows team members to build trust and support each other by addressing challenges with kindness and empathy

How I have demonstrated it: I have demonstrated compassion by being understanding toward my teammates' challenges and offering help when they need it

Virtue I have not demonstrated: Resilience

Why it's important: Resilience is important for bouncing back from setbacks and maintaining a positive outlook despite challenges

How I can demonstrate it: I can demonstrate resilience by staying motivated despite setbacks, learning from failures and encouraging others to persevere

Matthew:

Virtue I have demonstrated: Integrity

Why it's important: Integrity is important for establishing trust and credibility with others by aligning actions with our values and principles

How I have demonstrated it: I take ownership of my responsibilities and follow through on commitments such as meeting deadlines, participating in group meetings and helping other group members when needed

Virtue I have not demonstrated: Collaboration

Why it's important: It's essential for creating a cooperative team environment where everyone's contributions are valued

How I can demonstrate it: I can demonstrate collaboration by seeking input from others and working closely with teammates to achieve our common goals

Max:

Virtue I have demonstrated: Vision

Why it's important: Vision is important because it guides our efforts towards creating an impactful, lasting change

How I have demonstrated it: I have demonstrated this value by outlining the big picture for our project design and contributing ideas to improve our project

Virtue I have not demonstrated: Pragmatism

Why it's important: It is essential for making practical and realistic decisions that align with the team's goals and resources

How I can demonstrate it: I can demonstrate pragmatism by evaluating the feasibility of my ideas, considering potential challenges, and making adjustments based on practical considerations

Nick:

Virtue I have demonstrated: Inclusivity

Why it's important: Inclusivity ensures that everyone feels valued and heard. Inclusivity strengthens team cohesion and celebrates diverse perspectives.

How I have demonstrated it: I have demonstrated inclusivity by making sure all team members' opinions are included in discussions and decision making

Virtue I have not demonstrated: Curiosity

Why it's important: It's important for driving innovation and continuous learning because it encourages the exploration for new ideas and perspectives

How I can demonstrate it: I can demonstrate curiosity by constantly asking questions and exploring different approaches to improve our project

8 Closing Material

8.1 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 significant 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 tricky 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 unique knowledge and understanding of the project and what is needed. Next semester, we plan 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.

8.2 References

- Crawford et al., "Applying Web-Based Survey Design Standards" in Journal of Prevention and Intervention in the Community.
- World Wide Web Consortium (W3C), "Web Accessibility Initiative (WAI)" standards.
- Dillman, D. A., *Mail and Internet Surveys: The Tailored Design Method*.
- IEEE, "Challenges in Developing Research-Based Web Design Guidelines," *IEEE Xplore Digital Library,* 2015. [Online]. Available: https://ieeexplore.ieee.org/document/7304204.

8.3 Appendices

N/A

9 Team

9.1 Team Members

Alex Moeller - Project Manager

Nick Pinnello - Backend

Charlie Moreland - Backend

Max Strater - Frontend

Isabelle Raghavan - Frontend

Matthew Bennett - Cloud

Landon Gulotta - Cloud/Security

9.2 Required Skill Sets for Your Project

- Front-end Development
    - Figma
    - React
    - Github
    - Java
- Back-end Development
    - MySQL database
    - Node.js
    - Prisma
    - Github
- Cloud Development
    - NextAuth (SSO)
    - AWS ECS
    - AWS Secrets Manager
    - Amazon RDS
    - Github Actions

9.3 Skill Sets covered by the Team

Alex Moeller: Java, Object-Oriented Programming, React, TypeScript, HTML/CSS

Max Strater: React, TypeScript, HTML/CSS, Node.js, Next.js,

Landon Gulotta: Cloud Development with AWS applications and networking. Additionally as a cybersecurity major he has experience with security tooling and networking for the project.

Charlie Moreland: Java, PHP, Javascript, development of backend applications, Database Design. Experience with API Testing using Postman.

Isabelle Raghavan: React, Javascript, TypeScript, HTML/CSS

Matthew Bennett: Cloud Development with AWS applications and networking. React, Javascript, TypeScript, HTML/CSS

Nick Pinnello: Java, Object-Oriented Programming, React, TypeScript, HTML/CSS, Database/ SQL management

9.4 Project Management Style Adopted by the team

Our team has implemented the Agile development process with weekly meetings and 2 week sprints. The meetings consist of a meeting once a week with our team and a meeting once a week with our advisor.

9.5 Initial Project Management Roles

Front-end Development: Max Strater, Isabelle Raghavan

Back-end Development: Charlie Moreland, Nick Pinnello

Cloud Development: Matthew Bennett, Landon Gulotta

Project Management/Misc: Alex Moeller

Advisor/Client Communication: Alex Moeller

9.6 Team Contract (NOT DONE)

Team Members:

1) Nick Pinnello 2) Max Strater
3) Matthew Bennett 4) Alex Moeller
5) Charlie Moreland 6) Landon Gulotta

7) Isabelle Raghavan

Team Procedures

Team Meetings: Thursdays 2-3 Library

Advisor Meetings: Wednesdays 1-2 0115 Office and Lab Conference room

Our preferred method of communication is discord for communication with advisors and other university representatives and text messages for individual/ group communication.

When making team decisions we discuss pros and cons and go by majority vote.

We keep track of our hours with our report documents and hold each other accountable to ensure we work a certain amount weekly.

Participation Expectations

Advisor Meetings:

We expect attendance from everyone at the advisor 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

Leadership Roles:

Our team decided with seven members it would be beneficial to have a team member take on a project manager lead. We nominated Alex to this role in addition to his contribution to the cloud team. This role oversaw the integration of all three teams: Cloud, Frontend, Backend. This kept the focus of all three teams on the overall goals of the project and not falling into the trap of not seeing

the bigger picture. Alex also created a Jira board with tasks for each team member in terms of design items as well as planning items for the upcoming semester.

Max and Isabelle took on the leadership role of being in charge of User Interviews. Our frontend design had several meetings with different user groups to get feedback and they were in charge of collecting this feedback and scheduling these meetings.

Charlie and Nick took on the leadership roles of meetings with our client the University of Iowa. They worked with our advisor Dr. Rover to schedule these meetings and lead the execution of these meetings. They were in charge of all communications with our client.

Landon and Matthew were in charge of communication with Iowa State's ETG department. They scheduled meetings with the department and communicated all information from these meetings to the rest of the team.

Support Strategies:

Stand-up meetings during weekly meetings play a key role in supporting each other's progress. These short, focused meetings allow team members to quickly share updates on their work, discuss any challenges, and align on priorities. Each member reports on what they've accomplished, what they're currently working on, and any roadblocks they're facing. This structure helps identify potential issues early, allowing the team to provide assistance or resources to overcome obstacles.

Recognition Strategies:

Our team used Discord Channels as a strategy for recognizing each other's contributions. Team members would post major accomplishments to their respective discord team channels and team members would congratulate them and review their work. This allowed us to encourage and recognize one another while communicating and staying on the same page with our Advisor.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the

team

- **Alex Moeller (Project Manager):** Skilled in leadership and organization, Alex ensures team coordination and clear communication with the client.
- **Nick Pinnello (Backend Developer):** Strong in backend development, Nick specializes in secure database management and API design.
- **Charlie Moreland (Backend Developer):** Charlie focuses on backend optimization and reliable database systems.
- **Max Strater (Frontend Developer):** Max combines technical skill and creativity to design functional and visually appealing user interfaces.
- **Isabelle Raghavan (Frontend Developer):** Isabelle excels at creating intuitive, user-friendly designs with attention to detail.
- **Matthew Bennett (Cloud Engineer):** Matthew is experienced in designing scalable and efficient cloud infrastructure.

- **Landon Gulotta (Security Engineer):** Landon brings expertise in cybersecurity, ensuring the application is secure and resilient.

2. Strategies for encouraging and support contributions and ideas from all team members:

Regularly scheduled weekly meetings (both team and advisor-led) to discuss progress, challenges, and new ideas.

Open communication channels, such as Discord, to encourage informal brainstorming and collaboration.

Recognition of individual contributions during meetings and in project communications to motivate and foster engagement.

3. Procedures for identifying and resolving collaboration or inclusion issues

Team members can report concerns during meetings or directly to the project manager for confidential discussions.

A constructive feedback system ensures all team members feel heard and valued.

Persistent issues are escalated to the advisor for impartial mediation and resolution.

Goal-Setting, Planning, and Execution

Team Goals for the Semester:

Our primary goal is to create a fully functional application that meets all user and client requirements. Key items include:

Finalizing and integrating wireframe designs into the application.

Improving the user interface and overall experience by incorporating feedback from stakeholders.

Setting up and deploying a secure, scalable cloud infrastructure to ensure reliability and future growth.

Performing thorough testing to confirm the app's functionality, security, and usability.

Strategies for Planning and Assigning Work:

To stay organized and make steady progress, the team will:

Use Jira to manage tasks, making it easier to track responsibilities and deadlines.

Break down big milestones into smaller, manageable tasks to keep progress consistent.

Assign work based on each team member's skills while encouraging collaboration across different areas when needed.

Strategies for Staying on Task:

To ensure the project stays on track, we will:

Use Kanban boards to visually monitor progress, deadlines, and priorities.

Hold weekly sprint reviews to check in on progress, resolve challenges, and adjust plans as needed.

Include extra time in the schedule to deal with unexpected challenges, ensuring we meet our deadlines without unnecessary stres

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) Max Strater DATE 12/5/24

2) Landon Gulotta DATE 12/5/24

3) Alex Moeller DATE 12/5/24

4) Isabelle RaghavanDATE 12/5/24

5) Matthew Bennett DATE 12/5/24

6) Nick Pinnello DATE 12/5/24

7) Charlie Moreland DATE 12/5/24