

# IINSPIRE STEM Survey Visualization Tool

Final Design Document

SD24 - Team 35

Clients: University of Iowa, Dr. Diane Rover

Advisor: Dr. Diane Rover

Team Members/Roles

Jimmy Driskell - Front-end

Nathan Frank - Front-end

Kaitlin Hansen - Front-end

Lydia McCleary - Back-end

Thomas Nunez - Back-end

Abe Scheideman - Back-end

Team Email: [sdmay24-35@iastate.edu](mailto:sdmay24-35@iastate.edu)

Team Website: <https://sdmay24-35.sd.ece.iastate.edu/>

# Table of Contents

<b>Introduction</b> .....	3
Background.....	3
Problem Statement.....	3
Intended User and Uses.....	3
<b>Revised Design</b> .....	4
Functional Requirements.....	4
Nonfunctional Requirements.....	4
Engineering Standards.....	4
Security Concerns and Countermeasures.....	5
Evolution of the Design.....	5
<b>Implementation Details</b> .....	5
Detailed Design.....	5
Description of functionality.....	6
Implementation Notes.....	7
<b>Testing</b> .....	7
Unit Testing.....	7
Interface Testing.....	7
Integration Testing.....	8
System Testing.....	8
Regression Testing.....	8
Security Testing.....	8
Results.....	9
<b>Broader Context</b> .....	12
Public Health, Safety, and Welfare.....	12
Global, Cultural, and Social.....	12
Environmental.....	12
Economic.....	12
<b>Conclusions</b> .....	12
Overview.....	12
Progress Discussion.....	12
Value.....	13
Potential Future Steps.....	13
<b>Appendix 1 - Operation Manual</b> .....	14
<b>Appendix 2- Alternative/Initial Version of Design</b> .....	26
<b>Appendix 3- Code</b> .....	26

## Table of Figures

Figure 1- Student User Flow.....	6
Figure 2- Application Architecture.....	6
Figure 3- Front-end test coverage report.....	9
Figure 4- Postman API testing folders.....	9
Figure 5- Login Tests.....	10
Figure 6- Login Password Parameter Testing.....	11
Figure 7- Login Email Parameter Testing.....	11
Figure 8 - Landing Page.....	14
Figure 9 - Login or Sign Up options on the Landing Page.....	14
Figure 10 - Login Page.....	15
Figure 11 - Sign-up Page.....	15
Figure 12 - Admin Home Page.....	16
Figure 13 - Admin Results Page.....	16
Figure 14 - Admin Create New Program Page.....	17
Figure 15 - Admin Edit Survey Page.....	17
Figure 16 - Admin Access Page.....	18
Figure 17 - Reset User Password Page.....	18
Figure 18 - Student Home Page.....	19
Figure 19 - Survey General Information Page.....	19
Figure 20 - First Survey Question Page.....	20
Figure 21 - Bottom of a Survey Questions Page.....	20
Figure 22 - Final Survey Page.....	20
Figure 23 - Top of Results Page.....	21
Figure 24 - Bottom of Results Page.....	22
Figure 25 - Results Page with Comparisons between Two Survey Attempts.....	22
Figure 26 - Results Page with Comparisons between Two Survey Attempts in Bar Chart Form.....	22
Figure 27 - Results Page with Textual Analysis.....	23
Figure 28 - AWS Amplify.....	23
Figure 29 - AWS EC2.....	24
Figure 30 - AWS EC2 Deployment.....	24
Figure 31 - AWS EC2 Target Groups.....	25
Figure 32 - Postman Connection to EC2 Instance.....	25
Figure 33 - Original Design Figma Prototype.....	26

# Introduction

## Background:

IINSPIRE program aims to diversify the STEM workforce by providing individuals from underrepresented populations with the resources to learn STEM.

## 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 to administer the surveys and programmatically generate graphs and charts related to the results.

## Intended User and Uses:

Our system has two user types: admin/instructors and students.

Admins/Instructors are able to:

- View individual student's results from the surveys they have taken
- Create a new program for students to be in
- Edit the questions in a pre-existing survey
- Assign the admin role to another student
- Reset a student's password

Students are able to:

- Create an account
- Complete pre and post-program surveys for multiple programs
- Request and view graphs related to their survey results
- Download their survey 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 those interested in STEM and comes 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 they need 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 take the surveys and be more intentional about their answers and what they share in the surveys.

## Revised Design

### Functional requirements:

- Our web tool must be able to administer multiple surveys to each program participant throughout their experience in a program
- It must be possible for the survey to display pre-determined measures/questions
- 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 possible for an admin to export the data to a comma-separated values (.csv) file
- 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\***

## 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 the team and client regarding 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.
-

## ISO Standards

### ISO 29119 - Software Testing

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

### ISO 27001 - Information Security

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

## Security Concerns and Countermeasures

Our largest security concerns are protecting user data in the survey responses and their login credentials. We implemented HTTPS to ensure the safe and secure data transfer and invoked an SSL certificate for the site's encryption. For future iterations, greater database security should be implemented, as there can never be “too much security,” especially when protecting sensitive user data.

## Evolution of the Design

Since 491, our back-end architecture has expanded to include more AWS services and a more robust database schema. As we began developing this semester, we discovered more requirements for our database. Initially, we only had tables for storing user data, survey information, and survey responses. However, we quickly realized we needed to expand our database to handle role privileges (administrator vs. student/participant), various survey formats, and calculations for the user's responses on where they rate themselves on a numerical scale. On the AWS side, we integrated Route 53 to aid in hosting our site.

On the front-end side, we received continuous feedback from our clients at the University of Iowa on how to modify the user interface to achieve accessibility standards as well as match the client's vision. The front-end went through several iterations of the user interface, from what was planned last semester to what the final product is now. The initial design can be viewed in Figure 33.

## Implementation Details

### DETAILED DESIGN

Detailed design of our JavaScript Technologies: React (front-end framework), AnyChart (graph generation), jsPDF (PDF generation), Axios (HTTPS requests), JWT (secure data transfer between the front and back-end), Node.js (back-end server application), Express.js (server routing framework), bcrypt.js (password hashing), Jest (endpoint and React component testing), and SwaggerUI (endpoint documentation).

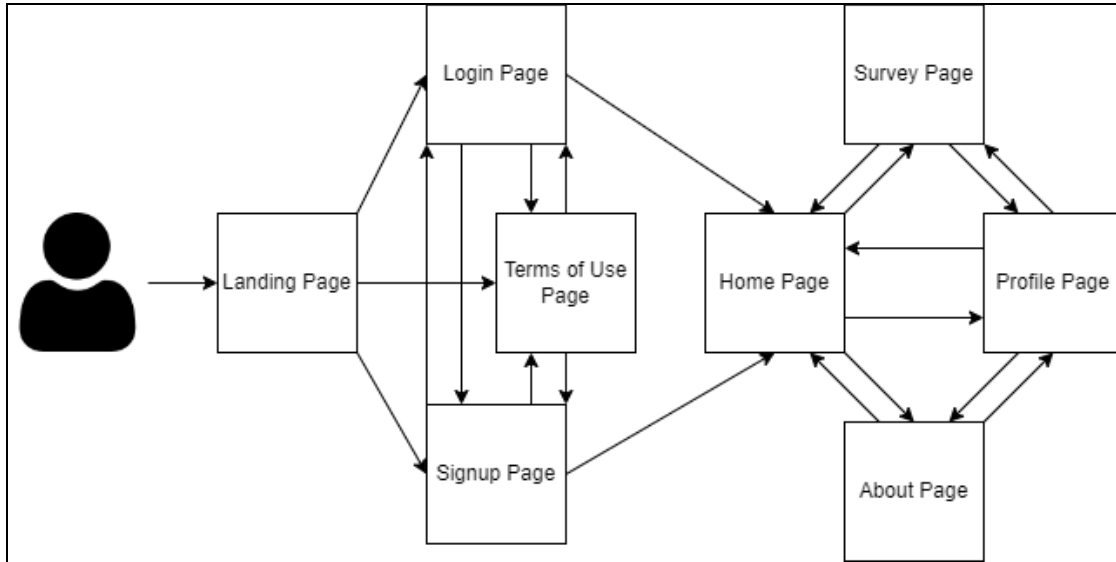


Figure 1- Student User Flow

Detailed Design of AWS Resources: Amplify (front-end React app cloud hosting), Route53 (domain registration), Certificate Manager (SSL certification), EC2 (back-end Node.js app cloud hosting), Secrets Manager (storing the password and host for the database), and RDS (MySQL cloud data storage).

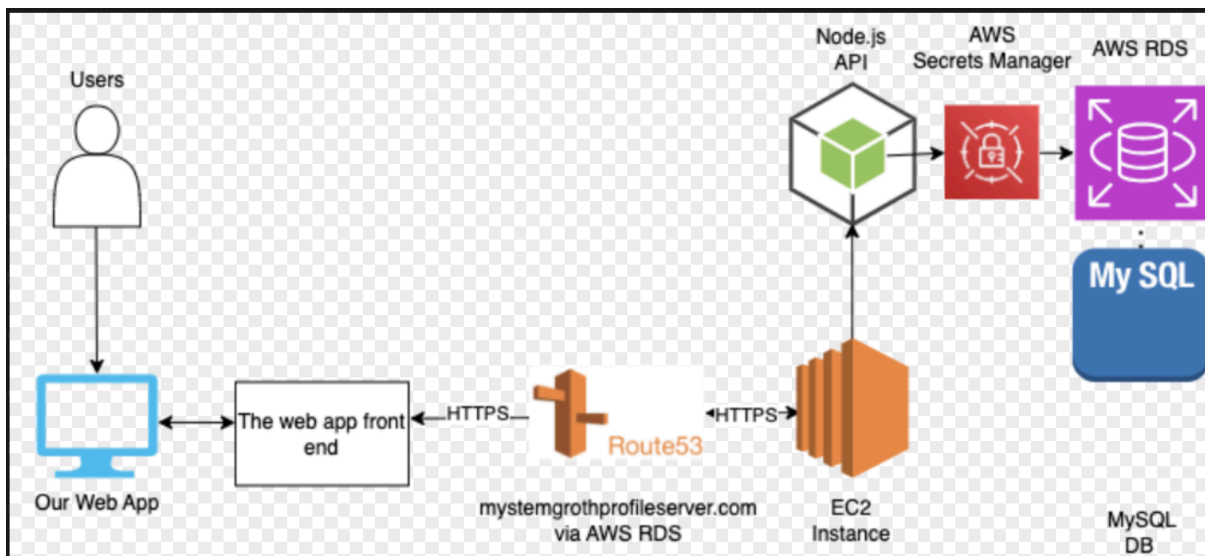


Figure 2- Application Architecture

### DESCRIPTION OF FUNCTIONALITY

Our thorough, detailed design is what allowed us to build the engaging functionality necessary for our users to complete surveys and view their data. Using JSON Web Tokens, we unlocked the functionality of creating accounts for individuals. Once a user creates a profile, they can interact with the main application. As with any web application, there needs to be an engaging user experience. To design the UI in a modular fashion, we took advantage of the plentiful resources for React developers to build interesting experiences. As for our core functionality of generating charts from survey data, we utilized AnyChart. Since we're a team of students, we were able to secure a license for the JavaScript library AnyChart, for free. We integrated the library directly into our front-end application to build our graphics from the student's survey data stored in

our MySQL database. This data was retrieved using the Axios library to format HTTPS requests to our Node.js server hosted on an EC2 instance.

Once our server application received the request, it used the Express.js framework to route a query to our MySQL server running on AWS RDS securely by using a password stored in AWS Secrets Manager. Once the charts were generated on the front-end using the survey results from our MySQL database, users have the ability to download a PDF of their results using the JavaScript library jsPDF. Lastly, admin users have a wider range of functionality that regular users don't. Using the same tech stack flow, admins can download CSVs of student survey data, assign new surveys to students, and even reset passwords. There are further plans to extend functionality for admins as well.

## IMPLEMENTATION NOTES

In order to define our application and the problems it solves, we conducted several rounds of client interactions with researchers at the University of Iowa. With each iteration, adjustments were made to the look and functionality of the application. Through these interactions, we were able to reach an implementation we are all proud of. By implementing our functionality purely with JavaScript and AWS, we have set a cohesive pattern for future development. One of the most crucial steps in our implementation was utilizing Postman to test our endpoints for retrieving data. Without this, we would have sparse visibility on if the features we were building would be useful. Having an ongoing conversation about what data the front-end team would need and how that manifestation would work from the back-end team was also crucial. Lastly, although there were some hiccups along the way, each team member pulled together to implement each feature.

Our plan was to implement an automation between taking a survey and generating charts based off of that data. In future implementations, our application could be used to serve thousands of students and mentors in getting a better grasp on how they can best use their interests and talents in STEM. With further user testing, AWS resource load balancing, and security updates, the baseline implementation we have accomplished could be a production-grade application.

## Testing

### UNIT TESTING

For the front-end created with React, the Jest Testing Library was used to write unit tests for React components, simulating user interactions and verifying their behavior. Jest was also used to test functions and simulate HTTP requests. For the back-end, Node.js code was tested using Jest to assess the correctness of API endpoints and the back-end logic. 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.

### 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.



## INTEGRATION TESTING

The Postman tool was used to test the Node.js back-end and React.js front-end. It mocked API calls to the back-end and mocked endpoints for the front-end. Postman was used to test the Node.js API and React.js front-end. It checked how well they worked together. Tests were set up in Postman to verify if they communicated correctly. Simulated requests were sent to the API to assess its responses. Postman's features, like variables and scripts, made testing straightforward. Batch testing was also possible, saving time. Postman ensured the Node.js API and React.js front-end integration was smooth and error-free.

## SYSTEM TESTING

Considering that our project was entirely software-based, the “system” we are testing is the entire web tool. Testing the entire system, in our case, involved 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 was present include when a user tries to log in and when a user imports data, expecting a result in graphics. Manual testing was performed to ensure that all functionality of the running application worked for the user.

## REGRESSION TESTING

A CI/CD pipeline on our code ran each time the code was pushed to the repository. This ensured there were no changes that caused features to stop working when changes were committed. After each iteration cycle, the project was also tested manually to ensure no breaking changes on any of the previous features or the features added from the iteration.

## SECURITY TESTING

### Types of Testing:

#### 1. Authentication and Authorization Testing

- **Objective:** Verify that only authorized users can access specific functionalities.
- **Process:**
  - User Authentication Testing:
    - Tested login mechanisms, including normal users and admin logins.
    - Verified the strength of password policies.
  - Authorization Testing:
    - Checked if users have appropriate permissions based on their roles.
    - Ensured that users cannot access unauthorized data or perform restricted actions.
      - For example, a non-admin user cannot access admin pages

#### 2. Data Encryption Testing

- **Objective:** Ensure data security during storage and transfer.
- **Process:**
  - Data Storage Encryption Testing:
    - Verified 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.

## RESULTS

The front-end did most of its testing manually, making sure that the changes worked as expected, and the unit tests helped more with the regression testing, ensuring that the new changes did not cause other issues to arise. A coverage report for the front-end unit tests can be seen in Figure 3. All known scenarios and errors that could arise, such as the back-end API going down, were accounted for during the testing.

File	% Stmts	% Branch	% Funcs	% Lines
All files	34.5	12.64	19.87	34.73
components	61.11	33.33	37.5	61.11
AdminNavBar.js	0	0	0	0
Card.js	75	50	33.33	75
Footer.js	100	100	100	100
LandingNavbar.js	75	50	50	75
context	50	100	0	50
AuthProvider.js	50	100	0	50
functions	81.81	50	66.66	82.75
graph.js	81.81	50	66.66	82.75
hooks	100	100	100	100
useAuth.js	100	100	100	100
routes	26.82	7.43	16.42	26.99
about.js	100	100	100	100
adminAssign.js	27.77	0	20	27.77
adminCreate.js	43.75	50	20	43.75
adminHome.js	100	100	100	100
adminPasswordReset.js	24.13	0	14.28	25.92
adminResults.js	21.66	100	7.14	21.66
adminSurvey.js	22.91	0	5.88	23.4
forgotPassword.js	100	100	100	100
home.js	100	100	100	100
landing.js	75	100	50	75
login.js	30	0	14.28	30
navbar.js	75	50	50	75
profile.js	21.33	4.16	20	21.33
requireAuth.js	100	50	100	100
signup.js	25.8	0	11.11	25.8
survey.js	25.51	4.34	9.8	25.64
termsOfUse.js	100	50	100	100

Figure 3- Front-end test coverage report

For the back-end, due to time constraints, we did most of our testing through Postman. We structured our API testing in two folders, one for local requests and one for main requests:

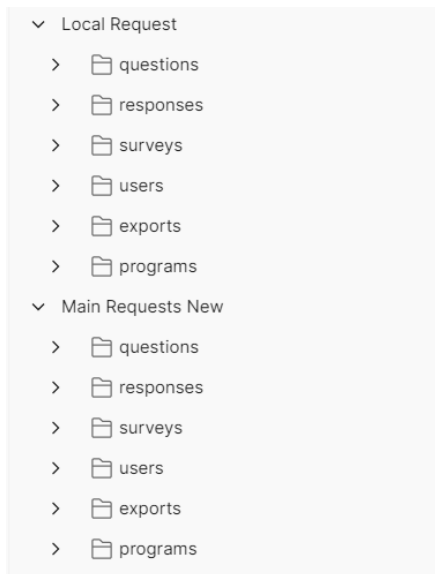


Figure 4- Postman API testing folders

We ran many tests through Postman to ensure each function worked properly. Figure 5 is an example of a failed login attempt:

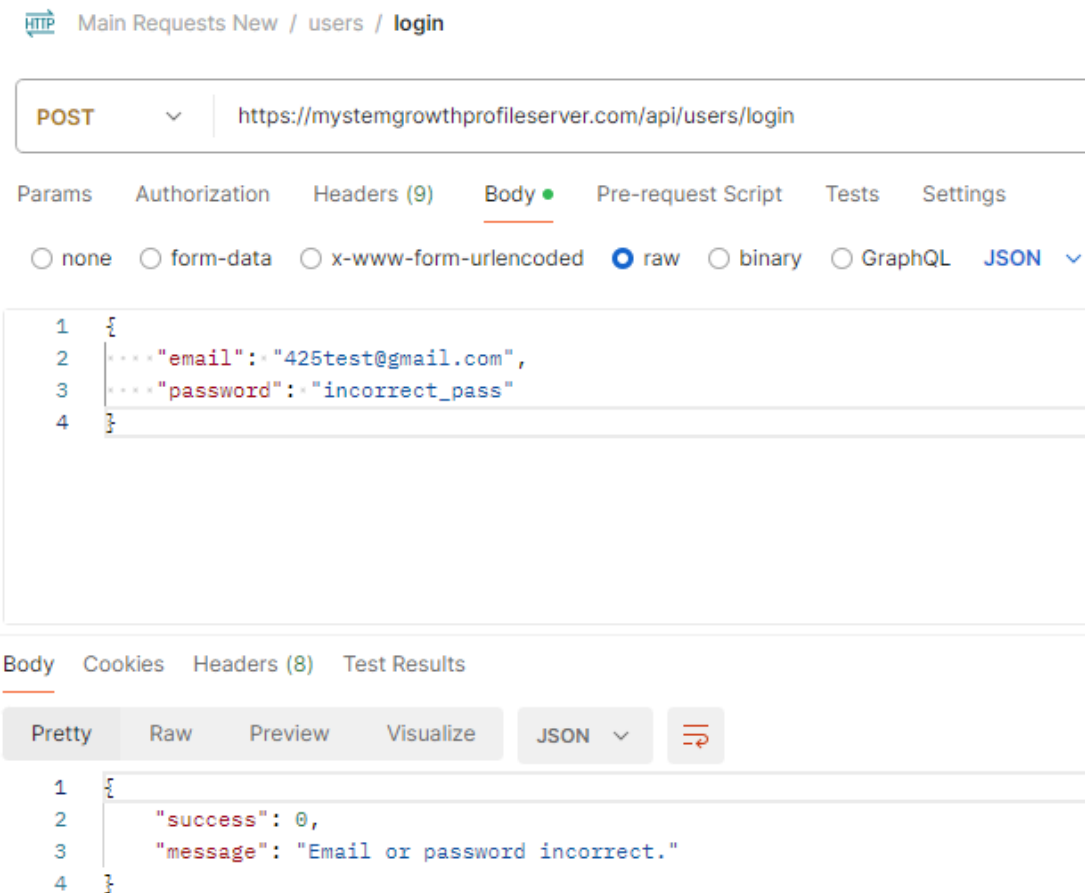


Figure 5- Login Tests

With this, we also tested our validation mechanisms to ensure that the input being sent in was the input we expected. For example, when an input such as "pas23sword" is sent in, the request should fail due to the only inputs allowed being "email" or "password":

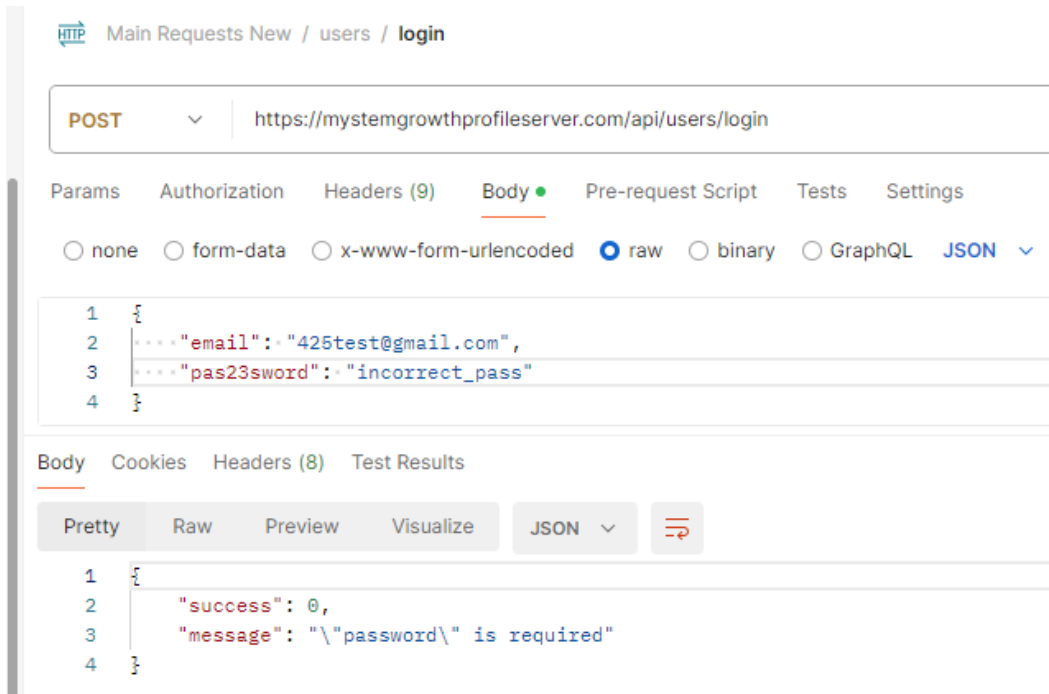


Figure 6- Login Password Parameter Testing

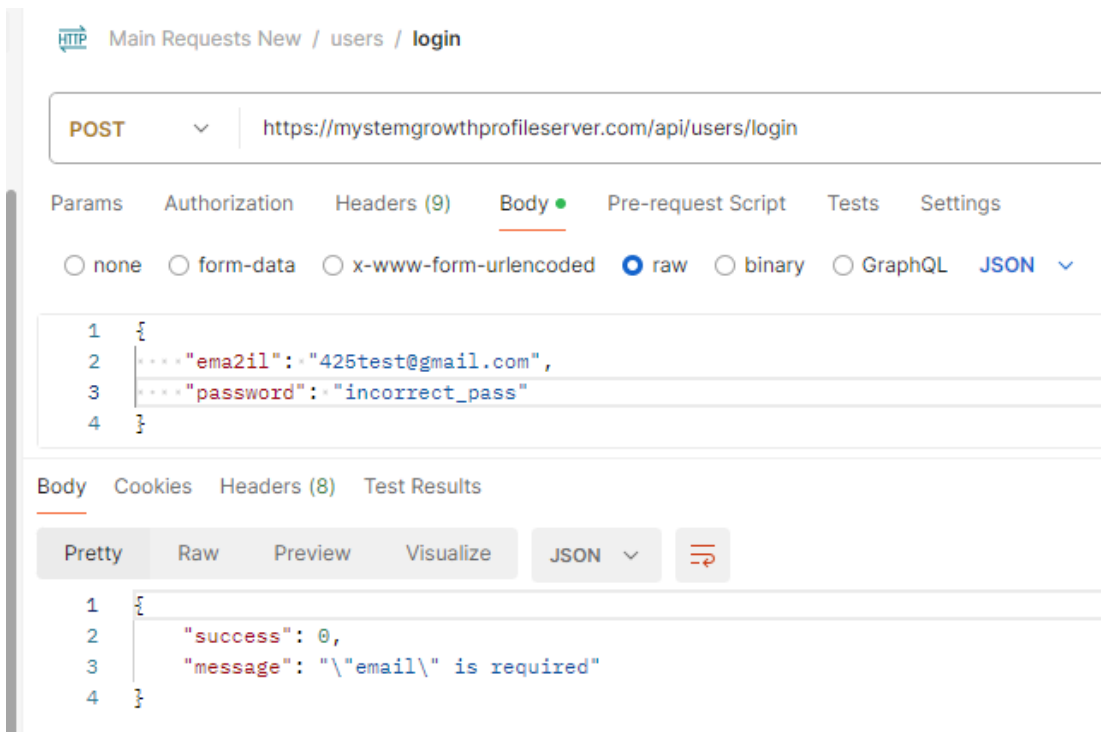


Figure 7- Login Email Parameter Testing

## Broader Context

### PUBLIC HEALTH, SAFETY, AND WELFARE

This web application will aid IINSPIRE and other programs in delivering their mission to empower students. Participants will be able to measure their strengths and growth throughout the duration of their program participation.

### GLOBAL, CULTURAL, AND SOCIAL

This project supports IINSPIRE and its mission to assist students in underserved communities and introduce them to STEM.

### 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.

### 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.

## Conclusions

### OVERVIEW

During this semester, we focused on the implementation of our project. We utilized the information we had gathered last semester regarding requirements, users, project context, client expectations and more to inform our decisions. During the previous semester, we came up with a high-level, end-to-end design for our project. We selected which frameworks, coding languages, and other technical tools we thought would best serve our project. At the end of last semester and over winter break, we further familiarized ourselves with these tools so that we could begin implementation right away. We followed an agile approach for our development. We met weekly as a team for standup-style meetings, along with meeting with our advisor on a weekly basis. We attempted to meet with our client every two weeks to demonstrate progress and receive timely feedback; however, due to scheduling conflicts, we would sometimes have 3 weeks between client meetings, followed by meetings with only a week in between. Overall, this approach still worked well for us. We entered the semester with a strong plan for how we wanted to approach development, and although we had to make many adjustments to our approach and design, we still finished the semester with a solid product.

### PROGRESS DISCUSSION

Throughout the course of the two semesters working on this project, there have been many different visions from us as a team and our client, for what our product should look like by the end of the spring semester. Ultimately, we all agreed on the goal of having our product ready for use by the client this summer with the IINSPIRE LSAMP 2024 cohort. We believe that the end product that we delivered to the client will meet the

needs for their use this summer. Administrators will be able to create program cohorts, view the program's survey results, and view individual students' survey results, along with being able to upgrade other users to admin status. Students will be able to register for a program, take surveys, view their results with graphic and textual descriptions, along with being able to download these results.

### VALUE

When discussing the progress made through the duration of this project, it can be seen how the product will bring value to its users and towards the problem it was meant to solve. Students participating in research with IINSPIRE LSAMP will be able to use our website to take their surveys and will be able to receive feedback and analysis in a much more timely manner than the previous methodology. Prior to this project, research personnel had to analyze the students' survey results and create analysis visuals by hand, which was very time-consuming and tedious for the research personnel. Furthermore, this process meant that students did not receive feedback until much after completing their surveys. By automatically generating graphics and other forms of survey feedback and analysis, our product is able to save both students and researchers valuable time.

### POTENTIAL FUTURE STEPS

As a team, we are happy with the progress we made with this product during our senior design cycle. However, there are many features our client and we envision for a future iteration of this project. In its current state, our product only allows students to take the default survey provided by our client however, in future iterations, we would like for administrators of the program to be able to customize surveys for different groups of students, either by editing the default questions or by uploading entirely new surveys. The options for requesting student data/responses on the admin side could also be expanded. Currently, admins are able to see an individual student's averages across the six research measures the survey questions are based on. Otherwise, they can request to see all the responses to each question from every student in a program cohort, along with the averages for each student in each of the 6 question groups. In the future, it would be useful for admins to be able to request data organized by different demographics, such as gender, age, or race. Additionally, it would be ideal for future iterations to provide more in-depth data to the admins, such as all of the raw survey response data organized into CSV form. Beyond user features, another potential future step would be to increase the website's security. Although we have taken basic security measures, having a team member dedicated to ensuring the cyber security of the product on a future iteration of this would be very beneficial.

# Appendix 1- Operation Manual

## 1. NAVIGATE TO URL

The first thing a user must do is navigate to our website at <https://www.mystemgrowth.com>. They will encounter the landing page as shown in Figure 8. From there they can select the dropdown menu in the upper right corner to either log in with an existing account or sign up to create a new account as shown in Figure 9.

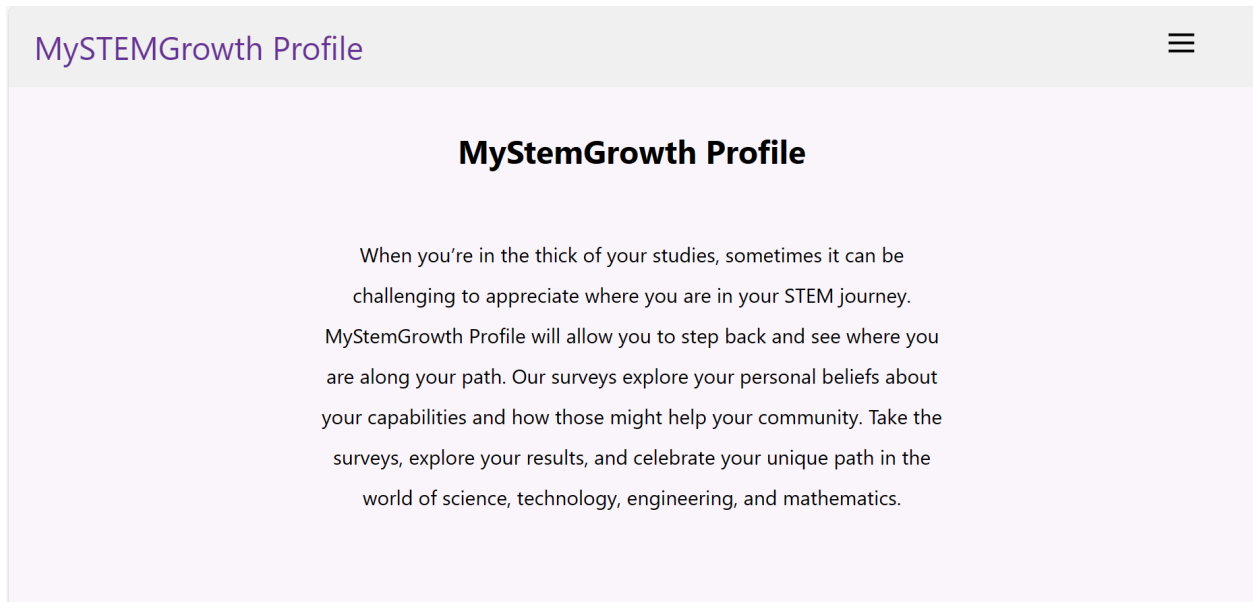


Figure 8 - Landing Page

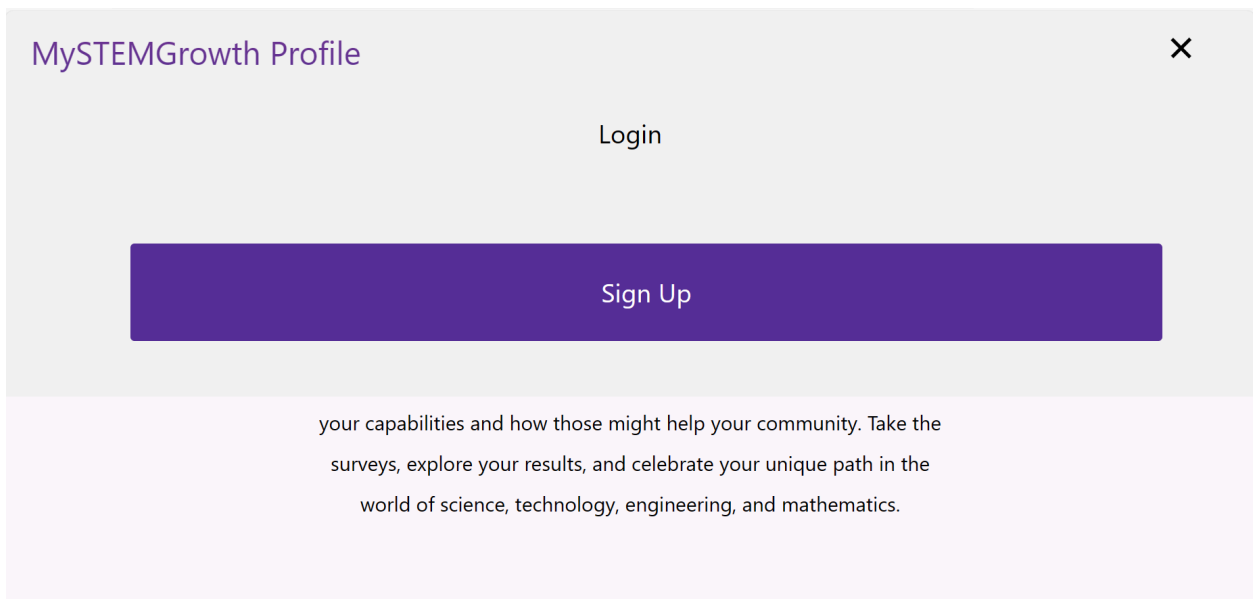
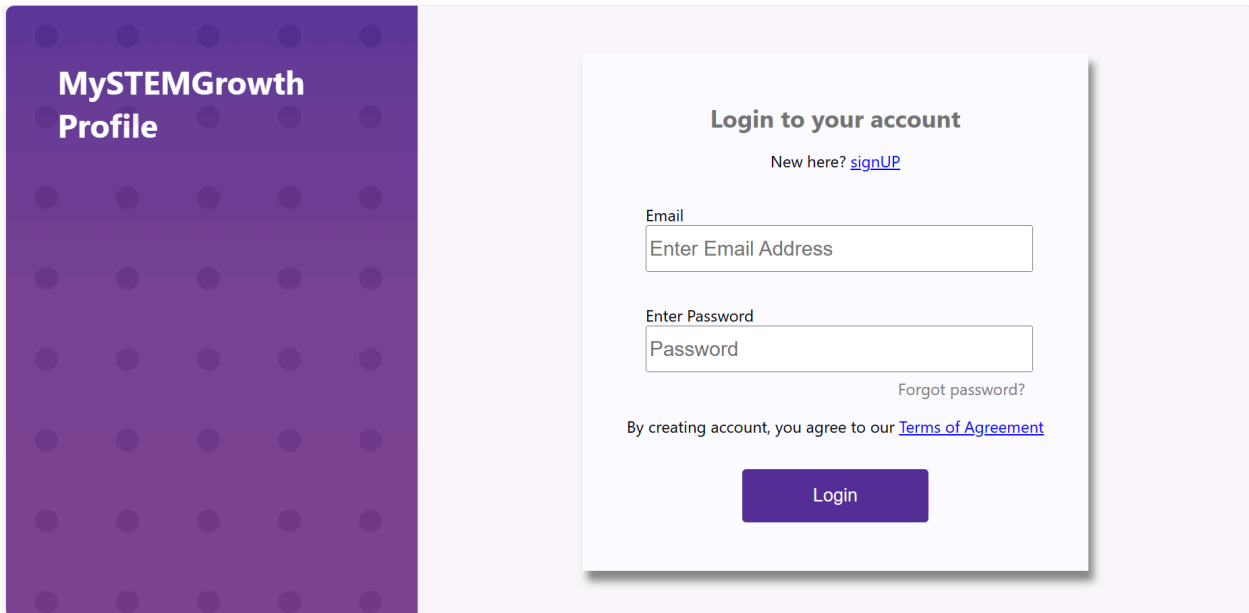


Figure 9 - Login or Sign Up options on the Landing Page

## 2. LOGIN OR SIGNUP

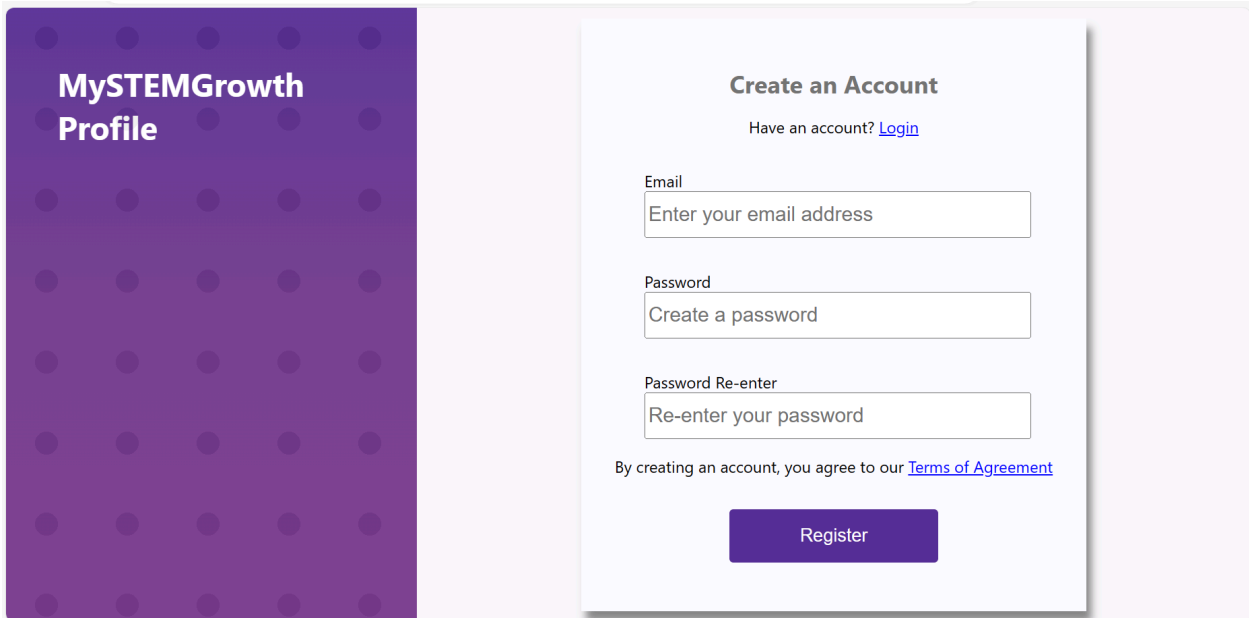
A user can either log in with their email and password if they already have an account using the screen shown in Figure 10 or sign up for one if they don't have an account, as shown in Figure 11.



The screenshot shows the login page for MySTEMGrowth Profile. On the left is a purple sidebar with the text "MySTEMGrowth Profile" and a grid of dots. The main content area is white and contains the following elements:

- Login to your account** (Section Header)
- Link: "New here? [signUP](#)"
- Form field: "Email" with placeholder "Enter Email Address"
- Form field: "Enter Password" with placeholder "Password"
- Link: "Forgot password?"
- Text: "By creating account, you agree to our [Terms of Agreement](#)"
- Button: "Login" (purple)

Figure 10 - Login Page



The screenshot shows the sign-up page for MySTEMGrowth Profile. On the left is a purple sidebar with the text "MySTEMGrowth Profile" and a grid of dots. The main content area is white and contains the following elements:

- Create an Account** (Section Header)
- Link: "Have an account? [Login](#)"
- Form field: "Email" with placeholder "Enter your email address"
- Form field: "Password" with placeholder "Create a password"
- Form field: "Password Re-enter" with placeholder "Re-enter your password"
- Text: "By creating an account, you agree to our [Terms of Agreement](#)"
- Button: "Register" (purple)

Figure 11 - Sign-up Page

**NOTE:** From here, the user's experience will diverge based on their user type: an administrator or student. We will first go over the admin experience.



### 3. ACCESS ADMIN HOME PAGE

Once an admin-type user logs in, they will be taken to the admin home page, as shown in Figure 12. From here, they can decide what action they would like to take.

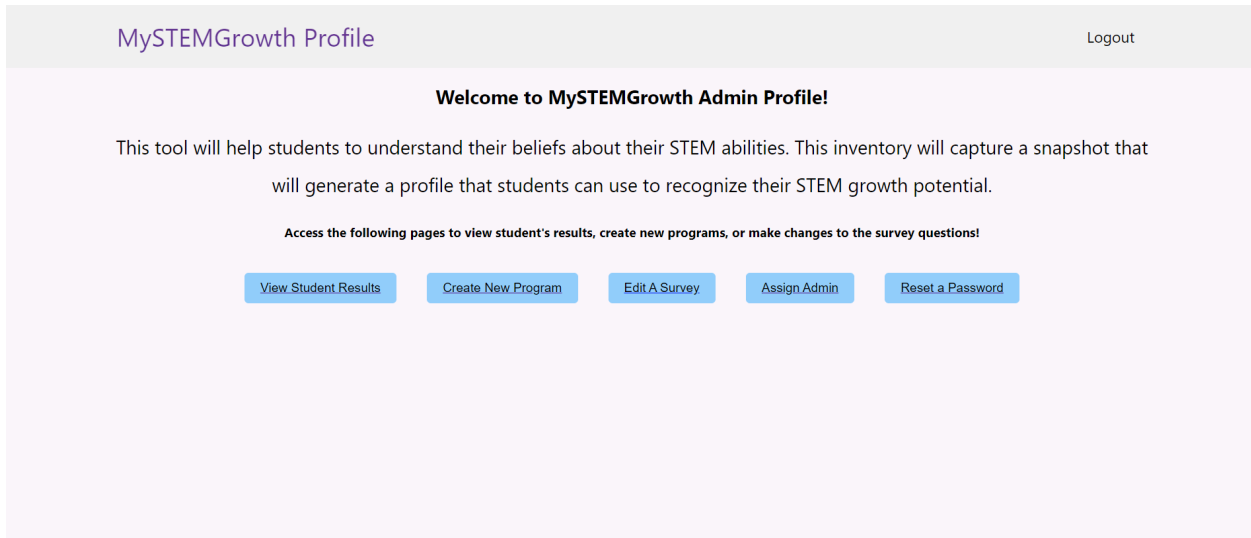


Figure 12 - Admin Home Page

### 4. VIEW STUDENT RESULTS

One action an admin can take is to view student survey results. By clicking on the “View Student Results” button, they are taken to the page shown in Figure 13 where they can request survey data for an individual student by email or for an entire program cohort by name.

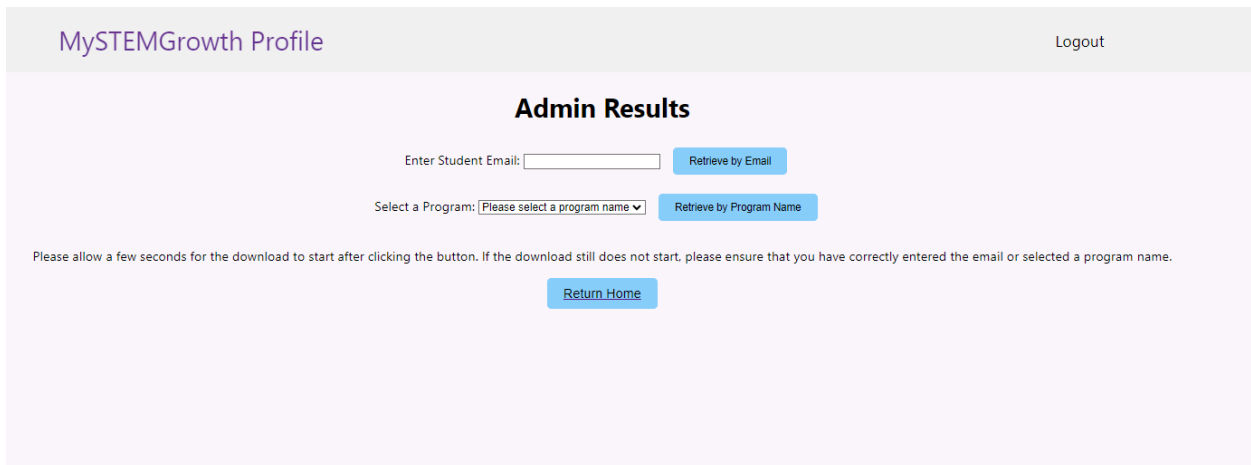


Figure 13 - Admin Results Page

### 5. CREATE NEW PROGRAM

From the home page, the administrator can also create new programs. After clicking on the “Create New Program” button, they are taken to the following page, as shown in Figure 13 where they can enter the desired new program name. Upon clicking the “Create New Program” button next to the textbox, a new program is created and students in that program will have access to the default survey questions.

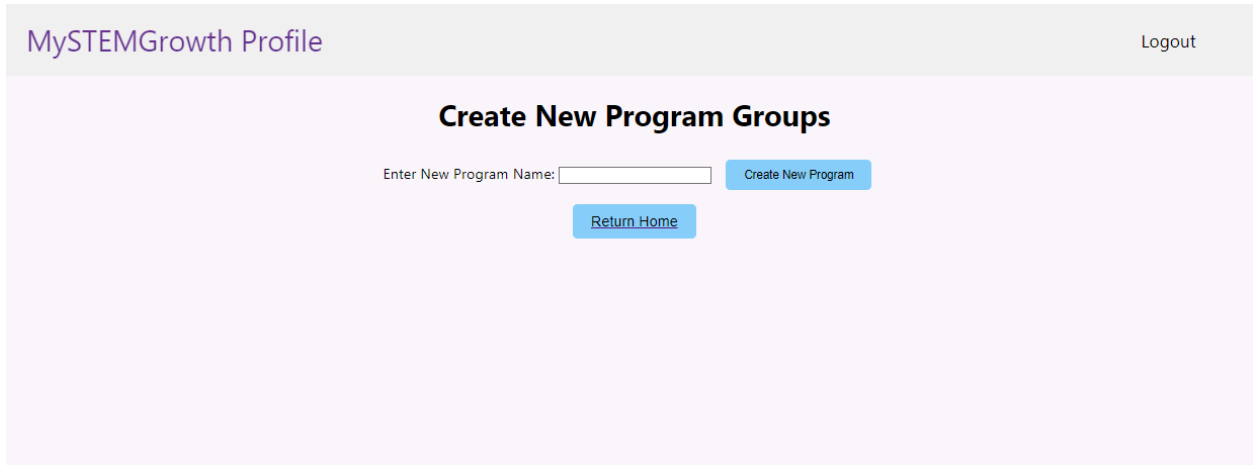


Figure 14 - Admin Create New Program Page

6. EDIT SURVEY

From the admin home page, a user can also edit the survey questions shown to students. After clicking the “Edit A Survey” button, the user will be shown the page in Figure 14. From the dropdown menu, they can select which survey they would like to edit. To edit the default survey shown to all students at this point in time, select the survey named “IINSPIRE Survey Final.” After selecting the survey name, the questions will be visible, and the admin can edit any question and save the changes.

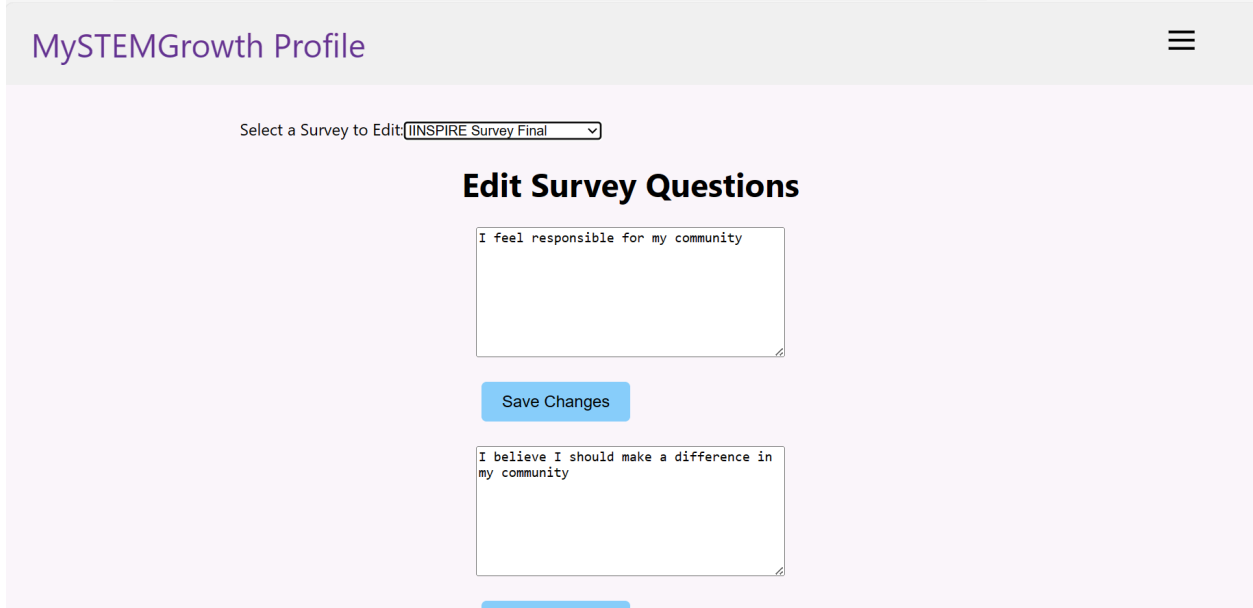


Figure 15 - Admin Edit Survey Page

7. GRANT ADMIN ACCESS TO OTHER USERS

From the admin home page, an admin can also assign administrator privileges to other users. After clicking on the “Assign Admin” button on the admin home page, they will be taken to the screen as shown in Figure 15. From here, they can enter the email address of an existing user that they would like to make an admin. After clicking on the “Assign Admin” button below the textbox, the provided user will now have admin access. The user can return to the home page using the “Return Home” button.

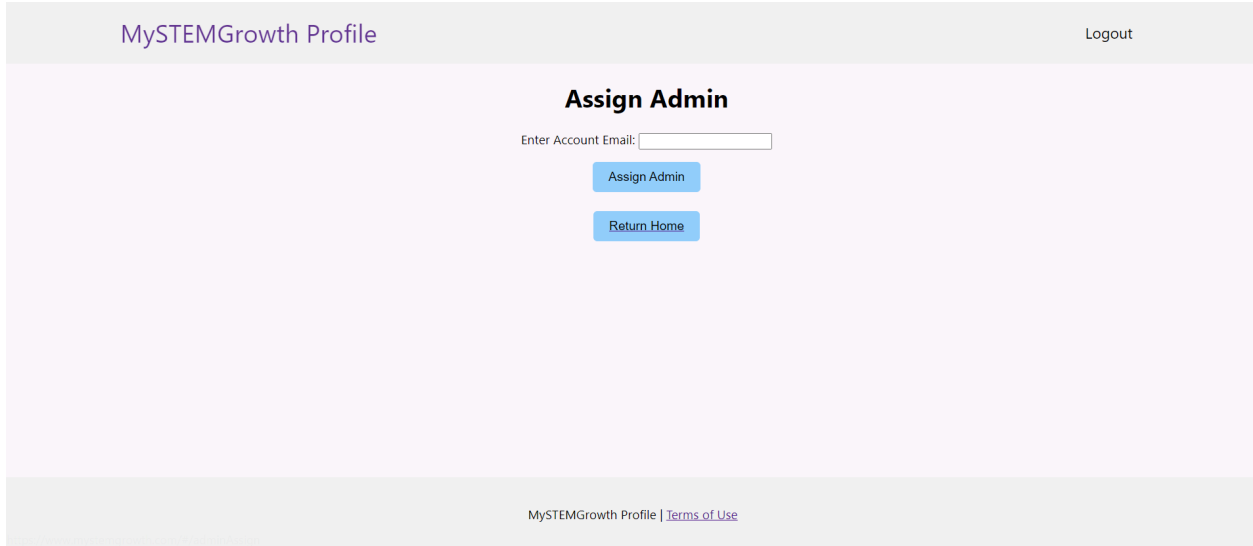


Figure 16 - Admin Access Page

## 8. RESET USER PASSWORD

From the admin home page, an admin can reset a user's password by clicking on the "Reset a Password" button. From there they will be taken to the page shown below in Figure 16. By following the prompts on the screen, the user can reset their password or the password of another user, such as a student who may have forgotten their password.

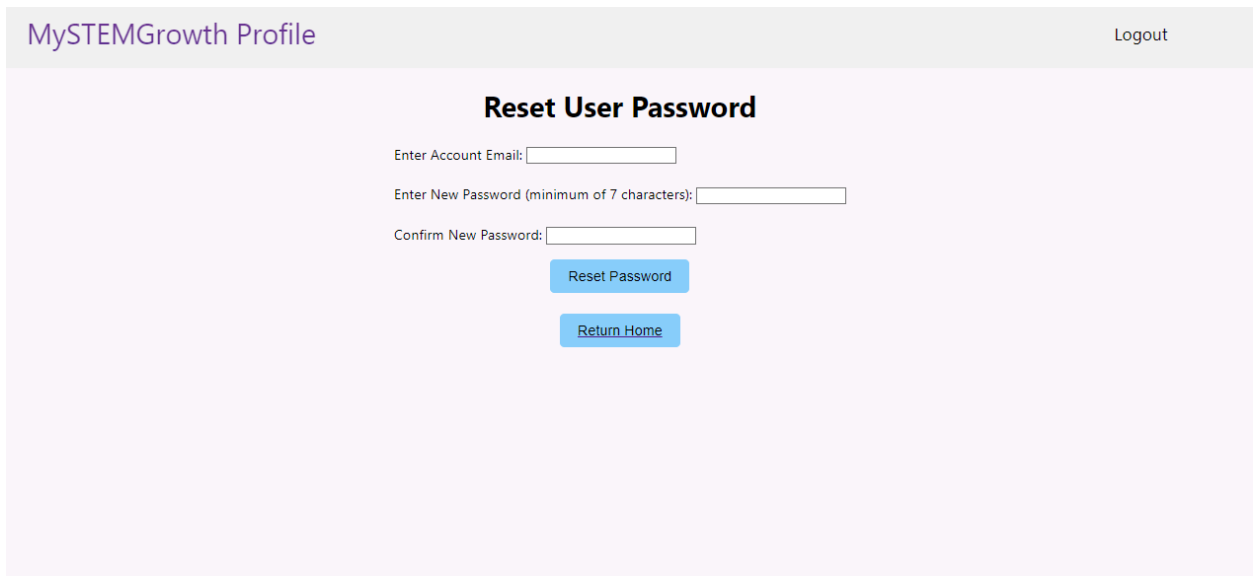


Figure 17 - Reset User Password Page

**NOTE:** This concludes the operation manual for the administrator user. Next we will be going over the student experience.

### 3. ACCESS HOME PAGE

After signing up or logging in, the student will be taken to the home page, as shown below in Figure 18. From here, they can take a survey, view their profile, read the About page, or log out.

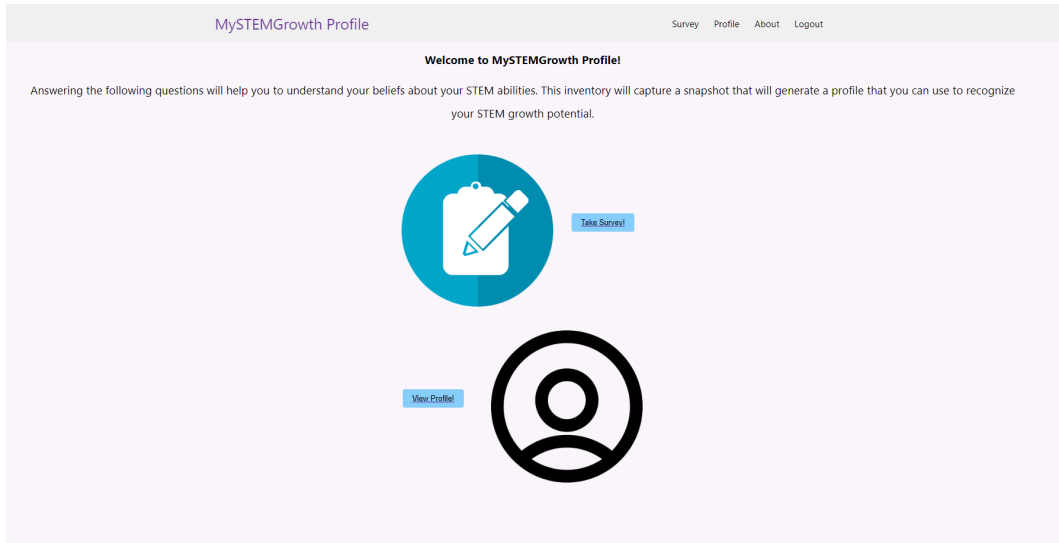


Figure 18 - Student Home Page

### 4. TAKE SURVEY

The student can access the survey page by either clicking the “Take Survey!” button or clicking “Survey” on the navigation bar. They will be taken to the general information page shown in Figure 19. After filling out the requested general information, students can click the “Next” button to be taken to the actual survey questions, as shown in Figure 20. There will be six pages of questions, and students can navigate to the next page by using the blue “Next” button at the bottom of each page as shown in Figure 21. When the student has completed the entire survey, there will be a “Finish Survey” button to click, as shown in Figure 22.

The screenshot shows the 'Survey' page with the heading 'General Questions'. It contains several form fields and a list of radio button options. The first field is a dropdown menu labeled 'Which program are you in? (Please select a program)'. Below it is a text input field for 'What is your age?'. The next section is 'Please indicate your gender (select all that apply)' with radio buttons for 'Woman', 'Man', 'Transgender', 'Non-binary/Non-conforming', and 'Prefer not to respond'. This is followed by 'Please indicate your race/ethnicity (select all that apply)' with radio buttons for 'African American', 'White', 'Hispanic/Latino(a)', 'Asian', 'American Indian/Alaska Native', 'Native Hawaiian/Other Pacific Islander', and 'Prefer not to respond'. The next field is 'How many higher education credits have you taken?'. This is followed by 'What is your STEM area of interest?' and 'Which institution of higher education are you attending (or planning on attending?)'. At the bottom left, there is a blue 'Next' button.

Figure 19 - Survey General Information Page

MySTEMGrowth Profile ☰

### Survey

Please indicate the level to which you agree or disagree with each statement

I feel responsible for my community	Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Agree
I believe I should make a difference in my community	Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Agree
I believe that I have a responsibility to help the poor and hungry	Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Agree
I am committed to serve in my community	Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Agree

Figure 20 - First Survey Question Page

MySTEMGrowth Profile ☰

Solving practical math problems	Dislike	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Like
Reading articles or books about engineering issues	Dislike	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Like
Solving computer software problems	Dislike	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Like
Working on a project involving engineering principles	Dislike	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Like
Solving complicated technical problems	Dislike	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Like
Learning new computer applications	Dislike	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Like

[Next](#)

Figure 21 - Bottom of a Survey Questions Page

MySTEMGrowth Profile ☰

Defend results to a critical audience	0	10	20	30	40	50	60	70	80	90	100
Identify implications for future research	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Identify and report limitations of study	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Identify and report limitations of study	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Synthesize results with regard to current literature	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Organize manuscript according to appropriate professional format and standards	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

[Finish Survey](#)

Figure 22 - Final Survey Page

## 5. VIEW RESULTS

After clicking the “Finish Survey” button, the student will be taken to the profile page, as shown in Figures 23 and 24. This page can also be accessed from the home page by clicking on the “View Profile!” button or from any page by clicking on “Profile” on the navigation bar. On this page, the student can view the results of their surveys. Using the first ‘Download’ button on the right, they can download their results to a pdf file. Using the “Data:” drop-down menu, if the student has taken multiple surveys, they can select which one they would like to see results organized by the date on which the survey was taken. Using the “Compare to:” drop-down menu, the student can choose which of their other survey responses they would like to compare the shown results to. This is also organized by date. As shown in Figure 25 below, when the student selects another survey to attempt to compare their results, the “compare to” survey results are shown in grey. By clicking on the “View as bar chart” button, the student can see their results in bar chart format instead of the default bubble chart, as shown in Figure 26. From there, the student can switch back to a bubble chart view using the button. Further, down the page, the student can use the “Show More” and “Show Less” buttons on the text boxes to read more/less about the textual analysis of their results, as shown in Figure 27.

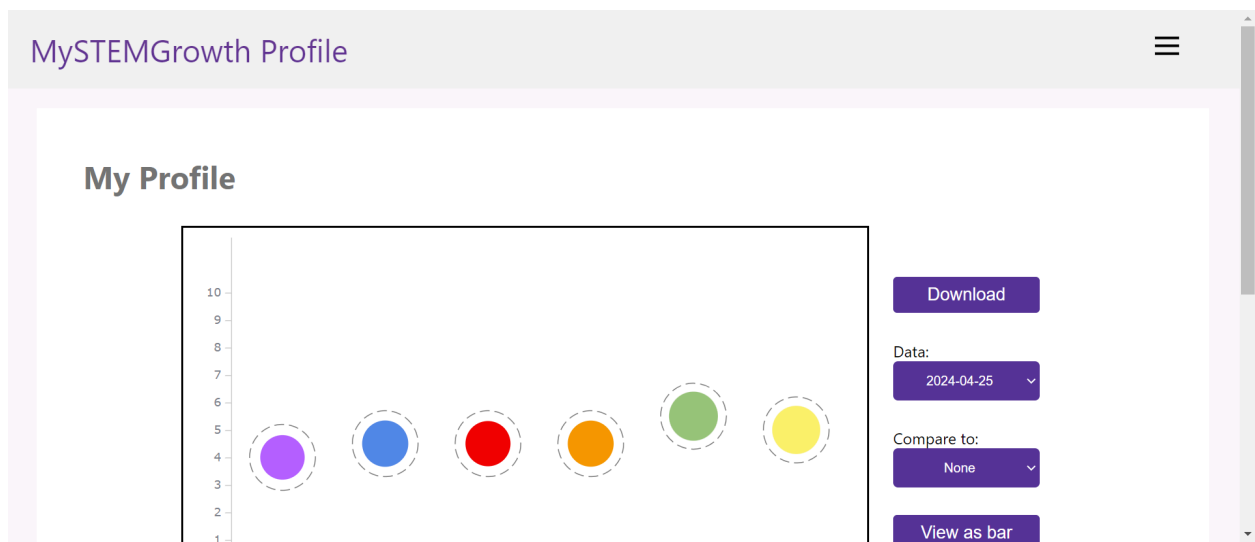


Figure 23 - Top of Results Page

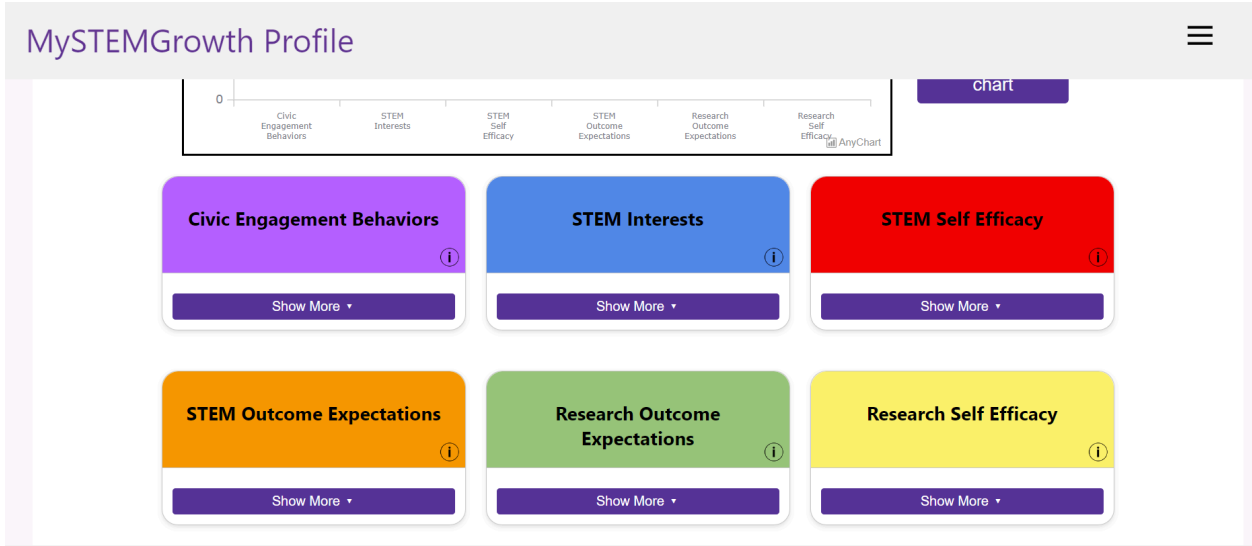


Figure 24 - Bottom of Results Page

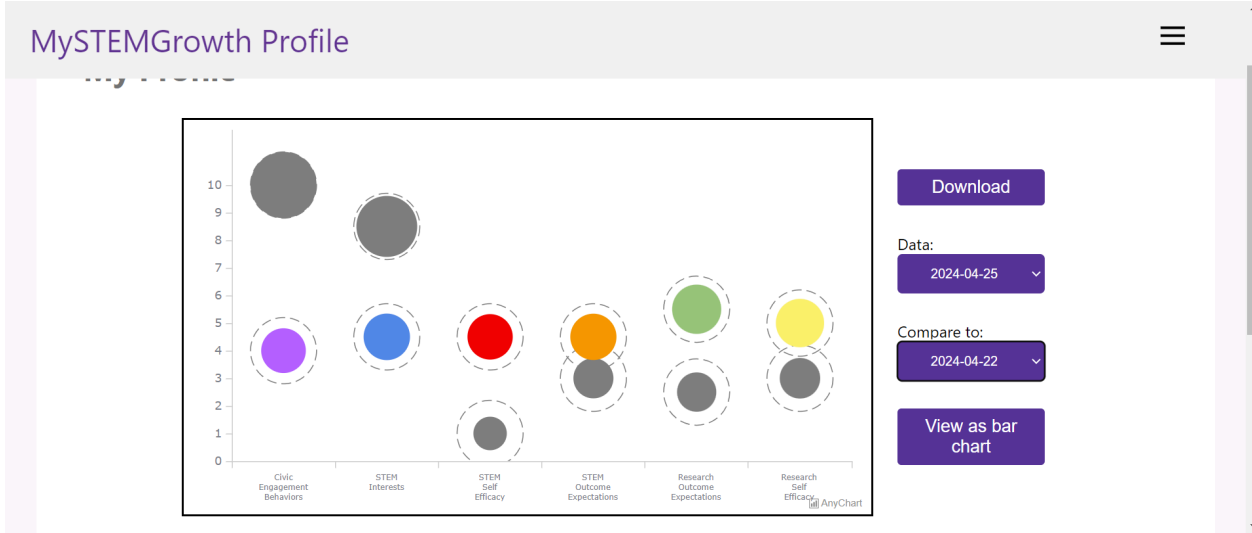


Figure 25 - Results Page with Comparisons between Two Survey Attempts

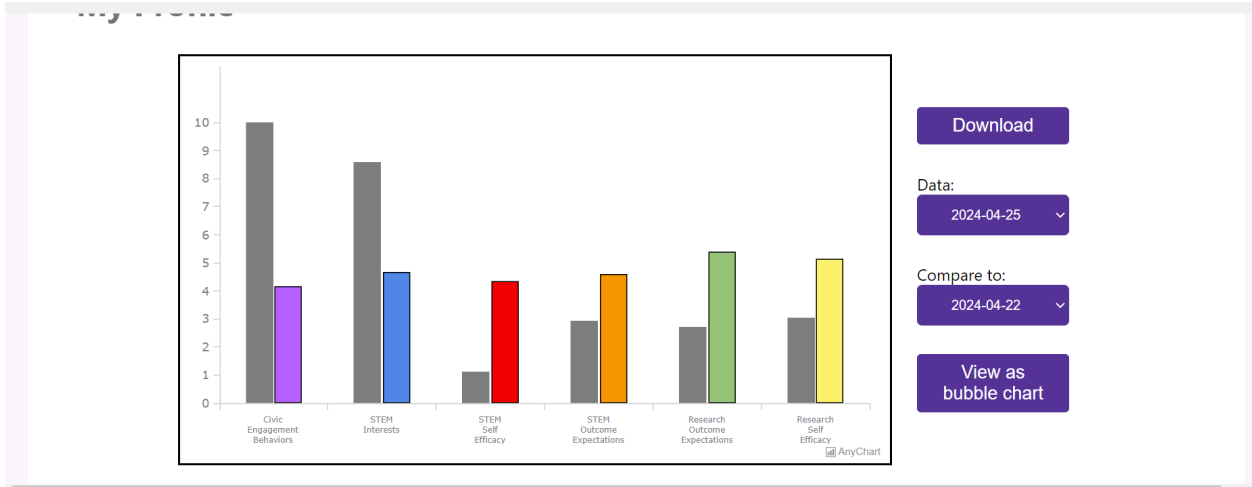


Figure 26 - Results Page with Comparisons between Two Survey Attempts in Bar Chart Form

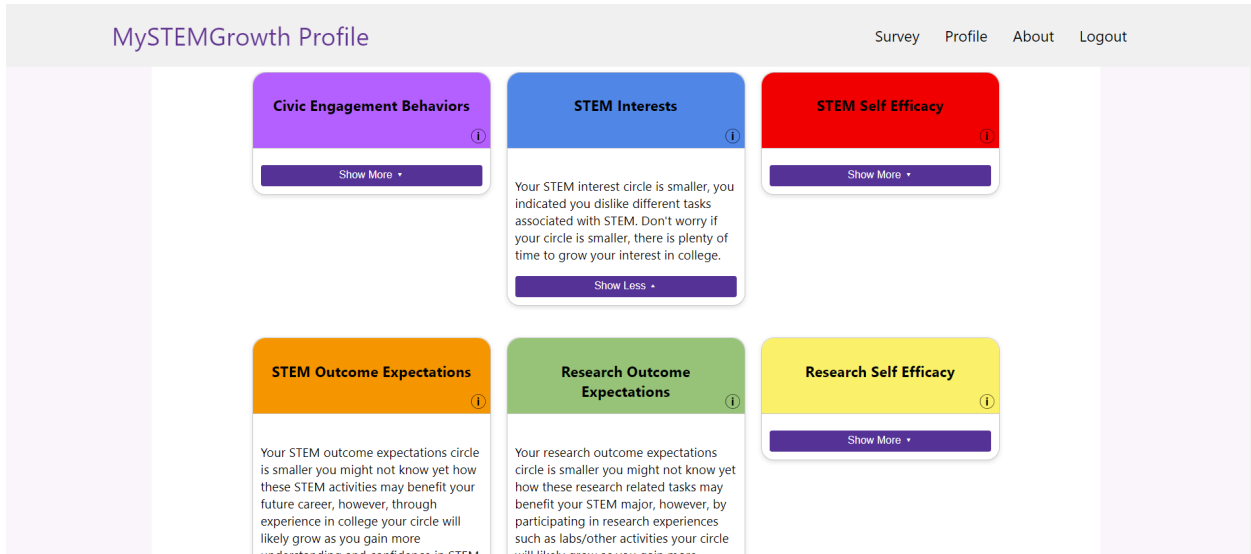


Figure 27 - Results Page with Textual Analysis

front-end instructions:

1. Navigate to the front-end folder of the project
  - a. Once there build the app by running `npm i` and wait for that to complete.
  - b. Run `npm start` and run the project on port 3001 for testing purposes.
  - c. Run `npm build` to get a production build that can be deployed
2. Host the application
  - a. Login to the [sdmay24-35@iastate.edu](mailto:sdmay24-35@iastate.edu) AWS account and navigate to the Amplify page.
  - b. Drag and drop the build folder in the front-end folder created by running `npm build` into the `team35frontend` project

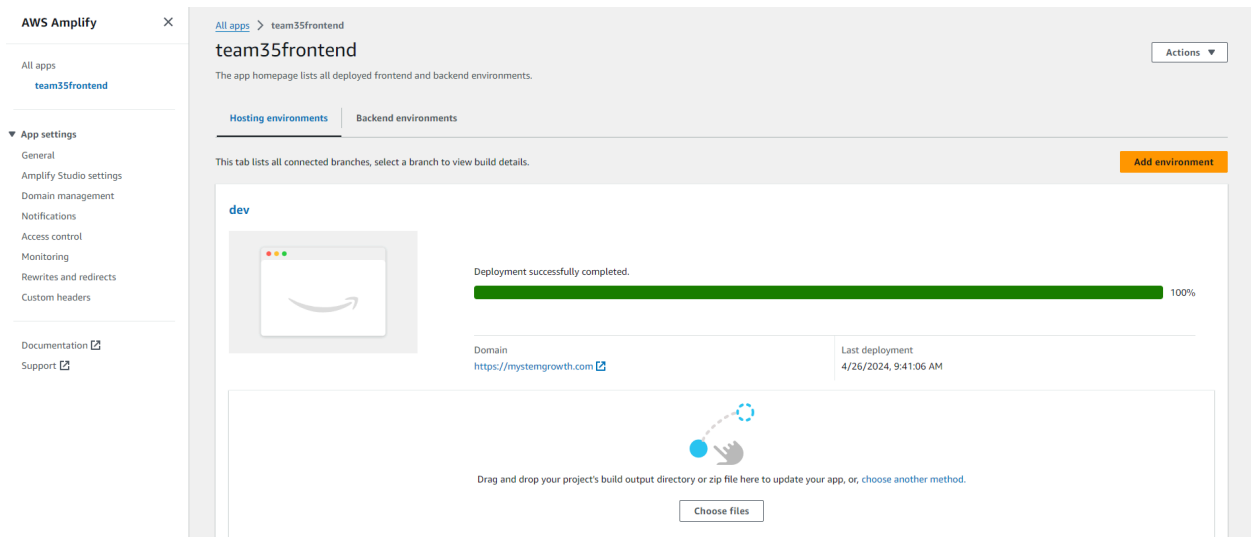


Figure 28 - AWS Amplify



3. Access the application
  - a. Navigate to <https://www.mystemgrowth.com> to access the application

Back-end Instructions:

1. Start the back-end on EC2 Instance:
  - a. Login to the [sdmay24-35@iastate.edu](mailto:sdmay24-35@iastate.edu) AWS account and navigate to the EC2 page.
  - b. Select the 'test-backend' instance and connect to it:

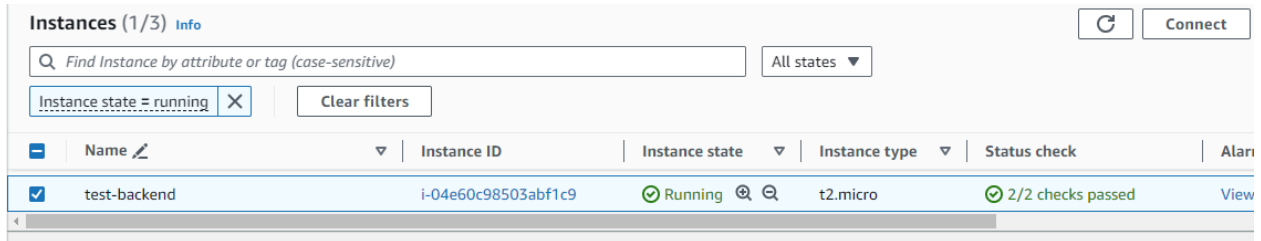


Figure 29 - AWS EC2

- c. Once connected, run the following commands:
  - i. `'cd sdmay24-35/backend'`
  - ii. `'pm2 list'` to check whether the back-end is running.
  - iii. If not running, run `'pm2 start app.js'`. This should start the back-end in the background.
  - iv. From there you can type `'pm2 monit'` to view the back-end's status and the logs of the back-end.

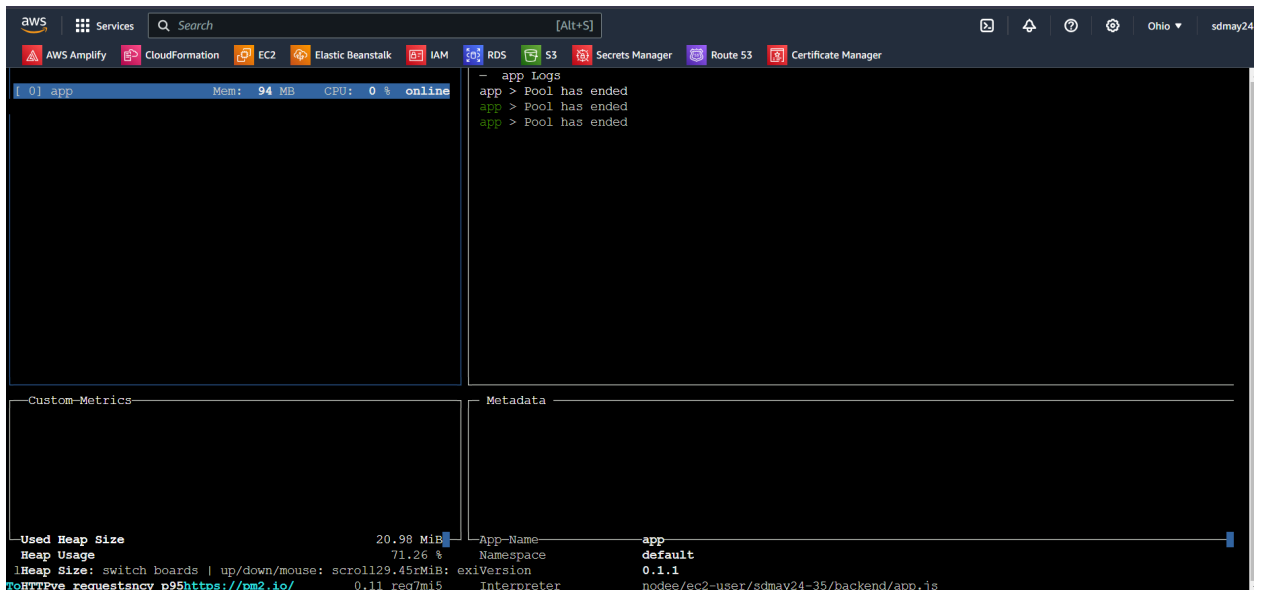


Figure 30 - AWS EC2 Deployment

- d. The back-end is now running. There will periodically be 'Pool has ended' logs, which are the health checks for the load balancer.

2. Navigate to 'target groups' in EC2 and check that the 'test-backend' target group is 'Healthy.' The back-end should be ready once it switches to this state after being turned on.

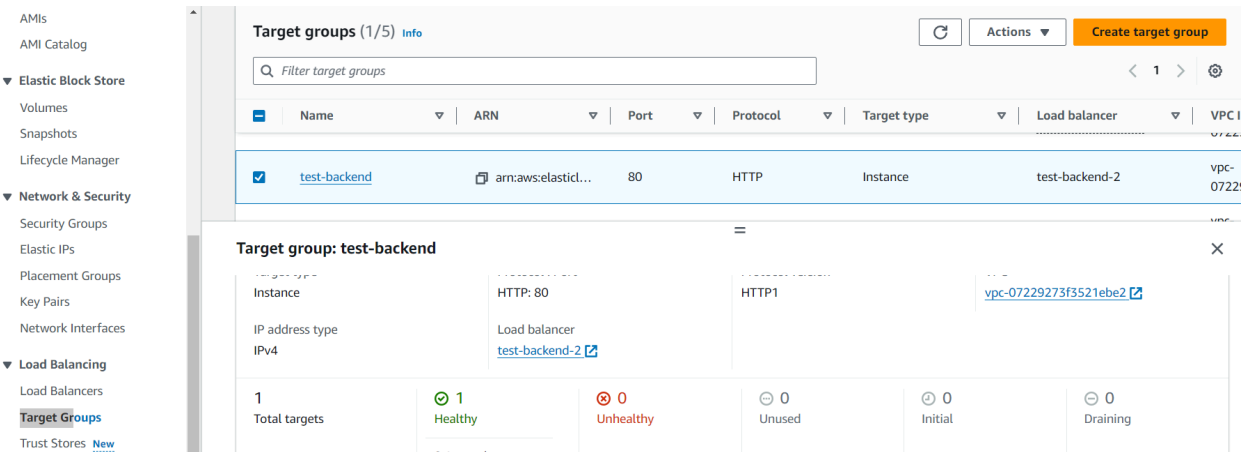


Figure 31 - AWS EC2 Target Groups

3. Now, to test that the back-end is working, you can utilize Postman to test any of the endpoints:

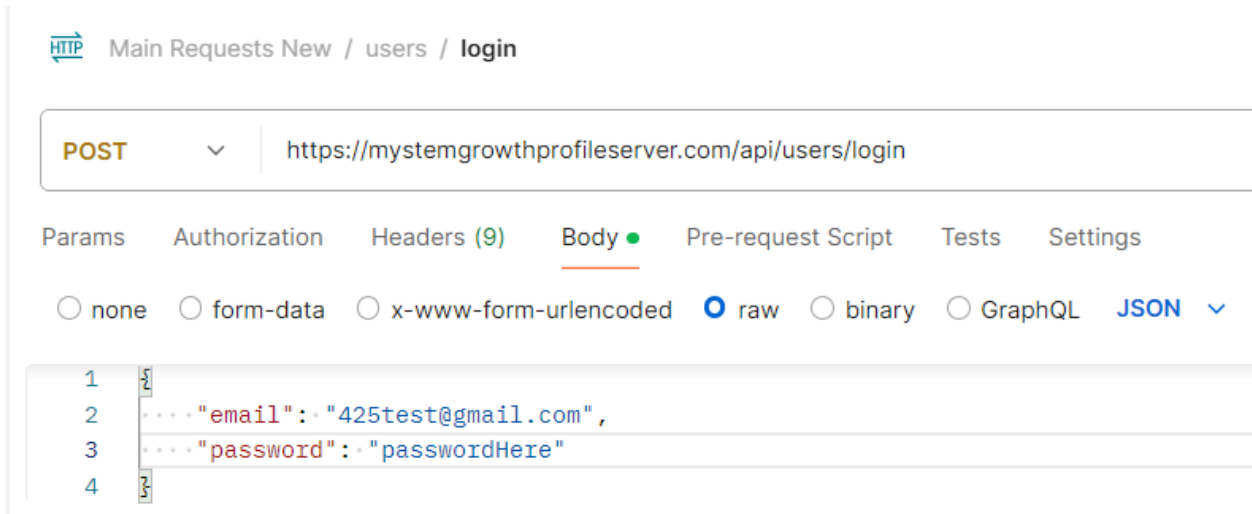


Figure 32 - Postman Connection to EC2 Instance

## Appendix 2- Alternative/Initial Version of Design

Last semester we created a design for the website using Figma. This was the initial prototype we made based on our requirements-gathering process and meetings with the client during the fall semester. An overview of the design can be seen in Figure 33 below. Although this version of our design was a great starting point, the more we learned about our product during implementation and through more frequent client meetings, the more we realized this design would not suit all of our needs. Our initial design only focused on the student user experience and did not include any considerations for the admin-type user. Additionally, the client and our team wanted to go in a different direction with the product's aesthetic.

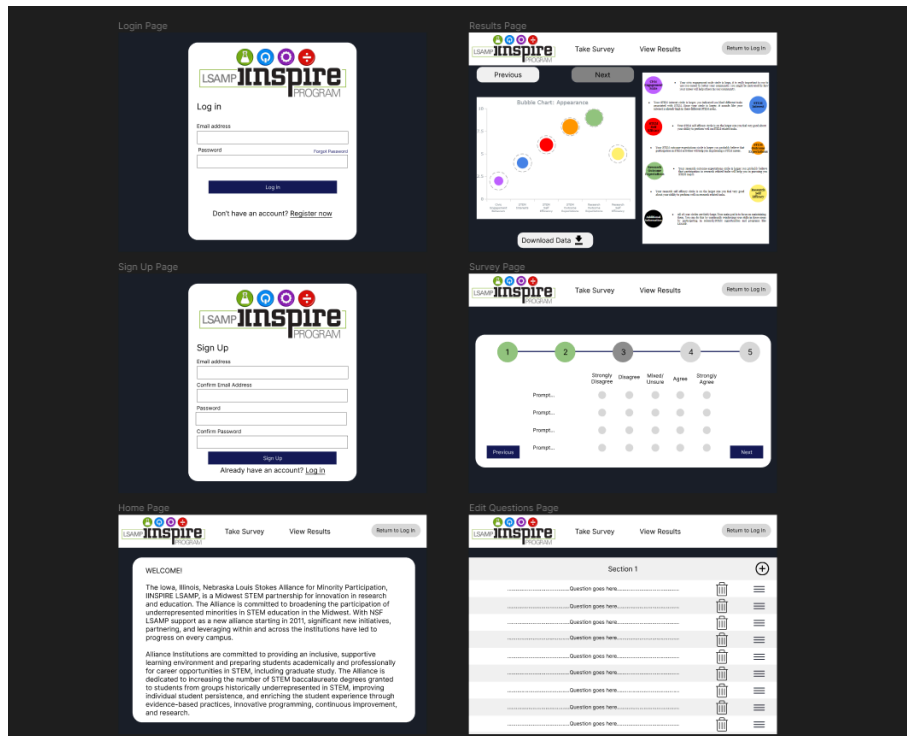


Figure 33 - Original Design Figma Prototype

## Appendix 3 – Code

<https://git.ece.iastate.edu/sd/sdmay24-35>