MySTEMGrowth Survey Tool

Design Document

Sdmay 26-27

Dr.Rover

Dr. Rover, Yiqi Liang

Ethan Buenting: Backend/Cloud

Ethan Van Caster: Developer

Nina Gadelha: Cybersecurity Developer/Testing

Ryan Mamrot: Developer

Sam Craft: Developer/Testing

Caleb Hemmestad: Backend/Cloud

https://sdmay26-27.sd.ece.iastate.edu/

Revised: 09/30/2025

3. Project Plan

3.1. Project Management/Tracking Procedures

Our team has chosen to adopt a mix of both agile and waterfall project management style. In an overall broad sense, we believe that the waterfall approach is the most beneficial. We think identifying all requirements and desires for the project will help give us a clear vision when we start designing/implementing. Ensuring the design/implementation of the project is clearly laid out and agreed upon will ensure testing/maintenance are applied to all specified areas.

However, when working with specific individual issues, we think it is more beneficial to our team and overall project to ensure we are practicing the agile management style. It is important to ensure the defining/planning of an issue is fully understood and agreed upon by our client and team. Once this is the case, we can proceed to the design phase. Once the design is complete, we can then test, and so on. When working with single features, we believe it is important to use this 'cyclical' motion to ensure we meet the desired requirements by our client and ensure we keep the overall flow of the project. Making small progress often for our features can allow the site to always be fully functional and reduces the chances of conflicts and incompatibility with other features/systems.

Our team has been utilizing the 'issue boards' feature available on Git in order to track progress. This feature allows us to create features we intend to add to the current state of the survey. We are able to label/categorize issues to ensure they get implemented in all required parts of the project (frontend, backend, cloud). We are also able to claim issues so we know who is responsible for what features and add timeline goals with dates. We can further organize details by labeling them with specific categories; such as security, UI, database, etc. Overall, the Git issue boards are an easy method we can use to ensure we are making progress in a timely manner and keep everyone organized and accountable.

3.2 Task Decomposition

Tasks are divided into 4 primary categories; Frontend, Backend, Security, and Cloud. Some of the tasks may overlap slightly, but we decide to place them within the list based on their primary area of needed development. The list is as follows:

- Frontend
    - Inform users of missed questions
    - Allow Program Coordinators to see who has completed survey
    - Set slider default to 50% for survey
    - Add change password feature
    - Mobile support
    - Explore using Qualtrics for survey
- Backend
    - Save survey progress for partial survey
    - Add delete user feature
    - Add tokens and cookies
    - Use hashing for login and new user

    - Fix CI/CD pipeline
    - Reduce object size
- Security
    - Change passwords to store as hashed
    - Attempt Breaches
    - Protect Vulnerable Endpoints
- Cloud
    - Switch URL Hosting to Digital Ocean
    - Transfer databases from AWS to Digital Ocean

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Milestone 1: Switch from AWS to Digital Ocean + CI/CD pipeline fix

While this milestone is not easily quantifiable, we do have a very clear definition of success. By the end of this milestone all aspects of the application should run on Digital Ocean; Database, URL Hosting, Frontend, Backend. The CI/CD pipeline will also be updated to ensure that all future features changed or added by the development team will be automatically updated to the new hosting application. This will allow our team to cease all operations on AWS and make an effort to create a more intuitive host for future administrators as well as lower costs.

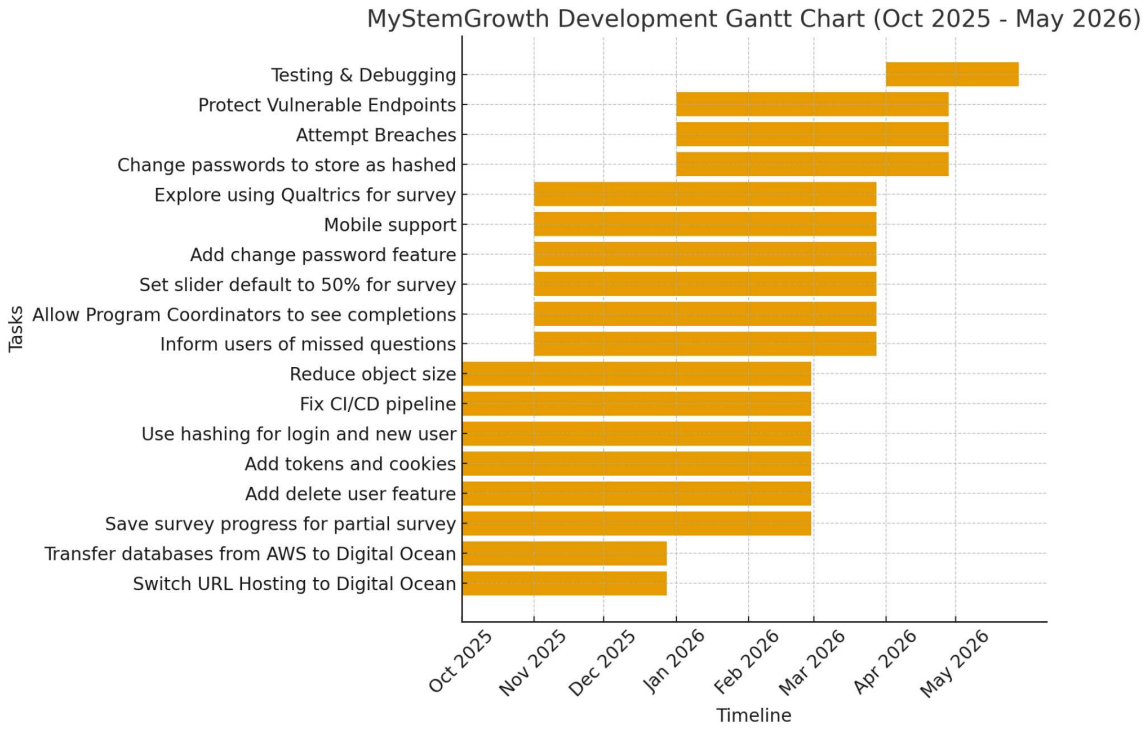
Milestone 2: Improving the Survey

We would look to quantify this milestone through user surveys and feedback from using the application. The goal would be to have over an 80% satisfaction rate with our user base. Throughout this milestone we would like to implement aforementioned features such as defaulting sliders to 50%, saving partial surveys for later completion, and informing users if a question was missed. The overall goal of this milestone would be to improve the user experience while taking the survey. While the current system is functional, it lacks some quality of life features that our team would like to implement.

Milestone 3: Security

Security is an important feature that any modern application should have, especially when said application is asking users questions identifying traits about themselves. The current application lacks most security features, including plaintext passwords and usernames when viewing the console, survey results being easy to find for other users, etc. The result of this milestone will be bringing the application up to expected security standards, ensuring that users information is only available to those who have permissions to access it, such as administrators and program coordinators.

3.4 Project Timeline/Schedule



3.5 Risks and Risk Management/Mitigation

The cloud transition from Amazon Web Services to Digital Ocean holds a few risks in losing user data between database services. This risk can be estimated to be about 25%, but can be easily mitigated by maintaining the original database until we can ensure that all data has been successfully transferred.

Most of the frontend and backend tasks carry virtually no risk. They are all software based features that can be tested on a Git branch to ensure that they work before they are implemented. The biggest risk is that the carry would be unpredictably affecting another feature of the application, but in that case the version of the application can be rolled back quite easily. Due to the little risk this imposes we estimate that the risk factor is about 5%.

The final risk that the group identified was the switch to Qualtrics. As a group we have not committed to the switch to Qualtrics for a survey tool for a couple of reasons. First is that we want to have a smooth user experience, and we worry that using Qualtrics would result in extra windows needing to be opened and interrupt the user's interaction with the application. The second issue is that we are unsure if we will be able to automatically update the database with information from Qualtrics to allow users to view their survey program, or for higher level users to view their subordinates survey information. Due to these factors, we assess the risk of Qualtrics to be 50%. If Qualtrics as a tool turns out to not be a valid approach, we can continue to use our custom implementation of the survey for data collection.

3.6 Personnel Effort Requirements

Task	Average time/week (per person)
Switch Clouds (AWS → Digital Ocean)	3
Improve security	3
Modify survey (questions + format)	3
Improve UI (for all users)	4
Improve storage	2
Add new user features	3

- Switch Clouds
  - Migrate all current data
- Improve security
  - Hash + salt passwords
  - Add tokens/cookies
  - Protect endpoints
  - Add authentication
- Modify Survey
  - Maybe migrate to Qualtrics
  - Fix answer methods
  - Fix questions (duplications/misspelling)
  - If keeping sliders - default at 50%
  - Add save progress button
  - Auto scroll to top
- Improve UI
  - Add modularity (correct format for small screens)
  - Change PC view from cards
  - Allow PCs to view who hasn't submitted survey (in managed groups)
  - Rework formatting for all views (students, PCs and admin)
  - Update About page
  - Add info to Resources page
- Improve storage
  - Delete unused fields
  - "Unsubscribe" from certain cloud options (saves money)
  - Remove abandon accounts
- Account features (all users)
  - Manage account info - (change password, email, name)
  - Delete account
  - Add a "test" mode for PCs and Admins
  - Add extra authentication method for Admin
- Fix Github CI/CD pipeline