

CSCE 606  
Fall 2023

MyMathApps  
Final Report

Nick Robert, Spencer Banasik, Nikhil Nehra,  
Arunim Samudra

## Summary:

The customer needs a website where they can sell math textbooks online. The primary users of this product would be calculus students and teachers. Users can pay for a product that provides all three textbooks for an amount of time determined by the product. Additionally, users can purchase additional products that are provided as a pdf and are a one-time purchase. Payments are sent to the owner of the online bookshop through PayPal, either directly or via a card.

Users may change their name or password, and an instructor system has been implemented. Instructors may create classes, add and remove students, see if they have subscriptions, and finally they may buy and manage subscriptions for their students. We've improved the documentation for setting up the project development environment and we have updated tests across the application which should make developing features easier.

## User Stories:

1. Description of *all* user stories (including revised/refactored stories in the case of legacy projects). For each story, explain how many points you gave it, and explain the implementation status, including those that did not get implemented. Discuss changes to each story as they went. Show lo-fi UI mockups/storyboards you created and then the corresponding screenshots, as needed to explain to stories.
  1. Instructor Class Management Page (80 points)
    - a. As an instructor

I want to create a class if I don't have one already and add/remove students as needed from my class  
So that I can effectively manage who's in my class.
    - b. Implementation status:
      - i. Complete
  2. Manage Transaction History, Subscriptions, and Free Suffixes (80 points)
    - a. As an admin

I want to view the most recent 100 transactions, with an option to view more,  
So that I can understand which textbooks are being bought and which are not being bought.

I want to be able to edit and delete user subscriptions  
So that I can effectively manage subscriptions on the website

I want to be able to define which email addresses with certain suffixes get subscriptions for free  
So that I can give free subscriptions to students from those given schools

- b. Implementation status:
    - i. Complete
- 3. User Profile/Account Changes (70)
  - a. As a user
    - I want to be able to view my profile and be able to change my first name, last name, and/or password
      - So that I can update my profile as needed.
  - b. Implementation status:
    - i. Complete
- 4. User PayPal Credit Card (50 points)
  - a. As a user
    - I want to pay for my products using a credit card
      - So that I can use my preferred payment method
  - b. Implementation status:
    - i. Complete
- 5. Instructor Purchases Textbook for Class (100 points)
  - a. As an instructor
    - I want to purchase textbooks subscriptions for students in my class
      - So that I can ensure that they have access to the class material
  - b. Implementation status:
    - i. Complete
- 6. Admins can manage users (20 points)
  - a. As an admin
    - I want to be able to see and edit user information
      - So that I can ensure everything is working properly on the website
  - b. Implementation status:
    - i. Complete
- 7. Admins can manage purchase codes (20 points)
  - a. As an admin
    - I want to be able to edit, add, and delete purchase codes
      - So that I can give discounts to certain customers when I want to
  - b. Implementation status:
    - i. Complete
- 8. Admins can check products (20 points)

a. As an admin

I want to be able to edit, add, and delete the products listed

So that I can effectively manage the website's store

b. Implementation status:

- i. Complete

9. View PDF Textbooks (50 points)

a. As a user

I want to be able to view the PDF Maple textbooks whenever I have bought them inside the browser

So that I can conveniently read them without requiring a download

b. Implementation status:

- i. Complete

10. Activation email (20 points)

a. As a user

I want to be able to activate my account from the activation email that was sent

So that I can buy subscriptions to the textbook

b. Implementation status:

- i. Mostly complete, requires contact with the current maintainer to hook it up to his email server

11. Converting Read Book Nav Bar into Drop Down Menu (50 points)

a. As a user

I want to be able to buy a subscription which covers all three calculus textbooks

So that I can view all textbooks through a single subscription

b. Implementation status:

- i. Mostly complete, requires subscription in nav bar to be in sync with the purchase made immediately after payment.

12. Admin can manage Classes (50 points)

a. As an admin

Admins can view a list of all classes and the instructor for each class along with a dropdown list of all Users who are students in each class. Admins cannot add or remove Users to classes.

b. Implementation status:

- i. Complete

13. Request Instructor Access (20 points)

a. As a user

b. Implementation status:

i. Complete

14. Fix and Expand Backend Testing for Email and Auth Fixes (50 points)

a. Complete Backend Testing for Email and Auth Module

b. Implementation Status:

i. Complete

15. Figure out the .env file for deployment (20 points)

a. There is no documentation at the moment for the .env file needed for deployment on Heroku. Figure out the deployment process and document it.

b. Implementation Status:

i. Complete

16. Book Module Fixes and Backend Testing Fixes and Expansion (50 points)

a. Complete backend testing for the module

b. Implementation status:

Complete

## Lofi Mockups:

Lofi mockups for some of the features we implemented-

### User Profile

A hand-drawn Lofi mockup of a user profile form on lined notebook paper. The form is titled "User Info". It includes fields for "First Name" (with a red box), "Last Name" (with a red box), "Password" (with a red box), and "Confirm Password" (with a red box). Below these fields is a large button labeled "Change Information". The entire form is enclosed in a rectangular border.

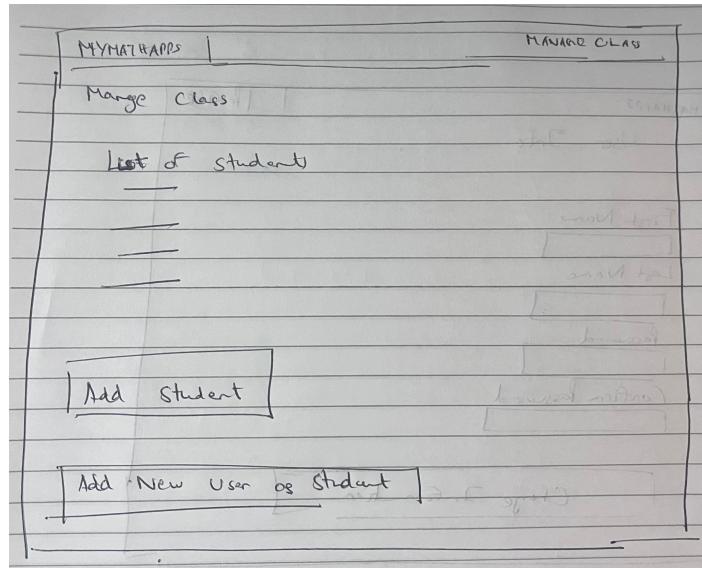
Lofi



A screenshot of a website's "User Information" form. At the top, there is a navigation bar with links for "MYMathApps", "Products", "About Us", "Read Book", "My Subscriptions", "Cart (0)", and "Spencer Banasik". The main form area has a title "User Information" and an email placeholder "Email: softwareengineering@gmail.com". Below it is a note "Edit your profile information here." and four input fields: "First Name" (Spencer), "Last Name" (Banasik), "Password" (\*\*\*\*\*), and "Confirm Password" (\*\*\*\*\*). A green "Change Information" button is at the bottom of the form.

Website

## Instructor- Class Management



Lofi

First Name	Last Name	Email	Has Active Subscription	You Own Their Subscription	Action
stu	test2	stutest2@test.com	FALSE	FALSE	<button>Remove Student</button>
stu	test3	test1@test.com	FALSE	FALSE	<button>Remove Student</button>
stu	test4	test2@test.com	FALSE	FALSE	<button>Remove Student</button>
stu	test5	test3@test.com	FALSE	FALSE	<button>Remove Student</button>

**Add Existing User as Student**  
If your student already has an account, enter their email below and we will add them to your class.

Email: jennith123@gmail.com

**Add New User as Student**  
If your student does not have an account, enter their information below and we will create an account for them and add them to your class automatically.

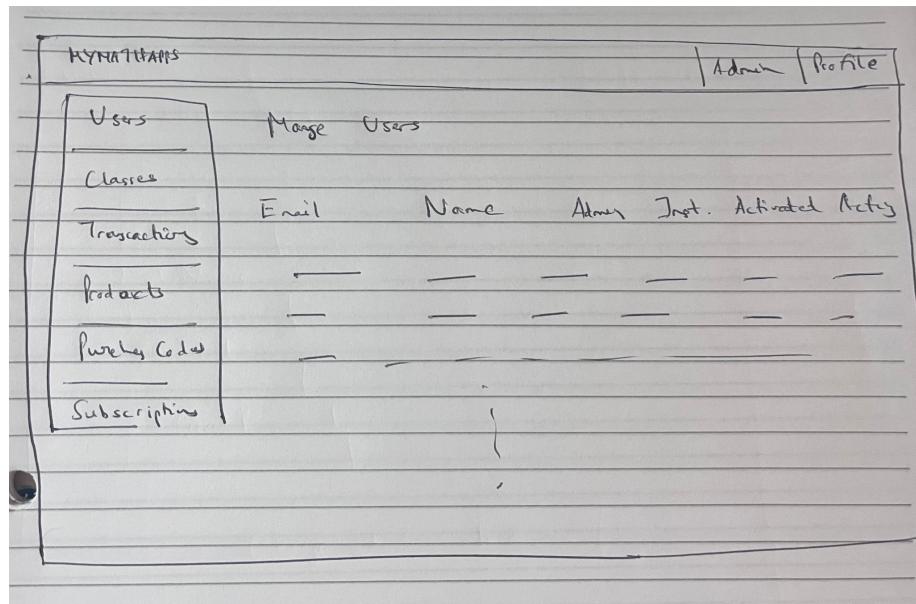
First Name: John

Last Name: Smith

Email: jennith123@gmail.com

Website

## **Admin- Manage Users**



Lofi



## Website

## Legacy:

This was a legacy project which was written with NodeJS and NestJS. There wasn't much documentation about setting up development environments, so it took us a week or two to figure out how to even get it running. During that time we spent time reading through the code to understand its general structure and how Nest and React worked. Once we had it running, we found that not much of the existing code had to be refactored except for the test files. Many of the existing spec and cucumber tests did not work properly, so we had to spend a lot of time fixing them.

## Roles:

Product Owner - Arunim Samudra

Scrum Master - Spencer Banasik

Members- Nicholas Robert, Nikhil Nehra

No changes were made to the roles during the project.

## Scrums:

In iteration 0, we analyzed the legacy code, understood the requirements for the project, and tried to set up the environment.

In iteration 1, we finished setting up our environment and started initial work on our features (User Profile Changes, Payment with Credit Card, Instructor Class Management, and Admin Check Transaction History). Only a little was accomplished in this project since the majority of time was spent on setting up the dev server so only 60 points were completed.

In iteration 2, we finished the admin check transaction history and payment using a credit card. The other two features were also in progress so a total of 190 points were finished.

In iteration 3, we finished Instructor Class Management. Simultaneously, we started working on other admin features like managing users, purchase code and products, viewing pdf textbooks, and activation email. A total of 110 points were completed in this iteration.

In iteration 4, we finished viewing PDF textbooks, activation email, admin features, instructor purchasing textbook and requesting instructor access for class backend which were all worth 190 points.

In iteration 5, we finished the remaining features like admin managing classes, subscriptions, and free subscriptions, converting the read book button into a drop-down, and instructor purchasing textbooks for the class frontend. Along with this, our focus was on completing the remaining backend test cases to increase our test coverage. In all, we completed 200 points in this iteration.

### Points Table:

Total:	Assigned	Completed	% Shared Effort
Nick	180	180	96
Spencer	180	180	96
Arunim	200	200	106.6666667
Nikhil	190	190	101.3333333

### Customer Meetings:

We met with the customer every Wednesday at 7 pm on Zoom (link <https://tamu.zoom.us/j/583606308?pwd=eDdLM2UrOGpKU0RKb0MyNWFsdWhCUT09>).

#### **1. Meeting on Sept 15:**

We had our first meeting with the customer on 15th September. It was just an initial meeting where we met the professor and the developer working on the project, got some idea about the website, and received the SRS document to understand more about the requirements.

#### **2. Meeting on Sept 27:**

We had the next meeting on 27th September. Throughout this iteration, we had problems setting up our development environment since it was a legacy project and there was not enough documentation on how to get started. So, in this meeting, we mainly resolved our doubts and got the dev environment running. After this meeting, we were able to work on our features.

#### **3. Meeting on Oct 4:**

In this meeting, we showed the Admin Check Transaction History in the backend, showed the project deployed on Heroku, and discussed issues with security related to access of admin pages by non-admin users.

#### **4. Meeting on Oct 11:**

We updated the client on our progress on user profile changes, and instructor class management and demoed the Payment using credit card feature. Additionally, we talked with the other developer about some technical considerations.

**5. Meeting on Oct 18:**

We gave a demo of the Protected Routes feature for Admin and other progress that we've made on our user stories. We also got further clarification on the Instructor functionality in this meeting.

**6. Meeting on Oct 25:**

We demoed the initial admin features to the professor and Andrew, the other developer working on the project. We also provided a short update on our progress on other features and both parties asked questions and received answers.

**7. Meeting on Nov 1:**

Demoed the Download PDF button feature and gained requirements for next feature to be implemented. We also had our questions that we had for the client and/or the developer answered.

**8. Meeting on Nov 8:**

Showed the instructor textbook purchase feature and explained some bugs that had been discovered, we also asked the other developer any questions we had.

**9. Meeting on Nov 15:**

Demoed the remaining admin features and the frontend for instructor purchase for student.

**10. Meeting on Nov 22:**

Members showed their features or some bugs that had been discovered, we also asked the other developer any questions we had. This meeting was short since it was the night before Thanksgiving.

### BDD/TDD Process:

A substantial number of tests existed before we took over the project, yet the majority were failing. To ensure the stability and reliability of each feature, our approach involved rectifying these failing tests prior to commencing work on any new functionality. We either addressed the existing tests to bring them to a passing state or introduced new test cases. Only after this process were we able to proceed with the implementation phase.

This approach helped us achieve more code coverage. In the end, we were able to achieve 92% test coverage and 98% test passed. It also helped us in identifying bugs early in the development process which led to fewer bugs in the final product.

### Configuration Management:

We had our own feature branches for each iteration. We would work on our set of features and when it came time for a release we would merge them together, ensure everything worked, and create a release branch. Development, release, and Heroku versions were handled by environment variables stored in a .env file, then handled by NestJS.

### Production Release Issues:

There were a few issues that we stumbled across when releasing to Heroku. The first was the server .env file, which had differing requirements depending on what database we were using. Setting up the database was another issue, though we were able to figure it out. We also had to change some hard-coded paths in the client and server files to match our URL. Finally, passing the .env file to the Heroku deployment was a precarious process that was error-prone and unsafe. It should be a priority to find a better or more automated way to provide the .env file to the remote Heroku repo.

### Tool Issues:

We had a lot of issues with the cucumber tests. Some tests do not work and we're unable to make them work due to time limitations. Additionally, running them has taken fifteen gigabytes of disk space. We attempted to use Ethereal as a dummy email service but were unsuccessful. There were no GitHub or Heroku-specific issues.

### Other Tools:

For our backend, we utilized NodeJS and NestJS. We did backend unit testing with Jest. Our front end used React and we used Cucumber and Selenium for our user testing. Axios was used to handle the HTTP requests, we used the React Semantic UI library for adding specific front-end elements like forms and buttons, BCrypt was used to hash passwords, and finally Nodemailer for handling requests involving the email service.

### Repository:

Our repository encompasses content from the Spring 2023 batch alongside current semester materials. It encapsulates:

- Source Code: All code for the application, including Spring 2023 content and current updates.

- Documentation: Detailed instructions for setting up the development server, deploying the application on Heroku, and executing tests.
- Configuration Files: Files necessary for environment setup and deployment, ensuring a smooth transition between development and deployment environments.

#### Deployment Process:

To fortify the deployment process and avoid missing dependencies, we've made significant enhancements:

- Detailed Documentation: Thorough step-by-step instructions for newcomers to clone the project, run the development server, and deploy the application on Heroku have been meticulously laid out.
- Deployment Steps Clarification: Clear delineation of specific steps required for deployment on Heroku, ensuring a hassle-free deployment process.
- Testing Procedures: Instructions for running Cucumber tests and backend tests have been included, ensuring that the testing suite can be effortlessly executed.

#### Links:

2. Links to your Project Management tool page, public GitHub repo, and Heroku deployment, as appropriate. Make sure these are up-to-date.
  - <https://github.com/orgs/MyMathApp-Fall2023/projects/2/views/1>
  - <https://github.com/MyMathApp-Fall2023/OnlineTextbook>
  - <https://my-math-apps-online-textbook-63eb858df6f6.herokuapp.com/>
3. Links to your presentation and demo video.
  - [https://www.youtube.com/watch?v=4cwD85eeLco&ab\\_channel=NicholasRobert](https://www.youtube.com/watch?v=4cwD85eeLco&ab_channel=NicholasRobert)