

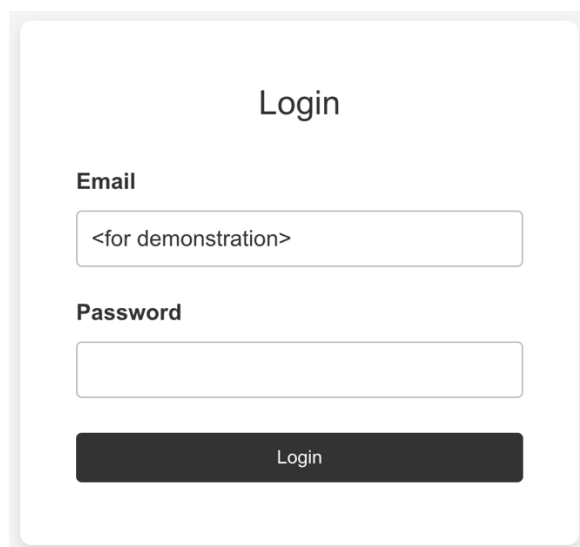
## Current Progress and Mechanisms

After discussing on our term 1 progress, we took advice from different parties such as professors, secondary school teachers and students. We are excited to announce the changes that could better fit into the problem that we are trying to solve through this report. The new design of the website improves the overall workflow and usability drastically. Details are showcased with detailed explanation in this chapter.

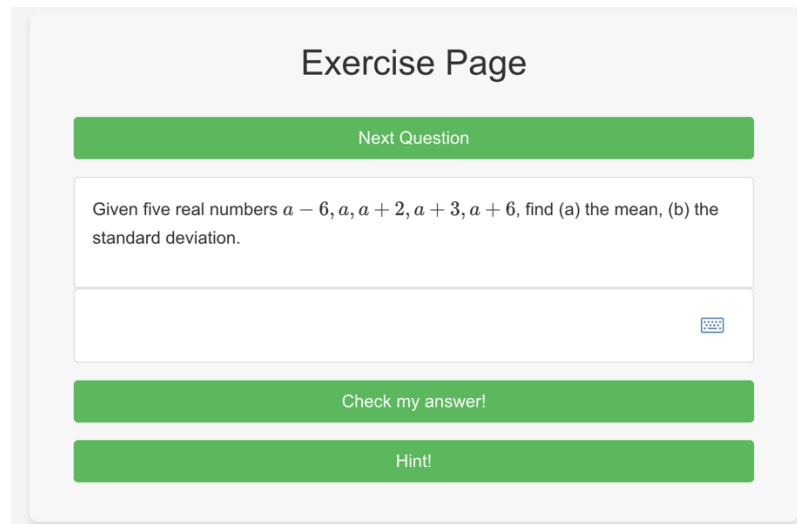
### 1. Website

#### a. Frontend (UI)

We host the website on localhost:3000 for demo purposes. The user interface (UI) has been improved with minimalistic ideas which makes it intuitive and easy to use. Compared to term 1, we switched from Material UI (MUI) library to native CSS. For which native CSS is highly customizable in terms of appearance. Furthermore, the holistic style is easier to manage for developers. By that, future development is more feasible using native CSS. Moreover, by considering the needs for different pages it would be clumsy if we use continue to build on a larger scale. For example,



The image shows a login form with a light gray border and rounded corners. At the top, the word "Login" is centered in a bold, dark gray font. Below it, the label "Email" is positioned to the left of a text input field. The input field contains the placeholder text "<for demonstration>". Below the email field, the label "Password" is positioned to the left of another text input field. At the bottom of the form, there is a dark gray button with the word "Login" in white text.

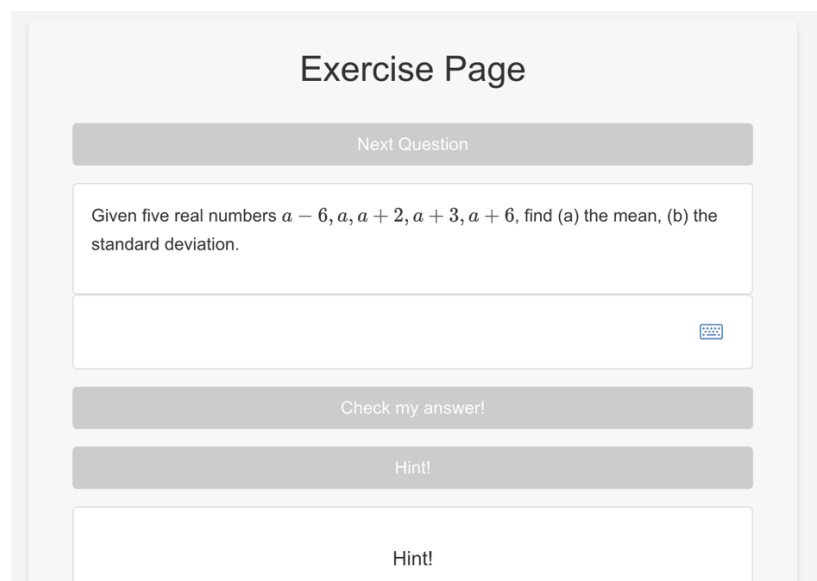


The image shows a web interface titled "Exercise Page". It features a green button labeled "Next Question" at the top. Below it is a text box containing the question: "Given five real numbers  $a - 6$ ,  $a$ ,  $a + 2$ ,  $a + 3$ ,  $a + 6$ , find (a) the mean, (b) the standard deviation." Below the text box is a white input field with a small keyboard icon on the right. At the bottom are two more green buttons labeled "Check my answer!" and "Hint!".

The box size for different text area should be different, it is easier to manage it by page rather than by component.

## b. Frontend (Interaction)

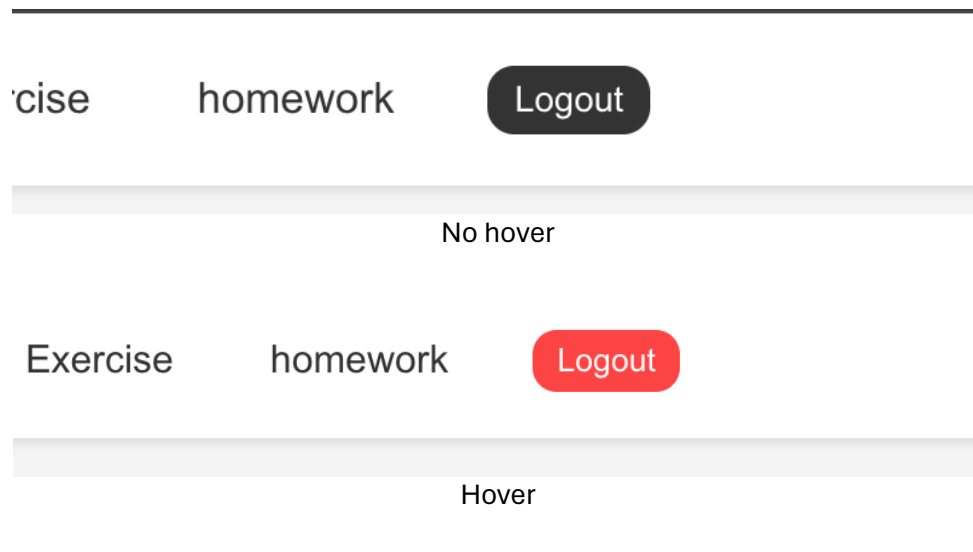
In this semester, more animations and interactions are added into the website. By utilizing React.js library, we can manage interactions using States. For example,



The image shows the same "Exercise Page" UI as above, but with a different styling. The buttons "Next Question", "Check my answer!", and "Hint!" are now grey. The text box and input field remain white. The "Hint!" button is now a separate white box at the bottom of the page.

The buttons will be disabled (grey in the above diagram) while some other component (get hint) is interacting with the user to prevent spamming, accident or unwanted behaviors produced by the user.

To prevent misclicking, the logout key is a special case which will turn red when hover.

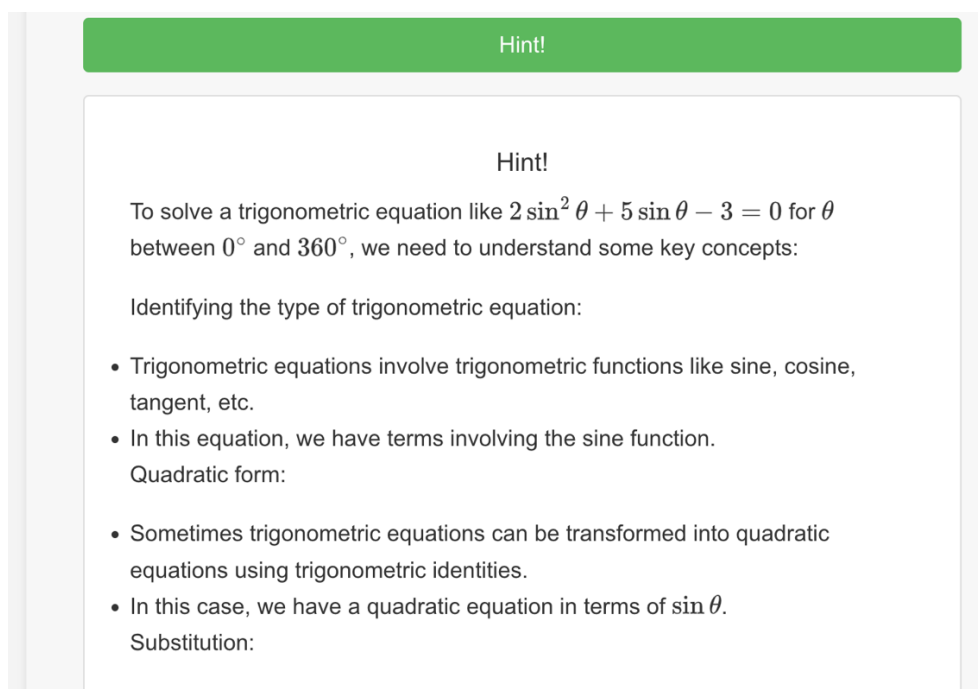
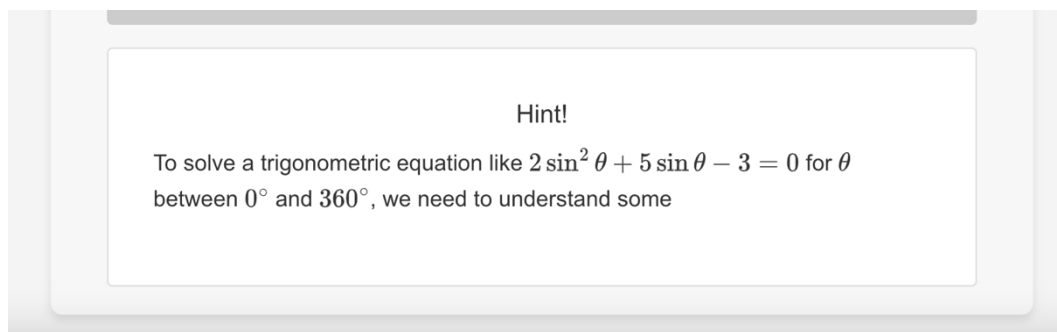


When some error or invalid input cannot be prevented by animation, such as invalid email format or wrong password, we often need to do validation and provide instant response to the user. For example,

A 'Create Account' form is shown with two input fields. The 'Username' field has a red error message 'Username is required' below it. The 'Email' field has a red error message 'Email is required' below it. The form is enclosed in a light grey border.

When the field is required for creating account, a red, sharp warning will be displayed on the screen for the invalid input made by users to prevent Garbage-In-Garbage-Out as those information is essential parts like user verification.

Another big improvement we introduced to the website is the way of the generative AI response. This semester, we added generative AI technology in our website. It is common for user to feel a sense of using generative AI tools if they see the words pop up on the screen sequentially. For the model we used, ChatGPT 3.5 by OpenAI support streaming data on the model output, so I picked streaming data response over non-stream response. For example,



### c. Website Structure and Authentication

Since we use the React.js library, there is a recommended way to handle the navigation over the webpages.



In the implementation, I used `<Route />` component provided by the React.js to handle browsing and navigating. The whole `<Router />` component is surrounded by a `<AuthProvider />` component to provide authentication information when necessary. A `<RequireAuth />` component will wrap the element of the page if the pages are intended to show it only to users that are logged-in.

For the protected route, the website first check the JSON Web Tokens (jwt) stored in the localStorage to verify the user is authenticated before letting the user to access the page. If it is invalid or expired , the website will redirect the user to the login page.

## d. Components

There are several components that builds up the website. It can be separated into two groups, one of them is “before login” group and the other one is “after login” group. Where the Navbar is a global component no matter the login status of the user. The different Navbar appearance will be discussed at each group.

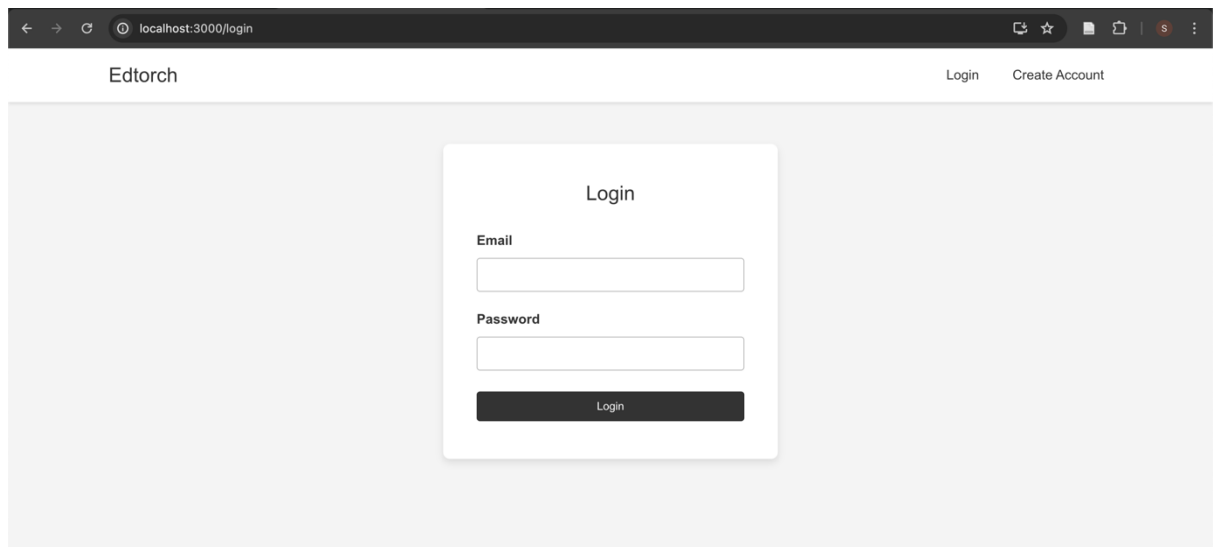
### Before login:

#### Navbar



We can see that it only has 2 buttons before login, Login and Create Account to prevent unnecessary visit to some of the pages (user will be redirected to /login if they visit by URL)

#### Login



The login component takes Email and password field as input, we do client-side validation on data. Including presence check and format check. We also do server-side password validation by sending the data to backend.

The image displays three sequential login form mockups, each enclosed in a light gray border. The first mockup, titled 'Login', shows empty input fields for 'Email' and 'Password', with red error messages 'Email is required' and 'Password is required' below them. The second mockup shows the 'Email' field filled with 'hello kitty' and a red error message 'Email is invalid' below it. The third mockup shows the 'Email' field filled with 'mark123@gmail.com' and the 'Password' field filled with four dots, with a red error message 'Invalid email or password. Please try again.' below the password field.

**Login**

Email

**Email is required**

Password

**Password is required**

Presence check

**Email**

hello kitty

**Email is invalid**

Format check

**Email**

mark123@gmail.com

**Password**

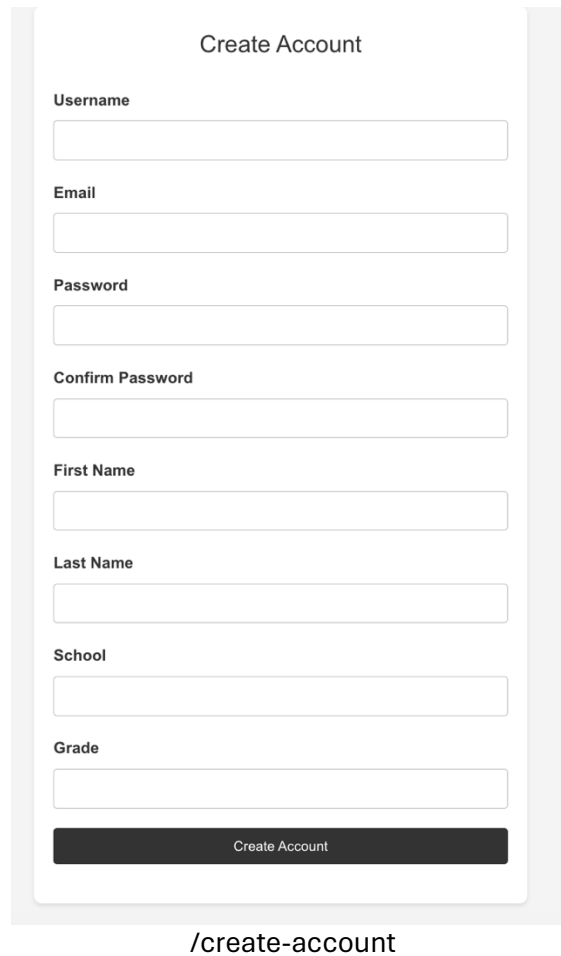
....

**Invalid email or password. Please try again.**

Password and email verification

If the login is successful, the user will be redirected to designated page.

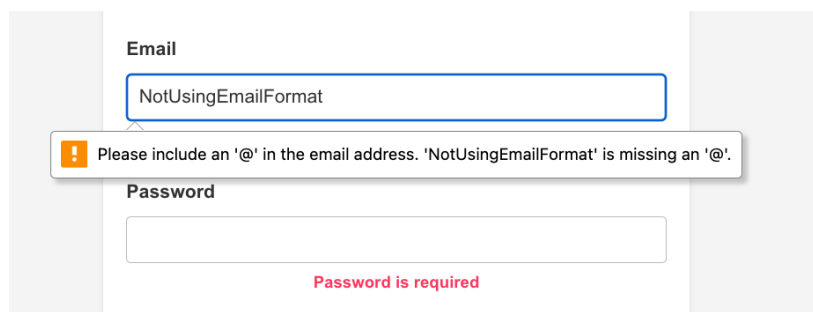
## Create Account



The image shows a 'Create Account' form with the following fields: Username, Email, Password, Confirm Password, First Name, Last Name, School, and Grade. Each field has a corresponding input box. At the bottom of the form is a dark button labeled 'Create Account'. The form is titled 'Create Account' at the top.

/create-account

This component takes the necessary information to create a student account. Same as the login page, it will do client-side validation such as presence check, datatype check, format check. For example,



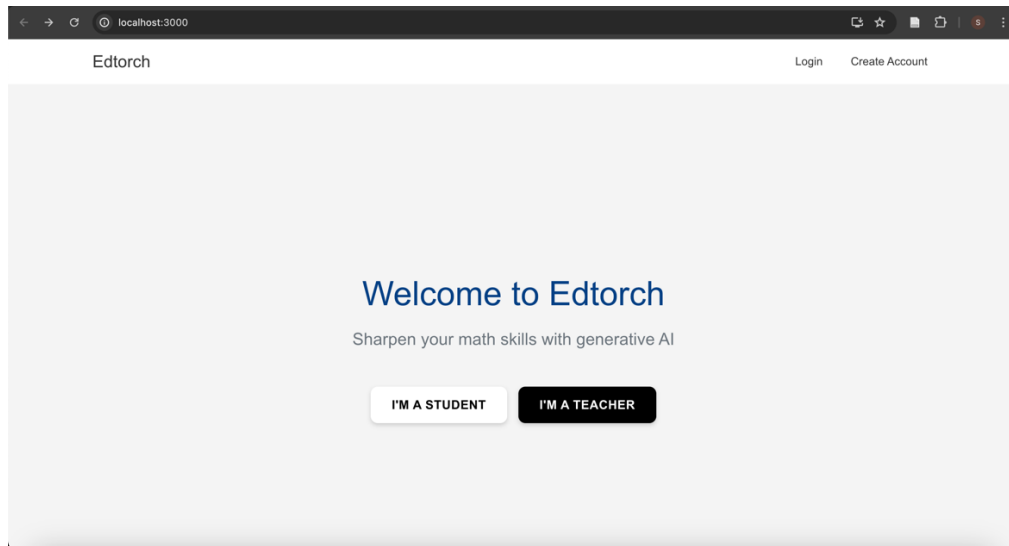
The image shows a validation error on the 'Email' field. The input box contains the text 'NotUsingEmailFormat'. A tooltip message appears below the field, stating: 'Please include an '@' in the email address. 'NotUsingEmailFormat' is missing an '@'.'. Below the email field is the 'Password' field, which is currently empty. A red error message 'Password is required' is displayed below the password field.

If successful, the user will be created and redirected to /login page.



## Home Page

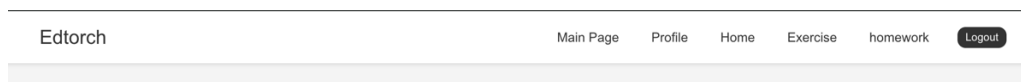
This is the home page for the user as it introduces the website, let user to login or create account.



(localhost:3000/)

**After login:**

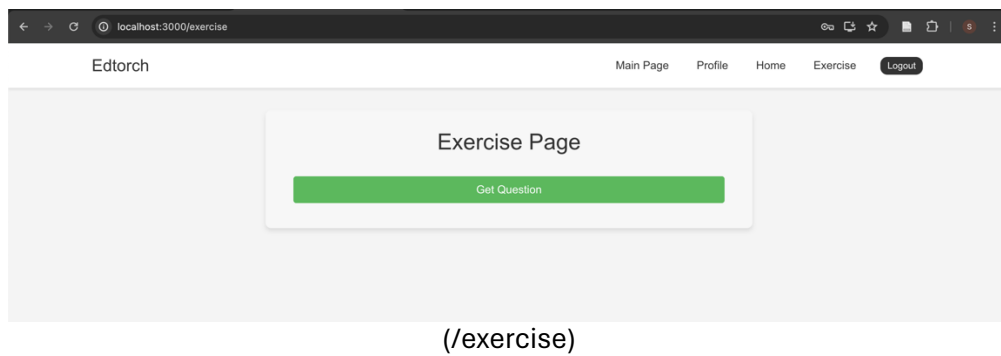
## Navbar



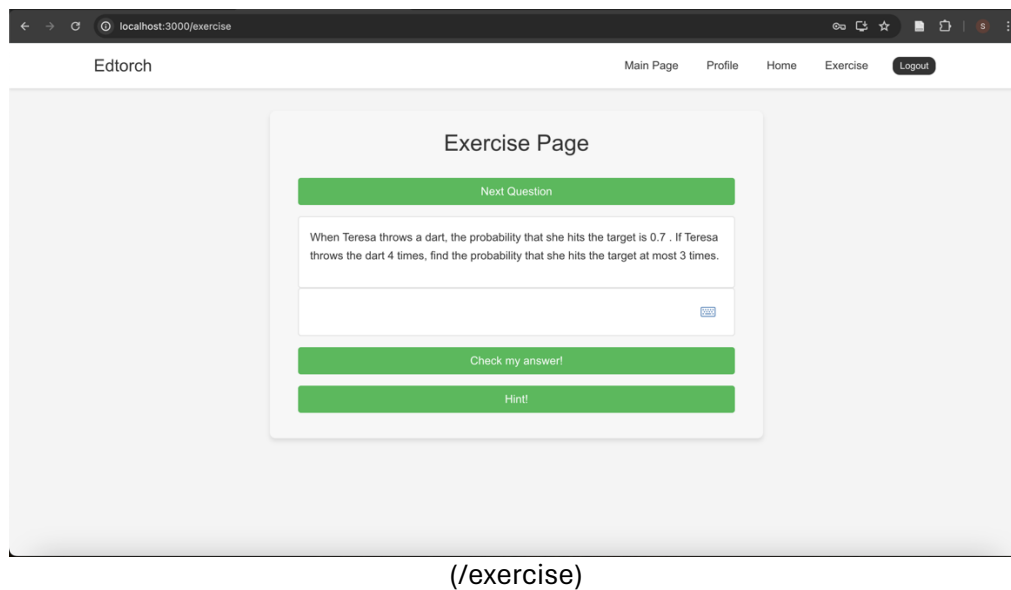
After log-in, there are more options on the Navbar, for which user can tap on them and jump to some other pages. Also, a logout button is added. Once the logout button is pressed, the jwt in localStorage will be discarded and the user will once again be redirected to the login page.

## Exercise

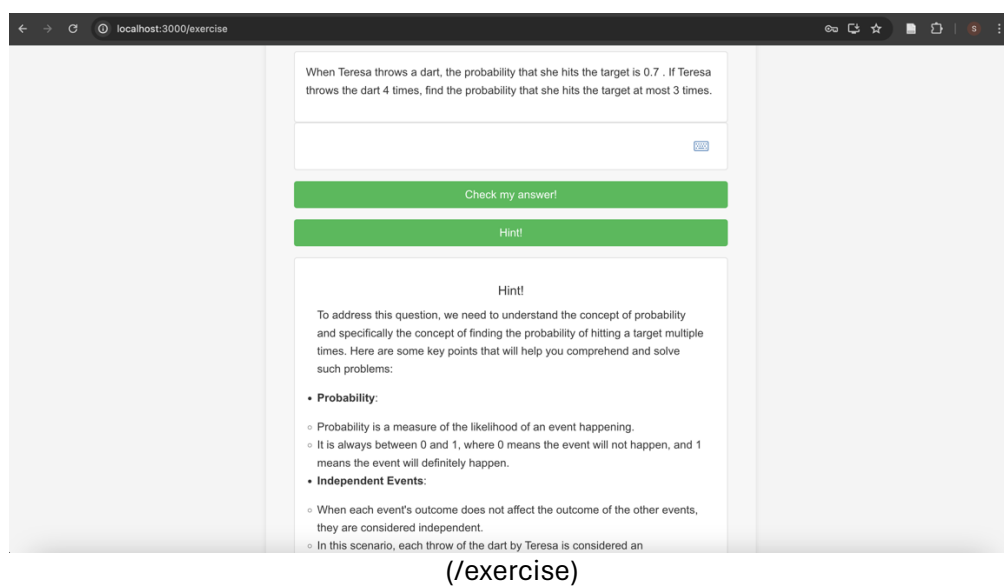
The student will then be redirected to a page where they can get the questions



After clicking the “Get Question” button, a question will be shown.

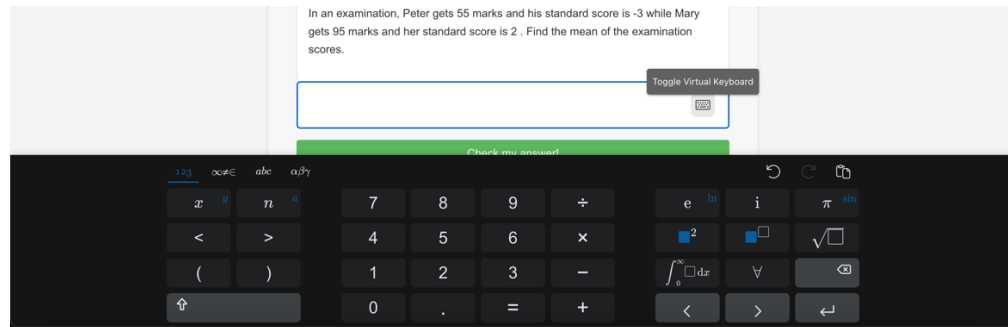


Student could use “Hint!” button to generate some hint to help them.

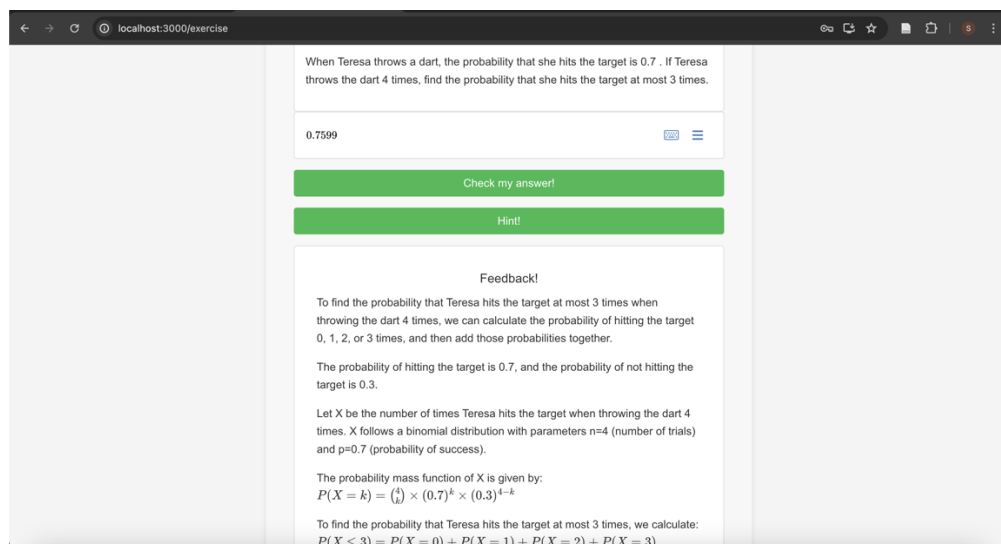


After they attempted the question, they can submit the question and check the answer.

Here, we are using a math virtual keyboard by Mathpix library which provides a decent amount of keys that satisfy the needs for students to input math formula for the answer,



(/exercise)



(/exercise)

After the student press submitted, they can press “Check my answer!”. This will call the backend to communicate with chatGPT API to check if they got the correct answer suggested by chatGPT 3.5 model. Detail will be discussed in the backend part.

## Exercise Page

Next Question

Simplify  $\frac{xy^7}{(x^{-2}y^3)^4}$  and express your answer with positive indices.

$$\frac{x^9}{y^5}$$



Check my answer!

Hint!

### Feedback!

Let's simplify the expression step by step:

1.  $(x^{-2}y^3)^4$  This is equivalent to  $x^{-8}y^{12}$  by applying the power rule for exponents.
2.  $\frac{xy^7}{x^{-8}y^{12}}$  Simplify by dividing the terms:  $\frac{x}{x^{-8}} \cdot \frac{y^7}{y^{12}}$  This simplifies to  $x^{1+8} \cdot y^{7-12} = x^9 \cdot y^{-5}$

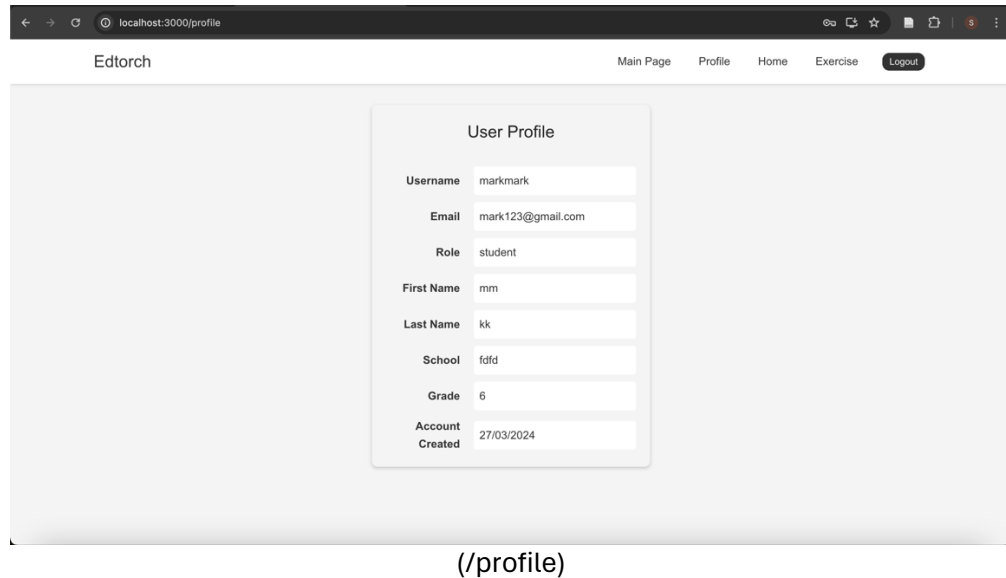
Now, we need to express the answer with positive indices:  $x^9 \cdot y^{-5} = \frac{x^9}{y^5}$

Therefore, the simplified expression is  $\frac{x^9}{y^5}$ , which matches the answer you provided. Well done!

(/exercise)

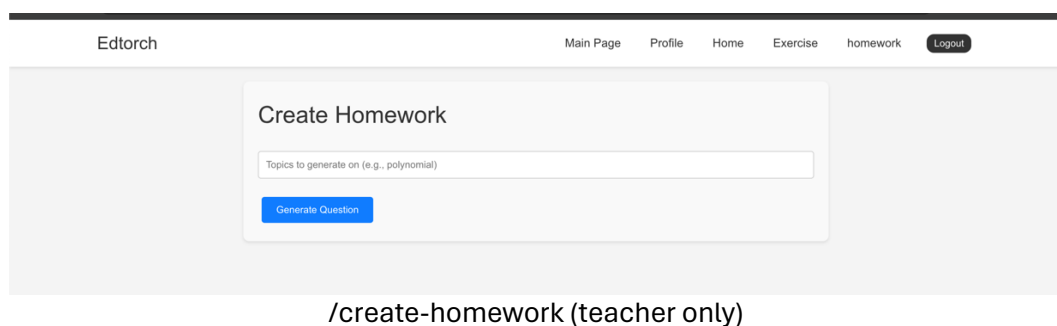
## Profile

A profile page for users, including teachers, to view their user profile.



## Homework

Teachers can generate homework with the help of the generative AI. First the authorization component will check whether or not the user has the role teacher. This component is only authorized for teachers, which means students cannot access this page. If they type the URL, they will be redirected back to the login page.



The teacher can then enter the topic(s) that they want the AI to generate on, for example factorization and the API will respond a question.

**Create Homework**

factorization

**Generated Question**

Expand the following expression:  $(2x - 3)(x + 4)$ .

Expand the following expression:  $(2x - 3)(x + 4)$ .

Add This Question   Regenerate Question

/create-homework (teacher only)

Teachers could modify the question by typing in the text area or regenerate the question if they want to. For report purposes, I added “This is the modified part for the report.” For illustration.

**Generated Question**

Factorize the following quadratic expression:  $3x^2 + 11x + 666$ . This is the modified part for the report.

Factorize the following quadratic expression:  $3x^2 + 11x + 666$ . This is the modified part for the report.

/create-homework (teacher only)

**Create Homework**

factorization

Generate Question

**Questions Added**

Factorize the following quadratic expression:  $3x^2 + 11x + 666$ . This is the modified part for the report.

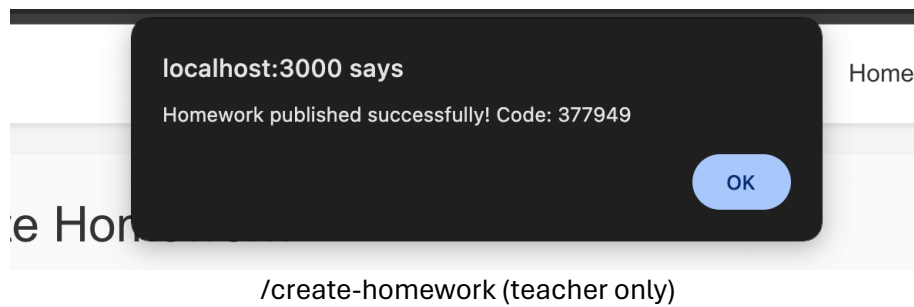
Factorize the following expression:  $x^2 + 5x + 6$ .

Factorize the following expression:  $2x^2 + 8x$ .

Publish Homework

/create-homework (teacher only)

After adding some questions, the teacher can publish the homework and get the code.



### e. Backend

We host the backend API server on localhost port 8000. There are mainly three categories: profile, question and user route which will be used for retrieving profile information, handling question related calls and handling user related calls respectively. Data such as images will be stored on the server as well. When frontend calls the backend API, it mainly returns JSON, a standard return format for API calls.

The code is separated into a models, routers, controllers which is a standard way for Express.js framework project.

We will discuss backend API functionality. More discussion on prompt engineering for the API endpoint that calls ChatGPT in the later section.



**question-router**

There are three API endpoints,

<b>/api/exercise/get-exercise</b>	<ul style="list-style-type: none"><li>• Responds with the question's ID and content if successful according to student score</li></ul>
<b>/api/exercise/check-exercise</b>	<ul style="list-style-type: none"><li>• Retrieves the question id that user want to get hint from the Question document using its ID</li><li>• Constructs a prompt asking ChatGPT API to evaluate a user-provided answer to the question</li></ul>
<b>/api/exercise/get-hint</b>	<ul style="list-style-type: none"><li>• Fetches a question by ID from the database.</li><li>• Sends a prompt to OpenAI requesting a conceptual explanation</li><li>• Streams the chunk response from ChatGPT API to the client</li></ul>

**homework-router**

There are two API endpoints,

<b>/api/homework/create-homework-question</b>	<ul style="list-style-type: none"><li>• Constructs a prompt to generate a single mathematical question on a specified topic in LATEX format.</li><li>• Streams the chunk response from ChatGPT API to the client</li></ul>
<b>/api/homework/publish</b>	<ul style="list-style-type: none"><li>• Receives multiple questions from the client</li><li>• Generates a unique code for the new homework assignment</li><li>• Creates a new homework</li><li>• Iterates over the provided questions, saving each to the HomeworkQuestion database</li></ul>

**user-router**

There are two API endpoints,

<b>/api/user/create-user</b>	<ul style="list-style-type: none"><li>• Checks if there contains duplicated email</li><li>• Hashes the user's password</li><li>• Saves the new user's information in the database</li><li>• Handles errors and returns appropriate error messages if something goes wrong (e.g., email already exists, database errors)</li></ul>
<b>/api/user/login</b>	<ul style="list-style-type: none"><li>• Finds a user in the database by email</li><li>• compare the provided password with the hashed password stored in the database</li><li>• If the password is correct, returns a JWT with the user's details and an expiration time</li><li>• Handles errors, such as incorrect email or password and server issues, and returns appropriate error messages</li></ul>

**profile-router**

There is only one API endpoints,

<b>/api/profile/get-profile</b>	<ul style="list-style-type: none"><li>• Retrieves the whole profile information of a user using id</li><li>• Handles errors, such as User not found or server error</li></ul>
---------------------------------	---

## 2. Prompt Engineering

Prompt engineering is an effective way to guide large language models to output the answer that better fits the use case. In this project, we have three API calls to ChatGPT for various of tasks, including Concept Explanation for students, Exercise Generation for teachers and Answer Evaluation for student attempts.

Each prompt is designed according to scenario in the backend express server. Together with the request data from frontend, a tailor-made prompt can be achieved according to different users' need.

### I. Concept Explanation Prompt

**Objective:** Guide the user to solve the math problem by their own by teaching underlying concepts that students might forget.

Prompt:

```
const prompt = [
  {
    role: "system",
    content:
      "You are a secondary school math tutor who is good at explaining concepts with simple words. \
      You will be given a question, however, You DO NOT discuss a particular question. \
      Instead, teach the concepts behind that needed to solve the question. \
      For example you want to discuss how to factorization, logarithm works rather than giving solid examples\
      Try to explain it with a simple manner with the help of bullet points. \
      Format all equation in your response using inline LATEX format with default '$'"
  },
  {
    role: "user",
    content: `Here is a math question: ${question} \n Please teach me.`
  }
]
```

There are three rationale that we want the model to follow. First, we instruct the model to focus on explaining the teach and learn rather than giving out answers directly. Second, as students are already really confused on solving problems, we want the model to explain it in a clear way, better in bullet points that will help them solve in a general approach. Third, the format of the output should be consistent otherwise unexpected rendering bug may occur in the frontend. Here's an example:

### Sample Question:

A box contains  $n$  white balls, 5 black balls and 8 red balls. If a ball is randomly drawn from the box, then the probability of drawing a red ball is  $\frac{2}{5}$ . Find the value of  $n$ .

**Sample Output:****Hint!**

To solve this problem, you need to understand the basic concept of probability and how it relates to the total number of outcomes.

Here are the key concepts you need to understand:

1. **Probability Formula:** The probability of an event happening is the number of ways it can happen divided by the total number of outcomes.
  - Probability  $P(A) = \frac{\text{Number of favorable outcomes}}{\text{Total number of outcomes}}$
2. **Total Number of Outcomes:** In this problem, the total number of outcomes is the sum of all different types of balls in the box: white balls, black balls, and red balls.
  - Total number of outcomes =  $n + 5 + 8 = n + 13$
3. **Number of Favorable Outcomes for Drawing a Red Ball:** Since the probability of drawing a red ball is given as  $\frac{2}{5}$ , the number of favorable outcomes for drawing a red ball is 2.
  - Number of favorable outcomes (for red balls) = 8
4. **Putting it Together:** Substitute these values into the probability formula and set it equal to  $\frac{2}{5}$  (as given in the question) to find the value of  $n$ .
  - $\frac{8}{n+13} = \frac{2}{5}$   
By solving this equation, you can find the value of  $n$  which satisfies the given conditions.

**II. Exercise Generation Prompt**

**Objective:** Generate single, topic-specific math question which is in latex format.

Prompt:

```
const prompt = [
  {
    role: "system",
    content: "You are a mathematics teacher in high school who frequently creates exercises for your students. \
Your exercises consist of the questions for topics that the user wants in text only. \
You should generate only one question at a time. \
Please ensure that the math equation inside your question is in inline LATEX format with dollar sign."
  },
  {
    role: "user",
    content: `Generate a question reagarding the topic ${req.body.topic} || "on any topic"`.
  }
]
```

There are three rationale. First the model has to be able to generate a variety of questions. Second, it should only give one question at a time in order to facilitate the rendering on the frontend. Third, in inline Latex format.

**Sample** (Topic would be polynomial):

## Create Homework

**Sample Output:**

### Generated Question

What is the degree of the polynomial  $P(x) = 3x^4 - 2x^2 + 5x - 1$ ?

### III. Answer Evaluation Prompt

**Objective:** Evaluate a user-provided answer to the math question and all math equation are correct in inline Latex format.

Prompt:

```
const prompt = [
  {
    role: "system",
    content:
      "You are a secondary school math teacher who is good at assessing answers \
      You will be given a math question and an attempt \
      Determine whether the attempt is correct. If yes, no explanation needed. \
      Otherwise, give steps and answer on how you would solve the problem.\
      Format all equation in your response using inline LATEX format with dollar sign"
  },
  {
    role: "user",
    content: `Here is a math question: ${question}\n
    Please evaluate my answer: ${req.body.answer}.`
  }
]
```

**Sample Input:**

Which is a wrong answer,

If  $81^x = 27^{2y}$  and  $x, y$  are non-zero integers, then  $x : y =$

4 : 6





**Sample Output:****Feedback!**

To determine whether the answer 4:6 is correct, we need to solve the given equations:

$$81^x = 27^{2y}$$

First, we express both sides with the same base:

$$3^{4x} = 3^{6y}$$

Now, since the bases are the same, we set the exponents equal to each other:

$$4x = 6y$$

Simplifying the equation further:

$$2x = 3y$$



From this simplified form, we can see that the ratio of x to y is 3:2, not 4:6 as provided in the answer. Thus, the correct ratio of x to y is 3:2.

**Sample Input and Output for correct result:**

### Exercise Page

Next Question

The solution of  $15 + 4x < 3$  or  $9 - 2x > 1$  is

$x < 4$   

Check my answer!

Hint!

**Feedback!**

Your answer is correct.

### 3. Database

MongoDB Atlas is used in this project. The main reason for us to migrate it on cloud is because the data can be shared across different regions and can access from anywhere permitted without worrying single hardware failure. Free tier is used throughout the process. The backend calls the database for user operation such as login and question related operations and get a question.

The database schema design is as follows:

```
+-----+
| Homework                               |
|-----|
| + questions: [ObjectId] (ref: HomeworkQuestion) |
| + createdBy: ObjectId (ref: User, required)      |
| + createdAt: Date (default: now)                 |
| + code: String (required, unique)                 |
+-----+
```

```
+-----+
| HomeworkQuestion                       |
|-----|
| + content: String (required)             |
| + homeworkId: ObjectId (ref: Homework, required) |
| + createdAt: Date (default: now)         |
+-----+
```

For homework and homeworkQuestion collection is related because each homework questions are stored independently as single questions and only leaves the id on the array of ObjectId in Homework collection. This will increase the scalability for future work if we want to rate each question from different homework.

```

+-----+
| Question |
+-----+
| + question: String (required, unique) |
+-----+

```

The question database is for students get question for the /exercise mentioned earlier.

```

+-----+
| User |
+-----+
| + username: String (required) |
| + email: String (required, unique) |
| + passwordHash: String (required) |
| + role: String (required, default: 'student') |
| + createdAt: Date (default: now) |
| + firstName: String (required) |
| + lastName: String (required) |
| + school: String (required) |
| + grade: Number (required) |
| + abilityScores: [Number] (conditionally required for students) |
+-----+

```

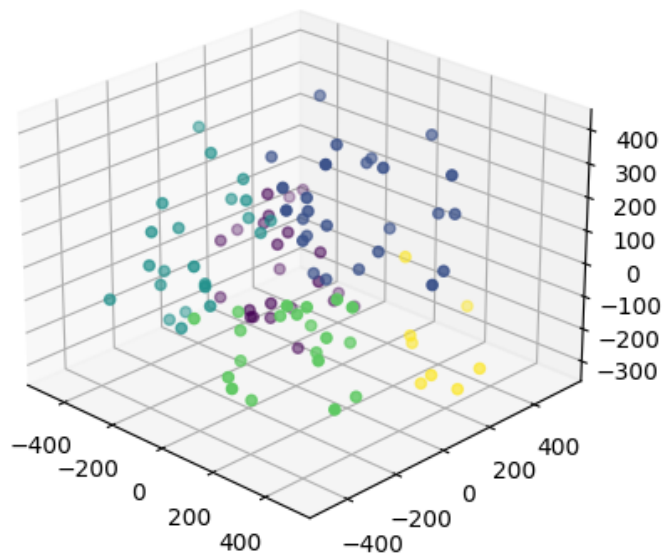
The abilityScores record how well each user (student) do when they attempt the questions in the database, calculate by the algorithm in the next section.

## 4. AI

### a. Question-classification

In semester 1, we created an alpha build of the question-classification artificial intelligence that can take questions text for input and output the question's labels.

The program takes plain text input with math notation mixed in. By regular expressions, we clean out and simplify math notations and numbers before doing word vectorization.



## 5. Data Collection Tool

As we need a lot of questions to populate our database, we decided to create an OCR interface to streamline the process of uploading questions to the database.

We made a tool that crops an image into smaller images since one image may have several questions. Then we would send all the cropped snippets to Mathpix, a software for turning the words in an image into latex format. After that, we store the latex string to the database hosted on MongoDB atlas so that we can access the data from different locations for better division of labor between two groupmates. We also consider the case that we might want to pause the cropping job so that we introduce a process.txt to keep track on the path that the image is processed.

**1.2 HKCEE MA 2007 – I – 10**

- (a) If the length of a piece of thin metal wire is measured as 5 cm correct to the nearest cm, find the least possible length of the metal wire.
- (b) The length of a piece of thin metal wire is measured as 2.0 m correct to the nearest 0.1 m.
- Is it possible that the actual length of this metal wire exceeds 206 cm? Explain your answer.
  - Is it possible to cut this metal wire into 46 pieces of shorter metal wires, with each length measured as 5 cm correct to the nearest cm? Explain your answer.

**1.3 HKCEE MA 2008 – I – 7**

John wants to buy the following items in a supermarket:

Item	Unit price	Quantity needed
Biscuit	\$8.2 per pack	4 packs
Chocolate	\$16.3 per box	3 boxes
Soft drink	\$4.8 per can	2 cans

- By rounding up the unit price of each item to the nearest dollar, estimate the total amount that John should pay.
- If John has only \$100, does he have enough money to buy all the items needed? Use the result of (a) to explain your answer.

**1.2 HKCEE MA 2007 – I – 10**

- (a) If the length of a piece of thin metal wire is measured as 5 cm correct to the nearest cm, find the least possible length of the metal wire.
- (b) The length of a piece of thin metal wire is measured as 2.0 m correct to the nearest 0.1 m.
- Is it possible that the actual length of this metal wire exceeds 206 cm? Explain your answer.
  - Is it possible to cut this metal wire into 46 pieces of shorter metal wires, with each length measured as 5 cm correct to the nearest cm? Explain your answer.

**1.3 HKCEE MA 2008 – I – 7**

John wants to buy the following items in a supermarket:

Item	Unit price	Quantity needed
Biscuit	\$8.2 per pack	4 packs
Chocolate	\$16.3 per box	3 boxes
Soft drink	\$4.8 per can	2 cans

- By rounding up the unit price of each item to the nearest dollar, estimate the total amount that John should pay.
- If John has only \$100, does he have enough money to buy all the items needed? Use the result of (a) to explain your answer.

**1.4 HKCEE MA 2009 – I – 4**

Round off 405.504 to

- the nearest integer,
- 2 decimal places,
- 2 significant figures.

As in the picture, the green box is for drag and select region and blue box is the image that will be cropped. We allow high customization when using the tool.

1. customized image path and snippets path

2. real time Mathpix result preview

3. store to MongoDB or not

4. record or not the image to the process.txt record