(People's feedback based on GEOMETRIX website):

(Real story): Before making the website, I did talk to a friend—a person who makes HTML websites and sells them online—about how to make a math website that would be comfortable for everyone. He said to include aesthetic, math-related colour, simple layouts/structures, easy-to-use, and better user-friendly designs. I considered those aspects and implemented them. Those are the feedbacks I got from others:

Those feedbacks were received at the beginning and during the developing process -

1. I think you should add some colour that is mainly for math users, maybe light blue or pale blue would be good.
2. Try to add a vertical navigation bar instead of a horizontal one because that would make your website unique. What do you think?
3. Use easy grids and fonts to improve the websites.
4. Easy interactions please.
5. Make it easier to use by adding background colour (e.g white would be better if it fits with blue).
6. Your texts are kind of mixed with the background colour, maybe you should change your text colour to black.
7. I recommend you to use external javascript files to make smooth transitions.
8. Maybe animations?
9. Animate the labels.
10. Add fade in and out effects.
11. A comment section, just like Reddit if you can make one.
12. Interactive buttons for different levels would enhance the website.
13. I don't recommend you add CSS and Javascript inside an HTML page because that would be too messy and unorganized. This is my suggestion, if you want to use yours, you can.

Those feedbacks were received when I finished the website -

1. It looks clean. Easy to get around and I hope you can make more functions.
2. The design's chill, not too flashy.
3. Honestly, I'd use this if I needed help with math. Some people would like this.
4. Cool how it's organized by level. Makes it less overwhelming.
5. The site is smooth. Doesn't feel laggy or old.
6. This feels more welcoming than other school sites tbh, it is more recommended if you can add more interactions to your web.

7. Not bad. I would send this to a friend struggling with geometry.
8. It's nice to see something that doesn't look like a boring worksheet.
9. Simple, but it works. Sometimes that's all you need for a website, if you can enhance it even more, that would be way better. :)
10. How did you make this, this is really the goat.
11. Layouts are clear and I didn't get lost trying to find things. I think most people would think so.
12. For real, I think this is way better than those sites full of ads and junk, I'm not lying. But more functions would be better.
13. Very creative.
14. Nice, you did a really good job.
15. Add a comment section to share thoughts.
16. If you can try to make the website a bit simple and add blue colour (because math fits blue), then that would be way better than the current version.

From the comments above, some people pointed out that more functions (e.g., more interactions) should be implemented to GEOMETRIX - I did. What I added is the feedback section where people can actively discuss their problems or share their thoughts throughout the website. Thus, adding more functions. Additionally, I noticed some people said the website has a simple, aesthetic layout, this is because I changed from other people's feedback repeatedly.

[CHECK BELOW - APPLYING THE FEEDBACKS TO MY WEBSITE & CODE TESTINGS]

1.

Using a Class for the Body Tag

<body class-"index">

Why this didn't work well:
<body>

First person said adding a class="index" makes it easier to target styles for different pages (e.g., to highlight the home link in the navigation).

I think it's better for improving the website; I agree.

Because it helps users feel oriented by making the navigation clearer. I mean, that's what every person said in my feedback section: make the website simple and clean.

Thus, I used his method and improved the website. **[Agreed]**

2.

Linking External Stylesheets

<link rel="stylesheet" href="MathWebsite.css" />

Why this didn't work well:
<style>

/* CSS written inside the HTML */
</style>

Another person replied that keeping everything inside the HTML makes the page harder to manage and slower to load.

Initially, I found writing '<style>' inside .html is difficult to see the separated code. Then I used a different code page, MathWebsite.css, to organize CSS code into a different section. When I applied his suggestion, I found everything was organized and clean.

Also, his suggestion is better because it keeps the code neat and consistent across different pages and linking to an external CSS file makes things easier to update and manage. **[Agreed]**

3.

Using External JavaScript Files

<script src="MathPolishment.js"></script>

Why this didn't work well:
<script>

// Scripts written inside the HTML
</script>

A third person said that putting all the scripts inside HTML can make the page message hard to debug.

I thought about this before, if I add javascript inside HTML it would be the same problem as CSS. So, I considered this as a recommendation. In other words, I think putting the JS in its own file keeps the HTML clean and makes it easier to reuse the code.

Why is it better? Because it is easier to read and scale the project as it grows. **[Agreed]**

4.

Clear Branding with Class for Titles

<h1 class="titles">LearnMATH</h1>

Why this didn't work well:
<h1>LearnMATH</h1>

A fourth person said without a class, I couldn't easily apply custom styles. However, I already knew this technique, but if I didn't know this, I would strongly agree with his opinion because 'class' is very crucial to design a good website.

Moreover, adding a class allows me to control the title's look and feel across the website.

He also highlighted that it creates a consistent look and helps reinforce the site's brand. **[Agreed]**

5.

Adding Animation to Section Titles

<h2 class=”fancy-heading” data-aos=”fade-up”>About GEOMETRIX</h2>

Why this didn’t work well:
<h2>About GEOMETRIX</h2>

Plain titles felt dull without animation, according to the fifth person.
</script>

I disagreed at first, however, but after that I think his method is fast and efficient.

He said it makes the page feel more interactive and modern.

[Agreed]

6.

Personalizing the Level Selector

<div class=”level-selector” data-aos-”zoom-in-up”>

Why this didn’t work well:

The person said without a class or animation, it looked boring and didn’t stand out. Adding a class and animation makes it more interactive, smooth/polished, and visually appealing.

He said why it’s better than my previous version: it improves the user experience by making it easier and more fun to choose a level. **[Agreed]**

7.

Fast Level Selection with Radio Buttons

<input type=”radio” ud=”advanced” name=”math-level”>

Why this didn’t work well:
<select><option>Advanced</option></select>

The person said dropdowns take more clicks and are harder to see. The reason is radio buttons show options right away and are easier to interact with. Because it is quicker for users to make a choice and keeps things simple.

I took his suggestion and applied to my website.

[Agreed]

8.

Level Descriptions that Appear When Needed

Why this didn’t work well:
Beginner level description

The person said always showing the description in my website takes up space and looks cluttered. I took his advice and used data-level to hide/show descriptions based on the user’s choice.

Why is it better?
Keeps the layout clean and gives the user a more personalized experience.

[Agreed]

9.

Community Post Form

<form class=”post-form” action=” “ method=”post”>

Why this didn’t work well:
<form>

The person suggested that no action or method meant the form wouldn’t work properly. This is why I changed it: including method and action ensures the form is secure and functional.

Why is it better?
He said it works properly and can be expanded later if needed.

[Agreed]

10.

Limiting Text Length for Better Clarity

maxlength=”500”

The person suggested that I make the comment section to 500 words maximum; however, I was thinking this would be too many words for a person to discuss (I was thinking about 200 words). In the end, his suggestion is better than mine because if a user wants to share thoughts or discuss thoroughly, then more words are able to speak clearly.

Moreover, adding a character limit keeps the content short and neat (500).

Why is it better?
He said the benefit is helping to avoid long, messy posts and makes it easier to read; 500 words is sufficient to use.

[Agreed]

11.

JavaScript Logic for Interactivity

```
document.addEventListener('DOMContentLoaded', function(){

// Key interactive and dynamic logic

});
```

(My original code was below-average than suggested one)

A person said putting everything inline on click or on input attributes made the code messy. That's why using event listeners after the page loads keeps things neat and more reliable. Moreover, it's easier to maintain, and the code works even when elements change.

[Agreed]

12.

Fade-in Effect for Main Content

```
mainContent.style.opacity = '0';

setTimeout(() => {
    mainContent.style.transition = 'opacity 0.5s ease';
    mainContent.style.opacity = '1';

}, 100);
```

A person suggested this fade-in effect makes the content appear smoothly, because it makes the website feel more polished and less jarring when loading. (Isn't that what the feedback section wants?)

[Agreed]

13.

Real-time Character Count

```
textarea.addEventListener('input', function(){
  const remaining = maxLength - this.value.length;

charCount.textContent = '{remaining} characters remaining';
});
```

A person highlighted static labels like "500 character" didn't change as users typed. Real-time feedback helps know how much they can still type. He said this is better because users have a clearer idea of their text length, improving the experience.

Personal reply: Since this would be a little bit complex, so I didn't implement this code.

[Moderate]

14.

Input Validation Before Submission

```
if (textarea.value.trim().length < 10) {

alert('Please write at least 10 characters for your post');
}
```

I agreed with this person's suggestion because if a person spams the comment with few or even no characters, it would be annoying and frustrating.

He said: “No validation meant users could submit empty or spammy posts.”

I changed: checking the text length before submission helps maintain quality content.

Why is it better?
The person said it keeps posts clean, encourages better contributions, and reduces spam.

I agreed with his suggestion and implemented it on my website. Initially, I didn’t have such functions to avoid those problems, thus using his method.

[Agreed]

15.

Animating Label Clicks for a Better Feel

```
label.addEventListener(‘click’, ‘function(){

This style.transform = ‘scale(1.05)’;
});
```

I tested this code and it works better than my previous one. The person said the label didn’t give any feedback when clicked in my previous version of code. So, I added a small zoom-in effect when the label is clicked adds a more interactive feel.

Why is it better?
It makes the interface feel more responsive and engaging.

I used this code for smooth transformations and animations.

[Agreed]

16.

Adding a Dynamic Footer with Branding

```
const footer = document.createElement('footer');

footer.innerHTML = '{newDate().getFullYear()} GEOMETRIX - LearnMATH';

document.body.appendChild(footer);
```

A person said my previous code didn't work well because hardcoding the footage in HTML meant I had to manually update it every year. He told me to make the footer dynamic by using his suggested code so the year updates automatically.

Why is it better?
He said less maintenance and ensures the footer always has the correct year.

I found this is a really smart way of updating the footer - I used it.

[Agreed]

[After those improvements and suggestions, I received a lot of good comments - you can see them at the beginning of the first page].