Author: Jesse Williams | Date: February 26th, 2025 | Version*: 1.3*

STEM Ranked

Abstract

This is a ranking system for STEM students at I.M. Terrell based on their performance in math and science classes, computer science projects, and weighted GPA, certificates, and extracurricular activities. There will also be an overall ranking based on your scores in every other category, determining the top STEM majors. Eventually, it could be implemented into our school's core award system every 6 weeks.

Table of Contents

Introduction

(October 20, 2025)

I actually started this project months prior with an idea: STEM majors seem to always be neglected at my school, and with no competitive drive, nothing was pushing us forward to be better. I wanted to create something that motivated the rest of the STEM department at I.M. Terrell to push forward and be the best, and hopefully leave a legacy for myself at this school when I inevitably leave next year. This idea became a concept of a ranking system, similar to something you would see in competitive spaces like Minecraft's PVP ranking system on mctiers.com.

I started by making a Google Form connected to a Google Sheets, with the idea that users could upload their accomplishments to me, ranked accordingly based on a specific criterion. The ranking list would update every month, giving people time to create new projects to blast themselves up on the leaderboard. I envisioned every Monday, when reviewing school-wide slideshows, a ranking system being presented in front of the student body that would motivate STEM students to rise to the top.

I lost interest in this project a little while later because I couldn't find a way to create a website unblocked by my school district, until now. My new goal is to use p5.js to use my HTML, CSS, and JavaScript skills to create a website that will display our school's top STEM

Overview

(October 20, 2025)

This project aims to create a ranking system for STEM students at I.M. Terrell based on their performance in math and science classes, computer science projects, and weighted GPA, certificates, and extracurricular activities. There will also be an overall ranking based on your scores in every other category, determining the top STEM majors.

Ranking will be done by either a team of moderators I appoint, a club, or myself. The I.M. Terrell STEM ranking system is based on merit and overall performance. You can only resubmit this form every month. Once your answers are flagged for review and one of our appointed moderators will check your submission and update your rank accordingly. You cannot go down in rank unless your new submission to a category is worse than your last one, or someone submits a better project than you for a category. You will be placed in a different tier depending on the level of merit shown in your submission. There can only be one HT1 and two LT1s for each category. In most categories, you will submit a screenshot. You can submit multiple items in different coding languages for categories like Computer Science. Projects for people in the top 25 in a category will be shown on the website. The top 10 in each category will potentially need to be fact-checked on responses—they will have to provide documentation. More categories will be added in the future.

Research

Coding

Entry 1.0

(October 21st, 2025)

I started by adding my school's logo with a link to their website in the top right corner of my website. I uploaded the logo to a file called assets and added it to my HTML file, then added the file parameters, and linked the website in my HTML file (Fig. 1.1). After, I changed the title and added the icon to the website image as well (Fig. 1.2).

```
        <a href="https://imterrell.fwisd.org/"
target="_blank">
          <img src="assets/IMT.png" alt="Terrell logo"
style="width: 40px;height: 40px;">
        </a>
```

Fig. 1.1 logo and link to IMT page

```
    <title>STEM Ranked</title>
    <link rel="icon" type="image/x-icon"
href="/assets/IMT.ico">
```

Fig. 1.2 this would change the tab name and the tab icon

I wanted to add a unique background that was a video. I couldn't find any videos that I could install on my school-issued MacBook, because of the amount of district-wide blocks. I found one on Canva and uploaded it to my assets folder. I started by making a new class in the body called main_body, where I could put all of my main elements. I referenced a video and made sure it played automatically when the site is launched and looped (Fig. 1.3). Then, I edited the CSS file to adjust some of the parameters for the video (Fig.1. 4). I'll adjust the size later and change the speed at which it scrolls.

```
  <body>
    <div class="main_body">
      <video autoplay loop muted playsinline class="back-
video">
        <source
        src="assets/STEM_Ranked_Background_1.mp4"
        type="video/mp4">
      </video>
```

Fig. 1.3 this plays a video I have in my assets of just a blue background

```
.back-video {
  position: fixed;
  width:100%;
  height: 100%;
  z-index: -1;
}
```

Fig. 1.4 I am actively working on fixing the class to fit the whole screen

I added a container class for the main area where it would display all the rankings. I then added subsequent courses for the actual individual rank displays: rank & card (Fig. 1.5). I went into my CSS file and set up the styling for the container. The important thing I did here was make the flex-direction column so that all of the elements in the container class would be set up in a column format (Fig. 1.6).

```
        <div class="container">
          <div class="rank">
            <div class="card">
              1.
```

Fig. 1.5 so far the class order

```
.container {
  display: flex;
  justify-content: flex-start;
  align-items: center;
  flex-direction: column;
  position: relative;
  margin: 0 auto;
  height: 300vh;
  width: 70%;
  background-color: #06132D60;
  padding: 5px 10px;
  border-width: 5px;
  border-style: solid;
  border-color: #091733;
  border-radius: 15px;
}
```

Fig. 1.6 the styling for the container class

When setting up the rank class, I added some effects when you hover over the element. Whenever you hover over it, it gets bigger and lights up a little (Fig. 1.7). I made the rank class so I could make subsequent rank objects with ease. I plan to implement a sound effect every time a button is hovered over, but that will be later.

```
.rank {
  display: flex;
  justify-content: flex-start;
  align-items: flex-start;
  width:100%;
  height: 30%;
  background-color: #0C1F44;
  border-width: 5px;
  border-style: solid;
  border-color: #091733;
  border-radius: 8px;
  cursor: pointer;
  transition: background-color 0.3s ease;
  margin-bottom: 10px;
}

.rank:hover {
  background-color: #06284D;
  transform: scale(1.02);
}
```

Fig. 1.7 rank class with hover effects

I added an image under the card class in the HTML file as a placeholder. Then I went back into the CSS file and made the shape a trapezoid by setting up a clip path (Fig. 1.8). This is currently what the website looks like (Fig. 1.9).

```
.card {
  width:25%;
  height: 100%;
  background-color: #FFEB3B;
  clip-path: polygon(0% 0%, 100% 0%, 85% 100%, 0% 100%);

}
```

Fig. 1.8 polygon is an upside-down trapezoid

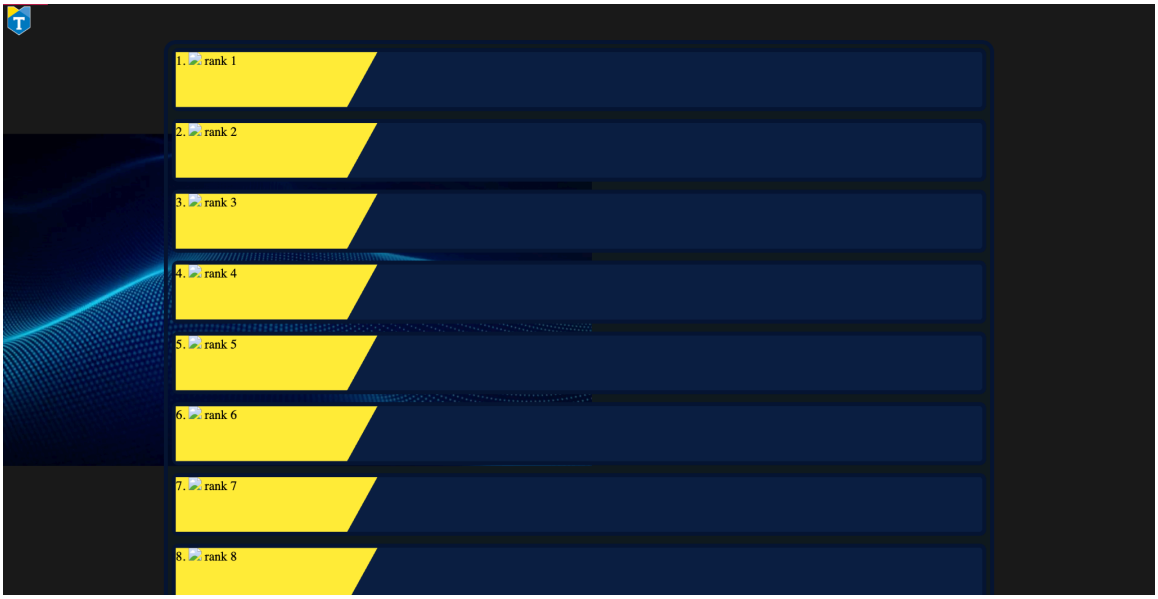

Fig. 1.9 current website on [p5.js]

After I fixed the issue of the background not working properly, I had a background that cut off after a certain point. So now I needed to make the video scroll with the user, but make it scroll slower so that it appears as if the image extends. I started by creating a script in my HTML file in JavaScript, that listens for whenever the user scrolls and then transforms the video by the scrollY multiplied by the scroll speed factor, which controlled how fast the background moved when scrolling (Fig. 1.10). Then I went into the CSS file, and made sure that it starts as 0 (Fig. 1.11).

```
    <script>
        const backVideo = document.querySelector('.back-
video');
        const scrollSpeedFactor = 0.5;

        window.addEventListener('scroll', () => {
            const scrollY = window.scrollY;
            backVideo.style.transform =
`translateY(${scrollY * scrollSpeedFactor}px)`;
        });
    </script>
```

Fig. 1.10 this is JavaScript in an HTML file

```
.back-video {
  position: absolute;
  right: 0;
  bottom: 200;
  min-width:200%;
  min-height: 200%;
  width: auto;
  height: auto;
  z-index: -1;
  object-fit: cover;
  overflow: hidden;
  transform: translateY(0);
}
```

Fig. 1.11 added transform line at the bottom

Bugs

Entry 1.0

(October 21st, 2025)

Whenever I was trying to configure the size of the background video in my CSS file, I kept running into issues making it fit the full screen and achieve the effect that I wanted. I tried various things, and nothing worked.

In the end, I fixed the issue by adding a minimum width and height, and adding an object fitting and hiding the overflow (Fig. 2.1).

```
.back-video {
  position: absolute;
  right: 0;
  bottom: 200;
  min-width:200%;
  min-height: 200%;
  width: auto;
  height: auto;
  z-index: -1;
  object-fit: cover;
  overflow: hidden;
```

Fig. 2.1 I chose not to make it a fixed position because I wanted it to scroll, just slightly

6.0 Future Work

1. Create a sorting system that parses through a html file full of responses from students
2. Display items from the responses file on the website
3. Change the display depending on the selected tab
4. Go through the schools stem department for approval
5. Go through the school's administration for approval

7.0 References

- p5.js (https://p5js.org/) - the designated compiler that runs the code