Website Overview

A Little Background

All websites are built using a collection of HTML, CSS, and JavaScript files. These files are stored on a server (generally one owned by either Amazon, Google, or Microsoft) and when your computer goes to a specific site it asks the server for the files associated with that site. The server sends the files to your computer, and your browser interprets them into the webpage that you see.

You can think of a web page like the human body. HTML files are the skeleton. They set the structure of the page through headings and sections (generally known as divs). CSS files are the skin and clothes that the page may wear. They are used to give text and shapes color, or even to give shapes their shape. If you ever want to go down a cool rabbit hole, check out the link below for drawing with CSS. And finally, JavaScript files are the muscles. JavaScript is used to make the page interactive: dancing lines, dancing bears, things that pop into and out of existence as you scroll down the page. All of that is JavaScript.

HTML = HyperText Markup Language
CSS = Cascading Style Sheets
JavaScript = JavaScript

Drawing with CSS: https://blog.prototypr.io/how-i-started-drawing-css-images-3fd878675c89

Frameworks and Languages

Programmers tend to be efficient (and/or lazy), so they don't like to do things twice. Most sites you will find these days are not built using just HTML, CSS, and JavaScript files. Instead, they are built using a framework. These frameworks are used to make coding faster and easier. The programmer codes in a different language and those files are turned into HTML, CSS, and JavaScript files by the framework. Some notable frameworks include Ruby on Rails (my preference), Django, Node.js, Laravel, and React (originally designed by an engineer at Facebook).

Each of these frameworks is built on a web language, so generally learning a new framework can mean learning a new language. When you use Ruby on Rails, you end up writing most of your code in Ruby, a language built to prioritize human understanding over computer understanding. When you use Django, your code is mostly Python, a language built to allow programmers to use fewer lines of code to achieve similar results.

There is no right answer in terms of what language and framework to use when building a site. Anyone who tells you different is wrong and is a snob. If you want a web app that has dynamic, changing web pages, you might choose React or Node.js. If you want a web app that does a lot with data, you might choose Django or another Python framework. If you want a web app where you can plug-and-play different code pieces and tools made by others, you might choose Ruby on Rails. An if you want a basic website with static, unchanging pages, you might choose Hugo or Jekyll as your framework.

Notes About the School's Site

The school's site is built using the Hugo framework, a static site framework built by engineers at Google. Most of the coding is done in Markdown, a language used to make writing HTML even easier. The Hugo framework takes the files in the app's folder and turns them into the necessary HTML, CSS, and JavaScript files that the server will then send to any computer that requests them.

Here's an example of a title in HTML:

```
<h1>The Title</h1>
```

And the same in Markdown:

```
# The Title
```

Here's a subtitle in HTML:

```
<h2>A Subtitle</h2>
```

And in Markdown:

```
## A Subtitle
```

Here's a paragraph in HTML:

```
<p>
    The school's site is built using the Hugo framework, a static site framework built by engineers at Google. Most     of the coding is done in Markdown, a language used to make writing HTML even easier.
</p>
```

And in Markdown:

The school's site is built using the Hugo framework, a static site framework built by engineers at Google. Most of the coding is done in Markdown, a language used to make writing HTML even easier.

Here's a table in HTML:

```
<table>
    <thead>
        <tr>
            <th>Header 1</th>
            <th>Header 2</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>Row 1, Col 1</td>
            <td>Row 1, Col 2</td>
        </tr>
        <tr>
            <td>Row 2, Col 1</td>
            <td>Row 2, Col 2</td>
        </tr>
```

```
        </tbody>
</table>
```

And in Markdown:

```
| Header 1 | Header 2 |
| --- | --- |
| Row 1, Col 1 | Row 1, Col 2 |
| Row 2, Col 1 | Row 2, Col 2 |
```

To be fair to HTML, Markdown loses a lot of the flexibility that is available in HTML, but if you want quick and easy-to-learn, it's hard to beat Markdown.

The website is built using a Hugo Theme called Alchemy. Alchemy is no longer maintained by it's developers (boo!), but there is a newer version with a lot of the same design: Wowchemy. Think of this theme as sort of the CSS handler of the website, it gives the colors and looks to the buttons for almost free.

Because the coding is done by school staff, the district pays $10 a year for the website, which is essentially only for hosting. The site is hosted by Netlify.

Hugo: https://gohugo.io/documentation/
Markdown Cheatsheet: https://www.markdownguide.org/cheat-sheet/
Wowchemy: https://wowchemy.com/
Netlify: https://app.netlify.com