Internship Report – Frontend Dev(Week3)

Name: Zainab
Father Name: Assad Qayyum
Date: 10th July, 2025
Internship Domain: Front-end Intern
Task: CSS - Variables in CSS (:root, var()), Custom Fonts

Task Overview: (Day4)

Today's task focused on understanding and applying **CSS Variables** using **:root and var(),** along with learning how to add and use **Custom Fonts** in web development. The goal was to explore how these tools can simplify styling and enhance visual design.

Content Covered:

- CSS Variables (:root, --name, var())
- Declaring and using custom properties
- Scoping and fallback values
- Custom Fonts using Google Fonts

Introduction:

CSS Variables and Custom Fonts help achieve maintainability and consistency by allowing developers to store reusable styling values and customize typography. Today I explored how to define global design tokens using variables and how to bring elegant, branded fonts into a website using Google Fonts.

1. Variables in CSS (:root, var()):

CSS variables (or **custom properties**) are user-defined **reusable values** for CSS properties like colors, font sizes, spacing, etc. This makes it **easier to update your design** — if you change the **variable value once**, it **updates everywhere** it's used.

:root	Represents the highest level (HTML root element) – best place to declare global variables.
--variable-name	Custom name you assign to a property (e.g., --main-color).
var(--name)	How you access the value of a variable.

Syntax:

These are defined using the **-- syntax** and can be accessed using the **var() function**.

```
:root {
  --primary-color: #3498db;
  --font-size-lg: 2rem;
}
```

And use it in your code like this:

```
h1 {
  color: var(--primary-color);
  font-size: var(--font-size-lg);
}
```

Why :root?

- **:root** is a special CSS selector that targets the **top-level HTML** element (<html>).
- When you define variables inside :root, they become **globally accessible** in all elements of the CSS.

You can even add fallback values: **color: var(--non-existing-var, black);**

2. Custom Fonts in CSS:

Custom fonts are **non-default fonts** used in your web pages to **improve visual appeal** and branding. Most developers use Google Fonts due to ease of use and wide variety.

There are two major ways to use them, either you load them from **Google Fonts** or add your **own .ttf file**.

A) Google Fonts:

Google Fonts is a free online library of web-optimized fonts provided by Google. It allows developers to easily add stylish, professional typography to their websites using a simple link.

How to Use Google Fonts:

Step 1: Go to: fonts.google.com

Step 2: Choose a font, e.g., Poppins

Copy the <link> tag and paste it in the <head> of your HTML

```
<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=swap"
rel="stylesheet">
```

Step 3: Use it in CSS like this:

```
body {
  font-family: 'Poppins', sans-serif;
}
```

B) @font-face – For Local Fonts

While Google Fonts lets you import fonts from an online library, @font-face allows you to load fonts from your local project folder — like .ttf, .woff, or .woff2 files saved inside your own files.

How to Use @font-face:

Step 1: Get a font file

- Download a font file like .ttf or .woff e.g. OpenSans-Regular.ttf
- Place it inside a folder in your project like: fonts/OpenSans-Regular.ttf

Step 2: Write @font-face in your CSS

```
@font-face {
  font-family: 'OpenSansCustom';
  src: url('fonts/OpenSans-Regular.ttf') format('truetype');
}
```

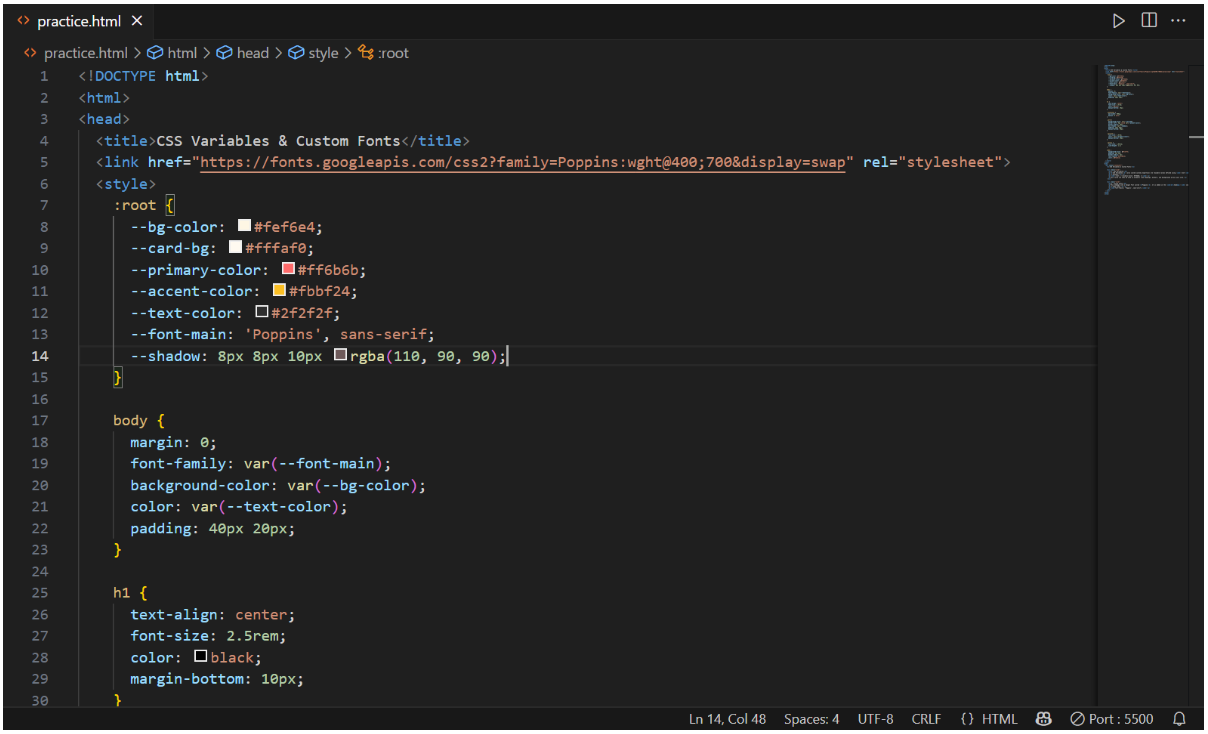
- font-family: You name the font (you can call it whatever you want)
- src: Path to your font file, relative to the CSS file.

Step 3: Use it like any other font

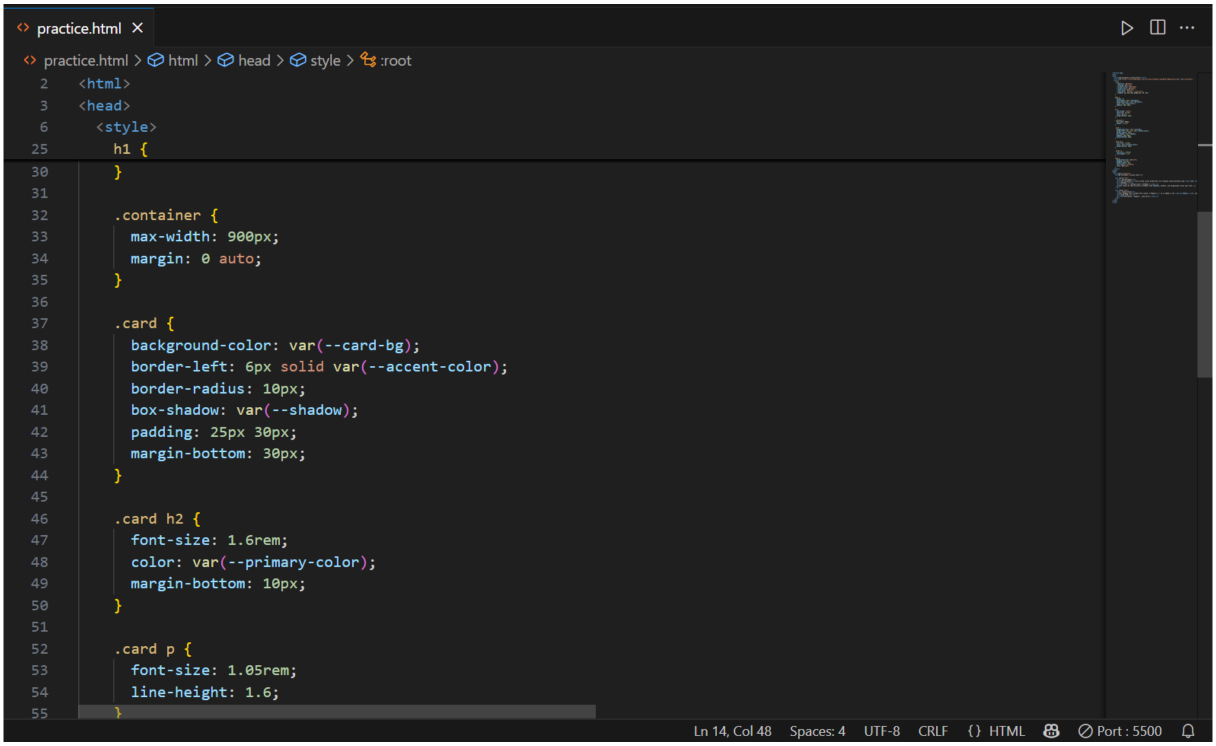
```
body {
  font-family: 'OpenSansCustom', sans-serif;
}
```

Practice Code:

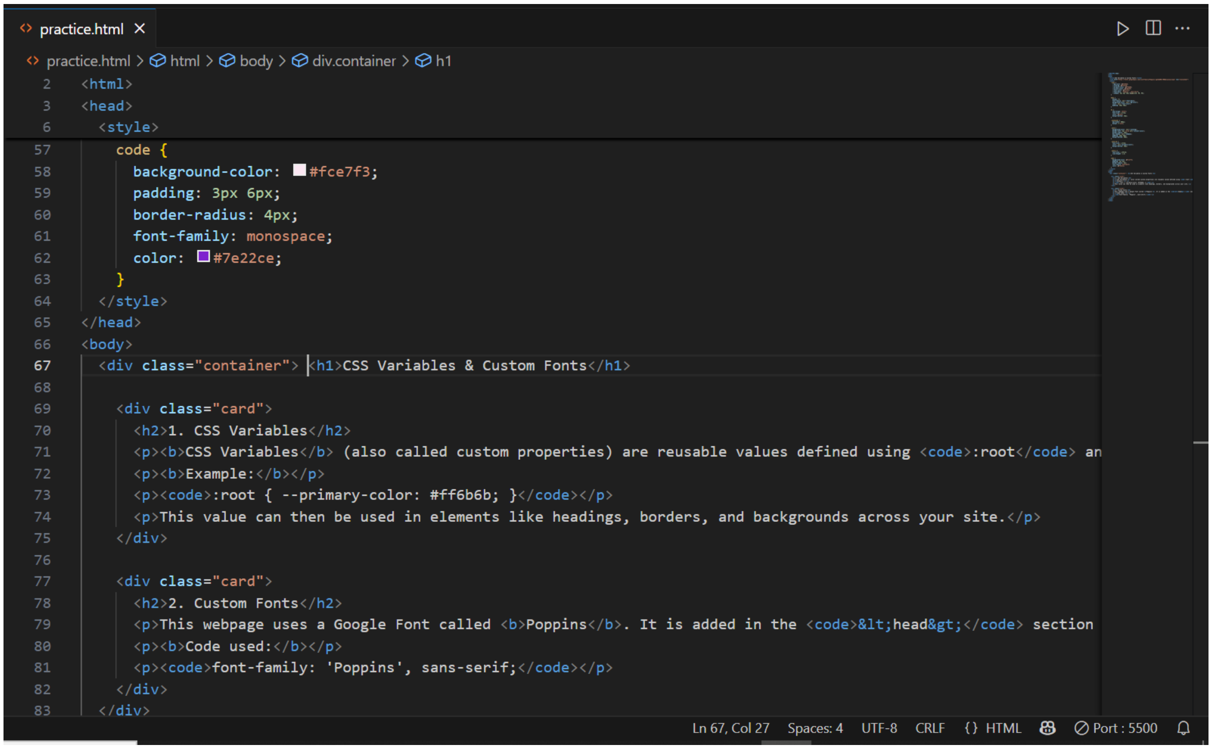
The This code demonstrates how to use **CSS Variables (:root, var())** for consistent styling and applies a **Google Font (Poppins)** to enhance typography, all within a colorful, card-based layout for easy learning.


practice.html
practice.html > html > head > style > :root
Ln 14, Col 48   Spaces: 4   UTF-8   CRLF   {} HTML   Port : 5500

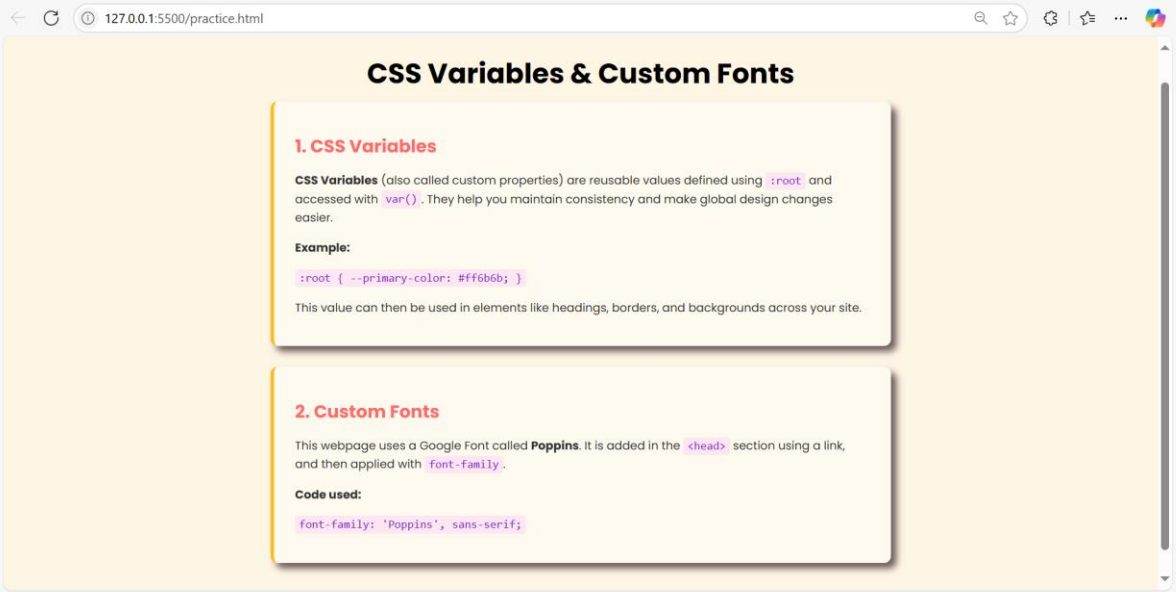
```
<!DOCTYPE html>
<html>
<head>
  <title>CSS Variables & Custom Fonts</title>
  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=swap" rel="stylesheet">
  <style>
    :root {
      --bg-color: #fef6e4;
      --card-bg: #fffaf0;
      --primary-color: #ff6b6b;
      --accent-color: #fbbf24;
      --text-color: #2f2f2f;
      --font-main: 'Poppins', sans-serif;
      --shadow: 8px 8px 10px rgba(110, 90, 90);
    }

    body {
      margin: 0;
      font-family: var(--font-main);
      background-color: var(--bg-color);
      color: var(--text-color);
      padding: 40px 20px;
    }

    h1 {
      text-align: center;
      font-size: 2.5rem;
      color: black;
      margin-bottom: 10px;
    }
```


practice.html
practice.html > html > head > style > :root
Ln 14, Col 48 Spaces: 4 UTF-8 CRLF {} HTML Port : 5500

```
<html>
<head>
  <style>
    h1 {
    }

    .container {
      max-width: 900px;
      margin: 0 auto;
    }

    .card {
      background-color: var(--card-bg);
      border-left: 6px solid var(--accent-color);
      border-radius: 10px;
      box-shadow: var(--shadow);
      padding: 25px 30px;
      margin-bottom: 30px;
    }

    .card h2 {
      font-size: 1.6rem;
      color: var(--primary-color);
      margin-bottom: 10px;
    }

    .card p {
      font-size: 1.05rem;
      line-height: 1.6;
    }
```


practice.html
practice.html > html > body > div.container > h1
Ln 67, Col 27 Spaces: 4 UTF-8 CRLF {} HTML Port : 5500

```
<html>
<head>
  <style>
    code {
      background-color: #fce7f3;
      padding: 3px 6px;
      border-radius: 4px;
      font-family: monospace;
      color: #7e22ce;
    }
  </style>
</head>
<body>
  <div class="container"> <h1>CSS Variables & Custom Fonts</h1>

    <div class="card">
      <h2>1. CSS Variables</h2>
      <p><b>CSS Variables</b> (also called custom properties) are reusable values defined using <code>:root</code> an
      <p><b>Example:</b></p>
      <p><code>:root { --primary-color: #ff6b6b; }</code></p>
      <p>This value can then be used in elements like headings, borders, and backgrounds across your site.</p>
    </div>

    <div class="card">
      <h2>2. Custom Fonts</h2>
      <p>This webpage uses a Google Font called <b>Poppins</b>. It is added in the <code>&lt;head&gt;</code> section
      <p><b>Code used:</b></p>
      <p><code>font-family: 'Poppins', sans-serif;</code></p>
    </div>
  </div>
```



CSS Variables & Custom Fonts

1. CSS Variables

CSS Variables (also called custom properties) are reusable values defined using `:root` and accessed with `var()`. They help you maintain consistency and make global design changes easier.

Example:

```
:root { --primary-color: #ff6b6b; }
```

This value can then be used in elements like headings, borders, and backgrounds across your site.

2. Custom Fonts

This webpage uses a Google Font called **Poppins**. It is added in the `<head>` section using a link, and then applied with `font-family`.

Code used:

```
font-family: 'Poppins', sans-serif;
```

Conclusion:

Today’s concepts—CSS Variables and Custom Fonts—are vital for scalable and professional web design. Variables help simplify and organize CSS, making large projects easier to maintain. Meanwhile, custom fonts enhance the visual personality of a website, giving designers freedom to match branding and style.