INTERNSHIP REPORT

WEEK 3 DAY 5

Submitted to:	**Ali Hyder**	Submission Date:	**11th July, 2025**
Internship Domain:	**Front Development**	Internship Name:	**ProSensia**
Student Name:	**Yasal Qamar**	Roll No.	**S25031**

CSS Pseudo-classes & Pseudo-elements

Objective

To learn how to implement **CSS Variables** using the **:root selector and var() function**, and to apply **Custom Fonts** through Google Fonts in modern web development. The goal was to simplify styling and ensure a more maintainable and visually appealing web design.

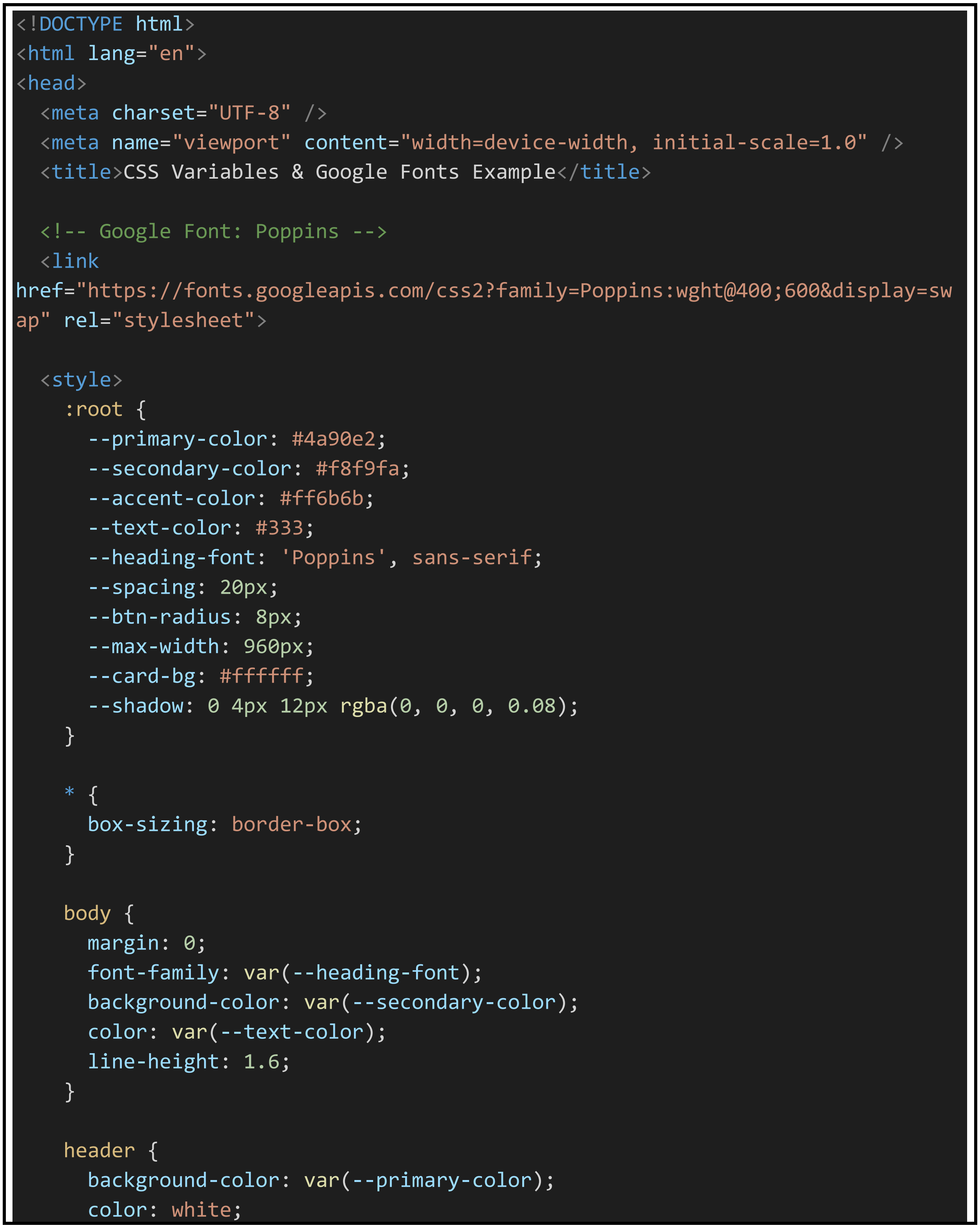
Introduction:

CSS Variables (Custom Properties) and **Google Fonts** are two essential features in modern front-end development. CSS Variables allow developers to store values (colors, font sizes, etc.) in one place and reuse them across stylesheets, improving consistency and maintainability. Google Fonts offers a library of open-source fonts to enhance typography and design.

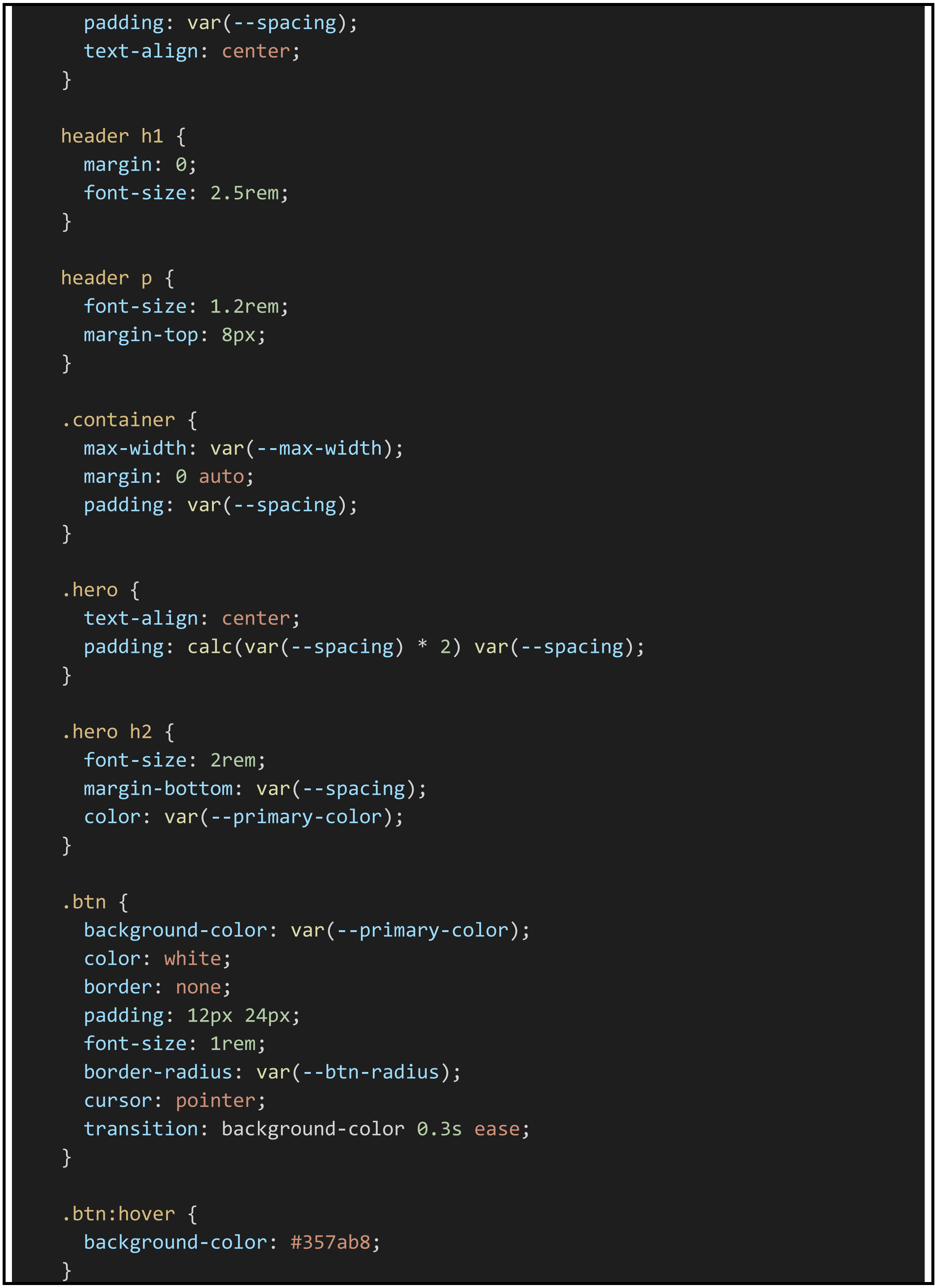
Key Learnings:

- How to declare global variables using :root in CSS.
- How to use var(--name) to access these variables.
- Advantages of using CSS Variables for design consistency and easier theme management.
- How to import and apply Google Fonts to improve website typography.
- Using Flexbox for responsive, side-by-side content layout.
- Building a neat and responsive card-based layout using variables and custom fonts.

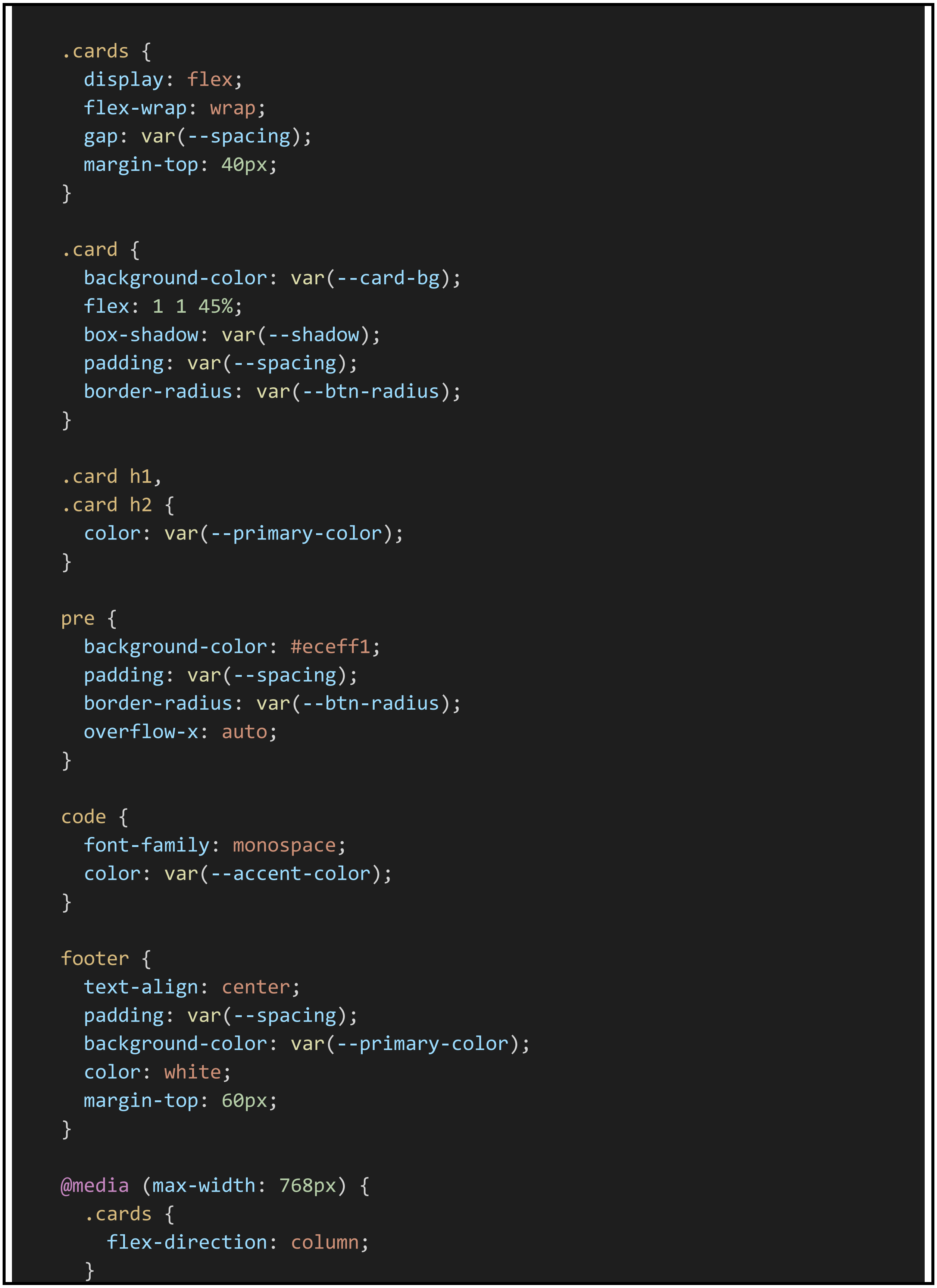
1. CODING



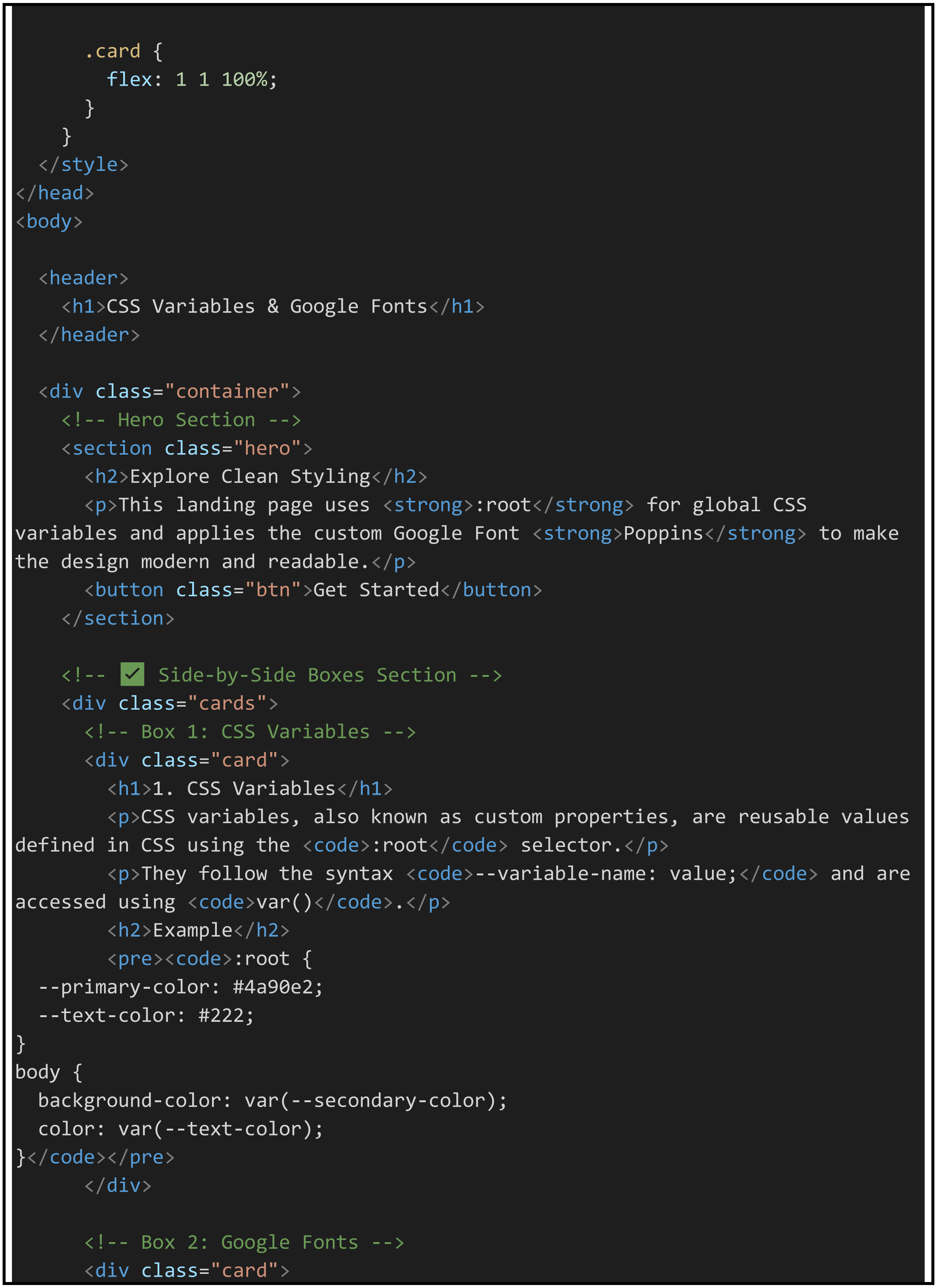
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>CSS Variables & Google Fonts Example</title>

  <!-- Google Font: Poppins -->
  <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600&display=sw
ap" rel="stylesheet">

  <style>
    :root {
      --primary-color: #4a90e2;
      --secondary-color: #f8f9fa;
      --accent-color: #ff6b6b;
      --text-color: #333;
      --heading-font: 'Poppins', sans-serif;
      --spacing: 20px;
      --btn-radius: 8px;
      --max-width: 960px;
      --card-bg: #ffffff;
      --shadow: 0 4px 12px rgba(0, 0, 0, 0.08);
    }

    * {
      box-sizing: border-box;
    }

    body {
      margin: 0;
      font-family: var(--heading-font);
      background-color: var(--secondary-color);
      color: var(--text-color);
      line-height: 1.6;
    }

    header {
      background-color: var(--primary-color);
      color: white;
```



```css
  padding: var(--spacing);
  text-align: center;
}

header h1 {
  margin: 0;
  font-size: 2.5rem;
}

header p {
  font-size: 1.2rem;
  margin-top: 8px;
}

.container {
  max-width: var(--max-width);
  margin: 0 auto;
  padding: var(--spacing);
}

.hero {
  text-align: center;
  padding: calc(var(--spacing) * 2) var(--spacing);
}

.hero h2 {
  font-size: 2rem;
  margin-bottom: var(--spacing);
  color: var(--primary-color);
}

.btn {
  background-color: var(--primary-color);
  color: white;
  border: none;
  padding: 12px 24px;
  font-size: 1rem;
  border-radius: var(--btn-radius);
  cursor: pointer;
  transition: background-color 0.3s ease;
}

.btn:hover {
  background-color: #357ab8;
}
```



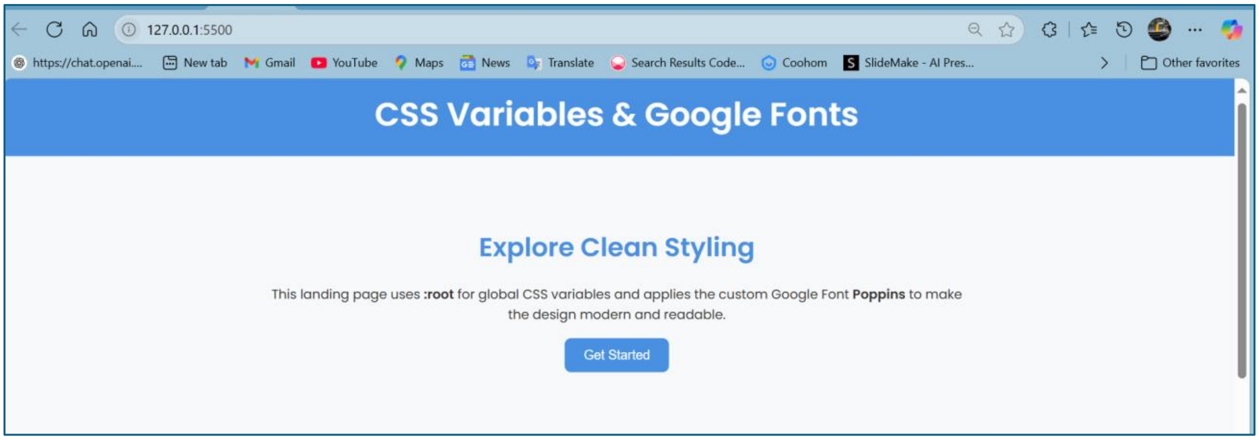
```css
.cards {
  display: flex;
  flex-wrap: wrap;
  gap: var(--spacing);
  margin-top: 40px;
}

.card {
  background-color: var(--card-bg);
  flex: 1 1 45%;
  box-shadow: var(--shadow);
  padding: var(--spacing);
  border-radius: var(--btn-radius);
}

.card h1,
.card h2 {
  color: var(--primary-color);
}

pre {
  background-color: #eceff1;
  padding: var(--spacing);
  border-radius: var(--btn-radius);
  overflow-x: auto;
}

code {
  font-family: monospace;
  color: var(--accent-color);
}

footer {
  text-align: center;
  padding: var(--spacing);
  background-color: var(--primary-color);
  color: white;
  margin-top: 60px;
}

@media (max-width: 768px) {
  .cards {
    flex-direction: column;
  }
```



```
      .card {
        flex: 1 1 100%;
      }
    }
  </style>
</head>
<body>

  <header>
    <h1>CSS Variables & Google Fonts</h1>
  </header>

  <div class="container">
    <!-- Hero Section -->
    <section class="hero">
      <h2>Explore Clean Styling</h2>
      <p>This landing page uses <strong>:root</strong> for global CSS
variables and applies the custom Google Font <strong>Poppins</strong> to make
the design modern and readable.</p>
      <button class="btn">Get Started</button>
    </section>

    <!-- ✅ Side-by-Side Boxes Section -->
    <div class="cards">
      <!-- Box 1: CSS Variables -->
      <div class="card">
        <h1>1. CSS Variables</h1>
        <p>CSS variables, also known as custom properties, are reusable values
defined in CSS using the <code>:root</code> selector.</p>
        <p>They follow the syntax <code>--variable-name: value;</code> and are
accessed using <code>var()</code>.</p>
        <h2>Example</h2>
        <pre><code>:root {
  --primary-color: #4a90e2;
  --text-color: #222;
}
body {
  background-color: var(--secondary-color);
  color: var(--text-color);
}</code></pre>
      </div>

      <!-- Box 2: Google Fonts -->
      <div class="card">
```

```
        <h1>2. Google Fonts</h1>
        <p>Google Fonts is a free web font library. You can include fonts in
your website by adding a link to the head of your HTML file.</p>
        <h2>Link Example</h2>
        <pre><code>&lt;link
href="https://fonts.googleapis.com/css2?family=Poppins&display=swap"
rel="stylesheet"&gt;</code></pre>
        <h2>CSS Usage</h2>
        <pre><code>body {
  font-family: 'Poppins', sans-serif;
}</code></pre>
      </div>
    </div>
  </div>

  <footer>
    <p>&copy; 2025 My Website. All rights reserved.</p>
  </footer>

</body>
</html>
```

Output:



CSS Variables & Google Fonts

Explore Clean Styling

This landing page uses **:root** for global CSS variables and applies the custom Google Font **Poppins** to make the design modern and readable.

Get Started

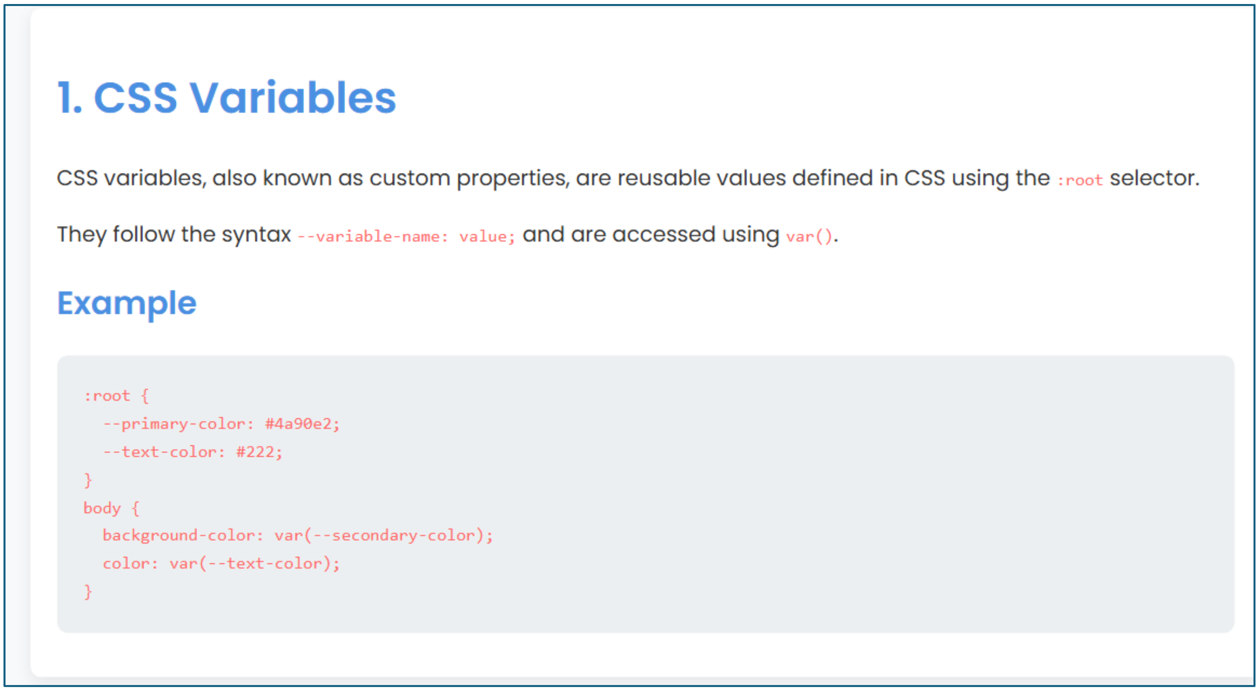
CSS Variables Implementation:

```
:root {
  --primary-color: #4a90e2;
  --secondary-color: #f8f9fa;

  --text-color: #333;

  --heading-font: 'Poppins', sans-serif;

}
```

These variables define the color scheme and font. Using :root ensures global availability across the page.

```
body {

  background-color: var(--secondary-color);

  color: var(--text-color);

  font-family: var(--heading-font);

}
```

Variables are reused using var() to apply consistent design styles.



1. CSS Variables

CSS variables, also known as custom properties, are reusable values defined in CSS using the `:root` selector.

They follow the syntax `--variable-name: value;` and are accessed using `var()`.

Example

```
:root {
  --primary-color: #4a90e2;
  --text-color: #222;
}
body {
  background-color: var(--secondary-color);
  color: var(--text-color);
}
```

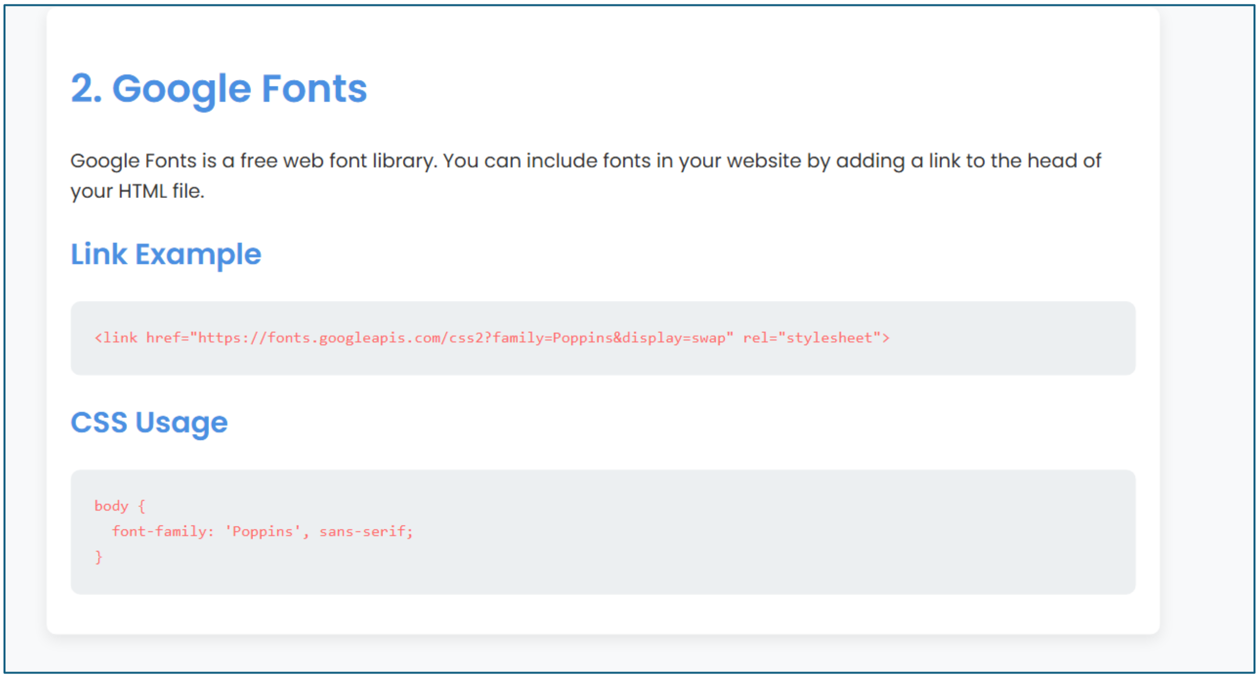
Google Fonts Integration:

```
<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600&display=swap" rel="stylesheet">
```

This imports the Poppins font from Google Fonts.

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

The font is applied globally to all text on the page, improving readability and modern appearance.



2. Google Fonts

Google Fonts is a free web font library. You can include fonts in your website by adding a link to the head of your HTML file.

Link Example

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

CSS Usage

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

Conclusion:

Today's concepts **CSS Variables** and **Custom Fonts** are vital for creating scalable and professional web designs. Variables help simplify and organize CSS, making large projects easier to maintain by allowing global control over colors, spacing, and typography. Meanwhile, custom fonts enhance the **visual personality** of a website, giving designers the freedom to match branding, tone, and style.

Mastering these tools ensures both **efficiency** in development and **elegance** in presentation.