1. **Enhance Visual Hierarchy and Navigation:**
    - **AI Action:**
        - Generate CSS code blocks to enforce a clear visual hierarchy.
            - Example: Use a function that takes heading level (1-6) as input and returns CSS with appropriate font-size, font-weight, and margin.
        - Create a reusable, responsive navigation component.
            - Example: Provide a data structure (JSON or object) representing the navigation menu, and have the AI generate the HTML, CSS, and JavaScript.
        - Implement accessible navigation.
            - Example: Generate ARIA attributes (aria-label, aria-current) dynamically based on context.
    - **Data Structure Example (for navigation):**

```
{
 "menuItems": [
  { "text": "Home", "url": "/", "active": true },
  {
   "text": "Products",
   "url": "/products",
   "children": [
    { "text": "Electronics", "url": "/products/electronics" },
    { "text": "Clothing", "url": "/products/clothing" }
   ]
  },
  { "text": "About Us", "url": "/about" }
 ]
}
```

2. **Optimize Visual Assets and Styling:**
    - **AI Action:**
        - Integrate with an image processing library (if available) to optimize images.
            - Example: Provide the AI with image paths, and have it generate code to resize, compress, and convert them (e.g., to WebP).
        - Generate CSS using variables or a preprocessor (like Sass) for consistency.
            - Example: Give the AI a color palette and font specifications, and have it create a CSS stylesheet.
        - Implement a responsive grid system.

        - Example: Provide the AI with parameters (number of columns, gap) and have it generate CSS grid code.
    - **Code Generation Example (CSS with variables):**

```
function generateCSS(colors, fonts) {
 let css = `:root {\n`;
 for (const [name, value] of Object.entries(colors)) {
  css += `  --${name}: ${value};\n`;
 }
 for (const [name, value] of Object.entries(fonts)) {
  css += `  --font-${name}: ${value};\n`;
 }
 css += `}\n`;
 // Add CSS rules using the variables
 css += `body { font-family: var(--font-primary); color: var(--text-color); }`;
 return css;
}
```

3. **Enhance Performance and Responsiveness:**
    - **AI Action:**
        - Inject loading="lazy" attribute into <img> tags.
            - Example: Have the AI parse HTML and modify image tags.
        - Integrate with a build tool (if applicable) to minify CSS and JavaScript.
            - Example: Provide the AI with file paths, and have it generate commands to run the minifier.
        - Add the viewport meta tag to HTML.
            - Example: Have the AI find the <head> tag in an HTML document and insert the meta tag.
    - **Code Modification Example (JavaScript):**

```
function addLazyLoad(html) {
 const parser = new DOMParser();
 const doc = parser.parseFromString(html, 'text/html');
 const images = doc.querySelectorAll('img');
 images.forEach(img => {
  if (!img.getAttribute('loading')) {
   img.setAttribute('loading', 'lazy');
  }
 });
 return doc.body.innerHTML; // Or doc.documentElement.outerHTML for the whole document
```

```
}
```

4. **Structure Content and Integrate CTAs:**
    - **AI Action:**
        - Generate semantic HTML5 markup for content sections.
            - Example: Provide the AI with a content outline, and have it create the HTML structure with <article>, <section>, etc.
        - Create reusable CTA button components.
            - Example: Define a CTA button style in CSS, and have the AI generate the HTML for buttons with that class.
        - Implement form validation.
            - Example: Provide the AI with form field definitions (name, type, validation rules), and have it generate the HTML and JavaScript for validation.
    - **Data Structure Example (for a form):**

```
{
  "fields": [
    { "name": "email", "type": "email", "label": "Email", "required": true },
    { "name": "password", "type": "password", "label": "Password", "required": true, "minLength": 8 },
    { "name": "submit", "type": "submit", "label": "Register" }
  ]
}
```

5. **Ensure Accessibility and SEO:**
    - **AI Action:**
        - Generate alt text for images.
            - Example: If you have an image analysis API, you could give the AI the image URL and have it fetch a description to use as alt text. Otherwise, prompt the user.
        - Add ARIA attributes for accessibility.
            - Example: Have the AI add aria-label to buttons and links based on their function.
        - Generate a sitemap.xml file.
            - Example: Provide the AI with a list of URLs, and have it format them correctly for the sitemap.
        - Implement structured data markup (JSON-LD).
            - Example: Provide the AI with data about the website or page (e.g., organization name, product information), and have it generate the

JSON-LD.

- **Code Generation Example (sitemap.xml):**

```
function generateSitemap(urls) {
  let xml = '<?xml version="1.0" encoding="UTF-8"?>\n';
  xml += '<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">\n';
  urls.forEach(url => {
    xml += `  <url>\n    <loc>${url}</loc>\n  </url>\n`;
  });
  xml += '</urlset>';
  return xml;
}
```