

# Lab 14: Web Design Application – AI-Assisted HTML/CSS/JS Generation

**Name:** YASHWANTH

**Enrollment Number:** 2503A51L42

**Assignment Number:** 14.4

## Task 1: Responsive Web Page Layout

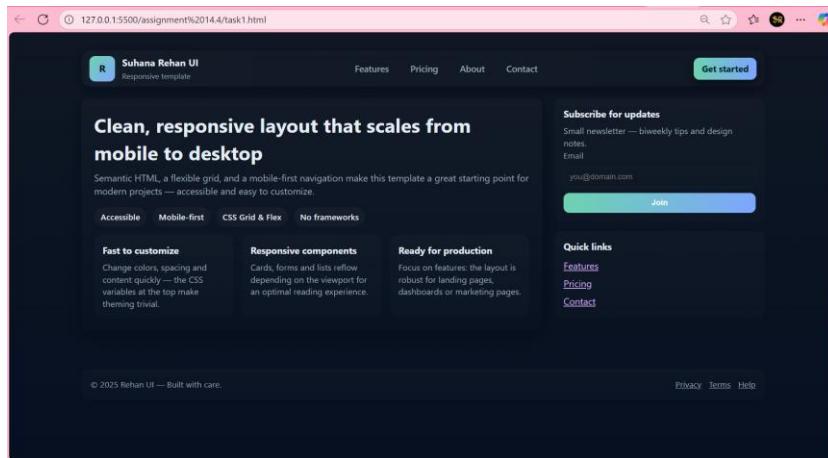
### Prompt:

Generate HTML and CSS for a responsive web page with a header, main content, and footer that works on desktop, tablet, and mobile.

### Code:

```
assignment 14.4 > task1.html > head > title
 5   <meta name="viewport" content="width=device-width,initial-scale=1" />
 6   <title>Responsive Web Page – Header · Main · Footer</title>
 7   <meta name="description" content="A clean responsive layout with header, main content, and footer." />
 8   <style>
 9     :root{
10       --bg:#0f1724; /* deep navy */
11       --card:#0b1220;
12       --muted:#9aa4b2;
13       --accent:#6dd3b0; /* teal-ish */
14       --accent-2:#7e6fff; /* soft blue */
15       --glass:rgba(255,255,255,0.03);
16       --radius:12px;
17       --max-width:1200px;
18       --gap:1.25rem;
19       font-family: Inter, ui-sans-serif, system-ui, -apple-system, "Segoe UI", Roboto, "Helvetica Neue", sans-serif;
20       color-scheme: dark;
21     }
22
23     /* Reset-ish */
24     *{box-sizing:border-box}
25     html,body{height:100%}
26     body{
27       margin:0;
28       background:linear-gradient(180deg,var(--bg),#071021);
29       color:#e6eff6;
30       -webkit-font-smoothing:antialiased;
31       -moz-osx-font-smoothing:grayscale;
32       line-height:1.45;
33       padding:2rem 1rem;
34     }
35
36     .container{
37       width:100%;
38       max-width:var(--max-width);
39       margin:0 auto;
40       padding:0 1rem;
41     }
42
43     header.site-header{
44       background:linear-gradient(90deg, rgba(255,255,255,0.02), rgba(255,255,255,0.01));
45       border-radius:calc(var(--radius) + 4px);
46       padding:0.6rem 0.9rem;
47       margin-bottom:1.25rem;
48       box-shadow: 0 6px 20px rgba(2,6,23,0.6);
49       position:sticky;
50       top:1rem;
51       z-index:50;
52     }
53
```

## Output:



## Observation:

- The layout looks neat and changes size smoothly on different screens.
- I learned how CSS media queries help in making pages responsive.
- It feels cool to see the same site fit my phone and laptop perfectly.

---

## Task 2: Interactive Button with JavaScript

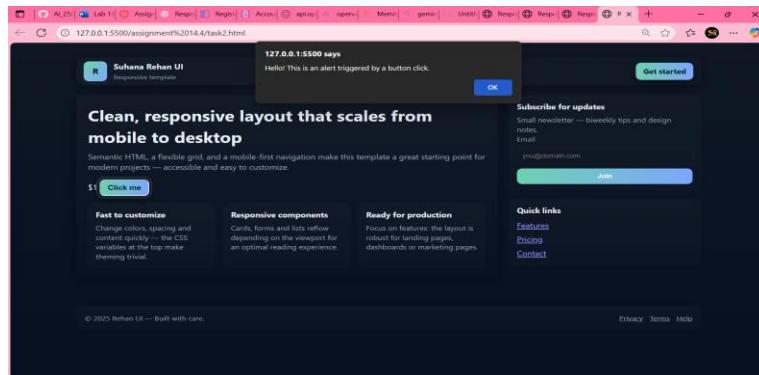
### Prompt:

Write JavaScript code that shows an alert message when a button is clicked, with comments explaining each step.

### Code:

```
assignment 14.4 > o task2.html > ⚡ html > ⚡ body > ⚡ div.container > ⚡ header.site-header > ⚡ div.nav-row > ⚡ div > ⚡ button.menu-button
151   </head>
152 <body>
153   <div class="container">
154     <header class="site-header" role="banner">
155       <div class="nav-row">
156         <a href="#" class="brand" aria-label="Homepage">
157           <div class="logo">R</div>
158           <div style="font-weight:700;letter-spacing:0.2px"> Suhana Rehan UI</div>
159           <div style="font-size:0.8rem;color:var(--muted);margin-top:2px">Responsive template</div>
160         </div>
161       </a>
162     </header>
163
164     <nav class="primary" role="navigation" aria-label="Primary">
165       <a href="#features">Features</a>
166       <a href="#pricing">Pricing</a>
167       <a href="#about">About</a>
168       <a href="#contact">Contact</a>
169     </nav>
170
171     <div style="display:flex;align-items:center;gap:0.6rem">
172       <a class="cta" href="#get-started">Get started</a>
173       <button class="menu-button" aria-expanded="false" aria-controls="primary-nav" aria-label="Toggle menu" data-mq="sm">
174         <!-- simple hamburger -->
175         <svg width="22" height="22" viewBox="0 0 24 24" fill="none" aria-hidden="true"><path stroke="currentColor" stroke-width="2" stroke-linecap="round" stroke-linejoin="round" d="M 12 2 L 12 22 L 22 22 M 12 22 L 12 22 L 22 22" /></svg>
176       </button>
177     </div>
178   </header>
179
180   <main>
181     <section class="hero">
182       <h1>Clean, responsive layout that scales from mobile to desktop</h1>
183       <p>Semantic HTML, a flexible grid, and a mobile-first navigation make this template a great starting point</p>
184       $1
185       <button id="alertBtn" class="cta" type="button">Click me</button>
186
187     <div class="grid-3">
188       <article class="card">
189         <h3>Fast to customize</h3>
190         <p>Change colors, spacing and content quickly – the CSS variables at the top make theming trivial.</p>
191       </article>
192
193       <article class="card">
194         <h3>Responsive components</h3>
195         <p>Cards, forms and lists reflow depending on the viewport for an optimal reading experience.</p>
196
197       </article>
198   </main>
```

## Output:



## Observation:

- The button actually reacts when I click it , that's fun!
  - JavaScript makes web pages feel alive and interactive.
  - I understood how addEventListener connects HTML and JS actions.
- 

## Task 3: Form with Validation

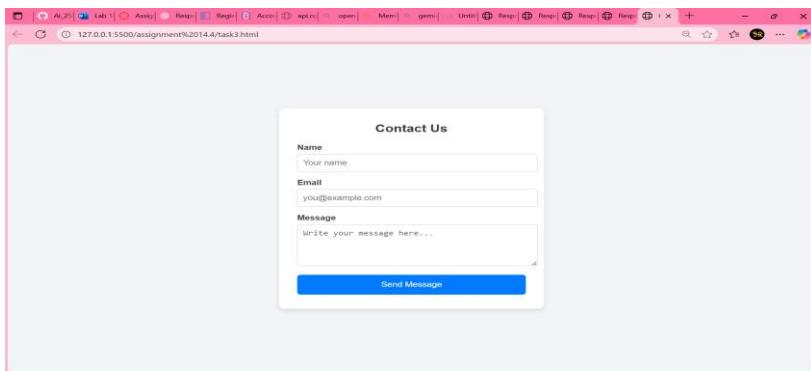
### Prompt:

Generate HTML, CSS, and JavaScript for a contact form with validation for name, email, and message fields with inline error messages.

### Code:

```
assignment 14.4 > <> task3.html > ⌂ html
 17   .contact-form {
 18     background: #fff;
 19     padding: 2rem;
 20     border-radius: 10px;
 21     box-shadow: 0 4px 10px rgba(0,0,0,0.1);
 22     width: 100%;
 23     max-width: 400px;
 24   }
 25   .contact-form h2 {
 26     margin-top: 0;
 27     text-align: center;
 28     color: #333;
 29   }
 30   .form-group {
 31     margin-bottom: 1rem;
 32   }
 33   label {
 34     display: block;
 35     margin-bottom: 0.3rem;
 36     font-weight: bold;
 37     color: #444;
 38   }
 39   input, textarea {
 40     width: 100%;
 41     padding: 0.6rem;
 42     border: 1px solid #ccc;
 43     border-radius: 6px;
 44     font-size: 1rem;
 45   }
 46   input:focus, textarea:focus {
 47     outline: none;
 48     border-color: #007bff;
 49     box-shadow: 0 0 4px rgba(0,123,255,0.2);
 50   }
 51   .error {
 52     color: red;
 53     font-size: 0.85rem;
 54     margin-top: 0.3rem;
 55   }
 56   button {
 57     width: 100%;
 58     background: #007bff;
 59     color: #fff;
 60     border: none;
 61     padding: 0.8rem;
 62     border-radius: 6px;
 63     font-size: 1rem;
 64     cursor: pointer;
 65   }
```

## Output:



A screenshot of a web browser window showing a contact form. The title of the form is "Contact Us". It contains three input fields: "Name" (with placeholder "Your name"), "Email" (with placeholder "you@example.com"), and "Message" (with placeholder "Write your message here..."). Below these fields is a blue "Send Message" button.

## Observation:

- The form doesn't submit unless all fields are correct ,that's helpful.
- I learned about regular expressions for checking email formats.
- It makes the page feel more professional and user friendly.

---

## Task 4: Dynamic Content Generation

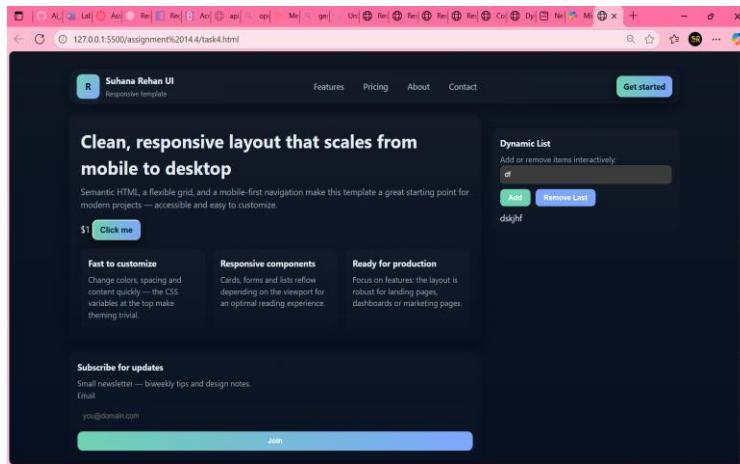
### Prompt:

Create JavaScript that dynamically adds or removes list items from an HTML list when buttons are clicked.

### Code:

```
assignment 14.4 > task4.html > html > body > script
  <p>Change colors, spacing and content quickly – the CSS variables
  </article>
  <article class="card">
    <h3>Responsive components</h3>
    <p>Cards, forms and lists reflow depending on the viewport for an
    </article>
  <article class="card">
    <h3>Ready for production</h3>
    <p>Focus on features: the layout is robust for landing pages, dash
    </article>
  </div>
  <section>
    <section class="card" style="margin-top:1.5rem">
      <h3>Dynamic List</h3>
      <p>Add or remove items interactively:</p>
      <input type="text" id="listItem" placeholder="Enter item" style=
        <div style="display:flex;gap:0.6rem;margin-bottom:0.6rem">
          <button onclick="#addListItem()" style="padding:0.5rem 1rem;border-
          <button onclick="#removeListItem()" style="padding:0.5rem 1rem;borde
        </div>
        <ul id="dynamicList" style="list-style:none;padding-left:0;margin:
          <li>Example item</li>
        </ul>
      </section>
    <aside class="sidebar">
      <div class="card">
        <h3 id="about">Subscribe for updates</h3>
        <p>Small newsletter – biweekly tips and design notes.</p>
        <form class="subscribe" onsubmit="event.preventDefault();alert('Than
          <label for="email" class="small">Email</label>
          <input id="email" type="email" placeholder="you@domain.com" aria-l
          <button type="submit">Join</button>
        </form>
      </div>
      <div style="height:1rem"></div>
      <div class="card">
        <h3>Quick links</h3>
        <ul style="list-style:none;padding:0;margin:0;display:flex;flex-dir
          <li><a href="#features">Features</a></li>
          <li><a href="#pricing">Pricing</a></li>
          <li><a href="#contact">Contact</a></li>
        </ul>
      </div>
    </aside>
  </main>
```

## Output:



## **Observation:**

- Clicking “Add Item” instantly shows a new item ,no refresh needed.
  - I learned that DOM manipulation can change web content in real time.
  - The appendChild() and removeChild() methods were simple but powerful.

## Task 5: Styled Modal Popup

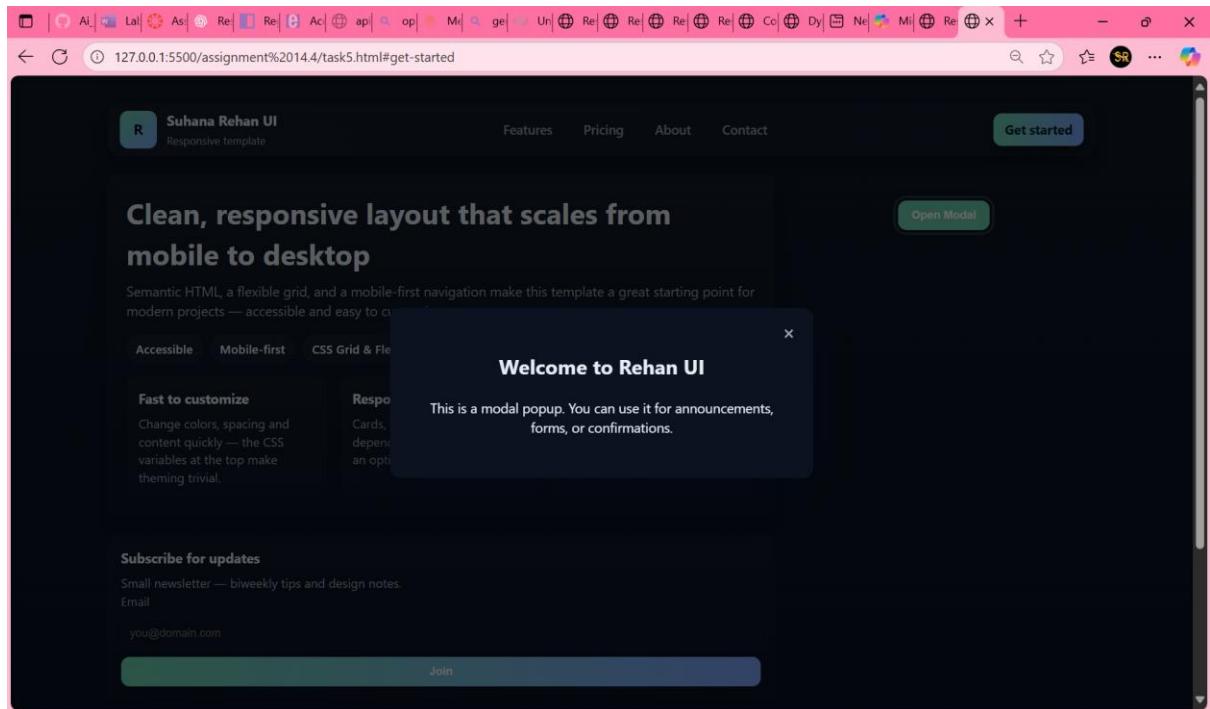
## Prompt:

Generate HTML, CSS, and JavaScript for a modal popup with overlay and close button that opens on button click.

## Code:

```
assignment 14.4 > task5.html > html > body > div.container > main
217     <h1>Clean, responsive layout that scales from mobile to desktop</h1>
218     <p>Semantic HTML, a flexible grid, and a mobile-first navigation make this template
219
220     <div class="features">
221         <span class="pill">Accessible</span>
222         <span class="pill">Mobile-first</span>
223         <span class="pill">CSS Grid & Flex</span>
224         <span class="pill">No frameworks</span>
225     </div>
226
227     <div class="grid-3">
228         <article class="card">
229             <h3>Fast to customize</h3>
230             <p>Change colors, spacing and content quickly – the CSS variables at the top make
231             </article>
232
233             <article class="card">
234                 <h3>Responsive components</h3>
235                 <p>Cards, forms and lists reflow depending on the viewport for an optimal reading
236             </article>
237
238             <article class="card">
239                 <h3>Ready for production</h3>
240                 <p>Focus on features: the layout is robust for landing pages, dashboards or mark
241             </article>
242         </div>
243     </section>
244
245     <!-- Modal Trigger Button -->
246     <div style="text-align:center; margin-top:2rem">
247         <button onclick="openModal()" style="padding:0.6rem 1rem; border-radius:10px; background-color:var
248
249     <!-- Modal Structure -->
250     <div id="modalOverlay" class="modal-overlay" role="dialog" aria-modal="true" aria-labelledby="modalTitle">
251         <div class="modal-content">
252             <button class="modal-close" onclick="closeModal()" aria-label="Close modal">&times;</button>
253             <h2 id="modalTitle">Welcome to Rehan UI</h2>
254             <p>This is a modal popup. You can use it for announcements, forms, or confirmations.</p>
255         </div>
256     </div>
257
258     <aside class="sidebar">
259         <div class="card">
260             <h3 id="about">Subscribe for updates</h3>
261             <p class="small">Small newsletter – biweekly tips and design notes.</p>
262             <form class="subscribe" onsubmit="event.preventDefault();alert('Thanks – this demo
263             <label for="email" class="small">Email:</label>
264             <input id="email" type="email" placeholder="you@domain.com" aria-label="Email ad
```

## Output:



## Observation:

- The popup appears nicely on top of the page when I click the button.
  - CSS overlay makes the background dim and the modal stand out.
  - Now I understand how modals can display messages or forms clearly.
-