

# 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

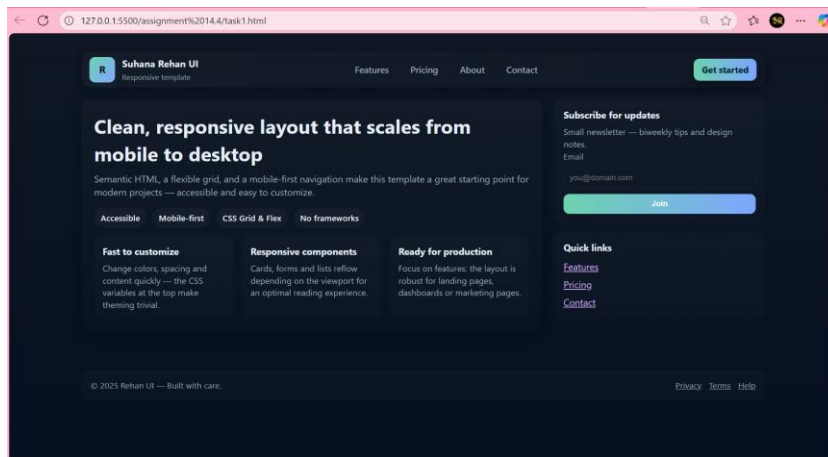
### Prompt:

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 > 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:#7ea6ff; /* 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:#e6eef6;
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

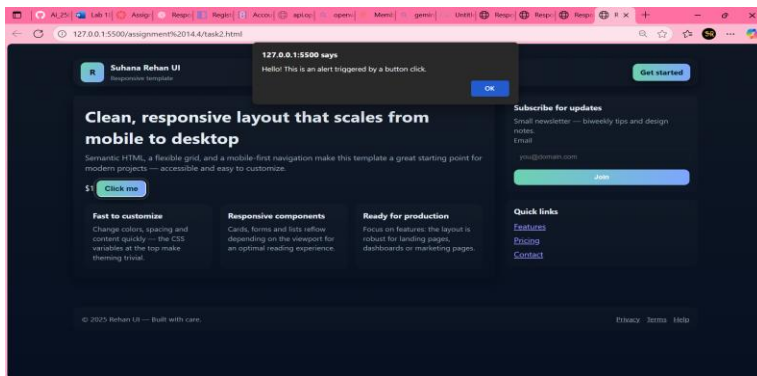
### Prompt:

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

### Code:

```
assignment 144 > 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>
159 <div style="font-weight:700;letter-spacing:0.2px"> Suhana Rehan UI</div>
160 <div style="font-size:0.8rem;color:var(--muted);margin-top:2px">Responsive template</div>
161 </div>
162 </a>
163 <nav class="primary" role="navigation" aria-label="Primary">
164 <a href="#features">Features</a>
165 <a href="#pricing">Pricing</a>
166 <a href="#about">About</a>
167 <a href="#contact">Contact</a>
168 </nav>
169 <div style="display:flex;align-items:center;gap:0.6rem">
170 <a class="cta" href="#get-started">Get started</a>
171 <button class="menu-button" aria-expanded="false" aria-controls="primary-nav" aria-label="Toggle menu">
172 <!-- Simple hamburger -->
173 <svg width="22" height="22" viewBox="0 0 24 24" fill="none" aria-hidden="true"><path stroke="currentCo
174 </button>
175 </div>
176 </div>
177 </div>
178 </div>
179 </div>
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
184 </p>
185 <div>
186 <div class="card">
187 <h3>Fast to customize</h3>
188 <p>Change colors, spacing and content quickly — the CSS variables at the top make theming trivial.</p>
189 </div>
190 <div class="card">
191 <h3>Responsive components</h3>
192 <p>Cards, forms and lists reflow depending on the viewport for an optimal reading experience.</p>
193 </div>
194 </div>
195 </div>
196 </div>
197 </div>
198 </div>
```

## 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

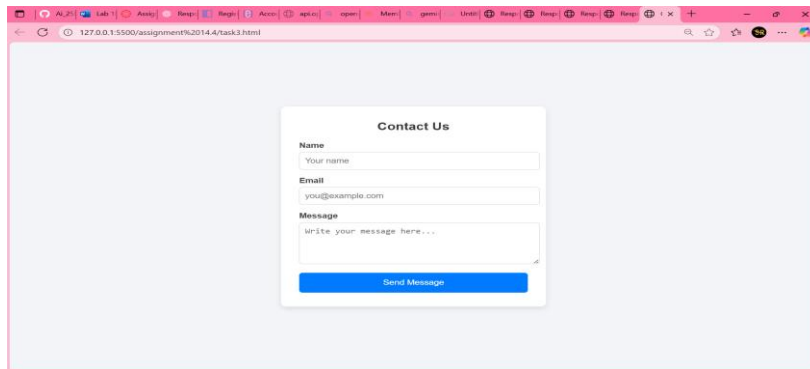
### Prompt:

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
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
- contact-form {
  background: #fff;
  padding: 2rem;
  border-radius: 10px;
  box-shadow: 0 4px 10px rgba(0,0,0,0.1);
  width: 100%;
  max-width: 400px;
}
- contact-form h2 {
  margin-top: 0;
  text-align: center;
  color: #333;
}
- form-group {
  margin-bottom: 1rem;
}
label {
  display: block;
  margin-bottom: 0.3rem;
  font-weight: bold;
  color: #444;
}
input, textarea {
  width: 100%;
  padding: 0.6rem;
  border: 1px solid #ccc;
  border-radius: 6px;
  font-size: 1rem;
}
input:focus, textarea:focus {
  outline: none;
  border-color: #007bff;
  box-shadow: 0 0 4px rgba(0,123,255,0.2);
}
.error {
  color: red;
  font-size: 0.85rem;
  margin-top: 0.3rem;
}
button {
  width: 100%;
  background: #007bff;
  color: #fff;
  border: none;
  padding: 0.8rem;
  border-radius: 6px;
  font-size: 1rem;
  cursor: pointer;
}
```

## Output:



## 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

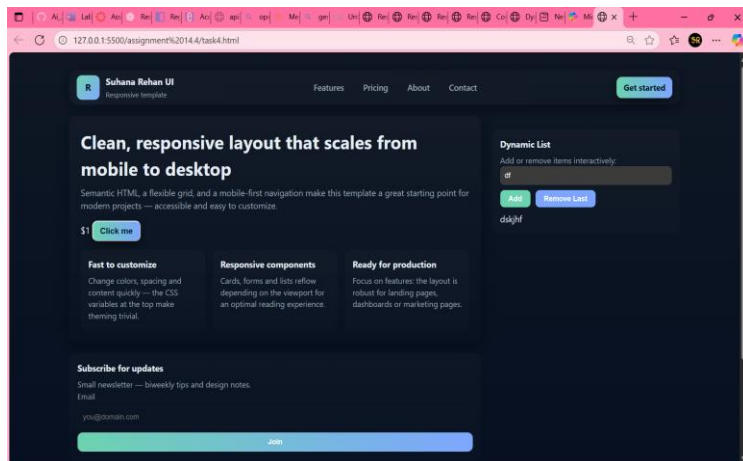
### Prompt:

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
193 <p>Change colors, spacing and content quickly – the CSS variables
194 </article>
195
196 <article class="card">
197 <h3>Responsive components</h3>
198 <p>Cards, forms and lists reflow depending on the viewport for an
199 </article>
200
201 <article class="card">
202 <h3>Ready for production</h3>
203 <p>Focus on features: the layout is robust for landing pages, dash
204 </article>
205 </div>
206
207 <section>
208 <section class="card" style="margin-top:1.5rem">
209 <h3>Dynamic List</h3>
210 <p class="small">Add or remove items interactively:</p>
211 <input type="text" id="listInput" placeholder="Enter item" style=
212 <div style="display:flex;gap:0.6rem;margin-bottom:0.6rem">
213 <button onclick="addListItem()" style="padding:0.5rem 1rem;border=
214 <button onclick="removeListItem()" style="padding:0.5rem 1rem;bord
215 </div>
216 <ul id="dynamiclist" style="list-style:none;padding-left:0;margin:
217 <li>Example item</li>
218 </ul>
219 </section>
220
221 <aside class="sidebar">
222 <div class="card">
223 <h3 id="about">Subscribe for updates</h3>
224 <p class="small">Small newsletter – biweekly tips and design notes.<
225 <form class="subscribe" onsubmit="event.preventDefault();alert('Than
226 <label for="email" class="small">Email</label>
227 <input id="email" type="email" placeholder="you@domain.com" aria-l
228 <button type="submit">Join</button>
229 </form>
230 </div>
231 <div style="height:1rem"></div>
232
233 <div class="card">
234 <h3>Quick links</h3>
235 <ul style="list-style:none;padding:0;margin:0;display:flex;flex-dire
236 <li><a href="#features">Features</a></li>
237 <li><a href="#pricing">Pricing</a></li>
238 <li><a href="#contact">Contact</a></li>
239 </ul>
240 </div>
241 </aside>
242 </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

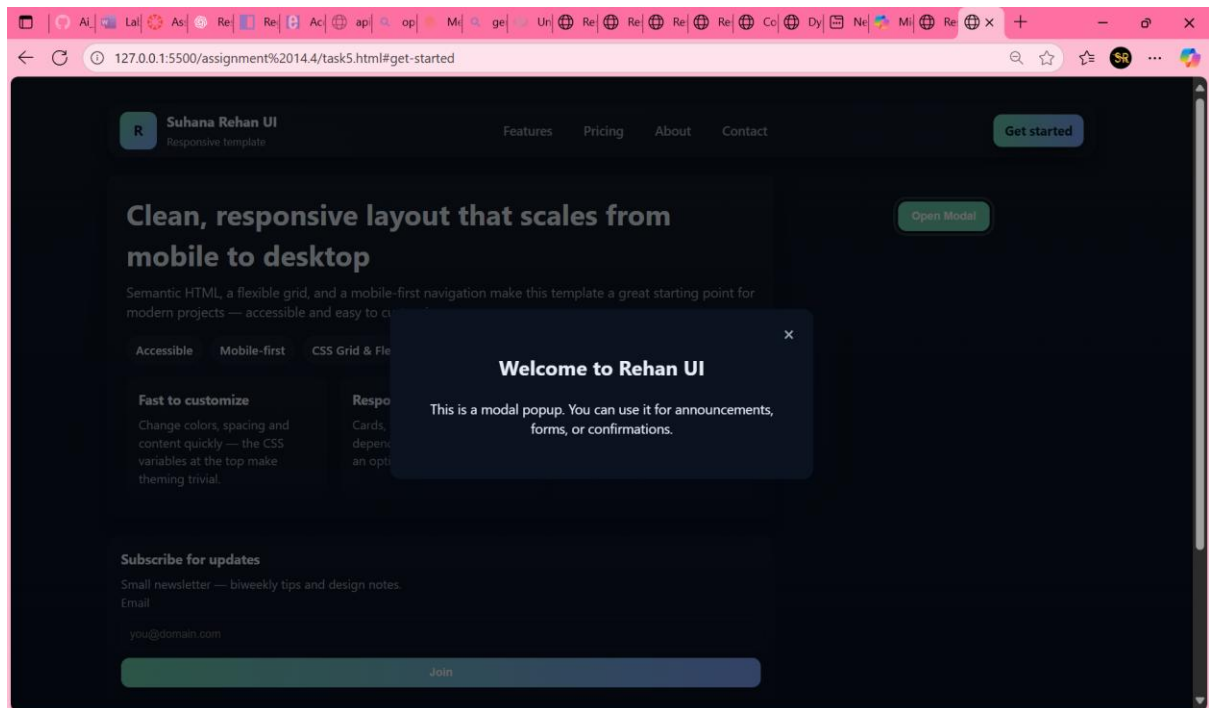
### Prompt:

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 a great starting point for
219 modern projects — accessible and easy to customize.
220
221 <div class="features">
222   <span class="pill">Accessible</span>
223   <span class="pill">Mobile-first</span>
224   <span class="pill">CSS Grid & Flex</span>
225   <span class="pill">No frameworks</span>
226 </div>
227
228 <div class="grid-3">
229   <article class="card">
230     <h3>Fast to customize</h3>
231     <p>Change colors, spacing and content quickly — the CSS variables at the top make
232   </article>
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   <article class="card">
238     <h3>Ready for production</h3>
239     <p>Focus on features: the layout is robust for landing pages, dashboards or mark
240   </article>
241 </div>
242 </section>
243
244 <!-- Modal Trigger Button -->
245 <div style="text-align:center;margin-top:2rem">
246   <button onclick="openModal()" style="padding:0.6rem 1rem;border-radius:10px;background:var
247 </div>
248
249 <!-- Modal Structure -->
250 <div id="modalOverlay" class="modal-overlay" role="dialog" aria-modal="true" aria-labelledby=
251   <div class="modal-content">
252     <button class="modal-close" onclick="closeModal()" aria-label="Close modal">&times;</but
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.
-