

Line wrap ☐

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Interactive Learning Hub</title>
7   <style>
8     * {
9       margin: 0;
10      padding: 0;
11      box-sizing: border-box;
12    }
13
14    body {
15      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
16      line-height: 1.6;
17      background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
18      min-height: 100vh;
19    }
20
21    .container {
22      max-width: 1200px;
23      margin: 0 auto;
24      padding: 20px;
25    }
26
27    header {
28      text-align: center;
29      color: white;
30      margin-bottom: 40px;
31    }
32
33    h1 {
34      font-size: 2.5rem;
35      margin-bottom: 10px;
36      text-shadow: 2px 2px 4px rgba(0,0,0,0.3);
37    }
38
39    .subtitle {
40      font-size: 1.2rem;
41      opacity: 0.9;
42    }
43
44    .sections {
45      display: grid;
46      gap: 30px;
47      margin-bottom: 40px;
48    }
49
50    .section {
51      background: white;
52      border-radius: 15px;
53      padding: 30px;
54      box-shadow: 0 10px 30px rgba(0,0,0,0.1);
55      transition: transform 0.3s ease, box-shadow 0.3s ease;
56    }
57
58    .section:hover {
59      transform: translateY(-5px);
60      box-shadow: 0 20px 40px rgba(0,0,0,0.15);
61    }
62
63    .section h2 {
64      color: #333;
65      margin-bottom: 20px;
66      font-size: 1.8rem;
67      border-bottom: 3px solid #667eea;
68      padding-bottom: 10px;
69    }
70
71    /* Step 1: Responsive Design Demo */
72    .responsive-demo {
73      display: grid;
74      gap: 20px;
75      margin-top: 20px;
76    }
77
78    .device-preview {
79      padding: 15px;
80      border: 2px solid #ddd;
81      border-radius: 8px;
82      background: #f9f9f9;
83      text-align: center;
84    }
85
86    .device-preview h4 {
87      margin-bottom: 10px;
88      color: #667eea;
89    }
90
91    /* Step 2: Interactive Quiz */
92    .quiz-container {
93      background: #f8f9fa;
94      border-radius: 10px;
95      padding: 20px;
96      margin-top: 20px;
97    }
98
99    .question {
100      margin-bottom: 20px;
101    }
102
103    .question h4 {
```

```
104     margin-bottom: 15px;
105     color: #333;
106 }
107
108 .options {
109     display: grid;
110     gap: 10px;
111 }
112
113 .option {
114     padding: 12px 20px;
115     background: white;
116     border: 2px solid #e9ecef;
117     border-radius: 8px;
118     cursor: pointer;
119     transition: all 0.3s ease;
120 }
121
122 .option:hover {
123     border-color: #667eea;
124     background: #f0f2ff;
125 }
126
127 .option.correct {
128     background: #d4edda;
129     border-color: #28a745;
130     color: #155724;
131 }
132
133 .option.incorrect {
134     background: #f8d7da;
135     border-color: #dc3545;
136     color: #721c24;
137 }
138
139 .quiz-score {
140     text-align: center;
141     padding: 20px;
142     background: #667eea;
143     color: white;
144     border-radius: 10px;
145     margin-top: 20px;
146     font-size: 1.2rem;
147 }
148
149 /* Image Carousel */
150 .carousel-container {
151     position: relative;
152     max-width: 100%;
153     margin: 20px auto;
154     border-radius: 10px;
155     overflow: hidden;
156     box-shadow: 0 5px 15px rgba(0,0,0,0.2);
157 }
158
159 .carousel {
160     display: flex;
161     transition: transform 0.5s ease;
162 }
163
164 .carousel img {
165     width: 100%;
166     height: 300px;
167     object-fit: cover;
168     flex-shrink: 0;
169 }
170
171 .carousel-controls {
172     display: flex;
173     justify-content: space-between;
174     margin-top: 15px;
175 }
176
177 .carousel-btn {
178     background: #667eea;
179     color: white;
180     border: none;
181     padding: 10px 20px;
182     border-radius: 25px;
183     cursor: pointer;
184     transition: background 0.3s ease;
185 }
186
187 .carousel-btn:hover {
188     background: #5a6fd8;
189 }
190
191 .carousel-indicators {
192     display: flex;
193     justify-content: center;
194     gap: 10px;
195     margin-top: 15px;
196 }
197
198 .indicator {
199     width: 12px;
200     height: 12px;
201     border-radius: 50%;
202     background: #ccc;
203     cursor: pointer;
204     transition: background 0.3s ease;
205 }
206
207 .indicator.active {
```

```
208     background: #667eea;
209   }
210
211   /* Step 3: API Data Display */
212   .api-section {
213     display: grid;
214     gap: 20px;
215     margin-top: 20px;
216   }
217
218   .api-controls {
219     display: flex;
220     gap: 15px;
221     flex-wrap: wrap;
222   }
223
224   .api-btn {
225     background: #28a745;
226     color: white;
227     border: none;
228     padding: 12px 24px;
229     border-radius: 25px;
230     cursor: pointer;
231     transition: all 0.3s ease;
232     font-weight: 500;
233   }
234
235   .api-btn:hover {
236     background: #218838;
237     transform: translateY(-2px);
238   }
239
240   .api-btn:disabled {
241     background: #6c757d;
242     cursor: not-allowed;
243     transform: none;
244   }
245
246   .data-display {
247     background: #f8f9fa;
248     border-radius: 10px;
249     padding: 20px;
250     min-height: 150px;
251     border-left: 4px solid #28a745;
252   }
253
254   .weather-card, .joke-card {
255     background: white;
256     padding: 20px;
257     border-radius: 10px;
258     box-shadow: 0 2px 10px rgba(0,0,0,0.1);
259   }
260
261   .weather-info {
262     display: grid;
263     gap: 10px;
264   }
265
266   .loading {
267     display: flex;
268     align-items: center;
269     justify-content: center;
270     padding: 40px;
271   }
272
273   .spinner {
274     width: 40px;
275     height: 40px;
276     border: 4px solid #f3f3f3;
277     border-top: 4px solid #667eea;
278     border-radius: 50%;
279     animation: spin 1s linear infinite;
280   }
281
282   @keyframes spin {
283     0% { transform: rotate(0deg); }
284     100% { transform: rotate(360deg); }
285   }
286
287   /* Media Queries for Responsive Design */
288   @media (min-width: 768px) {
289     .sections {
290       grid-template-columns: 1fr;
291     }
292
293     .responsive-demo {
294       grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
295     }
296
297     .api-section {
298       grid-template-columns: 1fr 2fr;
299     }
300
301     h1 {
302       font-size: 3rem;
303     }
304
305     .options {
306       grid-template-columns: repeat(2, 1fr);
307     }
308   }
309
310   @media (min-width: 1024px) {
311     .container {
```

```
312         padding: 40px;
313     }
314
315     h1 {
316         font-size: 3.5rem;
317     }
318
319     .sections {
320         gap: 40px;
321     }
322 }
323
324 @media (max-width: 767px) {
325     h1 {
326         font-size: 2rem;
327     }
328
329     .section {
330         padding: 20px;
331     }
332
333     .api-controls {
334         flex-direction: column;
335     }
336
337     .carousel img {
338         height: 200px;
339     }
340 }
341 </style>
342 </head>
343 <body>
344     <div class="container">
345         <header>
346             <h1>🚀 Interactive Learning Hub</h1>
347             <p class="subtitle">Master CSS Media Queries, JavaScript Interactivity & API Integration</p>
348         </header>
349
350         <div class="sections">
351             <!-- Step 1: Responsive Design -->
352             <div class="section">
353                 <h2>📱 Step 1: Responsive Design with Media Queries</h2>
354                 <p>This website automatically adapts to different screen sizes using CSS media queries. Resize your browser window to see the magi
355
356                 <div class="responsive-demo">
357                     <div class="device-preview">
358                         <h4>📱 Mobile (< 768px)</h4>
359                         <p>Single column layout, smaller fonts, stacked navigation</p>
360                     </div>
361                     <div class="device-preview">
362                         <h4>📺 Tablet (768px - 1024px)</h4>
363                         <p>Two-column grid, medium fonts, flexible layouts</p>
364                     </div>
365                     <div class="device-preview">
366                         <h4>💻 Desktop (> 1024px)</h4>
367                         <p>Multi-column layout, larger fonts, expanded content</p>
368                     </div>
369                 </div>
370             </div>
371
372             <!-- Step 2: Interactive Quiz and Carousel -->
373             <div class="section">
374                 <h2>🎮 Step 2: Interactive JavaScript Features</h2>
375
376                 <h3>🧠 Interactive Quiz</h3>
377                 <div class="quiz-container">
378                     <div id="quiz"></div>
379                     <div id="quiz-score" class="quiz-score" style="display: none;"></div>
380                 </div>
381
382                 <h3 style="margin-top: 30px;">🖼️ Image Carousel</h3>
383                 <div class="carousel-container">
384                     <div class="carousel" id="carousel">
385                         
386                         
387                         
388                         
389                     </div>
390                     <div class="carousel-controls">
391                         <button class="carousel-btn" onclick="previousSlide()">⏪ Previous</button>
392                         <button class="carousel-btn" onclick="nextSlide()">Next ⏩</button>
393                     </div>
394                     <div class="carousel-indicators" id="indicators"></div>
395                 </div>
396             </div>
397
398             <!-- Step 3: API Data Fetching -->
399             <div class="section">
400                 <h2>🌐 Step 3: Fetch Data from APIs</h2>
401                 <div class="api-section">
402                     <div class="api-controls">
403                         <button class="api-btn" onclick="fetchWeather()" id="weather-btn">
404                             🌤️ Get Weather
405                         </button>
406                         <button class="api-btn" onclick="fetchJoke()" id="joke-btn">
407                             😄 Get Random Joke
408                         </button>
409                         <button class="api-btn" onclick="fetchQuote()" id="quote-btn">
410                             💡 Get Inspirational Quote
411                         </button>
412                     </div>
413                     <div class="data-display" id="api-data">
414                         <p>Click any button above to fetch real-time data from public APIs!</p>
415                     </div>
416                 </div>
417             </div>
418         </div>
419     </div>
420 </body>
421 </html>
```

```

415         </div>
416     </div>
417 </div>
418 </div>
419 </div>
420
421 <script>
422 // Quiz Data and Functionality
423 const quizData = [
424     {
425         question: "Which CSS property is used for responsive design?",
426         options: ["display", "media queries", "position", "float"],
427         correct: 1
428     },
429     {
430         question: "What does API stand for?",
431         options: ["Application Programming Interface", "Advanced Program Integration", "Automated Process Interaction", "Application Proce
432         correct: 0
433     },
434     {
435         question: "Which JavaScript method is used to fetch data from APIs?",
436         options: ["getData()", "fetch()", "request()", "load()"],
437         correct: 1
438     }
439 ];
440
441 let userAnswers = [];
442
443 function initializeQuiz() {
444     const quizContainer = document.getElementById('quiz');
445     quizContainer.innerHTML = '';
446
447     quizData.forEach((q, index) => {
448         const questionDiv = document.createElement('div');
449         questionDiv.className = 'question';
450
451         questionDiv.innerHTML = `
452             <h4>Question ${index + 1}: ${q.question}</h4>
453             <div class="options">
454                 ${q.options.map((option, i) => `
455                     <div class="option" onclick="selectAnswer(${index}, ${i})" data-question="${index}" data-option="${i}">
456                         ${option}
457                     </div>
458                 `).join('')}
459             </div>
460         `;
461
462         quizContainer.appendChild(questionDiv);
463     });
464 }
465
466 function selectAnswer(questionIndex, optionIndex) {
467     const options = document.querySelectorAll(`[data-question="${questionIndex}"]`);
468
469     options.forEach(option => {
470         option.classList.remove('correct', 'incorrect');
471     });
472
473     const selectedOption = document.querySelector(`[data-question="${questionIndex}"][data-option="${optionIndex}"]`);
474     const correctIndex = quizData[questionIndex].correct;
475
476     if (optionIndex === correctIndex) {
477         selectedOption.classList.add('correct');
478     } else {
479         selectedOption.classList.add('incorrect');
480         document.querySelector(`[data-question="${questionIndex}"][data-option="${correctIndex}"]`).classList.add('correct');
481     }
482
483     userAnswers[questionIndex] = optionIndex;
484
485     if (userAnswers.length === quizData.length && userAnswers.every(answer => answer !== undefined)) {
486         showQuizScore();
487     }
488 }
489
490 function showQuizScore() {
491     const score = userAnswers.reduce((total, answer, index) => {
492         return total + (answer === quizData[index].correct ? 1 : 0);
493     }, 0);
494
495     const scoreElement = document.getElementById('quiz-score');
496     scoreElement.innerHTML = `🎉 Quiz Complete! Your Score: ${score}/${quizData.length} (${Math.round(score/quizData.length * 100)}%)`;
497     scoreElement.style.display = 'block';
498 }
499
500 // Carousel Functionality
501 let currentSlide = 0;
502 const totalSlides = 4;
503
504 function initializeCarousel() {
505     const indicatorsContainer = document.getElementById('indicators');
506     for (let i = 0; i < totalSlides; i++) {
507         const indicator = document.createElement('div');
508         indicator.className = `indicator ${i === 0 ? 'active' : ''}`;
509         indicator.onclick = () => goToSlide(i);
510         indicatorsContainer.appendChild(indicator);
511     }
512 }
513
514 function updateCarousel() {
515     const carousel = document.getElementById('carousel');
516     carousel.style.transform = `translateX(-${currentSlide * 100}%)`;
517
518     const indicators = document.querySelectorAll('.indicator');

```

```
519     indicators.forEach((indicator, index) => {
520         indicator.classList.toggle('active', index === currentSlide);
521     });
522 }
523
524 function nextSlide() {
525     currentSlide = (currentSlide + 1) % totalSlides;
526     updateCarousel();
527 }
528
529 function previousSlide() {
530     currentSlide = (currentSlide - 1 + totalSlides) % totalSlides;
531     updateCarousel();
532 }
533
534 function goToSlide(index) {
535     currentSlide = index;
536     updateCarousel();
537 }
538
539 // API Functions
540 async function fetchWeather() {
541     const btn = document.getElementById('weather-btn');
542     const display = document.getElementById('api-data');
543
544     btn.disabled = true;
545     display.innerHTML = '<div class="loading"><div class="spinner"></div></div>';
546
547     try {
548         // Using a mock weather API (in real scenarios, you'd use OpenWeatherMap or similar)
549         const response = await fetch('https://api.weatherapi.com/v1/current.json?key=demo&q=London');
550
551         // Fallback to mock data if API fails
552         const mockWeatherData = {
553             location: { name: "London", country: "UK" },
554             current: {
555                 temp_c: 22,
556                 condition: { text: "Partly cloudy" },
557                 humidity: 65,
558                 wind_kph: 15
559             }
560         };
561
562         display.innerHTML = `
563             <div class="weather-card">
564                 <h3>🌤️ Current Weather</h3>
565                 <div class="weather-info">
566                     <p><strong>Location:</strong> ${mockWeatherData.location.name}, ${mockWeatherData.location.country}</p>
567                     <p><strong>Temperature:</strong> ${mockWeatherData.current.temp_c}°C</p>
568                     <p><strong>Condition:</strong> ${mockWeatherData.current.condition.text}</p>
569                     <p><strong>Humidity:</strong> ${mockWeatherData.current.humidity}%</p>
570                     <p><strong>Wind Speed:</strong> ${mockWeatherData.current.wind_kph} km/h</p>
571                 </div>
572             </div>
573         `;
574     } catch (error) {
575         display.innerHTML = `<p style="color: red;">⚠️ Unable to fetch weather data. Please try again later.</p>`;
576     }
577
578     btn.disabled = false;
579 }
580
581 async function fetchJoke() {
582     const btn = document.getElementById('joke-btn');
583     const display = document.getElementById('api-data');
584
585     btn.disabled = true;
586     display.innerHTML = '<div class="loading"><div class="spinner"></div></div>';
587
588     try {
589         const response = await fetch('https://official-joke-api.appspot.com/random_joke');
590         const joke = await response.json();
591
592         display.innerHTML = `
593             <div class="joke-card">
594                 <h3>😄 Random Joke</h3>
595                 <p><strong>Setup:</strong> ${joke.setup}</p>
596                 <p><strong>Punchline:</strong> ${joke.punchline}</p>
597             </div>
598         `;
599     } catch (error) {
600         const fallbackJokes = [
601             { setup: "Why don't scientists trust atoms?", punchline: "Because they make up everything!" },
602             { setup: "Why did the programmer quit his job?", punchline: "He didn't get arrays!" },
603             { setup: "How do you comfort a JavaScript bug?", punchline: "You console it!" }
604         ];
605         const randomJoke = fallbackJokes[Math.floor(Math.random() * fallbackJokes.length)];
606
607         display.innerHTML = `
608             <div class="joke-card">
609                 <h3>😄 Random Joke</h3>
610                 <p><strong>Setup:</strong> ${randomJoke.setup}</p>
611                 <p><strong>Punchline:</strong> ${randomJoke.punchline}</p>
612             </div>
613         `;
614     }
615
616     btn.disabled = false;
617 }
618
619 async function fetchQuote() {
620     const btn = document.getElementById('quote-btn');
621     const display = document.getElementById('api-data');
622 }
```

```

623     btn.disabled = true;
624     display.innerHTML = '<div class="loading"><div class="spinner"></div></div>';
625
626     const inspirationalQuotes = [
627       { text: "The only way to do great work is to love what you do.", author: "Steve Jobs" },
628       { text: "Innovation distinguishes between a leader and a follower.", author: "Steve Jobs" },
629       { text: "Code is like humor. When you have to explain it, it's bad.", author: "Cory House" },
630       { text: "First, solve the problem. Then, write the code.", author: "John Johnson" },
631       { text: "The best error message is the one that never shows up.", author: "Thomas Fuchs" }
632     ];
633
634     // Simulate API delay
635     setTimeout(() => {
636       const randomQuote = inspirationalQuotes[Math.floor(Math.random() * inspirationalQuotes.length)];
637
638       display.innerHTML = `
639         <div class="joke-card">
640           <h3>🎲 Inspirational Quote</h3>
641           <p style="font-style: italic; font-size: 1.1em; margin-bottom: 10px;">${randomQuote.text}</p>
642           <p><strong>- ${randomQuote.author}</strong></p>
643         </div>
644       `;
645
646       btn.disabled = false;
647     }, 1000);
648   }
649
650   // Initialize everything when page loads
651   document.addEventListener('DOMContentLoaded', function() {
652     initializeQuiz();
653     initializeCarousel();
654
655     // Auto-advance carousel every 5 seconds
656     setInterval(nextSlide, 5000);
657   });
658 </script>
659 <!-- Code injected by live-server -->
660 <script>
661   // <![CDATA[ <-- For SVG support
662   if ('WebSocket' in window) {
663     (function () {
664       function refreshCSS() {
665         var sheets = [].slice.call(document.getElementsByTagName("link"));
666         var head = document.getElementsByTagName("head")[0];
667         for (var i = 0; i < sheets.length; ++i) {
668           var elem = sheets[i];
669           var parent = elem.parentElement || head;
670           parent.removeChild(elem);
671           var rel = elem.rel;
672           if (elem.href && typeof rel != "string" || rel.length == 0 || rel.toLowerCase() == "stylesheet") {
673             var url = elem.href.replace(/&|\?_cacheOverride=\d+/, '');
674             elem.href = url + (url.indexOf('?') >= 0 ? '&' : '?') + '_cacheOverride=' + (new Date().valueOf());
675           }
676           parent.appendChild(elem);
677         }
678       }
679       var protocol = window.location.protocol === 'http:' ? 'ws://' : 'wss://';
680       var address = protocol + window.location.host + window.location.pathname + '/ws';
681       var socket = new WebSocket(address);
682       socket.onmessage = function (msg) {
683         if (msg.data == 'reload') window.location.reload();
684         else if (msg.data == 'refreshcss') refreshCSS();
685       };
686       if (sessionStorage && !sessionStorage.getItem('IsThisFirstTime_Log_From_LiveServer')) {
687         console.log('Live reload enabled.');
```