

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for contents of text-input area

Checker Input

Show ☒ source ☐ outline ☐ image report [Options...](#)

Check by text input ☐ CSS

```

        e.preventDefault();
        fetchBooks(true);
    });
    genreSelect.addEventListener('change', () => fetchBooks(true));
    sortSelect.addEventListener('change', () => fetchBooks(true));
    loadMoreBtn.addEventListener('click', () => fetchBooks());

    // Initial load
    fetchBooks(true);
  });
</script>

</body>

```

[Check](#)

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

[Message Filtering](#)

Document checking completed. No errors or warnings to show.

Source

```

1.  ↵
2.  ↵
3.  <!DOCTYPE html>↵
4.  <html lang="en">↵
5.  ↵
6.  <head>↵
7.    <meta charset="UTF-8">↵
8.    <title>BookBase</title>↵
9.    <meta name="viewport" content="width=device-width, initial-scale=1">↵
10. ↵
11.  <!-- Favicons / CSS -->↵
12.  <link rel="icon" href="/static/img/favicon_bookbase.ico" type="image/x-icon">↵
13.  <link rel="preconnect" href="https://fonts.googleapis.com">↵
14.  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>↵
15.  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">↵
16.  <link rel="stylesheet" href="/static/books/css/style.css">↵
17.  <style>↵
18.    .brand-logo {↵
19.      transform: translateY(-2px);↵
20.    }↵
21.  ↵
22.    h1 {↵
23.      font-family: 'Great Vibes', cursive;↵
24.      font-weight: 500;↵
25.      letter-spacing: 0px;↵
26.      font-size: 2.5rem;↵
27.    }↵
28.  </style>↵
29. </head>↵
30. ↵
31. <body class="d-flex flex-column min-vh-100">↵
32. ↵
33.  <!-- Fixed Top Navbar -->↵
34.  <nav class="navbar navbar-expand-lg navbar-light bg-light border-bottom shadow-sm fixed-top">↵
35.    <div class="container-fluid d-flex align-items-center justify-content-between">↵
36. ↵
37.      <!-- Left: Logo -->↵
38.      <a class="navbar-brand me-3" href="/">↵
39.        ↵
40.      </a>↵
41. ↵
42.      <!-- Center: Tagline (hidden on < lg) -->↵
43.      <div class="d-none d-lg-block flex-grow-1 text-center">↵
44.        <p class="navbar-text navbar-tagline text-muted mb-0">↵
45.          Your personal book collection tracker↵
46.        </p>↵
47.      </div>↵
48. ↵
49. ↵
50.    <!-- Mobile toggler (hamburger) -->↵

```

```
51. <button class="navbar-toggler ms-3" type="button" data-bs-toggle="collapse" data-bs-  
target="#navbarContent">  
52.   aria-controls="navbarContent" aria-expanded="false" aria-label="Toggle navigation">  
53.   <span class="navbar-toggler-icon"></span>  
54. </button>  
55.   
56. <!-- Collapsible nav (wrap your links inside this) -->  
57. <div class="collapse navbar-collapse justify-content-end" id="navbarContent">  
58.     
59.   <ul class="navbar-nav mb-2 mb-lg-0">  
60.     <li class="nav-item">  
61.       <a class="nav-link active" href="/">Home</a>  
62.     </li>  
63.     <li class="nav-item">  
64.       <a class="nav-link" href="/about/">About Us</a>  
65.     </li>  
66.     <li class="nav-item">  
67.       <a class="nav-link"  
68.         href="/community/">Community</a>  
69.     </li>  
70.     
71.     
72.   <li class="nav-item">  
73.     <a class="nav-link" href="/collection/">  
74.       My Collection  
75.     <span class="badge rounded-pill bg-danger align-text-top">1</span>  
76.     </a>  
77.   </li>  
78.   <li class="nav-item">  
79.     <a class="nav-link" href="/profile/">Profile</a>  
80.   </li>  
81.   <li class="nav-item">  
82.     <a class="nav-link" href="/logout/">Logout</a>  
83.   </li>  
84. </ul>  
85. </div>  
86.   
87.   
88.   
89.   
90.   
91.   
92.   
93. </div>  
94. </nav>  
95.   
96. <!-- Main Content -->  
97. <main class="container my-5 pt-5 flex-grow-1">  
98.     
99.   <div class="d-flex flex-column flex-md-row justify-content-center align-items-center gap-2 mb-3 mt-4">  
100.     <h1 class="mb-0">  
101.       <span style="display:inline-flex; align-items:center;">  
102.           
103.         ooks To Discover  
104.       </span>  
105.     </h1>  
106.     
107.   <div class="input-group input-group-sm" style="width: 220px;">  
108.     <input type="text" id="searchInput" class="form-control" placeholder="Search..." aria-label="Search books">  
109.     <!-- styled search button -->  
110.     <button class="btn btn-custom" type="button" id="searchButton">Search</button>  
111.   </div>  
112. </div>  
113.   
114. <!-- Sort + Genre controls -->  
115. <div class="container px-3 px-md-4">  
116.   <div class="d-flex justify-content-between align-items-center mb-3 flex-wrap gap-2">  
117.     <div class="d-flex align-items-center gap-2">  
118.       <label for="genreSelect" class="mb-0">Genre:</label>  
119.       <select id="genreSelect" class="form-select form-select-sm" style="width: auto;">  
120.         <option value="">All genres</option>  
121.         <option value="fiction">Fiction</option>  
122.         <option value="science">Science</option>  
123.         <option value="history">History</option>  
124.         <option value="fantasy">Fantasy</option>  
125.         <option value="romance">Romance</option>  
126.       </select>  
127.     </div>  
128.     <div class="d-flex align-items-center gap-2 ms-auto">  
129.       <label for="sortSelect" class="mb-0">Sort by:</label>  
130.       <select id="sortSelect" class="form-select form-select-sm" style="width: auto;">  
131.         <option value="relevance">Relevance</option>  
132.         <option value="newest">Newest</option>  
133.       </select>  
134.     </div>  
135.   </div>  
136. </div>  
137.   
138. <!-- Loading spinner -->  
139. <div id="loadingSpinner" class="text-center my-3 d-none">  
140.   <div class="spinner-border text-primary" role="status" style="width: 3rem; height: 3rem;">  
141.     <span class="visually-hidden">Loading...</span>  
142.   </div>  
143. </div>  
144.   
145. <!-- Book grid + Load More -->  
146. <div class="container px-3 px-md-4">  
147.   <div class="row g-3" id="book-card-grid"></div>
```

```

148. <div class="text-center mt-4">↵
149.   <!-- Load More Button (inside container, before footer) -->↵
150.   <button id="loadMoreBtn" class="btn btn-outline-secondary d-none">Load More</button>↵
151. </div>↵
152. </div>↵
153. ↵
154. </main>↵
155. ↵
156. <!-- Sticky Footer -->↵
157. <footer class="container-fluid mt-5 mb-3">↵
158.   <div class="d-flex justify-content-between align-items-center px-4">↵
159.     <div></div>↵
160.     <div>↵
161.       <a href="https://www.x.com" target="_blank" class="mx-2"><i class="fa-brands fa-x-twitter fa-lg"></i>
162.     <a href="https://www.facebook.com" target="_blank" class="mx-2"><i
163.       class="fa-brands fa-square-facebook fa-lg"></i></a>↵
164.     </div>↵
165.     <div class="text-muted text-end">↵
166.       &copy; 2025 BookBase↵
167.     </div>↵
168.   </div>↵
169. </footer>↵
170. ↵
171. <!-- Bootstrap JS & Font Awesome -->↵
172. <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>↵
173. <script src="https://kit.fontawesome.com/147220bc71.js" crossorigin="anonymous"></script>↵
174. ↵
175. <script>↵
176.   document.addEventListener('DOMContentLoaded', () => {↵
177.     const container = document.getElementById('book-card-grid');↵
178.     const searchInput = document.getElementById('searchInput');↵
179.     const searchButton = document.getElementById('searchButton');↵
180.     const genreSelect = document.getElementById('genreSelect');↵
181.     const sortSelect = document.getElementById('sortSelect');↵
182.     const spinner = document.getElementById('loadingSpinner');↵
183.     const loadMoreBtn = document.getElementById('loadMoreBtn');↵
184. ↵
185.     let startIndex = 0;↵
186.     const batchSize = 12; // keep in sync with views.fetch_books default/maxResults↵
187.     let currentQuery = '';↵
188.     let currentSort = '';↵
189. ↵
190.     function fetchBooks(reset = false) {↵
191.       if (reset) {↵
192.         container.innerHTML = '';↵
193.         startIndex = 0;↵
194.       }↵
195. ↵
196.       const searchTerm = searchInput.value.trim();↵
197.       const genreTerm = genreSelect.value.trim();↵
198.       const query = searchTerm || genreTerm || 'book';↵
199.       const sort = sortSelect.value;↵
200. ↵
201.       currentQuery = query;↵
202.       currentSort = sort;↵
203. ↵
204.       spinner.classList.remove('d-none');↵
205.       loadMoreBtn.classList.add('d-none');↵
206. ↵
207.       fetch(`/get_books/?
q=${encodeURIComponent(query)}&order=${sort}&startIndex=${startIndex}&maxResults=${batchSize}`)↵
208.         .then(response => response.json())↵
209.         .then(data => {↵
210.           spinner.classList.add('d-none');↵
211. ↵
212.           if (!data.books || data.books.length === 0) {↵
213.             if (reset) {↵
214.               container.innerHTML = `
215.               <div class="text-center text-muted py-5">↵
216.                 <i class="fa-solid fa-book-open fs-1 d-block mb-3"></i>↵
217.                 <h5>No books found</h5>↵
218.                 <p class="small">Try a different search or check your spelling.</p>↵
219.               </div>`;↵
220.             }↵
221.             return;↵
222.           }↵
223. ↵
224.           // Show "Load more" only if we got a full page↵
225.           if (data.books.length === batchSize) {↵
226.             loadMoreBtn.classList.remove('d-none');↵
227.           }↵
228. ↵
229.           data.books.forEach((book) => {↵
230.             const thumbnail = book.cover_url || book.thumbnail || '/static/img/book-placeholder.png';↵
231.             const title = book.title || 'Untitled';↵
232.             const rawAuthor = book.author || 'Unknown Author';↵
233.             const bookId = book.id; // DB int or Google string↵
234. ↵
235.             let author = rawAuthor;↵
236.             if (rawAuthor !== 'Unknown Author') {↵
237.               const authorList = rawAuthor.split(',').map(name => name.trim());↵
238.               author = authorList.length > 2↵
239.                 ? `${authorList.slice(0, 2).join(', ')} +${authorList.length - 2} more`↵
240.                 : authorList.join(', ');↵
241.             }↵
242. ↵
243.             const card = `
244.             <div class="col-12 col-sm-6 col-md-4 col-lg-3 col-xl-2 d-flex">↵

```

```

245. <div class="card shadow-sm book-card" style="width: 100%; max-width: 100%;">↵
246.   ↵
253.   <div class="card-body d-flex flex-column p-2">↵
254.     <div>↵
255.       <h6 class="card-title mb-1">${title}</h6>↵
256.       <p class="card-text small text-muted">${author}</p>↵
257.     </div>↵
258.     <div class="mt-auto">↵
259.       <a href="/books/${bookId}/" class="btn btn-primary btn-sm w-100">View Details</a>↵
260.     </div>↵
261.   </div>↵
262. </div>↵
263. </div>`;↵
264. ↵
265.     container.insertAdjacentHTML('beforeend', card);↵
266.   });↵
267. ↵
268.     startIndex += data.books.length; // advance pagination↵
269.   })↵
270.   .catch(error => {↵
271.     spinner.classList.add('d-none');↵
272.     container.innerHTML = `<div class="text-center text-danger py-5">Error loading books.</div>`;↵
273.     console.error(error);↵
274.   });↵
275. }↵
276. ↵
277. // Event bindings↵
278. searchButton.addEventListener('click', () => fetchBooks(true));↵
279. searchInput.addEventListener('keydown', (e) => {↵
280.   if (e.key === 'Enter') {↵
281.     e.preventDefault();↵
282.     fetchBooks(true);↵
283.   }↵
284. });↵
285. genreSelect.addEventListener('change', () => fetchBooks(true));↵
286. sortSelect.addEventListener('change', () => fetchBooks(true));↵
287. loadMoreBtn.addEventListener('click', () => fetchBooks());↵
288. ↵
289. // Initial load↵
290. fetchBooks(true);↵
291. });↵
292. </script>↵
293. ↵
294. ↵
295. </body>↵
296. ↵
297. </html>

```

Used the HTML parser.

Total execution time 10 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 25.8.30