

Line wrap

```
1 <!DOCTYPE html>
2 <html lang="en" >
3 <head>
4   <meta charset="UTF-8">
5   <title>CodePen - Web Development Bot</title>
6   <link rel="stylesheet" href="./style.css">
7
8   <style>
9   body {
10    margin: 0;
11    font-family: Arial, Helvetica, sans-serif;
12    height: 100vh;
13    width: 100%;
14    background-color: rgba(4, 190, 109, 1);
15    background-image: url("smoke_PNG965.png"), url("smoke_PNG966.png"),
16      url("smoke_PNG967.png");
17    background-repeat: repeat-x;
18    background-position: 0 20%, 0 100%, 0 50%, 0 100%, 0 0;
19    background-size: 2000px, 2000px, 2000px;
20    animation: 120s smoke infinite linear;
21    box-sizing: border-box;
22  }
23
24  @keyframes smoke {
25    100% {
26      background-position: -5000px 20%, -800px 95%, 500px 50%, 1000px 100%,
27        400px 0;
28    }
29  }
30
31  .topnav {
32    overflow: hidden;
33    background-color: #000;
34    z-index: 100;
35  }
36
37  .topnav a {
38    float: left;
39    display: block;
40
41    color: #fff;
42    text-align: center;
43    padding: 14px 16px;
44    text-decoration: none;
45    font-size: 17px;
46  }
47
48  .topnav a:hover {
49    background-color: rgba(4, 190, 109, 1);
50
51    color: black;
52  }
53
54  .topnav a.active {
55    background-color: rgba(4, 190, 109, 1);
56    color: white;
57  }
58
59  .topnav .icon {
60    display: none;
61  }
62  main {
63    position: absolute;
64    top: 10%;
65    left: 50%;
66    transform: translateX(-50%);
67    margin: 1rem auto;
```

```
68 width: 60%;
69 height: 100%;
70 text-align: center;
71 padding: 1rem 2rem;
72 border-radius: 24px;
73 box-shadow: 0 6px 10px black;
74 }
75 #apikey {
76 width: 60%;
77 margin: 0.5rem;
78 border-radius: 10px;
79 padding: 4px 8px;
80 }
81 #prompt {
82 width: 80%;
83 margin: 0.5rem;
84 resize: none;
85 border-radius: 10px;
86 padding: 4px 8px;
87 }
88 button {
89 padding: 0.5rem 1rem;
90 border-radius: 10px;
91 cursor: pointer;
92 margin-top: 4px;
93 background: #f00;
94 color: #fff;
95 }
96 button:hover {
97 background: #fff;
98 color: #f00;
99 }
100
101 .user-message {
102 background-color: rgba(4, 190, 109, 1);
103 padding: 8px;
104 width: 70%;
105 align-self: start;
106 text-align: left;
107 border: 1px solid black;
108 color: black;
109 font-size: 0.9rem;
110 border-radius: 0;
111 margin-bottom: 4px;
112 line-height: 1;
113 }
114
115 .chatgpt-message {
116 background-color: transparent;
117 padding: 8px;
118 width: 100%;
119 align-self: start;
120 text-align: justify;
121 color: black;
122 font-size: 0.9rem;
123 border: 1px solid black;
124 box-shadow: 0 4px 8px black;
125 border-radius: 5px;
126 margin-bottom: 10px;
127 display: flex;
128 flex-direction: column;
129 justify-content: space-between;
130 align-items: start;
131 line-height: 1;
132 }
133
134 #content {
135 display: flex;
```

```
136 justify-content: space-between;
137 margin: 10px auto;
138 width: 100%;
139 height: auto;
140 background: transparent;
141 }
142
143 .preloader {
144   width: 48px;
145 }
146
147 #chathistory {
148   margin: 0 auto;
149   padding: 0.5rem;
150   width: 95%;
151   height: auto;
152   min-height: 10rem;
153   border: 1px solid black;
154   overflow-y: auto;
155 }
156
157 footer {
158   margin: 0 auto;
159   font-size: 1rem;
160   width: 100%;
161   color: white;
162   text-align: center;
163 }
164
165 @media screen and (max-width: 600px) {
166   .topnav a:not(:first-child) {
167     display: none;
168   }
169   .topnav a.icon {
170     float: right;
171     display: block;
172   }
173   main {
174     margin: 10px auto;
175     width: 90%;
176     height: auto;
177     text-align: center;
178     padding: 1rem;
179     border-radius: 24px;
180     box-shadow: 0 6px 10px black;
181   }
182   .preloader {
183     width: 28px;
184   }
185
186   #content {
187     margin: 10px auto;
188     border: 1px solid purple;
189     width: 100%;
190     height: auto;
191     display: flex;
192     flex-direction: column;
193     background: transparent;
194   }
195
196   #chathistory {
197     margin: 0.5rem auto;
198     padding: 0.5rem;
199     width: 90%;
200     height: 24rem;
201     overflow-y: auto;
202     box-shadow: 0 2px 4px black;
203   }
}
```

```
204 .user-message {
205   background-color: rgba(4, 190, 109, 1);
206   padding: 8px;
207   width: 90%;
208   align-self: start;
209   text-align: left;
210   border: 1px solid black;
211   color: black;
212   font-size: 0.8rem;
213   border-radius: 0;
214   margin-bottom: 4px;
215   line-height: 1;
216 }
217
218
219 .chatgpt-message {
220   background-color: transparent;
221   padding: 8px;
222   width: 100%;
223   align-self: start;
224   text-align: justify;
225   color: black;
226   font-size: 0.8rem;
227   border: 1px solid black;
228   box-shadow: 0 4px 8px black;
229   border-radius: 5px;
230   margin-bottom: 8px;
231   display: flex;
232   flex-direction: column;
233   justify-content: space-between;
234   align-items: start;
235   line-height: 1;
236 }
237 }
238
239 @media screen and (max-width: 900px) {
240   .topnav.responsive {
241     position: relative;
242   }
243   .topnav.responsive .icon {
244     position: absolute;
245     right: 0;
246     top: 0;
247   }
248   .topnav.responsive a {
249     float: none;
250     display: block;
251     text-align: left;
252   }
253   main {
254     margin: 10px auto;
255     width: 80%;
256     height: auto;
257     text-align: center;
258     padding: 1rem;
259     border-radius: 24px;
260     box-shadow: 0 6px 10px black;
261   }
262   .preloader {
263     width: 36px;
264   }
265 }
266 </style>
267 </head>
268 <body>
```

```
272 <div class="topnav" id="myTopnav">
273 <a href="https://codepen.io/MrCoder74">My Profile</a>
274 <a href="https://codepen.io/MrCoder74/pen/bGJKmgR">Multiple AI Tool</a>
275 <a href="#" class="active">Web Dev Bot</a>
276 <a href="http://www.vivacitydesign.net/index.php">My Web Agency</a>
277
278 <a href="javascript:void(0);" class="icon" onclick="myFunction()">
279 <i class="fa fa-bars"></i>
280 </a>
281 </div>
282
283 <main>
284 <h1>Web Dev Bot</h1>
285 <small>Generate your new web page <b>with just a click!</b></small>
286 <hr>
287 <div class="userinput">
288 <input type="text" id="apikey" placeholder="Your API key"><br>
289 <textarea rows="6" type="text" id="prompt" placeholder="Your prompt" autocomplete="off" autofocus></textarea><br>
290 <button class="btncopy" onclick="sendMessage()">Send</button>
291 <button class="btncopy" id="savetxt">Save</button>
292 </div>
293 <section id="content">
294
295 <div id="chathistory">
296
297 </div>
298 </section>
299 <footer>
300 Created and offered by Vivacity Design, august 2024.
301 </footer>
302 </main>
303
304 <script>
305 function myFunction() {
306 var x = document.getElementById("myTopnav");
307 if (x.className === "topnav") {
308 x.className += " responsive";
309 } else {
310 x.className = "topnav";
311 }
312 }
313
314 var time = new Date();
315 var hour = time.getHours();
316 var minute = time.getMinutes();
317
318 var saveButtont = document.getElementById("savetxt");
319 saveButtont.addEventListener("click", handleSaveButtontClick);
320
321 function handleSaveButtontClick() {
322 saveChatContentTxt();
323 }
324
325 function saveChatContentTxt() {
326 var chatContent = document.getElementById("chathistory").innerText;
327
328 chatContent = chatContent.replace(/Copy/g, "");
329
330 var currentTimestamp = new Date();
331 var datetime = currentTimestamp.toISOString();
332 var filename = "chatsession_v" + datetime.replace(/[-.:]/g, "");
333
334 var blob = new Blob([chatContent], { type: "text/plain" });
335
336 var url = URL.createObjectURL(blob);
337
338 var downloadLink = document.createElement("a");
339 downloadLink.href = url;
```

```
340 downloadLink.download = filename + ".txt";
341 downloadLink.click();
342 URL.revokeObjectURL(url);
343 }
344
345 let chatMemory = [];
346 chatMemory = createMemory([
347   {
348     role: "system",
349     content:
350       "You are a web developer bot. You don't talk, you don't explain your answers. Your only output is made of code to satisfy the received request. You will not explain the code, you will not introduce it, you will not greet or thank the user for the request, you will only produce a well structured code line by line without interruptions and without dividing it in sections."
351   }
352 ]);
353 console.log(chatMemory);
354
355 function createMemory(messages) {
356   const memory = [];
357   for (const msg of messages) {
358     memory.push({ role: msg.role, content: msg.content });
359   }
360   return memory;
361 }
362
363 async function sendMessage() {
364   const apikey = document.getElementById("apikey").value;
365
366   if (apikey === "") {
367     alert("No OpenAI API Key found.");
368   } else {
369     console.log(apikey);
370   }
371   const inputElement = document.getElementById("prompt");
372   const userInput = inputElement.value.trim();
373   if (userInput !== "") {
374     showMessage("Guest", userInput, "");
375     chatMemory = await getChatGPTResponse(userInput, chatMemory);
376
377     inputElement.value = "";
378   }
379 }
380
381 function showMessage(sender, message, tokens, downloadLink) {
382   const chatContainer = document.getElementById("chathistory");
383   const typingIndicator = document.getElementById("typing-indicator");
384
385   if (typingIndicator && sender === "VivacityGPT") {
386     chatContainer.removeChild(typingIndicator);
387   }
388
389   const messageElement = document.createElement("div");
390
391   if (sender === "Guest") {
392     messageElement.innerHTML = `+[${hour}:${minute}] - ${sender}: ${message}`;
393     messageElement.classList.add("user-message");
394   } else {
395     const timestampElement = document.createElement("p");
396     timestampElement.innerHTML = `-[${hour}:${minute}] - ${sender}: `;
397     timestampElement.classList.add("chatgpt-message");
398
399     messageElement.appendChild(timestampElement);
400
401     const iframe = document.createElement("iframe");
402     iframe.style.width = "100%";
403     iframe.style.height = "600px";
404     iframe.style.border = "1px solid black";
405
406     messageElement.appendChild(iframe);

```

```

407     const blob = new Blob(
408     [
409     [
410     <!DOCTYPE html>
411     <html lang="en">
412     <head>
413     <meta charset="UTF-8">
414     <meta name="viewport" content="width=device-width, initial-scale=1.0">
415     <title>Generated Code</title>
416     </head>
417     <body>
418     ${message}
419     </body>
420     </html>
421     ],
422     {
423     type: "text/html"
424     }
425     );
426
427     const url = URL.createObjectURL(blob);
428     iframe.src = url;
429
430     const separator = document.createElement("p");
431     separator.innerHTML = `${tokens}`;
432     messageElement.classList.add("chatgpt-message");
433     messageElement.appendChild(separator);
434
435     const downloadElem = document.createElement("div");
436     downloadElem.innerHTML = downloadLink;
437     messageElement.appendChild(downloadElem);
438   }
439
440   chatContainer.appendChild(messageElement);
441   chatContainer.scrollTop = chatContainer.scrollHeight;
442 }
443
444 async function getChatGPTResponse(userInput, chatMemory = []) {
445   const apiKey = document.getElementById("apikey").value;
446   console.log(apikey);
447   if (apikey === "") {
448     alert("No OpenAI API Key found.");
449   } else {
450     console.log(apikey);
451   }
452   const chatContainer = document.getElementById("chathistory");
453   const typingIndicator = document.createElement("p");
454   typingIndicator.id = "typing-indicator";
455   typingIndicator.innerHTML =
456     ''; // need to have this gif in your folder
457   chatContainer.appendChild(typingIndicator);
458
459   try {
460     const response = await fetch("https://api.openai.com/v1/chat/completions", {
461       method: "POST",
462       headers: {
463         "Content-Type": "application/json",
464
465         Authorization: "Bearer " + apiKey
466       },
467       body: JSON.stringify({
468         model: "gpt-4o-mini",
469         messages: [...chatMemory, { role: "user", content: userInput }]
470       })
471     });
472     if (!response.ok) {
473       throw new Error("Error while requesting to the API");
474     }

```

```
475 const data = await response.json();
476 if (
477     !data.choices ||
478     !data.choices.length ||
479     !data.choices[0].message ||
480     !data.choices[0].message.content
481 ) {
482     throw new Error("Invalid API response");
483 }
484
485 const chatGPTResponse = data.choices[0].message.content.trim();
486
487 var cleanResponse = chatGPTResponse.replace(
488     /(```html|```css|```javascript|```php|```python|```vb|```vb.net|cpp|java|csharp)(.*?)gs,
489     "$2"
490 );
491 console.log(chatGPTResponse);
492 cleanResponse = cleanResponse.replace(/```/g, "");
493 cleanResponse = cleanResponse.replace(/\*\*(.*?)\*\*/g, "$1");
494
495 const tokenCount = document.createElement("p");
496
497 if (data.usage.completion_tokens) {
498     const requestTokens = data.usage.prompt_tokens;
499     const responseTokens = data.usage.completion_tokens;
500     const totalTokens = data.usage.total_tokens;
501     const priceperTokenPrompt = 0.15 / 1000000; //uses gpt-4o-mini price of 0.15/Mt USD
502     const priceperTokenResponse = 0.6 / 1000000; //uses gpt-4o-mini price of 0.15/Mt USD
503     const priceperRequest = priceperTokenPrompt * requestTokens;
504     const priceperResponse = priceperTokenResponse * responseTokens;
505     const totalExpense = priceperRequest + priceperResponse;
506     tokenCount.innerHTML = `

---

Your request used ${requestTokens} tokens and costed ${priceperRequest.toFixed(
507         6
508     )}USD<br>This response used ${responseTokens} tokens and costed ${priceperResponse.toFixed(
509         6
510     )}USD<br>Total Tokens: ${totalTokens}. This interaction costed you: ${totalExpense.toFixed(
511         6
512     )}USD.`;
513 } else {
514     tokenCount.innerHTML = "Unable to track the number of used tokens.";
515 }
516
517 const blob = new Blob([cleanResponse], { type: "text/html" });
518 const url = URL.createObjectURL(blob);
519 const downloadLink = `
```