

Screenshots of Code:

```
Users > Ashley > Desktop > Week-12 > bakery.html > html > body.container > div.row > div.col > div#new-order.jumbotron > div.form-group > Input#date-due.form-control
# style.css
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Orders | Minimalist Pâtisserie</title>
8   <!-- Bootstrap -->
9   <link rel="stylesheet" href="style.css">
10  <link rel="stylesheet" href="node_modules/bootstrap/dist/css/bootstrap.css">
11  <!-- Google Fonts -->
12  <link rel="preconnect" href="https://fonts.gstatic.com">
13  <link href="https://fonts.googleapis.com/css2?family=Quicksand&family=Sofia&display=swap" rel="stylesheet">
14 </head>
15 <body class="container">
16   <!-- Navbar to make page responsive -->
17   <nav id="nav-bar" class="navbar navbar-expand-lg justify-content-between fixed-top navbar-light">
18     <a class="navbar-brand" href="#">
19       
20     </a>
21     <a class="navbar-brand" id="company-name" href="#">Minimalist Pâtisserie</a>
22     <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown">
23       <span class="navbar-toggler-icon"></span>
24     </button>
25     <div class="collapse navbar-collapse" id="navbarNavDropdown">
26       <ul class="navbar-nav">
27         <li class="nav-item">
28           <a class="nav-link" href="#">Home</a>
29         </li>
30         <li class="nav-item">
31           <a class="nav-link" href="#">About Us</a>
32         </li>
33         <li class="nav-item active">
34           <a class="nav-link" href="#">Orders<span class="sr-only">(current)</span></a>
35         </li>
36         <li class="nav-item">
37           <a class="nav-link" href="#">Contact</a>
38         </li>
39         <li class="nav-item dropdown">
40           <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown">Sign In</a>
41           <div id="bar" class="dropdown-menu">
42             <form class="form-inline">
43               <input class="form-control mr-sm-2" type="text" placeholder="Username">
44               <input class="form-control mr-sm-2" type="password" placeholder="Password">
45               <button id="submit" class="btn my-2 my-sm-0 btn-sm" type="submit">Sign In</button>
46             </form>
47           </div>
48         </li>
49       </ul>
50     </div>
51   </nav>
52   <div class="jumbotron">
53     <div class="row">
54       <div class="col">
55         <h1>Minimalist Pâtisserie</h1>
56       </div>
57       <div class="col">
58         <h2>Orders</h2>
59       </div>
60     </div>
61     <div class="row">
62       <div class="col">
63         <div class="form-group">
64           <input type="text">
65         </div>
66       </div>
67       <div class="col">
68         <div class="form-group">
69           <input type="text">
70         </div>
71       </div>
72       <div class="col">
73         <div class="form-group">
74           <input type="text">
75         </div>
76       </div>
77       <div class="col">
78         <div class="form-group">
79           <input type="text">
80         </div>
81       </div>
82       <div class="col">
83         <div class="form-group">
84           <input type="text">
85         </div>
86       </div>
87       <div class="col">
88         <div class="form-group">
89           <input type="text">
90         </div>
91       </div>
92       <div class="col">
93         <div class="form-group">
94           <input type="text">
95         </div>
96       </div>
97       <div class="col">
98         <div class="form-group">
99           <input type="text">
100        </div>
101      </div>
102    </div>
103    <div class="row">
104      <div class="col">
105        <div class="form-group">
106          <input type="text">
107        </div>
108      </div>
109      <div class="col">
110        <div class="form-group">
111          <input type="text">
112        </div>
113      </div>
114      <div class="col">
115        <div class="form-group">
116          <input type="text">
117        </div>
118      </div>
119      <div class="col">
120        <div class="form-group">
121          <input type="text">
122        </div>
123      </div>
124      <div class="col">
125        <div class="form-group">
126          <input type="text">
127        </div>
128      </div>
129      <div class="col">
130        <div class="form-group">
131          <input type="text">
132        </div>
133      </div>
134      <div class="col">
135        <div class="form-group">
136          <input type="text">
137        </div>
138      </div>
139      <div class="col">
140        <div class="form-group">
141          <input type="text">
142        </div>
143      </div>
144      <div class="col">
145        <div class="form-group">
146          <input type="text">
147        </div>
148      </div>
149      <div class="col">
150        <div class="form-group">
151          <input type="text">
152        </div>
153      </div>
154      <div class="col">
155        <div class="form-group">
156          <input type="text">
157        </div>
158      </div>
159      <div class="col">
160        <div class="form-group">
161          <input type="text">
162        </div>
163      </div>
164      <div class="col">
165        <div class="form-group">
166          <input type="text">
167        </div>
168      </div>
169      <div class="col">
170        <div class="form-group">
171          <input type="text">
172        </div>
173      </div>
174      <div class="col">
175        <div class="form-group">
176          <input type="text">
177        </div>
178      </div>
179      <div class="col">
180        <div class="form-group">
181          <input type="text">
182        </div>
183      </div>
184      <div class="col">
185        <div class="form-group">
186          <input type="text">
187        </div>
188      </div>
189      <div class="col">
190        <div class="form-group">
191          <input type="text">
192        </div>
193      </div>
194      <div class="col">
195        <div class="form-group">
196          <input type="text">
197        </div>
198      </div>
199      <div class="col">
200        <div class="form-group">
201          <input type="text">
202        </div>
203      </div>
204      <div class="col">
205        <div class="form-group">
206          <input type="text">
207        </div>
208      </div>
209      <div class="col">
210        <div class="form-group">
211          <input type="text">
212        </div>
213      </div>
214      <div class="col">
215        <div class="form-group">
216          <input type="text">
217        </div>
218      </div>
219      <div class="col">
220        <div class="form-group">
221          <input type="text">
222        </div>
223      </div>
224      <div class="col">
225        <div class="form-group">
226          <input type="text">
227        </div>
228      </div>
229      <div class="col">
230        <div class="form-group">
231          <input type="text">
232        </div>
233      </div>
234      <div class="col">
235        <div class="form-group">
236          <input type="text">
237        </div>
238      </div>
239      <div class="col">
240        <div class="form-group">
241          <input type="text">
242        </div>
243      </div>
244      <div class="col">
245        <div class="form-group">
246          <input type="text">
247        </div>
248      </div>
249      <div class="col">
250        <div class="form-group">
251          <input type="text">
252        </div>
253      </div>
254      <div class="col">
255        <div class="form-group">
256          <input type="text">
257        </div>
258      </div>
259      <div class="col">
260        <div class="form-group">
261          <input type="text">
262        </div>
263      </div>
264      <div class="col">
265        <div class="form-group">
266          <input type="text">
267        </div>
268      </div>
269      <div class="col">
270        <div class="form-group">
271          <input type="text">
27
```

```

1  bakery.html × JS bakery.js # style.css
2
3  Users > Ashley > Desktop > Week-12 > bakery.html > html > body.container > div.row > div.col > div#new-order.jumbotron > div.form-group > Input#date-due.form-control
4
5  46     </form>
6  47   </div>
7  48   </li>
8  49 </ul>
9  50 </div>
10 51 </nav>
11 52 <br><br><br>
12 53
13 54 <!-- New Order Form -->
14 55 <div class="row">
15 56   <div class="col">
16 57     <div id="new-order" class="jumbotron">
17 58       <h3>New Cake Order</h3>
18 59       <br>
19 60       <div class="form-group">
20 61         <label for="new-order-name">Name of Order:</label>
21 62         <input type="text" id="new-order-name" class="form-control" placeholder="e.g Class Party">
22 63       </div>
23 64       <div class="form-group">
24 65         <label for="new-order-baker">Baker:</label>
25 66         <div class="input-group">
26 67           <select class="custom-select" id="baker-name">
27 68             <option selected>Choose...</option>
28 69             <option value="Ashley">Ashley</option>
29 70             <option value="Sofia">Sofia</option>
30 71             <option value="Michael">Michael</option>
31 72             <option value="Daniela">Daniela</option>
32 73           </select>
33 74         </div>
34 75       </div>
35 76       <div class="form-group">
36 77         <label for="date-due">Event Date:</label>
37 78         <input type="date" id="date-due" class="form-control">
38 79       </div>
39 80     <br>
40 81     <div class="center">
41 82       <button id="create-new-order" type="button" class="form-control">Add New Order</button>
42 83     </div>
43 84   </div>
44 85 </div>
45 86 </div>
46 87 <!-- Child elements will be appended to this parent element -->
47 88 <div class="row">
48 89
49 90

```

```
bakery.html x JS bakery.js # style.css
Users > Ashley > Desktop > Week-12 > bakery.html > html > body.container > div.row > div.col > div#new-order.jumbotron > div.form-group > input#date-due.form-control

89 <!-- Child elements will be appended to this parent element -->
90 <div class="row">
91   <div class="col">
92     <div id="app"></div>
93   </div>
94 </div>
95 <!-- Footer -->
96 <footer class="footer">
97   <p class="footer-icons">
98     <!-- Facebook -->
99     <a href="#"><svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-facebook" viewBox="0 0 16 16">
100       <path d="M16 8.049c0-4.446-3.582-8.05-8.05-8.05C3.58 0-0.02 3.603-.02 8.05c0 4.017 2.926 7.347 6.75 7.951v-5.625h-2.03V8.05H6.75V6.275c0-2.017 1.195-3.131 3.02
101     </svg></a>
102     <!-- Instagram -->
103     <a href="#"><svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-instagram" viewBox="0 0 16 16">
104       <path d="M8 0C5.829 0 5.556.01 4.703.048 3.85.088 3.269.222 2.764.317 3.917 0 0-1.417 9.234 9.27 3.927 0 0 .42 2.76C.222 3.268.087 3.85.048 4.7.01 5.556 0
105     </svg></a>
106     <!-- Twitter -->
107     <a href="#"><svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-twitter" viewBox="0 0 16 16">
108       <path d="M5.026 15c6.038 0 9.341-5.041 9.341-9.341 0-.14-.002-.282-.006-.422A6.685 6.685 0 0 1 16 3.542a6.685 6.685 0 0 1-1.889 5.18 3.301 3.301 0 0 0 1.447-1.81
109     </svg></a>
110   </p>
111   <p class="footer-links">
112     <a class="link-1" href="#">Home</a>
113     <a href="#">About Us</a>
114     <a href="#">Orders</a>
115     <a href="#">Contact</a>
116   </p>
117   <p>Minimalist Pâtisserie LV &copy; 2021</p>
118 </div>
119 </footer>
120
121 <!-- JavaScript / Node Modules -->
122 <script src="node_modules/jquery/dist/jquery.js"></script>
123 <script src="node_modules/bootstrap/dist/js/bootstrap.js"></script>
124 <script src="bakery.js"></script>
125 </body>
126 </html>
127
```

```
bakery.html JS bakery.js x # style.css
Users > Ashley > Desktop > Week-12 > JS bakery.js > DOMManager > getNewCakeForm

1 //The utilities will store the function for generating new object ids, otherwise it will be made global
2 let utils = (function() {
3   let ids = 0;
4   function getNewId() {
5     return ids++;
6   }
7
8   return { getNewId };
9 })();
10
11 // Cake Model
12 class Cake {
13   constructor(id, flavor, frosting, layers, shapes) {
14     this.id = id;
15     this.flavor = flavor;
16     this.frosting = frosting;
17     this.layers = layers;
18     this.shapes = shapes;
19   }
20 }
21
22 // New Order Model
23 class Order {
24   constructor(id, name, baker, eventDate) {
25     this.id = id;
26     this.name = name;
27     this.baker = baker;
28     this.eventDate = eventDate;
29     this.cakes = [];
30   }
31
32   // Add new cake function
33   addCake(flavor, frosting, layers, shapes) {
34     this.cakes.push(new Cake(utils.getNewId(), flavor, frosting, layers, shapes));
35   }
36
37   // Remove cake function
38   removeCake(id) {
39     let cakeToDeleteIdx = this.cakes.map(cake => cake.id).indexOf(id);
40     if (cakeToDeleteIdx != -1) {
41       this.cakes.splice(cakeToDeleteIdx, 1);
42     }
43   }
44 }
45
46 //Where our orders will be stored
```

```
bakery.html JS bakery.js x # style.css
Users > Ashley > Desktop > Week-12 > JS bakery.js > DOMManager > getNewCakeForm

46 //Where our orders will be stored
47 let orderRepo = [];
48
49
50 class OrderService {
51
52   // Returns the index in the repo where an order with given id was found; -1 if not found
53   static getIdxForId(orderId) {
54
55     return orderRepo.map(a => a.id).indexOf(orderId);
56   }
57
58   static getAllOrders() {
59     return orderRepo;
60   }
61
62   static addOrder(order) {
63     orderRepo.push(order);
64   }
65
66   //We update the order by first locating it in the orderRepo
67   static updateOrder(order) {
68     let indexToUpdate = OrderService.getIdxForId(order.id);
69     if (indexToUpdate !== -1) {
70       orderRepo[indexToUpdate] = order;
71     }
72   }
73
74   static deleteOrder(id) {
75     let indexToUpdate = OrderService.getIdxForId(id);
76     orderRepo.splice(indexToUpdate, 1);
77   }
78
79   static deleteCake(orderId, cakeId) {
80
81     let indexToUpdate = OrderService.getIdxForId(orderId);
82     let orderToUpdate = orderRepo[indexToUpdate];
83     orderToUpdate.removeCake(cakeId);
84   }
85 }
86
87 //Frontend state
88 let currentState;
89
90 function updateState(newState) {
```

Ln 166, Col 54 Spaces: 4 UTF-8 LF JavaScript Prettier

```
bakery.html JS bakery.js x # style.css
Users > Ashley > Desktop > Week-12 > JS bakery.js > DOMManager > getNewCakeForm

90 function updateState(newState) {
91   //The new state is rendered in real-time
92   currentState = newState;
93   DOMManager.render();
94 }
95
96 //DOM Manager
97 class DOMManager {
98   static getOrderHeader(orderDesc) {
99     return `<div>
100       <div class="col-8">
101         <h4><span>Order Name:</span> <small class="text-muted">${orderDesc.name}</small><br>
102         <span>Baked by :</span> <small class="text-muted">${orderDesc.baker}</small><br>
103         <span>Event Date:</span> <small class="text-muted">${orderDesc.eventDate}</small></h4>
104       </div>
105       <div class="col-4">
106         <button class="delete-order btn btn-danger"
107           id="delete-order-${orderDesc.id}" data-order-id="${orderDesc.id}">Delete</button>
108       </div>
109     </div>`;
110   }
111 }
112
113 //We render the orders on the DOM programmatically with template literals + HTML by looping through each cake object in the order array
114 static getCakeMarkupForOrder(orderDesc) {
115   let cakeHtml = [];
116   orderDesc.cakes.forEach(cakeDesc => {
117     cakeHtml.push(`<div class="row">
118       <div class="col-8">
119         <ul>
120           <li><b>Flavor:</b> ${cakeDesc.flavor}</li>
121           <li><b>Frosting:</b> ${cakeDesc.frosting}</li>
122           <li><b>No. of Layers:</b> ${cakeDesc.layers}</li>
123           <li><b>Shape:</b> ${cakeDesc.shapes}</li>
124         </ul>
125       </div>
126       <div class="col-4">
127         <button class="delete-cake btn btn-danger" id="delete-cake-${cakeDesc.id}"
128           data-cake-id="${cakeDesc.id}" data-order-id="${orderDesc.id}">Delete</button>
129       </div>
130     </div>`);
131   });
132   return cakeHtml.join('');
133 }
134
135 static getNewCakeForm(orderDesc) {
```

Ln 166, Col 54 Spaces: 4 UTF-8 LF JavaScript Prettier

```
bakery.html JS bakery.js X # style.css
Users > Ashley > Desktop > Week-12 > JS bakery.js > DOMManager > getNewCakeForm
134 static getNewCakeForm(orderDesc) {
135   return `<div class="form-group" id="new-cake-form">
136     <label for="new-cake-flavor-${orderDesc.id}">Flavor:</label><br>
137     <div class="input-group">
138       <select class="custom-select" id="new-cake-flavor-${orderDesc.id}">
139         <option selected>Choose..</option>
140         <option value="Vanilla">Vanilla</option>
141         <option value="Chocolate">Chocolate</option>
142         <option value="Lemon">Lemon</option>
143         <option value="Red Velvet">Red Velvet</option>
144         <option value="Carrot">Carrot</option>
145       </select>
146     </div>
147   </div>
148   <div class="form-group">
149     <label for="new-cake-frosting-${orderDesc.id}">Frosting:</label><br>
150     <input class="form-control" type="text" id="new-cake-frosting-${orderDesc.id}" placeholder="Enter the color for your cake">
151   </div>
152   <div class="form-group">
153     <label for="new-cake-layers-${orderDesc.id}">Number of Layers:</label><br>
154     <div class="input-group">
155       <select class="custom-select" id="new-cake-layers-${orderDesc.id}">
156         <option selected>Choose..</option>
157         <option value="One">One</option>
158         <option value="Two">Two</option>
159         <option value="Three">Three</option>
160       </select>
161     </div>
162   </div>
163   <div class="form-group">
164     <label for="new-cake-shapes-${orderDesc.id}">Shape of Cake:</label><br>
165     <div class="input-group">
166       <select class="custom-select" id="new-cake-shapes-${orderDesc.id}">
167         <option selected>Choose..</option>
168         <option value="Square">Square</option>
169         <option value="Circle">Circle</option>
170         <option value="Rectangle">Rectangle</option>
171         <option value="Bundt">Bundt</option>
172       </select>
173     </div>
174   </div>
175   <div id="add-cake" class="form-group">
176     <button class="form-control btn" id="add-cake-for-order-${orderDesc.id}" data-order-id="${orderDesc.id}">Add New Cake</button>
177   </div>`;
178 }
179
```

Ln 166, Col 54 Spaces: 4 UTF-8 LF JavaScript Prettier

```
bakery.html JS bakery.js X # style.css
Users > Ashley > Desktop > Week-12 > JS bakery.js > DOMManager > getNewCakeForm
180 static getOrderBox(orderDesc) {
181   let orderHeader = DOMManager.getOrderHeader(orderDesc);
182   let addCakeForm = DOMManager.getNewCakeForm(orderDesc);
183   let currentCakes = DOMManager.getCakeMarkupForOrder(orderDesc);
184   return `<div class="card">
185     <div class="card-header">
186       ${orderHeader}
187     </div>
188     <div class="card-body">
189       ${addCakeForm}
190       ${currentCakes}
191     </div>
192   </div>`;
193 }
194
195 static render() {
196   // Render the app div based on current state
197   let newMarkup = currentState.map(this.getOrderBox);
198   $('#app').html(newMarkup);
199 }
200
201 static init() {
202   // Application initialization, event delegation hooks
203   let $nameInput = $('#new-order-name');
204   let $bakerInput = $('#baker-name');
205   let $dateInput = $('#date-due');
206
207   $('#create-new-order').on('click', () => {
208     OrderService.addOrder(new Order(utils.getNewId(), $nameInput.val(), $bakerInput.val(), $dateInput.val()));
209     updateState(OrderService.getAllOrders());
210   });
211
212   $('#app').on('click', (e) => {
213     // employing event delegation
214     let $target = $(e.target);
215     let targetId = $target.attr('id');
216
217     if (!targetId) return;
218
219     //Determining which of our buttons was clicked on by user + delegating an event to each of them
220     if (targetId.startsWith('delete-order')) {
221       let orderId = $target.data('orderId');
222       OrderService.deleteOrder(orderId);
223       updateState(OrderService.getAllOrders());
224     } else if (targetId.startsWith('delete-cake')) {

```

Ln 166, Col 54 Spaces: 4 UTF-8 LF JavaScript Prettier


```
bakery.html  JS bakery.js  # style.css
Users > Ashley > Desktop > Week-12 > JS bakery.js > DOMManager > getNewCakeForm
219 //Determining which of our buttons was clicked on by user + delegating an event to each of them
220 if (targetId.startsWith('delete-order')) {
221   let orderId = target.data('orderId');
222   OrderService.deleteOrder(orderId);
223   updateState(OrderService.getAllOrders());
224 } else if (targetId.startsWith('delete-cake')) {
225   let orderId = target.data('orderId');
226   let cakeId = target.data('cakeId');
227   OrderService.deleteCake(orderId, cakeId);
228   updateState(OrderService.getAllOrders());
229 } else if (targetId.startsWith('add-cake-for-order')) {
230   //How we will get the cake info we need to create a description of it
231   let orderId = target.data('orderId');
232   let flavor = $('#new-cake-flavor-${orderId}').val();
233   let frosting = $('#new-cake-frosting-${orderId}').val();
234   let layers = $('#new-cake-layers-${orderId}').val();
235   let shapes = $('#new-cake-shapes-${orderId}').val();
236
237   //We will update the order with the new cake
238   let orderToUpdate;
239   currentState.forEach(order => {
240     if (order.id === orderId) {
241       order.addCake(flavor, frosting, layers, shapes);
242       orderToUpdate = order;
243     }
244   });
245   if (orderToUpdate) {
246     OrderService.updateOrder(orderToUpdate);
247     updateState(OrderService.getAllOrders());
248   }
249 }
250 });
251
252 updateState(OrderService.getAllOrders());
253 }
254 }
255
256 // Application initialization
257
258 DOMManager.init();
```

Ln 166, Col 54 Spaces: 4 UTF-8 LF JavaScript Prettier

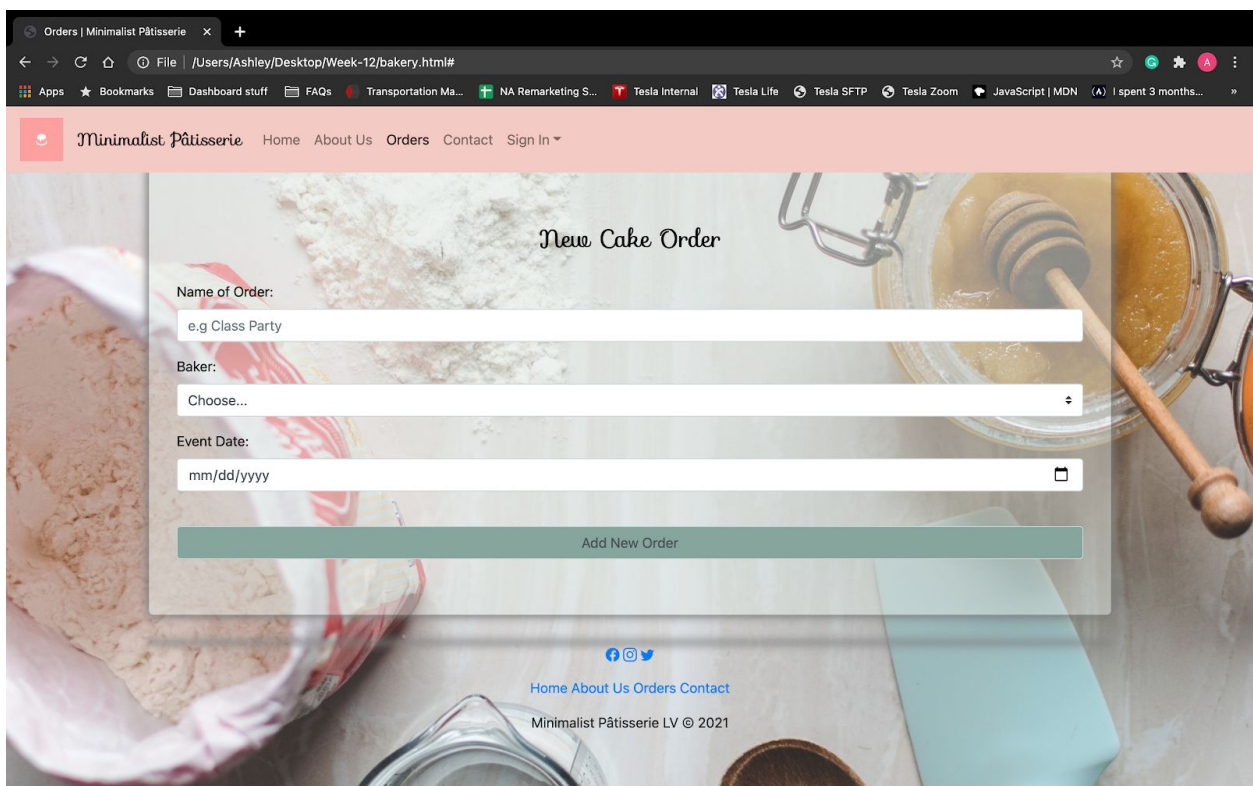
```
bakery.html  JS bakery.js  # style.css
Users > Ashley > Desktop > Week-12 > # style.css > #bar
1 body {
2   width: 100%;
3   height: auto;
4   min-height: 100vh;
5   background-size: 100% 100%;
6   background-position: top center;
7   font-family: 'Quicksand', sans-serif;
8   background: url(bakery/cake.jpg) no-repeat center center fixed;
9   background-size: cover;
10  background-position: center;
11 }
12
13 #new-order,
14 div.card-body div.card-header {
15   background-color: rgba(229, 235, 230, 0.5);
16   box-shadow: 0px 0px 10px 3px grey;
17   color: black;
18 }
19
20 #app {
21   background-color: rgba(229, 235, 230, 0.5);
22   box-shadow: 0px 0px 10px 3px grey;
23   color: black;
24 }
25
26 img {
27   width: 50px;
28   height: 50px;
29 }
30
31 h3 {
32   font-family: 'Sofia', cursive;
33   text-align: center;
34   font-size: 40px;
35 }
36
37 .dropdown-menu {
38   border: none;
39 }
40
41 .navbar-brand {
42   text-align: left;
43   font-family: 'Sofia', cursive;
44 }
45
46 .center {
```

Ln 68, Col 2 Spaces: 4 UTF-8 LF CSS Prettier

```
bakery.html  JS bakery.js  # style.css x
Users > Ashley > Desktop > Week-12 > # style.css > #bar

40
41 .navbar-brand {
42   text-align: left;
43   font-family: 'Sofia', cursive;
44 }
45
46 .center {
47   display: flex;
48   justify-content: center;
49   align-items: center;
50 }
51
52 .footer {
53   position: fixed-bottom;
54   width: 100%;
55   height: 90px;
56   background-color: none;
57   text-align: center;
58 }
59
60 #create-new-order,
61 #add-cake,
62 #submit {
63   background: #84A59D;
64 }
65
66 #bar {
67   background: #F9DED3;
68 }
69
70 #nav-bar {
71   background: #F5CAC3;
72 }
73
74
75
```

Screenshots of Running Application:



Orders | Minimalist Pâtisserie x +

File /Users/Ashley/Desktop/Week-12/bakery.html#

Apps ★ Bookmarks Dashboard stuff FAQs Transportation Ma... NA Remarketing S... Tesla Internal Tesla Life Tesla SFTP Tesla Zoom JavaScript | MDN (A) I spent 3 months... »

Minimalist Pâtisserie Home About Us Orders Contact Sign In ▾

New Cake Order

Name of Order:
Birthday Party

Baker:
Ashley

Event Date:
03/06/2021

Add New Order

Order Name: Birthday Party
Baked by : Ashley
Event Date: 2021-03-06

Delete

Orders | Minimalist Pâtisserie x +

File /Users/Ashley/Desktop/Week-12/bakery.html#

Apps ★ Bookmarks Dashboard stuff FAQs Transportation Ma... NA Remarketing S... Tesla Internal Tesla Life Tesla SFTP Tesla Zoom JavaScript | MDN (A) I spent 3 months... »

Minimalist Pâtisserie Home About Us Orders Contact Sign In ▾

Order Name: Birthday Party
Baked by : Ashley
Event Date: 2021-03-06

Delete

Flavor:
Choose..

Frosting:
Enter the color for your cake

Number of Layers:
Choose..

Shape of Cake:
Choose...

Add New Cake

Home About Us Orders Contact

Minimalist Pâtisserie LV © 2021

