

```
1  <!DOCTYPE html>
2  <html lang="fr">
3    <head>
4      <title>Kanap</title>
5
6      <meta charset="utf-8" />
7      <meta name="description" content="Plateforme incroyable de e-commerce" />
8
9      <link rel="preconnect" href="https://fonts.googleapis.com" />
10     <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
11     <link
12       href="https://fonts.googleapis.com/css2?
family=Montserrat:wght@100;200;300;400;500;600;700;800;900&display=swap"
13       rel="stylesheet"
14     />
15     <link href="../css/style.css" rel="stylesheet" />
16
17     <meta name="viewport" content="width=device-width, initial-scale=1" />
18   </head>
19
20   <body>
21     <header>
22       <div class="limitedWidthBlockContainer informations">
23         <div class="limitedWidthBlock">
24           <ul>
25             <li>
26               01 23 45 67 89
31             </li>
32             <li>
33               support@name.com
38             </li>
39             <li>
40               01 23 45 67 89
45             </li>
46           </ul>
47         </div>
48       </div>
49       <div class="limitedWidthBlockContainer menu">
50         <div class="limitedWidthBlock">
51           <a href="/index.html">
52             
57           </a>
58           <nav>
59             <ul>
60               <a href="/index.html"><li>Accueil</li></a>
61               <a href="/cart.html"><li>Panier</li></a>
62             </ul>
63           </nav>
64         </div>
65       </div>
```

```
66     
67 </header>
68
69 <main class="limitedWidthBlockContainer">
70   <div class="limitedWidthBlock">
71     <div class="titles">
72       <h1>Nos produits</h1>
73       <h2>Une gamme d'articles exclusifs</h2>
74     </div>
75     <section class="items" id="items">
76       <!-- <a href="/product.html?id=42">
77         <article>
78           
79             <h3 class="productName">Kanap name1</h3>
80             <p class="productDescription">Dis enim malesuada risus sapien gravida
nulla nisl arcu. Dis enim malesuada risus sapien gravida nulla nisl arcu.</p>
81           </article>
82         </a -->
83       </section>
84     </div>
85   </main>
86
87   <footer>
88     <div class="limitedWidthBlockContainer footerMain">
89       <div class="limitedWidthBlock">
90         <div>
91           
96         </div>
97         <div>
98           <p>10 quai de la charente <br />75019 Paris 19</p>
99         </div>
100        <div>
101          <p>Téléphone : 01 23 45 67 89</p>
102        </div>
103        <div>
104          <p>Email : support@name.com</p>
105        </div>
106      </div>
107    </div>
108    <div class="limitedWidthBlockContainer footerSecondary">
109      <div class="limitedWidthBlock">
110        <p>
111          © Copyright 2021 - 2042 | Openclassrooms by Openclassrooms | All
112          Rights Reserved | Powered by <3
113        </p>
114      </div>
115    </div>
116  </footer>
117  <script src="../js/index.js"></script>
118  <script src="../js/class.js"></script>
119 </body>
120 </html>
```

```
1  <!DOCTYPE html>
2  <html lang="fr">
3    <head>
4      <title>Nom du produit</title>
5
6      <meta charset="utf-8" />
7      <meta name="description" content="Plateforme incroyable de e-commerce" />
8
9      <link rel="preconnect" href="https://fonts.googleapis.com" />
10     <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
11     <link
12       href="https://fonts.googleapis.com/css2?
family=Montserrat:wght@100;200;300;400;500;600;700;800;900&display=swap"
13       rel="stylesheet"
14     />
15     <link href="../css/style.css" rel="stylesheet" />
16     <link href="../css/product.css" rel="stylesheet" />
17
18     <meta name="viewport" content="width=device-width, initial-scale=1" />
19   </head>
20
21   <body>
22     <header>
23       <div class="limitedWidthBlockContainer informations">
24         <div class="limitedWidthBlock">
25           <ul>
26             <li>
27               01 23 45 67 89
32             </li>
33             <li>
34               support@name.com
39             </li>
40             <li>
41               01 23 45 67 89
46             </li>
47           </ul>
48         </div>
49       </div>
50       <div class="limitedWidthBlockContainer menu">
51         <div class="limitedWidthBlock">
52           <a href="/index.html">
53             
58           </a>
59         <nav>
60           <ul>
61             <a href="/index.html"><li>Accueil</li></a>
62             <a href="/cart.html"><li>Panier</li></a>
63           </ul>
64         </nav>
65       </div>
```

```

66     </div>
67     
68 </header>
69
70 <main class="limitedWidthBlockContainer">
71     <div class="limitedWidthBlock">
72         <section class="item" id="oneProduct">
73             <article>
74                 <div class="item_img">
75                     <!--  -->
76                 </div>
77                 <div class="item_content">
78                     <div class="item_content_titlePrice">
79                         <h1 id="title"><!-- Nom du produit --></h1>
80                         <p>
81                             Prix : <span id="price"><!-- 42 --></span>€
82                         </p>
83                     </div>
84
85                     <div class="item_content_description">
86                         <p class="item_content_description_title">Description :</p>
87                         <p id="description">
88                             <!-- Dis enim malesuada risus sapien gravida nulla nisl arcu. -->
89                         </p>
90                     </div>
91
92                     <div class="item_content_settings">
93                         <div class="item_content_settings_color">
94                             <label for="color-select">Choisir une couleur :</label>
95                             <select name="color-select" id="colors">
96                                 <option value="">--SVP, choisissez une couleur --</option>
97                                 <!-- <option value="vert">vert</option>
98                                 <option value="blanc">blanc</option> -->
99                             </select>
100                         </div>
101
102                         <div class="item_content_settings_quantity" id="quantity">
103                             <label for="itemQuantity">
104                                 >Nombre d'article(s) (1-100) :</label>
105                             >
106                             <input
107                                 type="number"
108                                 name="itemQuantity"
109                                 min="1"
110                                 max="100"
111                                 value="0"
112                                 id="quantityValue"
113                             />
114                         </div>
115                     </div>
116
117                     <div class="item_content_addButton">
118                         <button id="addToCart">Ajouter au panier</button>
119                     </div>
120                 </div>
121             </article>
122         </section>
123     </div>
124 </main>
125
126 <footer>
127     <div class="limitedWidthBlockContainer footerMain">
128         <div class="limitedWidthBlock">
129             <div>
130                 
135 </div>
136 <div>
137     <p>10 quai de la charente <br />75019 Paris 19</p>
138 </div>
139 <div>
140     <p>Téléphone : 01 23 45 67 89</p>
141 </div>
142 <div>
143     <p>Email : support@name.com</p>
144 </div>
145 </div>
146 </div>
147 <div class="limitedWidthBlockContainer footerSecondary">
148     <div class="limitedWidthBlock">
149         <p>
150             © Copyright 2021 - 2042 | Openclassrooms by Openclassrooms | All
151             Rights Reserved | Powered by
152         </p>
153     </div>
154 </div>
155 </footer>
156 <script src="../js/product.js"></script>
157 <script src="../js/class.js"></script>
158 </body>
159 </html>
```

```
1  <!DOCTYPE html>
2  <html lang="fr">
3    <head>
4      <title>Cart</title>
5
6      <meta charset="utf-8" />
7      <meta name="description" content="Plateforme incroyable de e-commerce" />
8
9      <link rel="preconnect" href="https://fonts.googleapis.com" />
10     <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
11     <link
12       href="https://fonts.googleapis.com/css2?
family=Montserrat:wght@100;200;300;400;500;600;700;800;900&display=swap"
13       rel="stylesheet"
14     />
15     <link href="../css/style.css" rel="stylesheet" />
16     <link href="../css/cart.css" rel="stylesheet" />
17
18     <meta name="viewport" content="width=device-width, initial-scale=1" />
19   </head>
20
21   <body>
22     <header>
23       <div class="limitedWidthBlockContainer informations">
24         <div class="limitedWidthBlock">
25           <ul>
26             <li>
27               01 23 45 67 89
32             </li>
33             <li>
34               support@name.com
39             </li>
40             <li>
41               01 23 45 67 89
46             </li>
47           </ul>
48         </div>
49       </div>
50       <div class="limitedWidthBlockContainer menu">
51         <div class="limitedWidthBlock">
52           <a href="/index.html">
53             
58           </a>
59           <nav>
60             <ul>
61               <a href="/index.html"><li>Accueil</li></a>
62               <a href="/cart.html"><li>Panier</li></a>
63             </ul>
64           </nav>
65         </div>
```

```

66     </div>
67     
68 </header>
69
70 <main class="limitedWidthBlockContainer">
71     <div class="limitedWidthBlock" id="limitedWidthBlock">
72         <div class="cartAndFormContainer" id="cartAndFormContainer">
73             <h1>Votre panier</h1>
74             <section class="cart">
75                 <section id="cart_items">
76                     <!-- <article class="cart_item" data-id="{product-ID}">
77                         <div class="cart_item_img">
78                             
79                         </div>
80                         <div class="cart_item_content">
81                             <div class="cart_item_content_titlePrice">
82                                 <h2>Nom du produit</h2>
83                                 <p>42,00 €</p>
84                             </div>
85                             <div class="cart_item_content_settings">
86                                 <div class="cart_item_content_settings_quantity">
87                                     <p>Qté : </p>
88                                     <input type="number" class="itemQuantity" name="itemQuantity"
min="1" max="100" value="42">
89                                 </div>
90                                 <div class="cart_item_content_settings_delete">
91                                     <p class="deleteItem">Supprimer</p>
92                                 </div>
93                             </div>
94                         </div>
95                     </article> -->
96                 </section>
97
98                 <div class="cart_price">
99                     <div class="newFeatures">
100                         <!-- Trigger/Open The Modal -->
101                         <button id="myBtn">Vider le panier</button>
102                         <!-- The Modal -->
103                         <div id="myModal" class="modal">
104                             <!-- Modal content -->
105                             <div class="modal-content">
106                                 <div class="modalLineOne">
107                                     <div>Etes-vous certain de supprimer le panier?</div>
108                                     <span class="close">&times;</span>
109                                 </div>
110                                 <div class="modalLineTwo">
111                                     <button id="yesAnswer">OUI</button>
112                                     <button id="noAnswer">NON</button>
113                                 </div>
114                             </div>
115                         </div>
116                         <button id="backToIndex">Retour aux produits</button>
117                     </div>
118                     <p>
119                         Total (<span id="totalQuantity"><!-- 2 --></span> articles) :
120                         <span id="totalPrice"><!-- 84,00 --></span> €
121                     </p>
122                 </div>
123                 <div class="cart_order">
124                     <form method="get" class="cart_order_form">
125                         <div class="cart_order_form_question">
126                             <label for="firstName">Prénom: </label>
127                             <input type="text" name="firstName" id="firstName" required />
128                             <p id="firstNameErrorMsg">
129                                 <!-- ci est un message d'erreur -->
130                             </p>

```

```
131         </div>
132         <div class="cart_order_form_question">
133             <label for="lastName">Nom: </label>
134             <input type="text" name="lastName" id="lastName" required />
135             <p id="lastNameErrorMsg"></p>
136         </div>
137         <div class="cart_order_form_question">
138             <label for="address">Adresse: </label>
139             <input type="text" name="address" id="address" required />
140             <p id="addressErrorMsg"></p>
141         </div>
142         <div class="cart_order_form_question">
143             <label for="city">Ville: </label>
144             <input type="text" name="city" id="city" required />
145             <p id="cityErrorMsg"></p>
146         </div>
147         <div class="cart_order_form_question">
148             <label for="email">Email: </label>
149             <input type="email" name="email" id="email" required />
150             <p id="emailErrorMsg"></p>
151         </div>
152         <div class="cart_order_form_submit">
153             <input type="button" value="Commander !" id="order" />
154         </div>
155     </form>
156 </div>
157 </section>
158 </div>
159 </div>
160 </main>
161
162 <footer>
163     <div class="limitedWidthBlockContainer footerMain">
164         <div class="limitedWidthBlock">
165             <div>
166                 
171             </div>
172             <div>
173                 <p>10 quai de la charente <br />75019 Paris 19</p>
174             </div>
175             <div>
176                 <p>Téléphone : 01 23 45 67 89</p>
177             </div>
178             <div>
179                 <p>Email : support@name.com</p>
180             </div>
181         </div>
182     </div>
183     <div class="limitedWidthBlockContainer footerSecondary">
184         <div class="limitedWidthBlock">
185             <p>
186                 © Copyright 2021 - 2042 | Openclassrooms by Openclassrooms | All
187                 Rights Reserved | Powered by <3
188             </p>
189         </div>
190     </div>
191 </footer>
192 <script src="../../js/utils.js"></script>
193 <script src="../../js/class.js"></script>
194 <script src="../../js/cart.js"></script>
195 </body>
196 </html>
```



```
1  <!DOCTYPE html>
2  <html lang="fr">
3    <head>
4      <title>Confirmation</title>
5
6      <meta charset="utf-8">
7      <meta name="description" content="Plateforme incroyable de e-commerce">
8
9      <link rel="preconnect" href="https://fonts.googleapis.com">
10     <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
11     <link href="https://fonts.googleapis.com/css2?
family=Montserrat:wght@100;200;300;400;500;600;700;800;900&display=swap"
rel="stylesheet">
12     <link href="../css/style.css" rel="stylesheet" />
13     <link href="../css/confirmation.css" rel="stylesheet" />
14
15     <meta name="viewport" content="width=device-width, initial-scale=1">
16   </head>
17
18   <body>
19     <header>
20       <div class="limitedWidthBlockContainer informations">
21         <div class="limitedWidthBlock">
22           <ul>
23             <li>01 23 45 67 89</li>
24             <li>support@name.com</li>
25             <li>01 23 45 67 89</li>
26           </ul>
27         </div>
28       </div>
29       <div class="limitedWidthBlockContainer menu">
30         <div class="limitedWidthBlock">
31           <a href="/index.html">
32             
33           </a>
34           <nav>
35             <ul>
36               <a href="/index.html"><li>Accueil</li></a>
37               <a href="/cart.html"><li>Panier</li></a>
38             </ul>
39           </nav>
40         </div>
41       </div>
42       
43     </header>
44
45     <main class="limitedWidthBlockContainer">
46       <div class="limitedWidthBlock" id="limitedWidthBlock">
47
48         <div class="confirmation">
49           <p>Commande validée ! <br>Votre numéro de commande est : <span
id="orderId"><!-- 65431343444684674 --></span></p>
50         </div>
51
52       </div>
53     </main>
54
55     <footer>
56       <div class="limitedWidthBlockContainer footerMain">
57         <div class="limitedWidthBlock">
58           <div>
59             
60           </div>
```

```
61         <div>
62             <p>10 quai de la charente <br>75019 Paris 19</p>
63         </div>
64         <div>
65             <p>Téléphone : 01 23 45 67 89</p>
66         </div>
67         <div>
68             <p>Email : support@name.com</p>
69         </div>
70     </div>
71 </div>
72 <div class="limitedWidthBlockContainer footerSecondary">
73     <div class="limitedWidthBlock">
74         <p>© Copyright 2021 - 2042 | Openclassrooms by Openclassrooms | All Rights
Reserved | Powered by <3</p>
75     </div>
76 </div>
77 </footer>
78 <script src="../js/confirmation.js"></script>
79
80 </body>
81 </html>
```

```
1  /*****/
2  /*GENERAL*/
3  /*****/
4
5  body {
6      font-family: 'Montserrat', sans-serif;
7      margin: 0;
8      color: white;
9      min-width: 300px;
10 }
11 h1 {
12     font-size: 48px;
13     font-weight: 700;
14     margin-top: 135px;
15     text-align: center;
16 }
17 h2 {
18     font-weight: 400;
19     margin-bottom: 60px;
20     text-align: center;
21 }
22 .logo {
23     height: 100px;
24 }
25
26 /*maxwidth + centering*/
27 .limitedWidthBlockContainer {
28     display: flex;
29     justify-content: center;
30 }
31 .limitedWidthBlock {
32     width: 100%;
33     max-width: 1400px;
34     padding: 0 25px;
35 }
36
37 /*Colors*/
38 :root {
39     --main-color: #3498db;
40     --secondary-color: #2c3e50;
41     --text-color: #3d4c68;
42     --footer-text-color: #a6b0b3;
43     --footer-main-color: #3d424f;
44     --footer-secondary-color: #2d2f3e;
45     /*background-color: var(--main-color);*/
46 }
47
48
49
50
51 /*****/
52 /*Header*/
53 /*****/
54
55 /*informations*/
56 .informations {
57     font-size: 13px;
58     background-color: var(--main-color);
59 }
60 .informations img {
61     height: 17px;
62 }
63 .informations ul {
64     display: flex;
65     padding-left: 0;
66 }
```

```
67 .informations ul li {
68     display: flex;
69     list-style: none;
70     margin-right: 36px;
71 }
72 .informations__phone,
73 .informations__mail {
74     margin-right: 10px;
75 }
76
77 /*menu*/
78 .menu {
79     background-color: white;
80 }
81 .menu .limitedWidthBlock {
82     display: flex;
83     justify-content: space-between;
84 }
85 .menu a {
86     display: flex;
87     align-items: center;
88     text-decoration: none;
89 }
90 .menu nav {
91     display: flex;
92 }
93 .menu nav ul {
94     display: flex;
95     margin: 0;
96     line-height: 130px;
97 }
98 .menu nav ul li {
99     list-style: none;
100     margin-left: 50px;
101     color: var(--text-color);
102 }
103 .menu nav ul li:hover {
104     font-weight: 600;
105     color: #3498db !important;
106 }
107
108 /*banniere*/
109 .banniere {
110     width: 100%;
111     height: 550px;
112     object-fit: cover;
113     display: flex;
114 }
115
116
117
118
119 /*****/
120 /*Main*/
121 /*****/
122
123 main {
124     background-color: var(--main-color);
125 }
126
127 /*items*/
128 .items {
129     margin-bottom: 135px;
130     display: flex;
131     justify-content: space-around;
132     flex-wrap: wrap;
```

```
133 }
134 .items a {
135     width: 21%;
136     text-decoration: none;
137     color: unset;
138 }
139 .items a:hover article {
140     box-shadow: rgba(42, 18, 206, 0.9) 0 0 22px 6px;
141 }
142 .items article {
143     background-color: var(--secondary-color);
144     display: flex;
145     flex-direction: column;
146     align-items: center;
147     margin-bottom: 40px;
148     padding-top: 25px;
149     width: 100%;
150     border-radius: 10px;
151 }
152 .items article img {
153     width: 160px;
154     height: 160px;
155     object-fit: cover;
156     border-radius: 10px;
157 }
158 .items article h3 {
159     font-size: 22px;
160     font-weight: 700;
161     text-align: center;
162     text-transform: uppercase;
163 }
164 .items article p {
165     text-align: center;
166     padding-left: 23px;
167     padding-right: 23px;
168     box-sizing: border-box;
169     overflow: hidden;
170     text-overflow: ellipsis;
171
172     display: -webkit-box;
173     -webkit-line-clamp: 3;
174     -webkit-box-orient: vertical;
175     white-space: normal;
176 }
177
178 .items .error {
179     color: white;
180 }
181
182 .items .error a{
183     color: green;
184     text-decoration: underline;
185     font-weight: bold;
186 }
187
188
189
190 /*****/
191 /*Footer*/
192 /*****/
193
194 footer {
195     color: var(--footer-text-color);
196     display: flex;
197     flex-direction: column;
198     font-size: 14px;
```

```
199 }
200 footer .footerMain,
201 footer .footerSecondary {
202     width: 100%;
203 }
204
205 /*footerMain*/
206 .footerMain {
207     background-color: var(--footer-main-color);
208     padding: 50px 0;
209 }
210 .footerMain .limitedWidthBlock {
211     display: flex;
212     justify-content: space-between;
213 }
214 .footerMain .limitedWidthBlock div {
215     display: flex;
216     align-items: center;
217 }
218
219 /*footerSecondary*/
220 .footerSecondary {
221     background-color: var(--footer-secondary-color);
222     padding: 25px 0;
223 }
224
225
226
227
228 /*****/
229 /*Media queries*/
230 /*****/
231
232 @media (max-width: 991.98px) {
233     .informations ul {
234         display: flex;
235         justify-content: space-around;
236     }
237     .items a {
238         width: 42%;
239     }
240     .footerMain .limitedWidthBlock {
241         display: flex;
242         flex-direction: column;
243         align-items: center;
244     }
245     .footerSecondary .limitedWidthBlock p {
246         text-align: center;
247     }
248 }
249
250 @media (max-width: 575.98px) {
251     .informations ul {
252         flex-direction: column;
253         align-items: center;
254     }
255     .informations ul li {
256         margin-bottom: 14px;
257         margin-right: 0;
258     }
259     .menu .limitedWidthBlock {
260         display: flex;
261         flex-direction: column;
262         align-items: center;
263     }
264     .logo {
```

```
265         margin-top: 20px;
266     }
267     .menu nav ul {
268         padding: 0;
269         line-height: 80px;
270     }
271     .menu nav ul a:first-child li {
272         margin-left: 0;
273     }
274     .items a {
275         width: 75%;
276     }
277 }
```

```
1  /*****/
2  /*Product*/
3  /*****/
4
5  .errorQuantityAndColor {
6    color: red;
7    font-weight: bold;
8  }
9
10 .validQuantityAndColor {
11   color: green;
12   font-weight: bold;
13 }
14
15 /*Item Details*/
16 .item {
17   margin: 135px 0;
18   display: flex;
19   justify-content: center;
20 }
21 .item article {
22   display: flex;
23   flex-direction: column;
24   align-items: center;
25   width: 50%;
26 }
27 .item_img {
28   width: 100%;
29 }
30 .item_img img {
31   width: 100%;
32   border-radius: 15px;
33 }
34 .item_content {
35   width: 100%;
36   color: white;
37   display: flex;
38   flex-direction: column;
39   justify-content: space-between;
40 }
41 .item_content h1 {
42   text-align: left;
43   font-size: 48px;
44   font-weight: 700;
45   margin-top: 30px;
46   margin-bottom: 0;
47 }
48 .item_content_titlePrice,
49 .item_content_description,
50 .item_content_settings {
51   border-bottom: 1px solid white;
52 }
53 .item_content_titlePrice p {
54   margin-top: 4px;
55 }
56 .item_content_description {
57   display: flex;
58   flex-direction: column;
59 }
60 .item_content_description_title {
61   font-weight: 600;
62   font-size: 24px;
63   margin-bottom: 0;
64 }
65 .item_content_settings {
66   padding-top: 24px;
```



```
67     padding-bottom: 18px;
68 }
69 .item_content_settings_color {
70     margin-bottom: 10px;
71 }
72 .item_content_settings_color select,
73 .item_content_settings_quantity input {
74     color: var(--footer-secondary-color);
75     font-family: "Montserrat", sans-serif;
76     font-size: 14px;
77     border-radius: 20px;
78     border: 1px solid #767676;
79     text-align: center;
80     text-align-last: center;
81 }
82 .item_content_settings_color select:focus,
83 .item_content_settings_quantity input:focus {
84     outline: none;
85 }
86 .item_content_addButton {
87     display: flex;
88     justify-content: center;
89     padding-top: 32px;
90 }
91 .item_content_addButton button {
92     font-size: 22px;
93     border-radius: 40px;
94     border: 0;
95     background-color: var(--secondary-color);
96     color: white;
97     padding: 18px 28px;
98     cursor: pointer;
99 }
100 .item_content_addButton button:hover {
101     box-shadow: rgba(42, 18, 206, 0.9) 0 0 22px 6px;
102 }
103
104 /*API Connection error*/
105
106 #oneProduct .error {
107     color: white;
108     background-color: var(--secondary-color);
109     display: flex;
110     flex-direction: column;
111     align-items: center;
112     margin-bottom: 40px;
113     padding-top: 25px;
114     width: 100%;
115     border-radius: 10px;
116 }
117
118 #oneProduct .error a {
119     color: green;
120     text-decoration: underline;
121     font-weight: bold;
122 }
123
124 /*****/
125 /*Media queries*/
126 /*****/
127
128 @media (max-width: 991.98px) {
129     .item article {
130         width: 70%;
131     }
132     .item_content_settings_color,
```

```
133     .item__content__settings__quantity {
134         display: flex;
135         flex-direction: column;
136         align-items: center;
137     }
138     .item__content__settings__color {
139         margin-bottom: 26px;
140     }
141 }
142
143 @media (max-width: 575.98px) {
144     .item article {
145         width: 90%;
146     }
147 }
```

```
1  /*****/
2  /*GENERAL*/
3  /*****/
4
5  main {
6    padding-bottom: 90px;
7  }
8  h1 {
9    margin-bottom: 90px;
10 }
11 h2 {
12   text-align: start;
13 }
14
15 /*****/
16 /*Cart Items*/
17 /*****/
18 .cart_item {
19   display: flex;
20   justify-content: space-around;
21   border-top: 1px solid white;
22   padding-top: 35px;
23   margin-bottom: 30px;
24 }
25 .cart_item_img {
26   width: 25%;
27 }
28 .cart_item_img img {
29   width: 100%;
30   border-radius: 25px;
31 }
32 .cart_item_content {
33   width: 50%;
34   display: flex;
35   flex-direction: column;
36   justify-content: space-between;
37 }
38 .cart_item_content_titlePrice {
39   display: flex;
40   justify-content: space-between;
41 }
42 .cart_item_content_titlePrice h2,
43 .cart_item_content_titlePrice p {
44   margin: 0;
45 }
46 .cart_item_content_settings p {
47   margin: 0;
48 }
49 .cart_item_content_settings input {
50   color: var(--footer-secondary-color);
51   font-family: "Montserrat", sans-serif;
52   font-size: 14px;
53   border-radius: 20px;
54   border: 1px solid #767676;
55   text-align: center;
56   text-align-last: center;
57 }
58 .cart_item_content_settings input:focus {
59   outline: none;
60 }
61 .cart_item_content_settings_quantity {
62   display: flex;
63 }
64 .cart_item_content_settings_quantity p {
65   margin-right: 10px;
66   display: flex;
```

```
67 }
68 .cart_item_content_settings_delete p {
69     display: inline;
70 }
71 .cart_item_content_settings_delete p:hover {
72     cursor: pointer;
73     font-weight: 700;
74 }
75
76
77 /*****/
78 /*Price*/
79 /*****/
80 .cart_price {
81     border-top: 1px solid white;
82     padding-top: 35px;
83 }
84 .cart_price p {
85     text-align: end;
86     font-size: 22px;
87 }
88
89 /*****/
90 /*Quantity*/
91 /*****/
92 #cartQuantityErrorMsg {
93     display: none;
94     color: red;
95     font-weight: bold;
96 }
97
98 /*****/
99 /*Order*/
100 /*****/
101 .cart_order {
102     display: flex;
103     justify-content: center;
104     margin-top: 90px;
105 }
106 .cart_order_form {
107     width: 50%;
108 }
109 .cart_order_form_question {
110     display: flex;
111     flex-direction: column;
112     margin-bottom: 25px;
113 }
114 .cart_order_form_question input {
115     border-radius: 13px;
116     border: 0;
117     height: 26px;
118     margin-top: 4px;
119 }
120 .cart_order_form_question input:focus {
121     outline: none;
122 }
123 .cart_order_form_question p {
124     margin: 0;
125     color: #fbbcbcb;
126     font-size: 15px;
127     margin-left: 8px;
128 }
129 .cart_order_form_submit {
130     display: flex;
131     justify-content: center;
132 }
```

```
133 .cart_order_form_submit input {
134     font-size: 22px;
135     border-radius: 40px;
136     border: 0;
137     background-color: var(--secondary-color);
138     color: white;
139     padding: 18px 28px;
140     cursor: pointer;
141     margin-top: 40px;
142 }
143 .cart_order_form_submit input:hover {
144     box-shadow: rgb(42 18 206 / 90%) 0 0 22px 6px;
145 }
146
147 /******
148 /*New Features => "Vider le panier" button and "Retour aux produits" button */
149 /******
150
151 .newFeatures {
152     display: flex;
153     justify-content: center;
154     margin: 10px;
155     padding: 10px;
156 }
157 .newFeatures button {
158     font-size: 18px;
159     border-radius: 40px;
160     border: 0;
161     background-color: var(--secondary-color);
162     color: white;
163     padding: 10px 15px;
164     margin: 10px;
165     cursor: pointer;
166 }
167 .newFeatures button:hover {
168     box-shadow: rgba(42, 18, 206, 0.9) 0 0 22px 6px;
169 }
170
171 /* The Modal (background) */
172 .modal {
173     display: none; /* Hidden by default */
174     position: fixed; /* Stay in place */
175     z-index: 1; /* Sit on top */
176     left: 0;
177     top: 0;
178     width: 100%; /* Full width */
179     height: 100%; /* Full height */
180     overflow: auto; /* Enable scroll if needed */
181     background-color: rgb(0,0,0); /* Fallback color */
182     background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
183 }
184
185 /* Modal Content/Box */
186 .modal-content {
187     background-color: #fefefe;
188     margin: 15% auto; /* 15% from the top and centered */
189     padding: 20px;
190     border: 1px solid #888;
191     width: 80%;
192     color: black;
193     font-weight: bold;
194 }
195
196 /* Modal / The Close Button */
197 .close {
198     color: #aaa;
```

```
199     float: right;
200     font-size: 28px;
201     font-weight: bold;
202 }
203
204 .close:hover,
205 .close:focus {
206     color: black;
207     text-decoration: none;
208     cursor: pointer;
209 }
210
211 .modallLineOne {
212     display: flex;
213     justify-content: space-between;
214 }
215
216 .modallLineOne div {
217     align-self: center;
218 }
219
220 .modallLineTwo {
221     display: flex;
222     justify-content: space-around;
223 }
224
225 /******
226 /*API Connection error*/
227 /******
228
229 #cartAndFormContainer .error {
230     color: white;
231     background-color: var(--secondary-color);
232     display: flex;
233     flex-direction: column;
234     align-items: center;
235     margin: 40px 0;
236     padding-top: 25px;
237     width: 100%;
238     border-radius: 10px;
239 }
240
241 #cartAndFormContainer .error a {
242     color: green;
243     text-decoration: underline;
244     font-weight: bold;
245 }
246
247 #cartAndFormContainer .error h2 {
248     text-align: center;
249 }
250
251 /******
252 /*Media queries*/
253 /******
254
255 @media (max-width: 991.98px) {
256     .cart_item_content_titlePrice {
257         flex-direction: column;
258     }
259 }
260
261 @media (max-width: 575.98px) {
262     .cart_item {
263         flex-direction: column;
264         align-items: center;
```

```
265     }
266     .cart_item_img {
267         width: 50%;
268     }
269     .cart_item_content {
270         width: 90%;
271     }
272     .cart_item_content_titlePrice {
273         margin-bottom: 20px;
274     }
275     .cart_price p {
276         text-align: center;
277     }
278     .cart_order {
279         justify-content: center;
280     }
281     .cart_order_form {
282         width: 75%;
283     }
284 }
```

```
1  /*****  
2  /*Confirmation*/  
3  *****/  
4  .confirmation {  
5      display: flex;  
6      justify-content: center;  
7      margin-top: 85px;  
8      margin-bottom: 85px;  
9  }  
10  
11  .confirmation p {  
12      background-color: white;  
13      color: #3d424f;  
14      text-align: center;  
15      font-weight: 500;  
16      padding: 30px;  
17      border-radius: 25px;  
18  }
```



```
1  /* Recupération de l'ensemble des produits depuis l'API avec fetch
2  */
3
4  fetch("http://localhost:3000/api/products")
5
6  /* Conversion de la réponse au format JSON
7  */
8
9  .then(function (res) {
10    if (res.ok) {
11      return res.json();
12    }
13  })
14
15  /* Traitement des données avec une boucle for of
16  */
17  .then((data) => {
18
19    for (let check of data) {
20      let product = new Product(check);
21      document.getElementById("items").innerHTML +=
22        `<a href="./product.html?id=${product._id}">
23          <article>
24            
25            <h3 class="productName">${product.name}</h3>
26            <p class="productPrice">${product.price / 10}0€</p>
27            <p class="productDescription">${product.description}</p>
28          </article>
29        </a>`
30    }
31  })
32
33  /* Renvoi de l'erreur avec catch
34  */
35
36
37  .catch(function (error) {
38    let message = document.getElementById("items")
39    message.innerHTML += `
40      <article class="error">
41        <h2>Le contenu n'est pas disponible! Vérifier que vous avez lancé le serveur
42        local (sur le port 3000). Vous pourrez ensuite
43        <a href="index.html">recharger la page.</a></h2>
44      </article>`
45  })
```

```
1  /* Récupération de l'URL correspondant au produit sélectionné
2  */
3
4  const usedUrl = window.location.search;
5  const urlParams = new URLSearchParams(usedUrl);
6
7  /* Récupération de l'ID dans l'URL du produit sélectionné
8  */
9
10 const id = urlParams.get('id');
11
12
13 /* Recupération du produit sélectionné, grâce à l'ID, depuis l'API avec fetch
14 */
15
16 fetch("http://localhost:3000/api/products/" + id)
17
18   .then(res => res.json())
19
20   .then((data) => {
21
22     const product = new Product(data);
23
24     /* Affichage du produit sélectionné
25     */
26
27     document.getElementById("oneProduct").innerHTML = `<section class="item"
id="oneProduct">
28       <article>
29         <div class="item_img">
30           
31         </div>
32         <div class="item_content">
33
34           <div class="item_content_titlePrice">
35             <h1 id="title">${product.name}</h1>
36             <p>Prix : <span id="price">${product.price / 10}</span>0€</p>
37           </div>
38
39           <div class="item_content_description">
40             <p class="item_content_description_title">Description :</p>
41             <p id="description">${product.description}</p>
42           </div>
43
44           <div class="item_content_settings">
45             <div class="item_content_settings_color">
46               <label for="color-select">Choisir une couleur :</label>
47               <select name="color-select" id="colors">
48                 <option value="">--SVP, choisissez une couleur --</option>
49                 <option value=""></option>
50             </select>
51             </div>
52
53             <div class="item_content_settings_quantity" id="quantity">
54               <label for="itemQuantity">Nombre d'article(s) (1-100) :</label>
55               <input type="number" name="itemQuantity" min="1" max="100" value="0"
id="quantityValue">
56             </div>
57             </div>
58
59             <div class="item_content_addButton">
60               <button id="addToCart">Ajouter au panier</button>
61             </div>
62
63           </div>
64         </article>
```

```
65     </section>`;
66
67     /* Appel de la méthode d'instance pour le choix de la couleur du canapé (via le
68     menu déroulant) toujours en modifiant le HTML
69     */
70     product.selectColors(product);
71
72     /* Appel de la méthode d'instance pour vérifier les données du client, créer les
73     données et les envoyer au local storage
74     */
75     product.addToCart(product);
76
77     })
78
79     .catch(function (error) {
80         let message = document.getElementById("oneProduct");
81         message.innerHTML = `
82             <div class="error">
83                 <h2>Le contenu n'est pas disponible! Vérifier que vous avez lancé le serveur
84                 local (sur le port 3000). Vous pourrez ensuite
85                 <a href="index.html">recharger la page.</a></h2>
86             </div>`
87     })
```

```
1  /* Création d'une classe pour gérer les produits en POO (Programmation Orientée
   Objet)
2  */
3
4  class Product {
5      constructor(check) {
6          check && Object.assign(this, check);
7      }
8
9      /* Méthode d'instance permettant d'adapter le menu déroulant aux couleurs
      disponibles pour chacune des références
10     */
11
12     selectColors(product) {
13
14         document.getElementById("colors").innerHTML = `
15             <select name="color-select" id="colors">
16                 <option value="">--SVP, choisissez une couleur --</option>
17             </select>
18         </div>`;
19
20         let colorSelection = document.getElementById("colors");
21
22         for (let colorChoice of product.colors) {
23             let colorOption = document.createElement("option");
24             colorOption.setAttribute("value", colorChoice);
25             colorOption.innerHTML = colorChoice;
26             colorSelection.appendChild(colorOption);
27         }
28     }
29
30     /* Méthode d'instance permettant d'envoyer le choix du client au Local Storage
      (après vérification des données (couleurs et quantités) renseignées par le client)
31     */
32
33     addToCart(product) {
34
35         let addToCartBtn = document.getElementById("addToCart");
36         let messageStillPresent = false;
37         //Création du message apparaissant après vérification données client
38         let productValid = document.getElementById("quantity");
39         let productValidDiv = document.createElement("div");
40         productValid.appendChild(productValidDiv);
41
42         if (messageStillPresent) {
43             addToCartBtn.disabled = true;
44         }
45
46         // Effacer le message d'erreur si déjà présent
47
48         document.getElementById("colors").addEventListener("click", function
49 (event) {
50             event.preventDefault();
51             productValidDiv.innerHTML = ``;
52             messageStillPresent = false;
53
54             });
55         document.getElementById("quantityValue").addEventListener("click",
56 function (event) {
57             event.preventDefault();
58             productValidDiv.innerHTML = ``;
59             messageStillPresent = false;
60
61             });
62         addToCartBtn.addEventListener("click", function (event) {
63             event.preventDefault();
```

```
62         productValidDiv.innerHTML = ``;
63         messageStillPresent = false;
64     });
65
66
67     addToCartBtn.addEventListener("click", function (event) {
68         event.preventDefault();
69
70         let colorChoice = document.getElementById("colors").value;
71         let quantityChoice = document.getElementById("quantityValue").value;
72
73         let productAdded = {
74             kanapImg: product.imageUrl,
75             kanapName: product.name,
76             kanapPrice: product.price,
77             kanapId: product._id,
78             kanapQuantity: quantityChoice,
79             kanapColor: colorChoice,
80         };
81
82         // Vérification des données renseignées par le client (quantité et
couleur) et envoi vers le local storage si c'est correct
83
84         // => CAS N°1: les données sont bien renseignées et le Local Storage est
vide
85
86         if ((quantityChoice > 0) & (quantityChoice <= 100) & (colorChoice != "")
& (localStorage.getItem("all") == null)) {
87             let cart = []
88             cart.push(productAdded);
89             localStorage.setItem("all", JSON.stringify(cart));
90
91             productValidDiv.setAttribute("class", "validQuantityAndColor");
92             productValidDiv.innerHTML = `Félicitations, vous avez ajouté $
{product.name} en version ${colorChoice} au panier!`;
93
94             product.selectColors(product);
95             document.getElementById("quantityValue").value = 0;
96             messageStillPresent = true;
97         }
98
99         // => CAS N°2: les données sont bien renseignées et le Local Storage
n'est pas vide
100
101         else if ((quantityChoice > 0) & (quantityChoice <= 100) & (colorChoice !=
"")) & (localStorage.getItem("all") != null)) {
102             let cart = JSON.parse(localStorage.getItem("all"));
103             let missing = true;
104             let index;
105
106             for (let check of cart) {
107
108                 if ((productAdded.kanapId == check.kanapId) &&
(productAdded.kanapColor == check.kanapColor)) {
109
110                     missing = false;
111                     index = cart.indexOf(check);
112                     cart[index].kanapQuantity = +cart[index].kanapQuantity +
+productAdded.kanapQuantity;
113                     localStorage.setItem("all", JSON.stringify(cart));
114
115                     product.selectColors(product);
116                     document.getElementById("quantityValue").value = 0;
117                 }
118             }
119         }
```

```
120
121         if (missing) {
122             cart.push(productAdded);
123             localStorage.setItem("all", JSON.stringify(cart));
124
125             product.selectColors(product);
126             document.getElementById("quantityValue").value = 0;
127         }
128
129         productValidDiv.setAttribute("class", "validQuantityAndColor");
130         productValidDiv.innerHTML = `Félicitations, vous avez ajouté $
131 {product.name} en version ${colorChoice} au panier!`;
132         messageStillPresent = true;
133     }
134
135
136     // => CAS N°3: données mal renseignées => couleur OK mais quantité
incorrecte
137
138     else if (((quantityChoice <= 0) || (quantityChoice > 100)) &
139 (colorChoice != "")) {
140
141         productValidDiv.setAttribute("class", "errorQuantityAndColor");
142         productValidDiv.innerHTML = `Veuillez saisir une quantité entre 1 et
143 100!`;
144         messageStillPresent = true;
145     }
146
147     // => CAS N°4: données mal renseignées => couleur et quantité
incorrectes
148
149     else if (((quantityChoice <= 0) || (quantityChoice > 100)) &
150 (colorChoice == "")) {
151         productValidDiv.setAttribute("class", "errorQuantityAndColor");
152         productValidDiv.innerHTML = `Veuillez saisir une quantité entre 1 et
153 100 et sélectionner une couleur!`;
154         messageStillPresent = true;
155     }
156
157     // => CAS N°5: données mal renseignées => quantité OK mais couleur
incorrecte
158
159     else if ((quantityChoice > 0) & (quantityChoice < 100) & (colorChoice ==
160 "")) {
161         productValidDiv.setAttribute("class", "errorQuantityAndColor");
162         productValidDiv.innerHTML = `Veuillez sélectionner une couleur!`;
163         messageStillPresent = true;
164     }
165 }
```

```
1 // Récupération du panier
2
3 const cart = JSON.parse(localStorage.getItem("all"));
4
5 if (cart !== null) {
6   cart.sort(function (a, b) {
7     return (parseInt(a.kanapId) - parseInt(b.kanapId))
8   })
9 }
10
11 console.log(cart)
12
13 const regexForThree = (/^[A-Z-\s-éèëïäâù]{2,30}$/i)
14 const regexAddress = /^[A-Z-\s-éèëïäâù0-9]{5,100}$/i;
15 const regexEmail = (/^\b[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,50}\b/i)
16
17 const firstName = document.getElementById('firstName')
18 const lastName = document.getElementById('lastName')
19 const city = document.getElementById('city')
20 const address = document.getElementById('address')
21 const email = document.getElementById('email')
22
23
24 /* PARTIE 1 => Gestion du panier*/
25 if (cart !== null) {
26
27   // Tri du panier afin que les références soient triées par ID
28   cart.sort(function (a, b) {
29     return (parseInt(a.kanapId) - parseInt(b.kanapId))
30   })
31
32   cartView();
33   updateQuantityAndPrice();
34   customerModification();
35   clearOne();
36   backToIndex();
37   clearAll();
38
39   /* PARTIE 2 => Gestion de la commande*/
40   validateFirstName();
41   validateLastName();
42   validateCity();
43   validateAddress();
44   validateEmail();
45   customerInformationVerify()
46 }
47
48 if (cart == null || cart.length == 0 || cart.includes("")) {
49   emptyCart();
50 }
51
52
53 /* PARTIE 1 => Gestion du panier*/
54
55 // Affichage des produits ajoutés au panier
56 function cartView() {
57   // Parcourir le local storage pour consulter et afficher l'ensemble des choix du
  client
58   for (let product of cart) {
59
60     let subtotal = (product.kanapPrice / 10) * (product.kanapQuantity);
61
62     let article = document.createElement("article");
63     article.setAttribute("class", "cart_item");
64     article.setAttribute("data-id", product.kanapId);
65     article.setAttribute("id", "delete" + product.kanapId + product.kanapColor);
```

```

66     let articleList = document.getElementById('cart__items');
67     articleList.appendChild(article);
68     article.innerHTML = ` <div class="cart_item_img">
69         
70     </div>
71     <div class="cart_item_content">
72         <div class="cart_item_content_titlePrice">
73             <h2>${product.kanapName}</h2>
74             <p>Version: ${product.kanapColor}</p>
75             <p>${product.kanapPrice / 10}0€</p>
76         </div>
77         <div class="cart_item_content_settings">
78             <div class="cart_item_content_settings_quantity">
79                 <p>Qté : </p>
80                 <input type="number" data-id="${product.kanapId}"
data-version="${product.kanapColor}" class="itemQuantity" name="itemQuantity"
min="1" max="100" value="${product.kanapQuantity}">
81             </div>
82             <div id="cartQuantityErrorMsg">Merci d'indiquer une
quantité entre 1 et 100! Vous pouvez également supprimer la référence à l'aide du
bouton "Supprimer".</div>
83             <div class="cart_item_content_settings_delete">
84                 <p id="deleteItem${product.kanapId}"$
{product.kanapColor}>">Supprimer</p>
85             </div>
86             <div id="subtotal${product.kanapId}"$
{product.kanapColor}" class="cart_item_content_titlePrice">
87                 <h2>SOUS-TOTAL</h2>
88                 <p>${subtotal.toFixed(2)}€</p>
89             </div>
90         </div>
91     </div>
92 </article>`
93 }
94 }
95
96 // Actualisation des totaux de quantité et de prix lors de la modification des
quantités par le client sur la page panier
97 function updateQuantityAndPrice() {
98     let subTotalPrice;
99     let totalPrice = [];
100     let totalPriceSum;
101     let totalQuantity = [];
102     let totalQuantitySum;
103
104     for (let product of cart) {
105         subTotalPrice = (product.kanapPrice / 10) * (product.kanapQuantity);
106         totalPrice.push(subTotalPrice);
107         const reducer = (previousValue, currentValue) => previousValue + currentValue;
108         totalPriceSum = (totalPrice.reduce(reducer)).toFixed(2)
109     }
110
111     for (let product of cart) {
112         totalQuantity.push(parseInt(product.kanapQuantity));
113         const reducer2 = (previousValue, currentValue) => previousValue + currentValue;
114         totalQuantitySum = totalQuantity.reduce(reducer2)
115     }
116
117     document.getElementById('totalQuantity').innerHTML = `${totalQuantitySum}`;
118     document.getElementById('totalPrice').innerHTML = `${totalPriceSum}`;
119 }
120
121 // Mise à jour des modifications faites par le client
122 function customerModification() {
123     // Chercher les inputs dans lesquels la quantité peut être modifiée

```



```
124     let inputs = document.querySelectorAll('.itemQuantity');
125
126     inputs.forEach(input => {
127         // Ecouter les inputs trouvés
128         input.addEventListener('change', function () {
129
130             // Récupérer l'id dont la quantité a été modifiée
131             let modifiedId = input.dataset.id;
132             // Récupérer la couleur dont la quantité a été modifiée
133             let modifiedColor = input.dataset.version;
134             // Récupérer la nouvelle quantité
135             let newQuantity = input.value;
136
137             // Trouver quelle est la référence du panier où la valeur de l'input a changé
138             for (let product of cart) {
139
140
141                 if ((modifiedId == product.kanapId) && (modifiedColor ==
product.kanapColor)) {
142
143                     let index = cart.indexOf(product);
144
145                     if ((newQuantity <= 100) && (newQuantity > 0)) {
146
147                         cart[index].kanapQuantity = +newQuantity;
148                         // Intégrer la référence modifiée dans le panier
149                         localStorage.setItem("all", JSON.stringify(cart));
150                         // Mise à jour du sous-total
151                         let subtotal = (product.kanapPrice / 10) * (product.kanapQuantity);
152                         let subtotalId = "subtotal" + modifiedId + modifiedColor;
153                         document.getElementById(subtotalId).innerHTML = `
154                             <h2>SOUS-TOTAL</h2>
155                             <p>${subtotal.toFixed(2)}€</p>
156                         </div>`
157                         //Mise à jour des totaux
158                         updateQuantityAndPrice();
159                     }
160                     else {
161                         let cartQuantityErrorMsg =
document.getElementById('cartQuantityErrorMsg')
162                         cartQuantityErrorMsg.style.display = "unset";
163                         // Ci-dessous cela permet de revenir à la quantité tapée initialement et
ne pas garder input.value car sinon cela crée des bugs.
164                         input.value = cart[index].kanapQuantity;
165                         updateQuantityAndPrice();
166                     }
167                 }
168             }
169         });
170     });
171 }
172
173 // Suppression d'un produit (pour la couleur sélectionnée) avec le bouton
"Supprimer"
174 function clearOne() {
175
176     for (let product of cart) {
177
178         let index = cart.indexOf(product);
179         let articleDivSearch = document.getElementById("delete" + product.kanapId +
product.kanapColor);
180         let deleteButton = document.getElementById("deleteItem" + product.kanapId +
product.kanapColor);
181
182         deleteButton.addEventListener("click", function (event) {
183
```

```
184     event.preventDefault();
185     cart.splice(index, 1);
186     localStorage.setItem("all", JSON.stringify(cart));
187     articleDivSearch.parentNode.removeChild(articleDivSearch);
188     updateQuantityAndPrice();
189     emptyCart();
190   })
191 }
192 }
193
194 // Affichage du panier vide, d'un message l'indiquant, et ajout d'un bouton pour
// aller à la page d'accueil
195 function emptyCart() {
196   if (cart === null || cart.length === 0 || cart.includes("[]")) {
197     let sectionDivSearch = document.getElementById('cartAndFormContainer')
198     sectionDivSearch.innerHTML =
199       `<h1>Votre panier est vide!</h1>
200         <div class="newFeatures">
201           <button id="backToIndex">Retour aux produits
202         </button>
203         </div>`
204     backToIndex();
205   }
206 }
207
208 // Optimisation du retour à la page d'accueil en évitant de revenir en haut de la
// page
209 function backToIndex() {
210   document.getElementById('backToIndex').addEventListener('click', function (event)
211   {
212     event.preventDefault();
213     window.location.href = "index.html";
214   })
215 }
216
217 // Suppression de l'ensemble du panier
218 function clearAll() {
219   if (cart.length !== 0) {
220     openModal();
221     let yesAnswer = document.getElementById("yesAnswer")
222     let noAnswer = document.getElementById("noAnswer")
223     yesAnswer.addEventListener('click', function (event) {
224       event.preventDefault();
225       for (let i = 0; i < cart.length; i++) {
226         cart.splice(0, cart.length);
227       }
228       localStorage.setItem("all", JSON.stringify(cart));
229       emptyCart();
230     })
231     noAnswer.addEventListener('click', function (event) {
232       event.preventDefault();
233       var modal = document.getElementById("myModal");
234       modal.style.display = "none";
235     })
236   }
237 }
238
239 /* PARTIE 2 => Gestion de la commande*/
240
241 // Vérification du prénom
242 function validateFirstName() {
243   firstName.addEventListener('change', function () {
244     if (regExForThree.test(firstName.value)) { console.log("regExfirstName => OK") }
245     else { firstNameErrorMsg.innerHTML = `Saisie incorrecte! (Taper entre 2 et 30
// caractères sans caractère spécial)` }
246   })
247 }
```

```
246     })
247   }
248
249   // Vérification du nom
250   function validateLastName() {
251     lastName.addEventListener('change', function () {
252       if (regexForThree.test(lastName.value)) { console.log("regexlastName => OK") }
253       else { lastNameErrorMsg.innerHTML = `Saisie incorrecte! (Taper entre 2 et 30
caractères sans chiffre ni caractère spécial)` }
254     })
255   }
256
257   // Vérification de la ville
258   function validateCity() {
259     city.addEventListener('change', function () {
260       if (regexForThree.test(city.value)) { console.log("regexcity => OK") }
261       else { cityErrorMsg.innerHTML = `Saisie incorrecte! (Taper entre 2 et 30
caractères sans chiffre ni caractère spécial)` }
262     })
263   }
264
265   // Vérification de l'adresse
266   function validateAddress() {
267     address.addEventListener('change', function () {
268       if (regexAddress.test(address.value)) { console.log("regexAddress => OK") }
269       else { addressErrorMsg.innerHTML = `Saisie incorrecte! (Taper entre 5 et 100
caractères sans caractère spécial)` }
270     })
271   }
272
273   // Vérification de l'adresse mail
274   function validateEmail() {
275     email.addEventListener('change', function () {
276       if (regexEmail.test(email.value)) { console.log("regexEmail => OK") }
277       else { emailErrorMsg.innerHTML = `Votre adresse mail n'est pas valide.` }
278     })
279   }
280
281   // Vérification de l'ensemble des informations client, création de l'objet
nécessaire pour la requête POST
282   function customerInformationVerify() {
283     document.getElementById('order').addEventListener('click', function () {
284
285       const products = [];
286
287       for (let i = 0; i < cart.length; i++) {
288         products.push(cart[i].kanapId)
289       }
290
291       if (regexForThree.test(firstName.value) &&
292         regexForThree.test(lastName.value) &&
293         regexForThree.test(city.value) &&
294         regexAddress.test(address.value) &&
295         regexEmail.test(email.value)
296       ) {
297         const contact = {
298           firstName: firstName.value,
299           lastName: lastName.value,
300           city: city.value,
301           address: address.value,
302           email: email.value,
303         }
304       }
305       const send = {
306         products,
307         contact
```

```
308     }
309     postRequest(send);
310   }
311 })
312 }
313
314 // Mise en place de la requête POST
315 function postRequest(param) {
316   fetch("http://localhost:3000/api/products/order", {
317     method: "POST",
318     headers: {
319       'Accept': 'application/json',
320       'Content-Type': 'application/json'
321     },
322     body: JSON.stringify(param)
323   })
324   .then(function (res) {
325     if (res.ok) {
326       return res.json();
327     }
328   })
329   .then(function (res) {
330     //Effacer le local storage
331     localStorage.clear();
332     //Insérer les données récupérées
333     localStorage.setItem("orderId", res.orderId);
334
335     //Aller à la page de confirmation affichant le numéro de commande
336     document.location.href = "confirmation.html";
337   })
338   .catch(function (err) {
339     let message = document.getElementById("cartAndFormContainer");
340     message.innerHTML = `
341     <div class="error">
342       <h2>Une erreur s'est produite! Vérifier que vous avez lancé le serveur local
343       (sur le port 3000). Vous pourrez ensuite
344       <a href="cart.html">recharger la page</a> et finaliser votre commande.</h2>
345     </div>`
346   })
347 }
348
349
350
351
352
```

```
1  finalMessage();
2
3  // Récupération et affichage du numéro de commande
4  function finalMessage () {
5
6      let orderFind = document.getElementById('orderId');
7      //Récupérer le numéro de commande dans le local storage
8      let orderId = localStorage.getItem("orderId");
9      orderFind.innerText = orderId;
10     //Insérer le numéro de commande à la fin de l'URL
11     window.location.hash = orderId;
12     //Vider le local storage
13     localStorage.clear();
14 }
```

```
1  function openModal() {
2    // Get the modal
3    var modal = document.getElementById("myModal");
4
5    // Get the button that opens the modal
6    var btn = document.getElementById("myBtn");
7
8    // Get the <span> element that closes the modal
9    var span = document.getElementsByClassName("close")[0];
10
11   // When the user clicks on the button, open the modal
12   btn.onclick = function() {
13     modal.style.display = "block";
14   }
15
16   // When the user clicks on <span> (x), close the modal
17   span.onclick = function() {
18     modal.style.display = "none";
19   }
20
21   // When the user clicks anywhere outside of the modal, close it
22   window.onclick = function(event) {
23     if (event.target == modal) {
24       modal.style.display = "none";
25     }
26   }
27 }
```

```
1  const http = require('http');
2  const app = require('./app');
3
4  const normalizePort = val => {
5    const port = parseInt(val, 10);
6
7    if (isNaN(port)) {
8      return val;
9    }
10   if (port >= 0) {
11     return port;
12   }
13   return false;
14 };
15 const port = normalizePort(process.env.PORT || '3000');
16 app.set('port', port);
17
18 const errorHandler = error => {
19   if (error.syscall !== 'listen') {
20     throw error;
21   }
22   const address = server.address();
23   const bind = typeof address === 'string' ? 'pipe ' + address : 'port: ' + port;
24   switch (error.code) {
25     case 'EACCES':
26       console.error(bind + ' requires elevated privileges. ');
27       process.exit(1);
28     case 'EADDRINUSE':
29       console.error(bind + ' is already in use. ');
30       process.exit(1);
31     default:
32       throw error;
33   }
34 };
35
36 const server = http.createServer(app);
37
38 server.on('error', errorHandler);
39 server.on('listening', () => {
40   const address = server.address();
41   const bind = typeof address === 'string' ? 'pipe ' + address : 'port ' + port;
42   console.log('Listening on ' + bind);
43 });
44
45 server.listen(port);
```

```
1  const express = require('express');
2  const path = require('path');
3
4  const productRoutes = require('./routes/product');
5
6  const app = express();
7
8  app.use((req, res, next) => {
9    res.setHeader('Access-Control-Allow-Origin', '*');
10   res.setHeader('Access-Control-Allow-Headers', 'Origin, X-Requested-With, Content,
    Accept, Content-Type, Authorization');
11   res.setHeader('Access-Control-Allow-Methods', 'GET, POST, PUT, DELETE, PATCH,
    OPTIONS');
12   next();
13 });
14
15 app.use('/images', express.static(path.join(__dirname, 'images')));
16 app.use(express.static('images'));
17
18 app.use(express.urlencoded({extended: true}));
19 app.use(express.json());
20
21 app.use('/api/products', productRoutes);
22
23 module.exports = app;
```

```
1  const express = require('express');
2  const router = express.Router();
3
4  const productCtrl = require('../controllers/product');
5
6  router.get('/', productCtrl.getAllProducts);
7  router.get('/:id', productCtrl.getOneProduct);
8  router.post('/order', productCtrl.orderProducts);
9
10 module.exports = router;
```

```
1  const products = [  
2    {  
3      "colors": ["Blue", "White", "Black"],  
4      "_id": "107fb5b75607497b96722bda5b504926",  
5      "name": "Kanap Sinopé",  
6      "price": 1849,  
7      "imageUrl": "kanap01.jpeg",  
8      "description": "Ce canapé 2 personnes dispose d'une assise très confortable pour  
des moments agréables à petit prix.",  
9      "altTxt": "Photo d'un canapé bleu, deux places"  
10   },  
11   {  
12     "colors": ["Black/Yellow", "Black/Red"],  
13     "_id": "415b7cacb65d43b2b5c1ff70f3393ad1",  
14     "name": "Kanap Cyllène",  
15     "price": 4499,  
16     "imageUrl": "kanap02.jpeg",  
17     "description": "Pouvant accueillir 4 personnes, notre référence haut de gamme  
peut se transformer en lit 2 personnes.",  
18     "altTxt": "Photo d'un canapé jaune et noir, quatre places"  
19   },  
20   {  
21     "colors": ["Green", "Red", "Orange"],  
22     "_id": "055743915a544fde83cfd904935ee7",  
23     "name": "Kanap Calycé",  
24     "price": 3199,  
25     "imageUrl": "kanap03.jpeg",  
26     "description": "Ce canapé d'angle au design sobre vous permettra de faire dormir  
un invité si besoin.",  
27     "altTxt": "Photo d'un canapé d'angle, vert, trois places"  
28   },  
29   {  
30     "colors": ["Pink", "White"],  
31     "_id": "a557292fe5814ea2b15c6ef4bd73ed83",  
32     "name": "Kanap Autonoé",  
33     "price": 1499,  
34     "imageUrl": "kanap04.jpeg",  
35     "description": "Cette méridienne classique élève son potentiel à l'aide d'une  
transformation aisée en lit 1 personne.",  
36     "altTxt": "Photo d'un canapé rose, une à deux place"  
37   },  
38   {  
39     "colors": ["Grey", "Purple", "Blue"],  
40     "_id": "8906dfda133f4c20a9d0e34f18adcf06",  
41     "name": "Kanap Eurydomé",  
42     "price": 2249,  
43     "imageUrl": "kanap05.jpeg",  
44     "description": "Ce canapé 3 places à un prix exceptionnel dispose d'un design  
s'adaptant à tous les intérieurs.",  
45     "altTxt": "Photo d'un canapé gris, trois places"  
46   },  
47   {  
48     "colors": ["Grey", "Navy"],  
49     "_id": "77711f0e466b4ddf953f677d30b0efc9",  
50     "name": "Kanap Hélicé",  
51     "price": 999,  
52     "imageUrl": "kanap06.jpeg",  
53     "description": "Notre premier prix dispose d'un confort bluffant pour 2  
personnes.",  
54     "altTxt": "Photo d'un canapé gris, deux places"  
55   },  
56   {  
57     "colors": ["Red", "Silver"],  
58     "_id": "034707184e8e4eefb46400b5a3774b5f",  
59     "name": "Kanap Thyoné",  
60     "price": 1999,
```

```
61     "imageUrl": "kanap07.jpeg",
62     "description": "Le design attrayant et vintage de ce canapé 2 places vous
    permettra de compléter parfaitement votre intérieur.",
63     "altTxt": "Photo d'un canapé rouge, deux places"
64   },
65   {
66     "colors": ["Pink", "Brown", "Yellow", "White"],
67     "_id": "a6ec5b49bd164d7fbel0f37b6363f9fb",
68     "name": "Kanap orthosie",
69     "price": 3999,
70     "imageUrl": "kanap08.jpeg",
71     "description": "Au design cosy, ce canapé 3 places est facilement convertible en
    couchage 2 personnes.",
72     "altTxt": "Photo d'un canapé rose, trois places"
73   }
74 ];
75
76 exports.find = () => {
77   return new Promise((resolve, reject) =>
    resolve(JSON.parse(JSON.stringify(products))));
78 }
79
80 exports.findById = (id) => {
81   return new Promise((resolve, reject) =>
82     resolve(JSON.parse(JSON.stringify(products)).find(product =>
83       product._id === id)
84     )
85   );
86 }
87
88
89
```

```
1  const uuid = require('uuid/v1');
2  const Product = require('../models/Product');
3
4  exports.getAllProducts = (req, res, next) => {
5    Product.find().then(
6      (products) => {
7        const mappedProducts = products.map((product) => {
8          product.imageUrl = req.protocol + '://' + req.get('host') + '/images/' +
product.imageUrl;
9          return product;
10        });
11        res.status(200).json(mappedProducts);
12      }
13    ).catch(
14      () => {
15        res.status(500).send(new Error('Database error!'));
16      }
17    );
18  };
19
20  exports.getOneProduct = (req, res, next) => {
21    Product.findById(req.params.id).then(
22      (product) => {
23        if (!product) {
24          return res.status(404).send(new Error('Product not found!'));
25        }
26        product.imageUrl = req.protocol + '://' + req.get('host') + '/images/' +
product.imageUrl;
27        res.status(200).json(product);
28      }
29    ).catch(
30      () => {
31        res.status(500).send(new Error('Database error!'));
32      }
33    )
34  };
35
36  /**
37   *
38   * Expects request to contain:
39   * contact: {
40   *   firstName: string,
41   *   lastName: string,
42   *   address: string,
43   *   city: string,
44   *   email: string
45   * }
46   * products: [string] <-- array of product _id
47   *
48   */
49  exports.orderProducts = (req, res, next) => {
50    if (!req.body.contact ||
51        !req.body.contact.firstName ||
52        !req.body.contact.lastName ||
53        !req.body.contact.address ||
54        !req.body.contact.city ||
55        !req.body.contact.email ||
56        !req.body.products) {
57      return res.status(400).send(new Error('Bad request!'));
58    }
59    let queries = [];
60    for (let productId of req.body.products) {
61      const queryPromise = new Promise((resolve, reject) => {
62        Product.findById(productId).then(
63          (product) => {
64            if (!product) {
```

```
65         reject('Product not found: ' + productId);
66     }
67     product.imageUrl = req.protocol + '://' + req.get('host') + '/images/' +
product.imageUrl;
68     resolve(product);
69 }
70 ).catch(
71     () => {
72         reject('Database error!');
73     }
74 )
75 });
76 queries.push(queryPromise);
77 }
78 Promise.all(queries).then(
79     (products) => {
80         const orderId = uuid();
81         return res.status(201).json({
82             contact: req.body.contact,
83             products: products,
84             orderId: orderId
85         })
86     }
87 ).catch(
88     (error) => {
89         return res.status(500).json(new Error(error));
90     }
91 );
92 };
```

```
1  # Kanap #
2
3  This is the front end and back end server for Project 5 of the Web Developer path.
4
5  ### Back end Prerequisites ###
6
7  You will need to have Node and `npm` installed locally on your machine.
8
9  ### Back end Installation ###
10
11 Clone this repo. From the "back" folder of the project, run `npm install`. You
12 can then run the server with `node server`.
13 The server should run on `localhost` with default port `3000`. If the
14 server runs on another port for any reason, this is printed to the
15 console when the server starts, e.g. `Listening on port 3001`.
```