

main.js

```
1 var appURL = location.href;
2 var appPath = location.pathname;
3 var appRootPath = "/" + appPath.split("/")[1] + "/";
4
5 /**
6  *
7  * Bulma
8  */
9 (function() {
10     let burger = document.querySelector('.burger');
11     let menu = document.querySelector('#'+burger.dataset.target);
12     burger.addEventListener('click', function() {
13         burger.classList.toggle('is-active');
14         menu.classList.toggle('is-active');
15     });
16
17     onClickViewCatItem();
18
19     window.onload = onLoadActions;
20 })();
21
22 /**
23  * On Load Actions
24  *
25  */
26 function onLoadActions() {
27     toggleElemBy('id', 'products-tabs', true);
28 }
29
30
31 /**
32  * Show/Hide and element by id, class or CSS Selector
33  * @param by
34  * @param val
35  * @param hide
36  *
37  */
38 function toggleElemBy(by, val, hide) {
39     let elem = null;
40     if (by == 'id') {
41         elem = document.getElementById(val);
42     } else if (by == 'class') {
43         if (document.getElementsByClassName(val).length > 0)
44             {
```

main.js

```
45         elem = document.getElementsByClassName(val)[0];
46     }
47     } else if (by == 'css') {
48         elem = document.querySelector(val);
49     }
50
51     if (elem != null) {
52         if (hide == true) {
53             elem.style.display = "none";
54         } else {
55             elem.style.display = "block";
56         }
57     }
58 }
59
60 /**
61  *
62  * Navbar Toggle
63  */
64 document.addEventListener('DOMContentLoaded', () => {
65     // Get all "navbar-burger" elements
66     const $navbarBurgers =
67         Array.prototype.slice.call(document.querySelectorAll('.navbar-
68         burger'), 0);
69
70     // Check if there are any navbar burgers
71     if ($navbarBurgers.length > 0) {
72
73         // Add a click event on each of them
74         $navbarBurgers.forEach( el => {
75             el.addEventListener('click', () => {
76
77                 // Get the target from the "data-target" attribute
78                 const target = el.dataset.target;
79                 const $target = document.getElementById(target);
80
81                 // Toggle the "is-active" class on both the "navbar-
82                 burger" and the
83                 // "navbar-menu"
84                 el.classList.toggle('is-active');
85                 $target.classList.toggle('is-active');
```

```

86     }
87 });
88
89 /**
90  *
91  * Tabs Toggle
92  */
93 document.querySelectorAll("#tabNav li").forEach(function(navEl) {
94     navEl.onclick = function() { toggleTab(this.id,
95         this.dataset.target); }
96     // toggleTabContent(navEl.dataset.target);
97     if (navEl.className.includes("is-active")) {
98         toggleTab(navEl.id, navEl.dataset.target);
99     }
100 });
101 function toggleTab(selectedNav, targetId) {
102     let navEls = document.querySelectorAll("#tabNav li");
103     navEls.forEach(function(navEl) {
104         if (navEl.id == selectedNav) {
105             navEl.classList.add("is-active");
106         } else {
107             if (navEl.classList.contains("is-active")) {
108                 navEl.classList.remove("is-active");
109             }
110         }
111     });
112
113     toggleTabContent(targetId);
114 }
115
116 function toggleTabContent(targetId) {
117     let tabs = document.querySelectorAll(".tab-pane");
118
119     tabs.forEach(function(tab) {
120         if (tab.id == targetId) {
121             tab.style.display = "block";
122         } else {
123             tab.style.display = "none";
124         }
125     });
126 }
127
128 function hideAllCatsViewAllLink() {

```

main.js

```
129     let viewCatItemsBtns = document.querySelectorAll('a[id^="view-  
cat-item-"]');  
130     viewCatItemsBtns.forEach(function (el) {  
131         toggleElemBy('id', el.id, true);  
132     });  
133 }  
134  
135 function onClickViewCatItem() {  
136     let viewCatItemsBtns = document.querySelectorAll('a[id^="view-  
cat-item-"]');  
137  
138     viewCatItemsBtns.forEach(function (el) {  
139         el.addEventListener('click', () => {  
140  
141             toggleElemBy('id', 'page-products', true);  
142             toggleElemBy('id', 'products-tabs', false);  
143             hideAllCatsViewAllLink();  
144  
145             let catId = el.id.split("view-cat-item-")[1];  
146             document.getElementById("cat-" + catId).click();  
147  
148         });  
149     });  
150 }  
151  
152  
153 function onClickAddToCartBtn(el) {  
154     let productId = el.id.split("product-")[1];  
155     let addToCartUrl = appRootPath + 'Api/Cart';  
156     let qty = el.parentNode.parentNode.querySelector(".qtyInput  
input");  
157  
158     let qs = "addToCart=" + productId + "&" + "qty=" + qty.value;  
159     doSimpleAjax(addToCartUrl, qs, addToCartResult);  
160  
161 }  
162  
163 function addToCartResult(request)  
164 {  
165     // TODO: increase num on top cart btn  
166     let resp = JSON.parse(JSON.stringify(request.responseText));  
167     let checkoutLink =  
document.querySelectorAll("a[href$='Checkout']");  
168
```

main.js

```
169     if (checkoutLink.length < 1) {
170         updatePage();
171     }
172     console.log(resp);
173 }
174
175
176 function updateQty(el) {
177     console.log("updating item qty");
178     let cartUrl = appRootPath + 'Api/Cart';
179     let productNum = el.parentNode.parentNode.id.split('p-')[1];
180     let productQty = document.getElementById("qty-" +
        productNum).value;
181
182     console.log(productNum);
183
184     let qs = "updateQty=" + productNum + "&qty=" + productQty;
185     doSimpleAjax(cartUrl, qs, updateQtyResult);
186 }
187
188 function updateQtyResult(request)
189 {
190     // TODO: increase num on top cart btn
191     console.log('updated qty in cart', request.responseText);
192     updatePage();
193 }
194
195
196 function updateItemsQty(el) {
197     console.log("updating items qty");
198     let cartItemsTable = document.getElementById("cartItems");
199     let items =
        cartItemsTable.querySelector("tbody").getElementsByTagName("tr");
200     let cartUrl = appRootPath + 'Api/Cart';
201
202     for(let i=0; i<items.length; i++) {
203         let item = items[i];
204         let productNum = item.id.split('p-')[1];
205         let productQty = document.getElementById("qty-" +
            productNum).value;
206
207         let qs = "updateQty=" + productNum + "&qty=" + productQty;
208         doSimpleAjax(cartUrl, qs, updateQtyItemsResult);
209     }
```

main.js

```
210
211     updatePage();
212 }
213
214 function updateQtyItemsResult(request)
215 {
216     // TODO: increase num on top cart btn
217     console.log('updated qty in cart', request.responseText);
218 //     updatePage();
219 }
220
221
222 function deleteCartItem(el) {
223     let cartUrl = appRootPath + 'Api/Cart';
224     let productNum = el.parentNode.parentNode.id.split('p-')[1];
225
226     let qs = "removeItem=" + productNum;
227     doSimpleAjax(cartUrl, qs, deleteCartItemResult);
228 }
229
230 function deleteCartItemResult(request)
231 {
232     let req = request.responseText;
233     updatePage();
234
235     if (req != null && req.action == "REMOVE") {
236         let el = document.getElementById('p-' + req.actionReq);
237         console.log(el);
238         console.log(el.parentNode.parentNode);
239         let i = el.parentNode.parentNode.rowIndex;
240         document.getElementById("cartItems").deleteRow(i);
241         console.log('deleted from cart', req);
242     }
243 }
244
245 function doCheckout() {
246     location.href = appRootPath + "Checkout";
247 }
248
249 function updatePage() {
250     location.reload(true);
251 }
252
253 function gotoHomePage() {
```

main.js

```
254     location.href = appRootPath;
255 }
256
257 function doLogin() {
258     location.href = appPath + "?doLogin=";
259 }
260
261 function gotoSearchPg() {
262     let query = document.getElementById("sQuery");
263     let searchParams = "";
264     if (query.value != null && query.value != "") {
265         searchParams = "?query=" + query.value;
266     }
267     location.href = appRootPath + "Search" + searchParams;
268 }
269
270
271 function doSimpleAjax(address, data, handler)
272 {
273     var request = new XMLHttpRequest();
274     request.onreadystatechange = function() {handler(request);};
275     request.open("GET", (address + "?" + data), true);
276     request.send(null);
277 }
278
279
280
281
```