

# Diario di lavoro

Luogo	SAMT
Data	27.02.2019

## Lavori svolti

La prima cosa che ho fatto sta mattina è stato aggiornare il github, ovvero inserendo i file che mancano e quelli che già c'erano ma sono stati modificati, dopodiché ho modificato il file "Readme.md" permettendo una migliore navigazione nel progetto una volta aperto il git. Dopo ho creato il file "dati\_progetto.docx" che contiene il link al sito e i dati degli utenti già creati per testarlo.

Una volta messo tutto a posto e aggiornato, ho continuato la funzione che va a mettere i prodotti nel carrello della pagina per poterli visualizzare. Prima ho modificato la funzione che prende i valori del carrello per fare in modo che inserisca tutto in un array, per controllarlo più correttamente nella funzione dopo.

```

343 function getCart(){
344     xhttp.onreadystatechange = function () {
345         if (this.readyState === 4 && this.status === 200) {
346
347             //Prendo i valori passati dal server e li metto in un array
348             var obj = JSON.parse(xhttp.responseText);
349
350             //variabile che contiene tutte le chiavi
351             var codedKeys = new Array();
352
353             if(typeof (obj[0]) == "object"){
354                 for(var i = 0; i < obj.length; i++){
355                     var codedKey = encode(obj[i]['nome_prodotto'], obj[i]['prezzo_prodotto'], obj[i]['quantita_prodotto']);
356                     codedKeys.push(codedKey);
357                 }
358             }else{
359                 var codedKey = encode(obj['nome_prodotto'], obj['prezzo_prdotto'], obj['quantita_prodotto']);
360                 codedKeys.push(codedKey);
361             }
362             //richiamo la funzione che inserisce i prodotti
363             modifyCart(codedKeys);
364         }
365     }
366     xhttp.open( method: "POST", url: "/gestionevendita2018/customer/getCart", async: true);
367     xhttp.send();
368 }

```

Figura 1 funzione getCart

Prima di codificare i valori creo un array (riga 351) e dopodiché i valori li inserisco dentro esso, e alla fine lo passo con tutti i dati all'interno.

```

374 function modifyCart(codedKeys){
375     //Decodifico tutte le chiavi
376     var keys = new Array();
377     for(var i = 0; i < codedKeys.length; i++) {
378         keys.push(decode(codedKeys[i]));
379     }
380
381     //Prendo il corpo del div in cui inserire i campi
382     var divBody = document.getElementById( 'elementId: 'cart-list');
383     divBody.innerHTML = "";
384
385     xhttp.onreadystatechange = function () {
386         if (this.readyState === 4 && this.status === 200) {

```

*Figura 2 JavaScript modifyCart 1*

All'inizio della funzione prendo tutti i valori delle chiavi le decodifico e le inserisco in un array. Dopodiché prendo il corpo in cui inserire ogni prodotto. Una volta controllo che sia andato a buon fine la richiesta al php.

```

389 var obj = JSON.parse(xhr.responseText);
390
391 //Prendo i div principale per i prodotti
392 var divContainer = document.createElement( tagName: "div");
393 divContainer.setAttribute( qualifiedName: "class", value: "single-cart-item");
394
395 var a = document.createElement( tagName: "a");
396 a.setAttribute( qualifiedName: "class", value: "product-image");
397 var img = document.createElement( tagName: "img");
398 var src = "http://saminfo.ch/gestionevendita2018/" + obj[0]["img"];
399 img.setAttribute( qualifiedName: "class", value: "cart-thumb");
400 img.setAttribute( qualifiedName: "src", src);
401 img.setAttribute( qualifiedName: "alt", obj[0]['nome_prodotto']);
402 img.setAttribute( qualifiedName: "title", obj[0]['nome_prodotto']);
403 a.appendChild(img);
404
405 var div = document.createElement( tagName: "div");
406 div.setAttribute( qualifiedName: "class", value: "cart-item-desc");
407 var span = document.createElement( tagName: "span");
408 span.setAttribute( qualifiedName: "class", value: "product-remove");
409 var i = document.createElement( tagName: "i");
410 i.setAttribute( qualifiedName: "class", value: "fa fa-close");
411 i.setAttribute( qualifiedName: "aria-hidden", value: "true");
412 span.appendChild(i);
413 div.appendChild(span);
414
415 var span = document.createElement( tagName: "span");
416 span.setAttribute( qualifiedName: "class", value: "badge");
417 span.appendChild(document.createTextNode( data: "Prodotto"));
418 div.appendChild(span);

```

Figura 3 JavaScript modifyCart 2

```

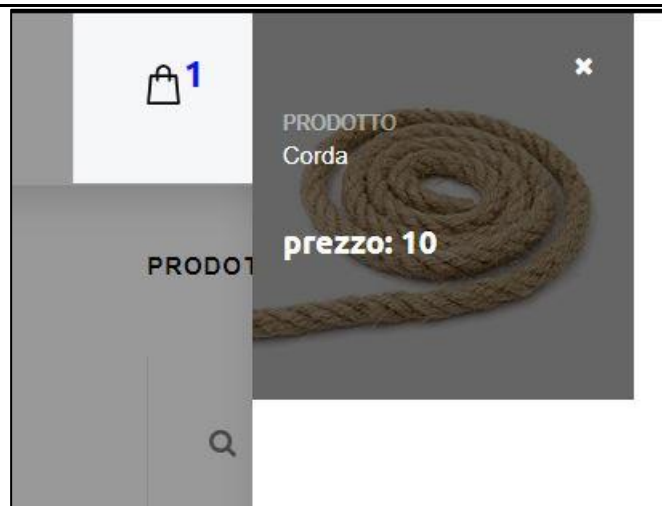
419
420 var h6 = document.createElement( tagName: "h6");
421 h6.appendChild(document.createTextNode(obj[0]['nome_prodotto']));
422 div.appendChild(h6);
423
424 var p = document.createElement( tagName: "p");
425 p.setAttribute( qualifiedName: "class", value: "price");
426 p.appendChild(document.createTextNode( data: "prezzo: " + obj[0]['prezzo_prodotto']));
427 div.appendChild(p);
428
429 a.appendChild(div);
430 divContainer.appendChild(a);
431 divBody.appendChild(divContainer);
432 }
433 }
434
435 xhr.open( method: "POST", url: "/gestionevendita2018/product/getProduct", async: true);
436 xhr.setRequestHeader( name: "Content-Type", value: "application/x-www-form-urlencoded");
437 xhr.send( body: "&name=" + keys[0] + "&price=" + keys[1] + "&quantity=" + keys[2]);

```

Figura 4 JavaScript modifyCart 3

Le ultime 2 immagini mostrano la creazione delle parti della mostra del prodotto nel carrello, con le relative classi e informazioni che devono mostrare.

Le informazioni del prodotto vengono mostrate correttamente, come mostrato nell'immagine qui sotto.



*Figura 5 mostra info prodotto*

Questo metodo però riscontra un problema, spiegato nella sezione “Problemi riscontrati e soluzioni adottate”.

#### Problemi riscontrati e soluzioni adottate

Il problema che ho riscontrato oggi è come uno già avuto in precedenza e riguarda la richiesta in post alla funzione che prende i prodotti del carrello e li inserisce nella pagina. Il problema è che la richiesta, essendo asincrona dato che la sincrona è deprecata, è troppo veloce e di conseguenza va a prendere solo l'ultima disponibile e non tutte quelle possibili. Ho pensato ad una possibile soluzione, ma non sono riuscito ad attuarla oggi per mancanza di tempo, essendo che fare la mostra mi ha preso molto tempo. La soluzione potrebbe essere di prendere già i prodotti direttamente nella funzione del controller, quindi già nella richiesta in “getCart” si potrebbe prendere i dati dal carrello dell'utente e dopodiché prendere i prodotti e passare direttamente quelli senza dover prendere tramite JavaScript i dati delle chiavi dei prodotti e richiederli con un'altra funzione.

#### Punto della situazione rispetto alla pianificazione

Ho portato avanti il carrello ma non è ancora completo. Devo completare in modo ottimale la parte d'inserimento dei prodotti, ovvero inserendo anche quale negozio e venditore lo imposta.

#### Programma di massima per la prossima giornata di lavoro

Completare l'implementazione del carrello.