

# Progetto Tweb

Silvestro Stefano Frisullo  
mat. 832813

31 agosto 2019

## Indice

<b>1</b>	<b>Tema e sezioni del sito</b>	<b>1</b>
<b>2</b>	<b>Funzionalità</b>	<b>1</b>
2.1	Login . . . . .	1
2.2	Registrazione . . . . .	2
2.3	Gestione del contenuto generato dall'utente . . . . .	2
2.3.1	Wishlist . . . . .	2
2.3.2	Carrello . . . . .	3
<b>3</b>	<b>Results</b>	<b>3</b>

# 1 Tema e sezioni del sito

Il tema del sito è un e-commerce di scarpe. È possibile scegliere tra diversi modelli, classici o sportivi e per uomo o donna. Per ogni modello è presente nome, descrizione, taglia disponibile e prezzo.

Le sezioni *commerciali* del sito sono:

- Man
- Woman
- Classic
- Sport
- Sneakers

La parte superiore presenta 4 pulsanti:

- Home
- Login/Logout
- Wishlist
- Carrello

## 2 Funzionalità

### 2.1 Login

Appena l'utente accede ad una qualsiasi pagina del sito, se non ha una sessione valida attiva, viene reindirizzato su *login.html*. Questa pagina prevede due opzioni: il login tramite username e password e la registrazione. L'utente può loggarsi inserendo username e password oppure essere autenticato dal sistema se in possesso del cookie di login persistente. I file coinvolti sono: *login.js* e *login.php*. Nel caso di login tradizionale javascript manda una ajax request al server, il quale farà query sul database, e in caso affermativo autenticerà l'utente, creerà la sessione e il cookie di login persistente valido 24 ore. Il cookie è un hash con sha256 di username + timestamp Unix + un numero casuale tra 0 e 1000000000 generato con *random\_int*. Nel caso di autenticazione con cookie, *login.js* manderà ajax request col contenuto di tale cookie e se uguale a quello nel database l'utente è accettato.

## 2.2 Registrazione

Per la registrazione sono coinvolti i file *login.js* e *register.php*. Javascript manda i campi non vuoti al server che controlla che l'username non sia già occupato e libero crea il profilo utente. Successivamente l'utente può loggarsi.

## 2.3 Gestione del contenuto generato dall'utente

I contenuti generabili dall'utente sono la **wishlist** e il **carrello**.

### 2.3.1 Wishlist

La **wishlist**, a cui si accede cliccando sull'icona del cuore, è gestita dai file *wishlist.html*, *wishlist.js*, *wishlist.php*, *add-wishlist.php*, *remove-wishlist.php*. Per scelta progettuale, la wishlist è memorizzata all'interno di una tabella nel database e non è possibile aggiungere lo stesso oggetto più volte. Riprendendo lo stile della home, la wishlist permette di visualizzare le scarpe, aggiungerle o rimuoverle. Javascript manda un'ajax request senza dati a *wishlist.php*, il quale prenderà da `$_SESSION` l'id dell'utente. Con l'id utente fa query e ricava tutte le scarpe, crea gli oggetti *Shoe* corrispondenti e le restituisce in formato JSON secondo questa sintassi:

---

```
[{ "id":"3", //id univoco scarpa
  "model":"Classic",
  "price":"200",
  "description":"perfect for classy man",
  "image":"images/scarpa-classica.jpg" // percorso relativo
    immagine
}]
```

---

in caso di successo, javascript rappresenta ogni scarpa in html e inserisce il codice nel corpo della pagina. In *wishlist.html*, ogni scheda ha un bottone *Remove*, che tramite *removeWishlist* manderà una ajax request a *remove-wishlist.php*, con id del prodotto da rimuovere dalla wishlist (quindi ci sono due ID, uno è dell'utente ed è salvato in `$_SESSION`, l'altro è della scarpa e si trova in `$_GET`). *remove-wishlist.php* farà una query sul database e rimuoverà dalla wishlist la scarpa con quell'ID restituendo "ok". A quel punto javascript eliminerà il nodo DOM dell'oggetto e aggiornerà la pagina. Per aggiungere un oggetto alla wishlist, si hanno due opzioni: trascinarlo sull'icona del cuore oppure cliccarci sopra, e in seguito cliccare su "Add to wishlist". I file che gestiscono questo caso sono *wishlist.js* e *add-wishlist.php*. Il primo si occupa dell'animazione e delle chiamate ajax, mentre il secondo tramite id

della scarpa fa una query al database. Dato che non è possibile inserire più di una volta lo stesso oggetto, nel caso in cui sia già presente il server restituisce "duplicate", e l'utente è notificato che il prodotto è già in wishlist. Altrimenti, se tutto va bene, javascript rimuove la scheda del prodotto dalla pagina, in modo da dare un feedback visivo della corretta esecuzione dell'operazione, ma senza essere troppo invadenti. Stesso funzionamento per quando si aggiunge alla wishlist dalla pagina del prodotto *shop-single.html*, ma in questo caso il file javascript responsabile della ajax request è *single-item.js*.

### 2.3.2 Carrello

Diversamente dalla *wishlist*, il carrello è implementato tramite variabili di sessione. Gli oggetti possono essere aggiunti o rimossi, e l'aggiunta come per la wishlist può avvenire trascinando gli oggetti o cliccando sul bottone relativo. L'array che contiene i dati del carrello è `$_SESSION[ "items" ]` e i file coinvolti sono *cart.html*, *cart.js*, *cart.php*, *add-cart.php*, *remove-cart.php*. In fondo alla pagina è indicato il costo totale degli oggetti nel carrello e il bottone "Buy!" simula una vendita fittizia, fa comparire un'immagine di conferma all'utente e svuota il carrello.

## 3 Results

Here are my results.  
bsadbsabuidsabui

1. First thing
2. Second thing
  - A sub-thing
  - Another sub-thing
3. Third thing

Apples	Green
Strawberries	Red
Oranges	Orange



Figura 1: My test image