Relazione progetto finale di Tecnologie Web Anno 2018/2019

Studente: *Bushaj Aldo*

Matricola: *847091*

Introduzione

Il tema scelto è quello di un e-commerce, chiamato “AB Watches”, abilitato alla vendita di orologi, il sito in questione permette di eseguire le fondamentali operazioni che un utente ha a disposizione su un qualsiasi e-commerce, si ha a disposizione per ogni utente registrato una lista preferiti, un carrello dove verranno inseriti tutti i prodotti che si desidera acquistare e una sezione, in cui è possibile visualizzare tutte le recensioni di uno specifico prodotto, da parte dei soli utenti registrati.

Il sito è così composto, nella pagina principale (siamo in index.php) si avranno solo due casi possibili:

1. Il caso del “login”, cioè si dovrà accedere con le credenziali (email e password) di un utente registrato in precedenza e quindi già presente nel database.
2. Il caso del signUp (registrazione), questo è il caso in cui si vuole far registrare un nuovo utente (compilando correttamente gli opportuni campi) che quindi naturalmente non è ancora presente nel database.

Un messaggio nella query string notificherà l’esito della registrazione o del login, una volta entrati(siamo in HomePage/index.php), nella home page troveremo tutti gli orologi acquistabili su AB Watches con i relativi nomi e prezzi, per visualizzare nel dettaglio un particolare prodotto e aggiungerlo al carrello, o alla lista preferiti, o ancora aggiungere una recensione, basterà cliccare l’immagine, a questo punto (siamo in HomePage/recensioni.php),vedremo un’immagine ingrandita dell’orologio selezionato a sinistra dello schermo, mentre in alto nella parte centrale troveremo tre pulsanti con le seguenti funzionalità:

1. Il primo a sinistra (recante la dicitura “inserisci recensione”) permette appunto di inserire una recensione, una volta premuto si aprirà una finestra a comparsa dove sarà possibile scrivere la recensione, in basso si trovano due pulsanti con i quali avremo la possibilità di aggiungere il commento (pulsante “Add ”) o annullare l’inserimento del commento (pulsante “close”).
2. Il pulsante centrale (con la raffigurazione di un cuore) permette di aggiungere il prodotto alla propria lista preferiti, quindi si verrà avvisati visivamente riguardo

l’esito dell’operazione, l’unica restrizione è che lo stesso articolo può essere inserito una sola volta nella lista.
3. Infine il terzo pulsante a destra (raffigurante l’icona di un carrello) permette di aggiungere un numero arbitrario di articoli anche con lo stesso nome.

Spostandoci nella pagina che visualizza la lista dei preferiti (siamo in preferiti.php) verranno visualizzati per ogni riga il nome del prodotto e il relativo prezzo, inoltre per ogni prodotto ci saranno due pulsanti con i quali e possibile spostare il prodotto nel carrello o eliminarlo definitivamente dalla lista dei preferiti.

Andando nella pagina che gestisce il carrello (siamo in carrello.php) avremo tre colonne raffiguranti diverse informazioni per ogni prodotto e un pulsante con le seguenti funzionalità:

1. **Name:** contiene il nome del prodotto con relativa immagine.
2. **Price:** contiene la somma del prezzo di tutti i prodotti con lo stesso nome.
3. **N° item:** contiene il numero di prodotti con lo stesso nome.
4. **Pulsante:** nella parte destra della pagina si trova un pulsante (raffigurante il cestino) il quale, se il numero di prodotti è > 1 diminuirà di uno il numero di prodotti, altrimenti se il numero di prodotti è = 1 verrà eliminato dal carrello.

Nella parte inferiore avremo il prezzo totale degli orologi presenti nel carrello e un pulsante recante la dicitura acquista, che appunto permette l’acquisto degli oggetti nel carrello, i quali, se l’acquisto va a buon fine, vengono eliminati dal carrello, e si ritorna alla home page con un messaggio (tramite query string) che notifica l’avvenuto acquisto dei prodotti.

Organizzazioni cartelle

Le cartelle sono così suddivise:

1. La cartella Tweb progetto contiene index.php che si occupa del login e della registrazione degli utenti, all’interno troviamo le cartelle php, css e js i quali contengono i file che gestiscono la pagina del login.
2. La sottocartella HomePage contiene tutti i file che si occupano della gestione delle funzionalità del sito, recensioni, carrello ecc., all’interno della sottocartella HomePage abbiamo la cartella assets che contiene le immagini e tutti i file js, php, css, mentre nella cartella headerFooter troviamo appunto l’header e il footer.

Funzionalità

Nel sito si avrà diverse volte la necessità di accedere al database per inserire dei prodotti, o accedere a quelli già inseriti ed effettuare diverse operazioni, l’accesso quindi verrà effettuato cliccando il pulsante “login” che si collegherà al database tramite il file dbServer.php incluso all’occorrenza , il file login.php (php/login.php) quindi andrà a recuperare l’email e la password(criptata tramite la funzione password_hash) memorizzandole nelle apposite variabili, a questo punto potremo interrogare il database con la seguente query :

SELECT * FROM users WHERE user_email = $email;

quindi ci assicureremo di avere almeno un risultato, il che significa che l'utente è già registrato, verificheremo quindi con la funzione "password_verify" che la password digitata sia corretta, nel caso in cui fosse incorretta si verrà reindirizzati alla pagina del login, altrimenti si memorizzeranno nell'array associativo $_SESSION tutte le informazioni dell'utente cioè email, password, nome e cognome quindi si vierrà reindirizzati nella home page all'interno del sito.

Se non si è registrati si darà la possibilità di registrarsi con il click del pulsante "sign Up", che solo dopo aver verificato con la funzione **preg_match()** che il nome e il cognome digitati soddisfino il pattern " ***/^[a-zA-Z]*$/*** ", si passerà alla validazione dell'email con la funzione

" **filter_var($email,FILTER_VALIDATE_EMAIL)** ", quindi si verificherà che l'email non sia già presente nel database in quanto univoca, e solo a questo punto verrà verificato che il campo "Password" corrisponda al campo "Confirm password", onde evitare digitazioni involontarie di password errate, quindi si potrà procedere con la codificazione, tramite la funzione "**password_hash**", della password in modo da renderla incomprensibile all'interno del database, infine verrà inserita con la seguente query:

INSERT INTO users (user_first, user_last, user_email, user_psw) VALUES ('$name', '$surname', '$email', '$hashPsw');

quindi verranno memorizzate nell'array associativo $_SESSION tutte le informazioni dell'utente che quindi verrà reindirizziato nella home page all'interno del sito.

L'ultimo caso quello del log out, verrà gestito nel file logOut.php, che verificherà l'avvenuto click del pulsante "log out" e quindi con la funzione " **session_start()** " si riprenderà la sessione corrente(in quanto era già stata avviata in precedenza), si de-allocheranno tutte le variabili lasciando ancora la sessione attiva con la funzione " **session_unset()** " , infine verrà cancellata esplicitamente la sessione con la funzione " **session_destroy()** " e si verrà reindirizzati alla pagina iniziale.

Caratteristiche

Per la realizzazione di AB Watches si è pensato all'implementazione di un sito web il più chiaro e semplice possibile sia per quanto riguarda la sua comprensione che per quanto riguarda il suo utilizzo, sono stati scelti colori non contrastanti e pulsanti autoesplicativi, inoltre è stato utilizzato il design responsive in grado di adattarsi graficamente automaticamente al dispositivo col quale viene visualizzato, nella home page appena loggati avremo in alto i prodotti in evidenza mentre più in basso l'intera lista dei prodotti, per "tracciare" l'utente durante l'interazione con il sito sono state utilizzate le sessioni PHP, le quali conterranno le informazioni necessarie all'identificazione di un generico utente, le sessioni verranno create(o si riprendono quelle già create) nei file login.php , signUp.php, logOut.php e header.php, dove si verificherà opportunamente che la sessione sia settata con la funzione "**isset()"**, il caso positivo indica che l'utente ha già effettuato il login e quindi potrà utilizzare il sito, oltre alla validazione dei dati in input da parte del server(come illustrato precedentemente) valideremo i dati in input anche con i due form login e signUp, il form contenuto nel div con **id="login"** invierà i dati al file login.php, solo dopo aver verificato che tutti i campi siano stati compilati, mediante il pulsante con **id="submitLog"**,

allo stesso modo il form contenuto nel div con **id=”signUp”,** invierà i dati, solo se tutti compilati, tramite il pulsante con **id=”submitSign”**.

Per quanto riguarda la sicurezza, verrà gestita la protezione da inserimenti di stringhe malevole usando la funzione **htmlspecialchars()** la quale converte i caratteri speciali in elementi HTML, per quanto riguarda il database non verranno mostrate le password degli utenti grazie alle funzioni illustrate precedentemente, la protezione da SQL Injection verrà gestita dalle funzioni **prepare()** ed **execute(),** infatti la funzione prepare() mette a disposizione uno strumento più avanzato e più sicuro per l'esecuzione delle query che dovrà essere eseguita tramite l'oggetto execute().

Front end

In questa sezione verrà illustrato semplicemente il compito di ogni file, mentre successivamente verrà illustrato come avviene la comunicazione tra front e back end.

La maggior parte delle funzionalità del sito vengono gestite dai file contenuti nella cartella homePage, precisamente su assets dai file php e javascript.

Il primo file javascript che troviamo (siamo all’interno della cartella Tweb progetto) è quello nella sottocartella js chiamato “insertData.js”, questo file si occuperà semplicemente dell’apertura, e della chiusura, delle finestre che permettono l’inserimento dei dati, per effettuare il login o il sign up.

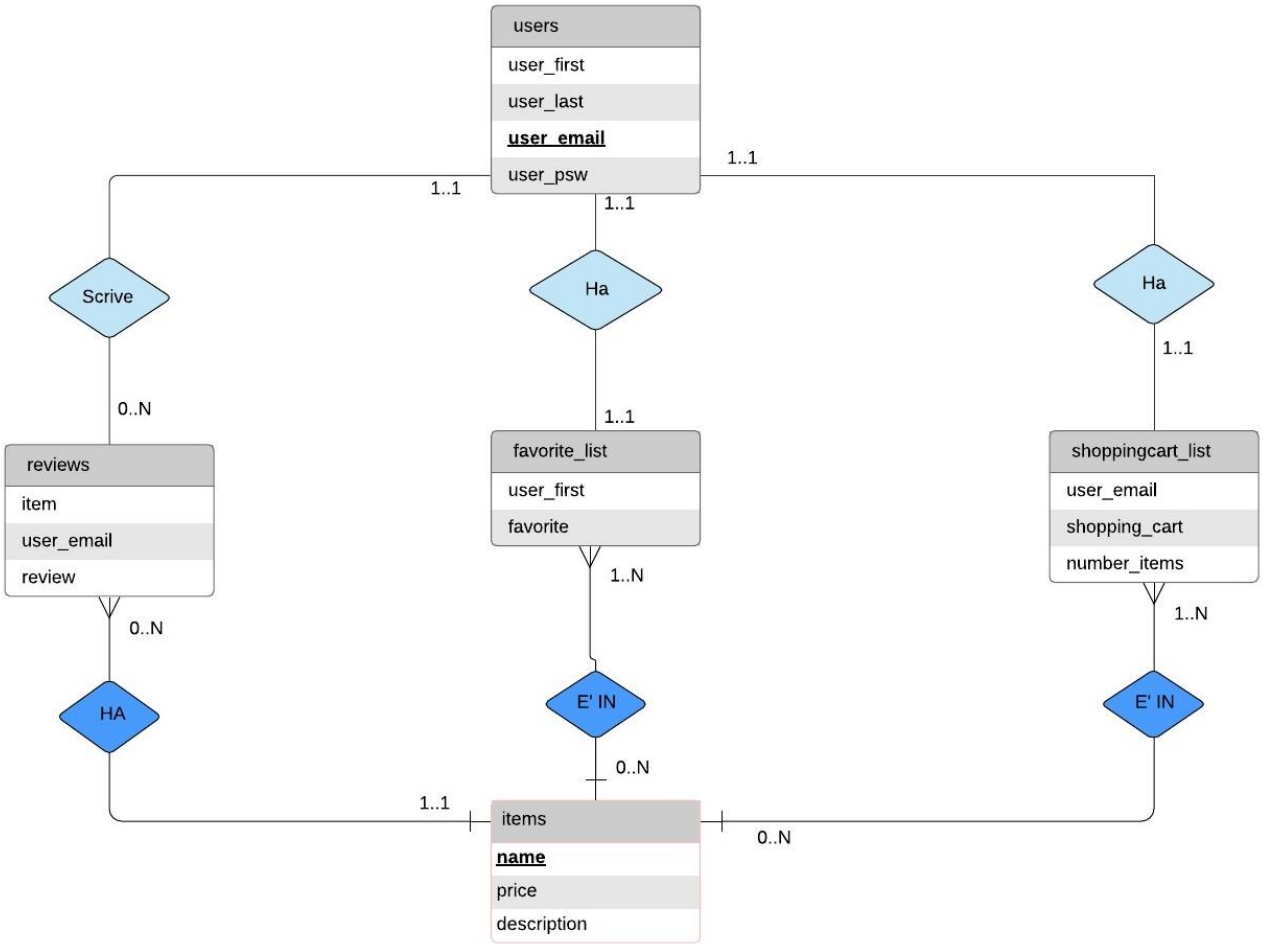
Spostandoci in “HomePage” avremo i file recensioni, preferiti e carrello (tutti con l’estensione .php) i quali contengono codice essenziale come titolo della pagina, nome dell’orologio ecc., questi file verranno successivamente modificati all’occorrenza, mantenendo uno stile unobtrusive, dai file che si trovano andando nella sottocartella “assets” e poi js la quale contiene:

- Onde evitare ripetizioni di operazioni comuni a tutti file vengono illustrate qui, in tutti i file si attenderà con la funzione **$(document).ready( )** che il DOM sia stato completamente caricato prima di effettuare qualsiasi altra operazione.
- **navbar-dropdown.js:** si occupa dell’animazione del corpo della pagina e della gestione del pulsante scroll to top.
- **loadReviews.js:** con questo file si gestirà il caricamento e la visualizzazione delle recensioni, le quali vengono caricate, solo quando la relativa pagina verrà aperta, questo è possibile grazie alla funzione **window.location.href.indexOf('recensioni.php')** la quale richiama la funzione **loadReviews()** , che recupererà dal database tutte le recensioni scritte su quel particolare prodotto ogni volta che la pagina recensioni.php verrà aperta.
- **loadFavorite.js:** questa classe si occuperà di tutte le operazioni (visualizzazione, aggiunta ai preferiti ecc.) per quanto riguardo la lista preferiti, l’email dell’utente loggato in quel momento viene memorizzato nel campo value di un’ input nascosto, questo valore verrà usato come illustrato successivamente per recuperare i dati dal database, troviamo anche la funzione **getFavoriteDiv** che richiede due parametri, nome e prezzo dell’orologio, e li inserirà opportunamente in due div preimpostati che conterranno l’immagine e le informazione di un particolare orologio.

- **loadShoppingCart.js:** questa classe si occuperà di tutte le operazioni riguardanti il carrello, avremo la funzione **getShoppingCartDiv** che restituirà i div che contengono l'immagine e le informazione di ogni orologio, inoltre avremo tre variabili, **dataDefault** che conterrà i div preimpostati per il pulsante acquista, prezzo totale orologi nel carrello e infine le informazioni che ogni colonna rappresenta, la variabile **email** che recupererà l'email dell'utente loggato come illustrato nel punto precedente, e la variabile **num** impostata a 0 che distingue il primo elemento del carrello dagli altri.

Back end

Per la memorizzazione permanente dei dati è stato usato un database chiamato "tweb" illustrato nel seguente schema (i campi sottolineati in grassetto sono le chiavi):



Le tabelle sono cosi composte.

- **Tabella items:** sono presenti tutti i prodotti acquistabili (identificati da un nome univoco) su AB Watches con le relative informazioni.
- **Tabella users:** vengono memorizzati tutti gli utenti identificati da un'indirizzo email univoco e relativo nome, cognome e password.

- **Tabella reviews:** contiene tutte le recensioni di un particolare prodotto da parte di tutti gli utenti registrati su AB Watches.
- **Tabella favorite_list:** contiene la lista dei preferiti dell'utente attualmente loggato, qui non possono essere inseriti più prodotti con lo stesso nome.
- **Tabella shoppingCart_list:** contiene la lista dei prodotti nel carrello dell'utente loggato, qui a differenza della lista dei preferiti è possibile trovare più articoli con lo stesso nome.

Comunicazione front/back end

L'accesso e la manipolazione dei dati del database sono effettuati dai file php getFavorite, getReviews, getShoppingCart(con il seguente percorso: HomePage/assets/php), grazie all'interazione con i file js descritti precedentemente.

Procediamo descrivendo le operazioni fondamentali per le quali è necessario interagire con il database (in tutti i file js i dati vengono inviati tramite AJAX con il metodo POST):

- **Recensioni:** ogni volta in cui viene aperta la pagina recensioni.php vengono caricate tutte le recensioni del prodotto selezionato, questo avviene grazie alla funzione **loadReview()** che accetta come parametro il nome dell'orologio e l'email dell'utente loggato, questa funzione invierà il contenuto dei parametri al file getReviews.php grazie alla tecnologia AJAX , nel file php verranno gestiti il caso del caricamento dei commenti con la seguente query:
  SELECT user_email,review FROM reviews WHERE item = $watch;

  mentre quello dell'inserimento dei commenti con la seguente query:

  INSERT INTO reviews (item, user_email, review) VALUES ($jsonOb->item,$jsonOb->user_email, $jsonOb->review);

  in caso di successo (callback) nel file javascript chiudo la finestra dell'inserimento delle recensioni e carico con la funzione **loadReviews** la lista delle recensioni aggiornata, per evitare l'inserimento di commenti duplicati si distingue la fase in cui inizia l'invio dei dati e quella in cui finisce con la variabile is_sending.

- **Lista preferiti:** gli orologi nella lista preferiti vengono caricati con la funzione **loadFavorite(),** la quale invia tramite AJAX l'email al file getFavorite.php, quest'ultimo dopo aver eseguito la seguente query

  SELECT favorite_list.favorite ,items.price FROM favorite_list INNER JOIN items ON favorite_list.favorite = name WHERE user_email = $email;

  inserisce i dati in un array e lo restituisce in formato JSON grazie alla funzione PHP **json_encode(),** a questo punto nella funzione callback non ci resta che convertire tramite **JSON.parse()** l'oggetto e aggiungerlo al DOM con la funzione **append()**.
  Il caso dell'aggiunta o rimozione di un prodotto dalla lista dei preferiti viene effettuata inviando tramite AJAX email, nome dell'orologio e una variabile add impostata a 0 nel caso in cui l'utente voglia eliminare l'articolo, mentre a 1 nel caso in cui l'utente voglia aggiungere l'articolo ai preferiti, la query utilizzata per eliminare uno specifico prodotto della lista è:

DELETE FROM favorite_list WHERE favorite = $watch AND user_email = $email;

mentre per inserire l'articolo nella lista, prima si dovrà verificare che il prodotto non sia già presente con la seguente query:

SELECT * FROM favorite_list WHERE user_email = $email AND favorite = $watch;

successivamente se la funzione **rowCount()** restituisce un numero < 1(il prodotto non è già presente) si potrà procedere con l'inserimento dell'articolo nella lista con la seguente query:

INSERT INTO favorite_list (user_email, favorite) VALUES ($email, $watch);

- **Carrello:** gli orologi nel carrello vengono caricati con la funzione **loadShoppingCart(),** la quale invia tramite AJAX al file getShoppingCart.php l'email dell'utente loggato, quest'ultimo recupererà dal database tutti gli orologi nel carrello con la query:

  SELECT shoppingcart_list.shopping_cart ,items.price, shoppingcart_list.number_items FROM shoppingcart_list INNER JOIN items ON shoppingcart_list.shopping_cart = name WHERE user_email = $email;

  dopodiché verrà inserito in un array il risultato della query e una volta convertito con la funzione **json_encode()** verrà restituito, a questo punto nella funzione di callback la funzione **JSON.parse()** convertirà l'oggetto e verrà aggiunto al DOM con la funzione **append(),** il calcolo del prezzo totale degli orologi nel carrello, verrà effettuato con la funzione **eval("price*number")**, quindi verranno aggiunti gli zeri finali con la funzione **toLocaleString("en",{useGrouping: false,minimumFractionDigits: 3}).**

  Gli articoli possono essere inseriti nel carrello cliccando l'apposito pulsante in recensioni.php oppure spostandoli dalla lista preferiti, in entrambi i casi verranno aggiunti tramite AJAX, inviando al file getShoppingCart.php, l'email, il nome dell'orologio e la variabile add impostata a 1(indica che devo aggiungere l'articolo al carrello), il file PHP dopo aver verificato che la variabile sia impostata a 1 , aggiungerà(se non è già presente) l'articolo al database con la seguente query:

  INSERT INTO shoppingcart_list (user_email, shopping_cart,number_items) VALUES ($email, $watch,1);

  dove il numero 1 all'interno della query, indica che è presente un solo prodotto con il nome specificato, altrimenti se è già presente verrà usata la query:

  UPDATE shoppingcart_list SET number_items = '$number' WHERE user_email = $email AND shopping_cart = $watch;

che provvederà semplicemente ad aggiornare il numero degli articoli con il nome specificato, nel caso in cui l'utente abbia spostato l'articolo dalla lista dei preferiti, nella funzione di callback si provvederà ad eliminare l'articolo dalla lista dei preferiti facendo un ulteriore chiamata AJAX e quindi inviando nuovamente email, nome orologio e variabile add settata a 0 al file getfavorite.php, che eliminerà l'articolo dalla lista dei preferiti, nella funzione di callback di quest'ultima chiamata AJAX, verrà semplicemente aperta la pagina del carrello con la funzione **open("carrello.php",'_self').**

Per eliminare i prodotti viene usata la stessa logica utilizzata per la lista dei preferiti, la sola differenza è che qui non si ha bisogno di inviare anche la variabile add, basterà solo l'email e il nome dell'orologio, questo per differenziare l'evento in cui viene cliccato il pulsante per eliminare un articolo da quello in cui si preme il pulsante per acquistare gli articoli nel carrello, infatti in quest'ultimo caso verrà inviata la variabile add impostata a 0, quindi il file php eliminerà tutti gli articoli presenti nel carrello, e la funzione di callback non dovrà far altro che andare nuovamente nella Home Page con la funzione **open("index.php?status=items_buyed",'_self')**