RELAZIONE TecnologieWeb 2020/2021

Studente: Giorgio Mecca

Matricola: 880847

--Descrizione/Presentazione Sito

Best-E.Commerce-Ever è un sito di Ecommerce in cui un utente può registrarsi (sia come utente normale che come Azienda) e mettere in vendita / comprare prodotti. Il sito dopo aver effettuato il login/ registrazione mostrerà all'utente tutti i prodotti disponibili; l'utente potrà su questi effettuare una ricerca in base al nome o alla categoria di questi. Cliccando su di un prodotto si visualizzerà la sua pagina contenente: l'immagine (se disponibile), i dati del prodotto (Nome, descrizione, prezzo con uno sconto se specificato), i dati del venditore e due bottoni per inserire tale prodotto nel carrello o nella lista dei preferiti. Dal Banner in alto sarà possibile muoversi verso il carrello, i dati dell'utente, la lista dei preferiti, degli ordini, la pagina per vendere un prodotto, e sarà possibile effettuare il logout. Nel carrello si potrà rimuovere un prodotto oppure specificare una quantità per ogni prodotto per poi effettuare l'ordine che sposterà i prodotti nella lista degli ordini. Nella pagina dei dati verranno visualizzati i dati dell'utente, una lista dei suoi indirizzi(da cui sarà possibile aggiungerci un altro) , e dei collegamenti per il carrello, la lista dei preferiti, gli ordini effettuati e una lista di oggetti messi in vendita. La lista dei preferiti ci permetterà di eliminare un prodotto da essa oppure spostare questo nel carrello, mentre la lista degli ordini visualizzerà soltanto i prodotti acquistati e la data di acquisto ma non sarà possibile effettuare operazioni. Nella pagina per vendere un prodotto si dovrà inserire i dati relativi ad esso in dei campi input per poi validare l'inserimento; i dati del prodotto potranno essere modificati in seguito nella lista dei prodotti in vendita in cui sarà anche possibile inserire uno sconto (in percentuale) oppure rimuovere il prodotto dal Database.

--Funzionalità

-Il login è effettuabile tramite una pagina html(login.html) a cui verremo reindirizzati se non è presente una nostra sessione, questa pagina ci chiederà Email e Psw e con il premere di un button la pagina js collegata(login.js) avvierà una richiesta ajax alla funzione login.php che una volta controllato i dati immessi (se l'utente esiste e la psw è giusta) creerà una sessione(session_start()) e setterà i dati di essa come l'id ed un booleano che indicherà se l'utente è comune oppure è un'azienda. Per il logout viene richiamata la funzione logout.php che si occuperà di distruggere la sessione(session_unset(); session_destroy()) e reindirizzarci alla pagina di login
-Dalla pagina di login è possibile cliccare su Registrati e verrà mostrato un nuovo form di richiesta dati. Una volta inseriti tutti i dati necessari (Solo Mail e Psw sono obbligatori) verrà effettuato un controllo lato client su di essi (mail accettabile e le 2 psw coincidono); se superano i test allora verrà effettuata una richiesta ajax alla funzione registration php che setterà i dati in modo da renderli accettabili e sicuri per un DB(htmlspecialchars() e $db->quote()) e infine li inserirà nel DB; non è necessario fare controlli se l'utente è già registrato in quanto il DB non accetta 2 mail uguali(campo Email UNIQUE)
-L'utente può generare contenuto inserendo un nuovo prodotto in vendita. Questo è possibile dalla pagina sellPage.php raggiungibile dal link nel banner. Questa conterrà vari campi input in cui inserire i dati del prodotto (Nome, descrizione, immagine, quantità, prezzo ed una categoria tra quelle presenti), l'immagine è l'unico campo non obbligatorio mentre bisogna ricordarsi che se la quantità è 0 questo prodotto non verrà mostrato nella home. Cliccando sul button Inserisci si andrà ad effettuare dei controlli lato client sugli input e ci sarà in seguito al successo dei controlli una richiesta ajax per la funzione insertProduct.php che una volta controllati i dati inserirà un nuovo prodotto nel DB (è possibile inserire più prodotti potenzialmente identici). Questi dati sono modificabili tramite la pagina salesList.php raggiungibile dalla pagina profile.php. Qui verrà mostrata la lista di oggetti messi in vendita dall'utente della sessione e potrà modificare i dati(verranno comunque controllati lato client es: il nome e la descrizione non possono essere null oppure i numeri come prezzo e sconto non possono essere negativi).

--Caratteristiche

All'inizio di una qualunque pagina php del sito e anche nelle funzioni php dove si ha bisogno dei dati dell'utente viene effettuato un controllo sulla sessione. Questa è inizialmente avviata con session_start() ad ogni file e viene effettuato un controllo !isset($_SESSION) e !isset($_SESSION["ID]) se uno di questi risulta positivo indica un problema con la sessione e quindi reindirizzerà la pagina sulla pagina di login.
La sessione viene terminata se si accede alla funzione logout.php che userà session_inset(); sesion_destroy() per terminare la sessione e reindirizzerà al login.
La validazione degli input avviene sia lato Client che Server. Lato Client viene solo controllato che i dati non siano nulli o che siano dei valori accettabili es: mail che contenga '@' oppure il prezzo di un prodotto che non sia negativo. Lato Server ogni qualvolta si ha necessità di salvare dei dati nel DB allora viene utilizzata la funzione htmlspecialchars() per disabilitare i tag html, questo andrò anche a ridurre i possibili attacchi XSS poichè i tag saranno disabilitati quando saranno passati al js per poi essere stampati, mentre quando si ha bisogno di eseguire una query con dei parametri viene utilizzata la funzione $db->quote(valore) che prepara i valori per essere passati ad una query evitando così attacchi di SQLInjection.
L'animazione/interazione con l'utente avviene nella home page (index.php) in cui è possibile effettuare un drag&drop di un prodotto nel carrello spostandolo in alto a destra sulla scritta o immagine del Carrello, inoltre per l'utente è anche possibile scegliere dall'input range sulla destra il numero di elementi che verranno visualizzati per pagina.

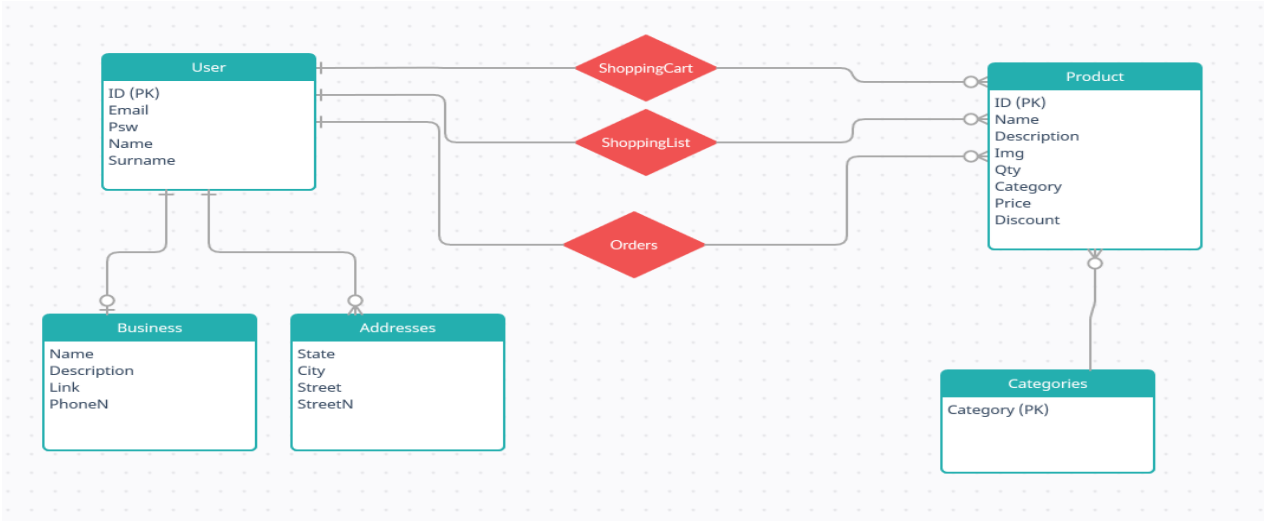
--Stile Unobtrusive

Nel progetto viene adottato uno stile Unobtrusive in quanto viene separata la presentazione dal comportamento. Questo viene mediato dal javascript che utilizzerà delle librerie ajax per richiedere funzioni php che restituiranno i dati sotto forma di scrittura json, in questo modo non si adottano costrutti come “form” e il comportamento di determinati tag html(es:button) viene settato dal js che all'avvio della pagina imposterà una determinata funzione all'azione del soggetto. Molti oggetti (es: liste del carrello, ordini, preferiti) vengono create da determinate funzioni js che dopo aver ricevuto in input un json (output php) creeranno questi oggetti ed aggiungeranno loro ad elementi html (es: $(elemento).append(nuovo_oggetto))

--Organizzazione

I file sono organizzati in 5 sottocartelle della directory principale Progetto. Le cartelle contengono i file CSS, i file HTML, le immagini (Img),i file javascript(js) e i file php; nella cartella php è presente una sotto cartella "function"
 che appunto conterrà le funzioni php mentre nella cartella php ci saranno le pagine php.

--DATABASE



Il Database si Basa su 5 Tabelle, "User": l'utente del sito, questo può essere un utente classico o un'azienda relazionandosi con "Business" che avrà l'ID dell'utente come chiave primaria ed esterna;

un utente può essere associato a 0 o più indirizzi "Addresses". I prodotti in vendita sono descritti dalla tabella "Product", questa possiede una relazione con chiave esterna Category che indica una categoria presente nella tabella "Categories". I prodotti si relazionano con User creando 3 Liste tra cui : Il carrello "Shopping Cart", Lista dei preferiti ("ShoppingList"), e gli ordini effettuati ("Orders").
/* Nella cartella del progetto (progetto/php/) è presente un file php ("create_db_example") che crea il database e inserisce alcuni dati fittizi (di prova) */

--Funzioni PHP:

addAddress.php
- Aggiunge un indirizzo legato all'utente della sessione
- metodo: GET / argomenti: State, City, Street, StreetN(Numero Civico)
- output: { "result" : "TRUE" || "FALSE",
  "StrErr": "Errore" }

addElement.php
- Aggiunge un elemento ad una lista specificata(Carrello/ Prefetiti)
- metodo: GET / argomenti: To("Cart","List"), ID_Product
- output: { "result" : "TRUE" || "FALSE",
  "StrErr": "Errore" }

InsertProduct.php
- Inserisce un Prodotto nel database (mettere in vendita)
- metodo: GET / argomenti: Name, Desc(Descrizione), Img(non obbligatorio), Qty, Cat(categoria), Price
- output: { "result" : "TRUE" || "FALSE",
  "StrErr": "Errore" }

login.php
- effettua il login di un utente/ crea una sessione con il suo ID
- metodo: POST / argomenti: Email, Psw
- output: { "result" : "TRUE" || "FALSE",
  "StrErr": "Errore" }

modifyProduct.php
- Modifica i dati di un prodotto messo in vendita dall'utente della sessione
- metodo: GET / argomenti: ID_Product, Name, Desc, Qty, Price, Discount
- output: { "result" : "TRUE" || "FALSE",
  "StrErr": "Errore" }

order.php
- Effettua un ordine/ acquista tutti gli elementi che sono presenti nel carrello
- metodo: GET / argomenti: per ogni Prodotto: ?ID_Product= Qty
- output: { "result" : "TRUE" || "FALSE",
  "StrErr": "Errore" }

registration.php
- Effettua la registrazione si un nuovo utente, se possibile crea la sessione(effettua il login)
- metodo: POST / argomenti: Email, Psw, Type = "User" || "Business"

User -> argomenti: Name, Surname
Business -> argomenti: Name, Desc, Link, Tel
- output: { "result" : "TRUE" || "FALSE",
  "StrErr": "Errore" }

removeElement.php
- Rimuove un elemento dalla lista dei preferiti o dal carrello
- metodo: GET / argomenti: ID_Product, From ("Cart","List")

- output: { "result" : "TRUE" || "FALSE",
  "StrErr": "Errore" }

removeProduct.php
- Rimuove un Prodotto(che l'utente della sessione aveva messo in vendita) dal DB
- metodo: POST / argomenti: ID_Product
- output: { "result" : "TRUE" || "FALSE",
  "StrErr": "Errore" }

cart.php
- Restituisce i prodotti presenti nel carrello dell'utente della sessione
- output: [{ "ID_Product":"Italia",
"Name":"città",
"Description":"via",
"Img":"via",
"Qty":"via",
"Category":"via",
"Price":"via",
"Discount":"numero civico"}
,{prodotto2},{...}] (array di oggetti product)

favorites.php
- Restituisce i prodotti presenti nella lista dei preferiti dell'utente della sessione
- output: array di oggetti product

orders.php
- Restituisce i prodotti ordinati dall'utente della sessione
- output: array di oggetti product
-

product.php
- Restituisce i prodotti disponibili nel DB
- output: array di oggetti product

sale.php
- Restituisce i prodotti messi in vendita dall'utente della sessione
- output: array di oggetti product

search.php
- Restituisce i prodotti ricercati in base all'id oppure ad una stringa che li rappresenta
- metodo: GET / argomenti: ID_Product || Str_Product
- output: array di oggetti product

address.php
- Restituisce gli indirizzi associati all'utente della sessione
- output: [{ "State":"Italia",
"City":"città",
"Street":"via",
"StreetN":"numero civico"}
,{indirizzo2},{...}]

categories.php
- Restituisce tutte le categorie presenti nel DB
- output: [{"Category":"Abbigliamento"},{"Category":"Tecnologia"}]

search.php
- Restituisce i prodotti ricercati
- metodo: GET / argomenti: State, City, Street, StreetN(Numero Civico)
- output: {

"result" : "TRUE" || "FALSE",
"StrErr": "Errore"
}

personalData.php

- Restituisce i dati dell'utente della sessione
- output: { "Type": "User" || "Business",

"Data": { "ID":"6",
"EMail":"pina@gmail.com",
"Psw":"1234",
"Name":"Pina",
"Surname":null }
}

seller.php

- Restituisce i dati del venditore di un prodotto
- metodo: GET / argomenti: ID_Seller
- output: [{ "EMail":"Azienda2@Azienda.com","Name":null,"Surname":null},
{"Name":"Azienda2",

"Description":"Nuova azienda nel commercio Droni della IOT Generation", "Link":"Azienda2.com",
"PhoneN":"7756219776"}
]

Logout.php

- Effettua il logout di un utente, distrugge la sessione creata e reindirizza la pagina sul login