Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAMT |
| Data | 08.04.2019 |

|  |
| --- |
| Lavori svolti |
| Oggi ho iniziato la pagina dell’amministratore.  L’amministratore ha la possibilità di aggiungere e gestire i negozi e i relativi gestori di essi.  Per implementare questa funzionalità ho per prima cora creato una pagina, molto simile a quella del venditore, ma in questo caso non vengono mostrate le categorie ma solo i negozi.  L’immagine qui sotto mostra la base della pagina dell’amministratore.    Figura 1 pagina iniziale amministratore  Una volta impostata la pagina ho aggiunto il modal che si va ad occupare della registrazione, il cui codice è mostrato nelle immagini che seguono.      Il codice mostra che la base del modal, come le classi e la struttura, è la stessa dei modal creati in precedenza per i prodotti, la differenza è il contenuto, in questo caso vengono inseriti i dati per fare la registrazione del prodotto e del venditore che lo gestisce.  Essendo una registrazione la maggior parte del codice era già esistente, mi è bastato rinominare nomi e id dei componenti in modo corretto e poi riportare le funzioni JavaScript utilizzate nella pagina di registrazione dei compratori.  Una volta fatto ho dovuto creare 3 funzioni in PHP, 2 da model e una controller.  La prima funzione l’ho creata nella classe controller “Admin” ed è quella che viene richiamata dal form al click del bottone submit. La funzione si occupa di prendere i dati inseriti dall’utente e passarli alle relative funzioni che le inseriscono nel database, come mostrato dalla foto qui sotto.    Figura 2 registerShopDealer registrazione negozio e venditore  L’immagine mostra solo la parte interno all’”if” che controlla che ci sia il metodo POST, il controllo presente in ogni funzione.  Come si può notare la funzione fa riferimento a 2 diversi model, e 2 funzioni di essi, questo perché non deve registrare solo un nuovo negozio, ma anche salvare un nuovo venditore per quel negozio. Le funzioni non hanno niente di speciale, si occupano semplicemente di inserire i dati presi come argomento all’interno delle tabelle, come mostrato dalle raffigurazioni che seguono.    Figura 3 insertDealer funzione di inserimento del venditore    Figura 4 insertShop inserimento di un negozio e il suo venditore  Una volta fatto questo ho impostato la visualizzazione dei negozi all’interno della pagina.  La struttura di base è la stessa di quella dei prodotti ma in questo caso vengono solo scritti i titoli, che mostrano il nome del negozio, e un link per archiviare il negozio, come mostrato sotto.      Figura shopDetails inserimento  Le immagini mostrano la funzione JavaScript che va a prendere i dati dei negozi e li inserisce nel div corretto. La funzione controller e model sono molto semplici, si occupano di andare a prendere i dati dello shop selezionato e del gestore di esso, e poi passarli alla funzione JavaScript.  Quelle mostrate sono le funzioni model e controller che prendono i dati.    Figura model getShops che prende le informazioni dei negozi    Figura controller getShops che recupera i dati e li stampa  Una volta presi e stampati i dati in JSON la funzione JS mostrata sopra inserisce i dati all’interno della pagina.  Una volta fatto ho implementato il form nel modal per modificare i file. Il modal è quasi uguale a quello per aggiungere ma senza gli input della password, che non si può modificare, e con degli input nascosti in più che contengono le variabili delle chiavi del negozio e del gestore, per fare in modo che se vengono modificati riesce a prenderli lo stesso.  La prima cosa che viene fatta è riempire gli input con i dati del negozio e del gestore richiesti, questo con una funzione JavaScript che li prende, come mostrato sotto.    Figura shopDetails inserimento dei dati del venditore e del suo negozio  La funzione controller richiamata a riga 268 prende semplicemente i dati dal model, che invece fa una query con una doppia join per prendere tutti i dati sia del negozio che del venditore. Le immagini mostrano le 2 funzioni in questione.    Figura controller getShopsDealer prende i valori del negozio e del venditore    Figura model getShopDealer prende i dati del negozio e del suo gestore dal database e le ritorna  Le ultime funzioni per ultimare la modifica dei dati sono quelle che si occupano di modificare effettivamente i dati nel database. Come al solito la funzione controller della classe Admin, come tutti gli altri controller di questa pagina, sono semplicemente da passaggio per modificare i file, come mostrato nell’immagine.    Figura controller modifyShop, prende i dati e richiama le funzioni che modificano dati  L’immagine, come alcune precedenti, mostra solo il codice dopo il controllo del POST.  Le 2 funzioni che modificano i dati sono quasi uguali se non per i dati che inseriscono, che sono per forza di cose diverse, e sono mostrate nelle immagini che seguono.    Figura 12 model modifyDealer, modifica i dati con i nuovi ricevuti    Figura model modifyShop, modifica i dati con i nuovi ricevuti  Dopo questo ho lavorato alla creazione del file pdf, essendo che la parte precedente è stata molto veloce perché era per la maggior parte codice già precedente implementato.  Per fare il pdf ho messo in un form tutto il contenuto del carrello e ho inserito degli input nascosti che contenevano i dati da portare nel pdf, ovvero il nome, il prezzo e la quantità del prodotto, la quantità e il prezzo totali e la scelta del negozio di ritiro.  Il form va a richiamare una funzione nel controller “customer” che prende i dati e crea il pdf.            Figura controller doCheckout funzione che crea il pdf  La funzione dopo aver preso i dati controlla che effettivamente siano creati e non nulli ma in caso lo sono ritorna alla pagina iniziale.  La visualizzazione definitiva del file pdf è mostrata nell’immagine che segue.    Figura file pdf definitivo  Si può notare che c’è un errore nella scritta “Quantità” questo perché il charset non riesce a leggere la “à” e scrive quell’altra lettera. |  |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Ho avuto alcuni problemi nello scrivere le query, inizialmente perché sbagliavo il nome dei campi ma poi anche con la riga per prendere i dati del negozio singolo con il suo gestore, questo perché provavo ad utilizzare una “full outer join” che in mysql non è più utilizzabile. |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| Oggi ho completato la pagina dell’amministratore e la creazione del pdf  Devo ancora gestire il caso in cui 2 negozi facciano riferimento allo stesso prodotto e in uno dei 2 diminuisca la quantità. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Completare la documentazione. |