Diario di lavoro

Luogo	SAMT
Data	25.02.2019

Lavori svolti

Questa mattina inizialmente ho dovuto aggiustare un problemino con la mostra dei prodotti alla ricerca come mostrato nella sezione “Problemi riscontrati e soluzioni adottate”.
Una volta fatto mi sono accorto che era scaduta la mia prova gratuita di phpStorm, e di conseguenza ho cercato una soluzione, e ho trovato la versione per studenti che è gratuita, di conseguenza ho effettuato la registrazione con la mail di scuola e ho scaricato quella, in modo che non mi si chiuda l’editor ogni mezzora. Questo mi ha preso un po' di tempo perché ho dovuto registrarmi e il sito ha dato un po' di problemi a prendere la mail di scuola.
Una volta completata in modo corretto la tabella “compra” ho dovuto cambiare la query che inserisce i dati al suo interno, per inserire anche prezzo e quantità del prodotto.

```php
public function insertData($prodName, $prodPrice, $prodQuantity, $custMail){

    //Connetto al database
    $conn = $this->connection->sqlConnection();

    //echo "Connected successfully";
    $sql = $conn->prepare("INSERT INTO compra (nome_prodotto, prezzo_prodotto, quantita_prodotto, email_cliente, data)
    VALUES (:prodName, :prodPrice, :prodQuantity, :custMail, now())");
    $sql->bindParam(':prodName', $prodName, PDO::PARAM_STR);
    $sql->bindParam(':prodPrice', $prodPrice);
    $sql->bindParam(':prodQuantity', $prodQuantity, PDO::PARAM_INT);
    $sql->bindParam(':custMail', $custMail, PDO::PARAM_STR);

    //Eseguo la query
    $sql->execute();

    //Se ci sono dei valori
    if($sql->rowCount() > 0){
        $conn = null;
        return true;
    } else {
        $conn = null;
        return false;
    }
}
```

Figura 1 funzione inserimento dati nella tabella compra

Una volta completata la funzione di inserimento ho iniziato a lavorare alla mostra dei prodotti nel carrello della pagina. Ma prima ho aggiunto una nuova colonna nella tabella chiamata “quantità_richiesta”, che contiene un numero che si incrementa ogni volta che l’utente inserisce quel prodotto nel carrello, per fare in modo che ne possa ordinare più di uno fino al massimo disponibile. Dopo aver fatto questo ho cambiato anche lo schema E-R e ho dovuto modificare la funzione di inserimento dei dati nl database un'altra volta.

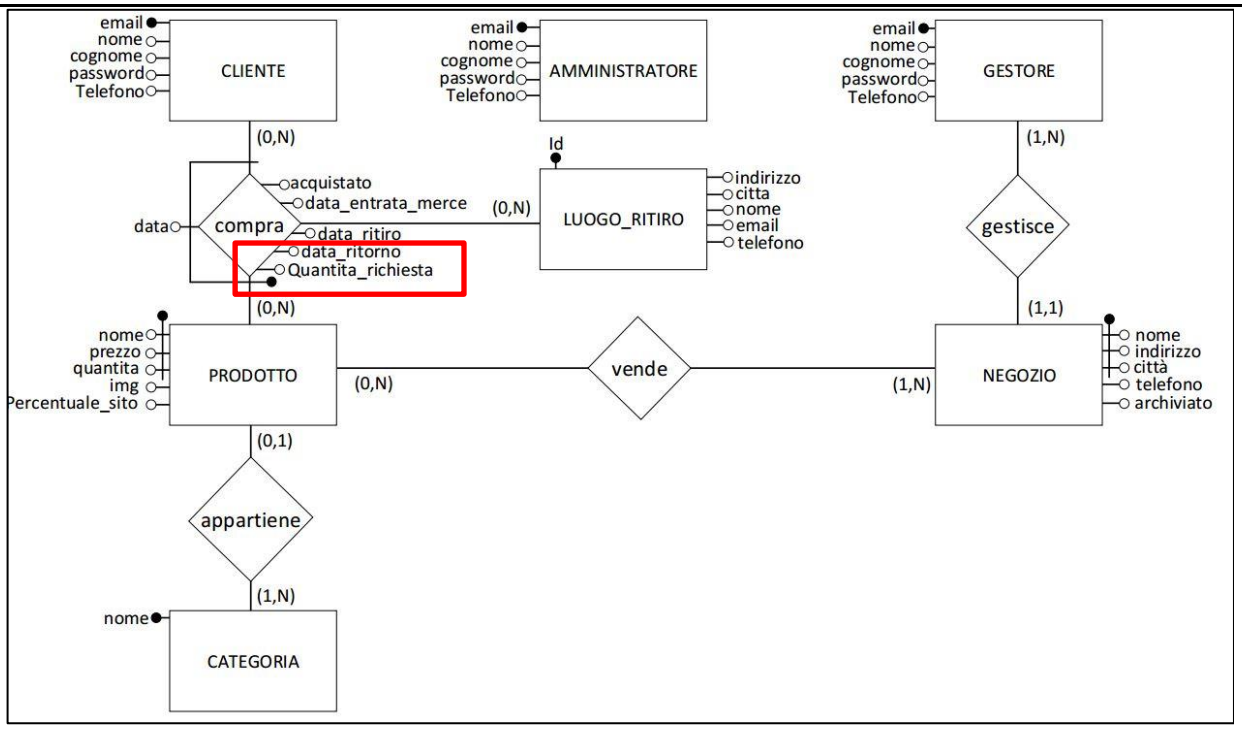

Figura 2 schema E-R con quantita_richiesta

Come si può notare ho semplicemente aggiunto un campo nella tabella ponte “compra”. Dopodiché ho dovuto modificare ulteriormente l’inserimento dei dati nella funzione, per fare in modo che un utente possa inserire nella tabella, ovvero il carrello, più volte lo stesso prodotto.

```
27          $quantity = 0;
28          $buyCount = $this->getDataByProduct($prodName, $prodPrice, $prodQuantity);
29          if(count($buyCount) > 0){
30
31              //Setto la nuova quantità e modifico il campo nella tabella
32              $quantity = $buyCount['quantita_richiesta'] + 1;
33              $sql = $conn->prepare("UPDATE compra SET quantita_richiesta = :quantity, data = now() WHERE nome_prodotto LIKE :prodName
34              AND prezzo_prodotto LIKE :prodPrice AND
35              quantita_prodotto LIKE :prodQuantity AND email_cliente LIKE :custMail");
36              $sql->bindParam(':prodName', $prodName, PDO::PARAM_STR);
37              $sql->bindParam(':prodPrice', $prodPrice);
38              $sql->bindParam(':prodQuantity', $prodQuantity, PDO::PARAM_INT);
39              $sql->bindParam(':custMail', $custMail, PDO::PARAM_STR);
40              $sql->bindParam(':quantity', $quantity, PDO::PARAM_INT);
41          }else{
42
43              //se il campo è nuovo inserisco la nuova riga
44              $quantity = 1;
45              $sql = $conn->prepare("INSERT INTO compra (nome_prodotto, prezzo_prodotto, quantita_prodotto, email_cliente, data, quantita_richiesta
46          VALUES (:prodName, :prodPrice, :prodQuantity, :custMail, now(), :quantity)");
47              $sql->bindParam(':prodName', $prodName, PDO::PARAM_STR);
48              $sql->bindParam(':prodPrice', $prodPrice);
49              $sql->bindParam(':prodQuantity', $prodQuantity, PDO::PARAM_INT);
50              $sql->bindParam(':custMail', $custMail, PDO::PARAM_STR);
51              $sql->bindParam(':quantity', $quantity, PDO::PARAM_INT);
52          }
```

Figura 3 controllo prodotto inserito

Prima di fare la query viene controllato se il prodotto è stato già inserito nel carrello dal cliente, questo viene fatto dalla funzione “getDataByProduct” (riga 28) che controlla se esiste già nella tabella quel prodotto messo dall’utente corrente e solo nel carrello, in caso positivo, quindi se la funzione ritorna un valore nell’array, modifica il valore della quantità richiesta incrementandolo, altrimenti inserisce il valore nuovo.
Qui sotto è mostrata la funzione che prende i dati.

```php
    public function getDataByproduct($prodName, $prodPrice, $prodQuantity){

        //Connetto al database
        $conn = $this->connection->sqlConnection();

        //echo "Connected successfully";
        $sql = $conn->prepare("SELECT * FROM compra WHERE nome_prodotto LIKE :prodName
        AND prezzo_prodotto LIKE :prodPrice AND quantita_prodotto LIKE :prodQuantity AND email_cliente LIKE :custMail");
        $sql->bindParam(':prodName', $prodName, PDO::PARAM_STR);
        $sql->bindParam(':prodPrice', $prodPrice);
        $sql->bindParam(':prodQuantity', $prodQuantity, PDO::PARAM_INT);
        $sql->bindParam(':custMail', $_SESSION['customer'], PDO::PARAM_STR);

        //Eseguo la query
        $sql->execute();

        $dataArray = array();
        //Se ci sono dei valori
        if($sql->rowCount() > 1) {

            // Ciclo tutti i valori
            while ($row = $sql->fetch()) {
                array_push( &array: $dataArray, $row);
            }
            //Se c'è un solo valore lo inserisco
        }else if($sql->rowCount() == 1) {
            $dataArray = $sql->fetch();
        }
        $conn = null;
        return $dataArray;
    }
}
```

Figura 4 model getDataByProduct

Come si vede la funzione ritorna un array, nel caso non dovessero esserci i dati richiesti di conseguenza ritorna un array vuoto.
Questa è stata l'ultima cosa che ho fatto, quindi purtroppo non sono ancora riuscito a completare il carrello.

Problemi riscontrati e soluzioni adottate

Il primo problema che ho avuto è stato la mostra dei prodotti dopo la ricerca, perché avendo cambiato i dati che prende la query, ovvero quado ho messo che prende solo i dati del prodotto, e quindi ho dovuto cambiare l'indice nell'array che gli arriva alla funzione model, mettendo solo "nome" al posto di "nome_prodotto" al controllo dei dati.
Un secondo problema che ho avuto è stato con le foreign key, perché nella tabella "compra" ho dovuto inserire le 2 colonne "prezzo_prodotto" e "quantita_prodotto" che sono le 2 chiavi che ho aggiunto la scorsa lezione, e senza quelle 2 non potevo prendere il prodotto coretto dopo aver inserito i dati nella tabella per mostrarli nel carrello. Per farlo ho dovuto cancellare la tabella e ricrearle, perché è il metodo più rapido avendo già salvato in un file il la creazione della tabella, mi è bastato copiarla e incollarla nel database.
Ho avuto dei problemini con la modifica delle query, ho dovuto provarli hardcoded su "phpMyAdmin" per controllare dov'era l'errore e spesso era un campo inserito male o un dato non valido che passava.

Punto della situazione rispetto alla pianificazione

Ho messo a posto 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.