Diario di lavoro

Luogo	SAMT
Data	12.02.2019

Lavori svolti

Questa mattina ho completato l'inserimento dei dati del prodotto all'interno del database. È stato un processo semplice, a parte un problemino, spigato nella sezione "Problemi riscontrati e soluzioni adottate". Per inserire dati vengono richiamati il controller e il model appositi, spiegati qui sotto.

```
11      public function insertProduct()
12      {
13          if ($_SERVER["REQUEST_METHOD"] == "POST") {
14
15              require_once 'application/models/product.php';
16              $product = new ProductModel();
17
18              // Connssione all'ftp
19              $connFTP = ftp_connect( host: "efof.ftp.infomaniak.com");
20              $login = ftp_login($connFTP,  username: "efof_gestvend",  password: "GestVend_Admin_2018");
21
22              $image = isset($_FILES['imageQuestion'])? $_FILES['imageQuestion'] : null; //Prendo il nome del file
23              $name = $image['name'];
24
25              //Prendo il percorso temporaneo del file e gli cambio nome
26              $tmpName = $image['tmp_name'];
27              $newName = 'application/img/'. $name;
28              rename($tmpName, $newName);
29
30              //Imposto i permessi per il file
31              ftp_chmod($connFTP,  mode: 0664, $newName);
32
33              $category = isset($_POST["category"])? $_POST["category"] : null;
34              $title = isset($_POST["title"])? $_POST["title"] : null;
35              $prize = isset($_POST["prize"])? $_POST["prize"] : null;
36              $quantity = isset($_POST["quantity"])? $_POST["quantity"] : null;
37
38              //Se entrambi i campi non sono vuoti
39              if ($category != null && $title != null && $prize != null && $quantity != null && $image != null) {
40
41                  $var = $product->insertProduct($category, $title, $prize, $quantity, $newName, $newName);
42              }
43
44              header( string: "location: ". URL ."dealer/details");
45          }else{
46              header( string: "location: javascript://history.back()");
47          }
48      }
```

Figura 1 model inserimento prodotto

In questo caso il model svolge un ruolo abbastanza importante, prima stabilisce una connessione con l'FTP (righe 19-20) e dopodiché prende il file che è stato caricato, questo file viene salvato in un file temporaneo, infatti il passaggio successivo (righe 26-28) si occupa di prendere il valore del percorso del file e sostituirlo con il percorso nel ftp in cui inserire il file. Una volta fatto deve impostare i permessi per il file per tutti gli utenti (riga 31), la modalità "0664" imposta lettura e scrittura per il creatore del file e solo lettura per gli altri utenti, in modo che dal sito si possano visualizzare le immagini. La struttura dei permessi nel file è quella mostrata sotto.

Figura 2 permessi file FTP

Dopo aver impostato tutto prendo i valori degli altri campi inseriti e se sono validi richiamo la funzione del controller per inserire i dati. Una volta fatto tutto riapro la pagina di insrimento.

```
40  public function insertProduct($category, $title, $prize, $quantity, $image){
41      //Connetto al database
42      $conn = $this->connection->sqlConnection();
43
44      //Prendo i dati dell'utente in base alla mail
45      $sql = $conn->prepare("INSERT INTO prodotto(nome_categoria, nome, prezzo, quantita, img) values(:categoria, :nome, :prezzo, :quantita, :img)");
46      $sql->bindParam(':categoria', $category, PDO::PARAM_STR);
47      $sql->bindParam(':nome', $title, PDO::PARAM_STR);
48      $sql->bindParam(':prezzo', $prize, PDO::PARAM_INT);
49      $sql->bindParam(':quantita', $quantity, PDO::PARAM_INT);
50      $sql->bindParam(':img', $image, PDO::PARAM_STR);
51
52      //Eseguo la query
53      if(!$sql->execute()){
54          $conn = null;
55          $_SESSION['ExistData'] = true;
56          return $_SESSION['ExistData'];
57      }
58
59      $conn = null;
60      $_SESSION['created'] = true;
61      return $_SESSION['created'];
62  }
```

Figura 3 controller inserimento prodotto

Il controller si occupa semplicemente di inserire i dati passati nel database, sempre on PDO. La cosa particolare è alla fine, se l'inserimento non va a buon fine (righe 53-57) allora chiudo la connessione e ritorno una variabile di sessione per segnalare l'errore, mentre se va tutto a buon fin imposto un'altra variabile che segnala la corretta razione del prodotto nel database. La variabile di sessione creata nel controller viene usata nella pagina di inserimento per segnalare o meno l'errore, come nel caso del login, ma in questa pagina non si può applicare il popup, quindi ho dovuto gestirlo in modo diverso.

Figura 4 errore prodotto già esistente

Nel caso ci sia l'errore viene segnalato sotto al form on la scritta mostrata, ma se tutto va a buon fine allora vado nella pagina principale.
Un'ultima cosa che ho dovuto modificare nella pagina è stata il fatto che quando viene selezionata un'immagine vene mostrata al posto dell'immagine di base, ed è semplicemente bastato inserire una funzione che modifica la source dell'immagine con quello nuovo all'"onchange" dell'input, come mostrato qui sotto.

```
239     function setImage(file){
240         //Controlo che abbia selezionato un file
241         if (file.files && file.files[0]) {
242 
243             //creo un lettore di file
244             var reader = new FileReader();
245 
246             //Quando viene caricato un file
247             reader.onload = function (e) {
248                 //Modifico l'attributo src dell'immagine niserendo quelo del file selezionato
249                 $('#image')
250                     .attr('src', e.target.result)
251             };
252 
253             //Setto i dati del file per l'URL
254             reader.readAsDataURL(file.files[0]);
255         }
256     }
```

Figura 5 mostra immagine selezionata

Un ulteriore cosa che ho dovuto gestire è il fatto che non vengano mostrati tutti i prodotti esistenti ma solo quelli che lui vene nel determinato negozio, per questo ho dovuto creare una nuova funzione che faccia una join per prendere solo i dati adatti, la funzione è la stessa l'unica differenza è la query, che è la seguente:

```
SELECT * from prodotto p
inner join vende v on v.nome_prodotto = p.nome
inner join negozio n on n.nome = v.nome_negozio
      and n.indirizzo = v.indirizzo_negozio
      and n.citta = v.citta_negozio
      WHERE n.email_gestore = :email
```

La query qui mostrata va a prendere tutto dalla tabella prodotto controllando anche le altre tabelle, partendo dal fondo, pria controlla che prende le informazioni dal negozio gestito dall'utente corrente, dopodiché prende tutti i valori nella tabella ponte "vende" che corrispondono a quelli del negozio e infin quelli in "prodotto" che corrispondono alla tabella vende, in questo modo vengono ritornati tutti i dati delle 3 tabella che corrispondono all'utente attuale.
L'ultima cosa he ho fatto è stata fare in modo che vengano mostrati i prodotti di una determinata categoria quando essa viene selezionata nella lista a sinistra, e vengono mostrati tutti se viene cliccato "Tutto".
Per farlo ho creato 2 nuove funzioni, uno che fa la richiesta per prendere i prodotti adatti mentre l'altra per inserire le informazioni, in JavaScript, la 3 funzione contiene il for con l'inserimento dei div mentre le altre 2 sono come quella mostrata qui sotto con la differenza nella funzione che richiamano.

```
function getProductsByCategory(category) {
    xhttp.onreadystatechange = function () {
        if (this.readyState === 4 && this.status === 200) {
            //Prendo i valori passati dal server e li metto in un array
            var obj = JSON.parse(xhttp.responseText);

            //Richiamo la funzione per inserire i prodotti
            insertProducts(obj);
        }
    }
    xhttp.open( method: "POST", url: "/gestionevendita2018/product/getProductsByCategory", async: true);
    xhttp.setRequestHeader( name: "Content-Type", value: "application/x-www-form-urlencoded");
    xhttp.send( body: "&category="+ category);
}
```

Figura 6 funzione mostra prodotti

Questa funzione va a richiamare quella che prende i prodotti in base alla categoria e gli passa la categoria richiesta, viene richiamata al click sul link della categoria.

```
public function getProductsByCategory(){

    if ($_SERVER["REQUEST_METHOD"] == "POST") {

        //Prendo la classe model
        require_once 'application/models/product.php';
        $product = new ProductModel();

        //Prendo il valore della categoria
        $category = isset($_POST["category"])? $_POST["category"] : null;

        $categories = $product->getProductsByCategory($category);

        //Stampo con json i valori dei coach
        header( string: 'Content-Type: application/json');
        echo json_encode($categories);
    }else{
        header( string: "location: javascript://history.back()");
    }
}
```

Figura 7 model selezione prodotti con categoria

La funzione sopra mostrata si occupa di prendere dal controller apposito nella classe “ProductModel” tutti i prodotti in base alla categoria passata e stamparlo come JSON i modo che Ajax possa prenderlo e riutilizzarlo.

```php
public function getProductsByCategory($category){
    //Prendo tutti i prodotti dell'utente corrente
    $data = $this->getDealerProducts();

    //Creo un array vuoto ee faccio passare tutti i dati presi
    $dataArray = array();
    foreach ($data as $value){

        //Se la categoria del prodott è corretta lo inserisco nell'array
        if($value['nome_categoria'] == $category){
            array_push( &array: $dataArray, $value);
        }
    }

    //Ritorno l'array
    return $dataArray;
}
```

Figura 8 model che seleziona i prodotti in base alla categoria

La funzione non va a fare la query ma prende tutti i prodotti dalla funzione creata in precedenza e dopodiché li filtra con un foreach per ritornare, in un array, solo quelli della categoria richiesta.

Problemi riscontrati e soluzioni adottate

Prima di far funzionare tutto ho avuto un problemino con il passaggio dell'immagine tramite il form, questo perché per passare dei file bisogna inserire un attributo speciale nel tag <form>, come mostrato nella foto.

```
<form action="<?php echo URL ?>product/insertProduct" method="POST" enctype="multipart/form-data">
```

La parte da inserire per farlo funzionare è "**enctype="multipart/form-data"**".
Un altro problema che ho trovato è la visualizzazione delle categorie nelle pagine in cui mostro anche i prodotti. A quanto pare avendo 2 funzioni che utilizzano Ajax l'unica che viene presa è l'ultima richiamata, mentre la prima chiamata POST non va a buon fine, per questo motivo nella pagina vengono mostrati tutti i prodotti ma non le categorie. Purtroppo questo problema non sono riuscito a metterlo a posto neanche con l'aiuto del professor Sartori, e di conseguenza ancora non è impostato.
Dopo un po' però ho trovato un modo, ovvero richiamare la funzione alla fine dell'inserimento delle categorie in modo che solo dopo vengano inseriti i prodotti e non allo stesso modo, in questo modo funziona.

Punto della situazione rispetto alla pianificazione

Ho finito la mostra dei prodotti quindi mi sono portato avanti.

Programma di massima per la prossima giornata di lavoro

Completare la pagina di modifica de prodotti inseriti.