

Diario di lavoro

Luogo	SAMT
Data	29.01.2019

Lavori svolti

Questa mattina ho completato il login, ho finito la funzione che ritorna i valori in base ai dati inseriti, ovvero se i dati dell'utente sono validi o meno, come spiegato sotto.

```

34      if($sql->rowCount() > 0){
35          // Ciclo tutti i valori
36          while ($row = $sql->fetch()) {
37              //Controlla che la password sia corretta
38              if (!(strcmp($pass, $row["password"]))) {
39                  // LOGIN VENDITORE
40                  $_SESSION['customer'] = $mail;
41                  $checkSession = "customer";
42              } else {
43                  // PASSWORD ERRATA
44                  $_SESSION['wrongPass'] = "wrongPass";
45                  $checkSession = "wrongPass";
46              }
47          }
48          //Altrimenti
49      }else{
50          // UTENTE INESISTENTE
51          $_SESSION['noUser'] = "noUser";
52          $checkSession = "noUser";
53      }
54
55      //Chiudo la sessione
56      $conn = null;
57      //Ritorno il valore
58      switch ($checkSession) {
59          case "customer":
60              return $_SESSION['customer'];
61          case "wrongPass":
62              return $_SESSION['wrongPass'];
63          case "noUser":
64              return $_SESSION['noUser'];
65          default:
66              return "ERRORE";
67      }

```

Figura 1 model login

La parte iniziale dell'immagine viene subito dopo aver eseguito la query con il comando "\$sql->execute()" e si occupa di controllare che la query abbia prodotto dei risultati, in caso contrario crea la variabile di sessione in cui viene mostrato che non ci sono utenti. Se invece la query da dei risultati con la mail passata, li cicla, ovvero prende l'unico disponibile, e fa un controllo con la

password passata, a dipendenza del risultato crea una variabile di sessione adatta. Se la password è corretta crea la variabile con la mail altrimenti con un "messaggio di errore".

Una volta fatto chiudo la connessione, mettendo la variabile di essa a null, poi con uno switch controllo e a dipendenza del valore della variabile di controllo passo quella di sessione adeguata.

Una volta fatto questo nella funzione controller vengono svolti ulteriori controlli per poi riaprire la pagina corretta.

```

40 public function loginD()
41 {
42     if ($_SERVER["REQUEST_METHOD"] == "POST") {
43
44         //Prendo la classemodel
45         require_once 'application/models/dealer.php';
46         $dealer = new Dealer();
47
48         //Prendo le variabili passate dal post.
49         $email = isset($_POST["email"])? $_POST["email"] : null;
50         $pass = isset($_POST["pass"])? $_POST["pass"] : null;
51         //$pass = hash('sha512', $pass);
52
53         //Se entrambi i campi non sono vuoti
54         if ($email != null && $pass != null) {
55
56             //controllo il login
57             $var = ($dealer->checkLogin($email, $pass));
58
59             //Se viene ritornata la mail apre la pagina dell'utente
60             if(!strcmp($var, $email)){
61                 header( string: "location: ". URL);
62             //Altrimenti torna alla pagina di login
63             }else{
64                 header( string: "location: ". URL ."login/index");
65             }
66         }
67     }
68 }

```

Figura 2 controller login

La funzione dopo aver preso i dati e averli inviati al model, immagazzina il ritorno in una variabile, che poi viene controllata, se è la mail allora apre la pagina dell'utente, che momentaneamente non è completa per cui porta alla pagina principale, ma se non è come la mail ritorna alla pagina di login in cui verrà mostrato il problema, questa stessa funzione è stata copiata per 3 volte, una per ogni tipo di utente, ed è praticamente uguale, l'unica differenza è che fanno riferimento a dei model diversi e aprono pagine diverse.

Una volta fatto il login se dovesse esserci un errore esso viene segnalato con un popup che mostra un messaggio di errore.

Questo sistema l'ho creato con un template preso online a questo link: <https://codyhouse.co/gem/simple-confirmation-popup> in questo link si può sia vedere la demo che andare alla pagina i github in cui sono presenti tutti i file utili per il popup.

Qui sotto è mostrato come appare a schermo il popup una volta mostrato, il bottone non è centrato al 100% ma è un problema su cui ho deciso di non perdere troppo tempo essendo che il resto funziona perfettamente e sono già indietro.

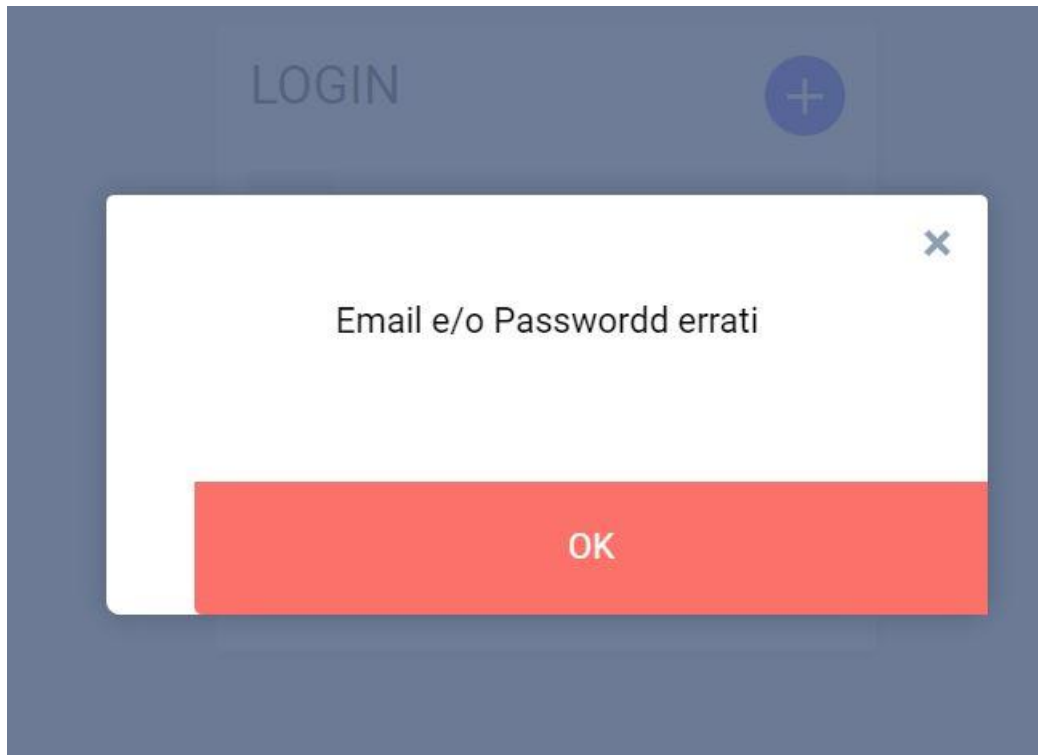


Figura 3 popup login errato

Inizialmente per funzionare viene parte il codice PHP mostrato nell'immagine sottostante.

```

6      <?php
7      if(isset($_SESSION['wrongPass'])){
8          print "<script type='text/javascript'>
9              linkClick();
10         </script>";
11
12         unset($_SESSION['wrongPass']);
13     }else if(isset($_SESSION['noUser'])){
14         print "<script type='text/javascript'>
15             linkClick();
16         </script>";
17
18         unset($_SESSION['noUser']);
19     }
20
21     ?>
    
```

Figura 4 PHP attivazione popup

Il codice controlla inizialmente se una delle due variabili di sessione esiste, in caso di risposta positiva stampa uno script che fa partire una funzione nel file JavaScript apposito per il popup, la funzione è quella mostrata nell'immagine che segue.

Una volta finito il codice viene disattivata la variabile di sessione in modo che se si aggiorna la pagina non viene mostrato nuovamente, e inutilmente, il popup.

Come ultima cosa ho ultimato la registrazione inserendo l'invio della mail ma ancora riscontra alcuni problemi per cui lo completerò definitivamente domani.

```

1  jQuery(document).ready(function($){
2
3      //close popup
4      $('.cd-popup').on('click', function(event){
5          if( $(event.target).is('.cd-popup-close') || $(event.target).is('.cd-popup') || $(event.target).is('.ok-close')) {
6              event.preventDefault();
7              $(this).removeClass( @: 'is-visible');
8          }
9      });
10     //close popup when clicking the esc keyboard button
11     $(document).keyup( e: function(event){
12         if(event.which=='27'){
13             $('.cd-popup').removeClass( @: 'is-visible');
14         }
15     });
16 });
17
18 function linkClick(){
19     $('.cd-popup').addClass( @: 'is-visible');
20 }

```

Figura 5 JQuery popup

La funzione che viene richiamata è quella in fondo, che va a rendere visibile il div per il popup e una volta fatto si vede il popup come mostrato nella foto di sopra, sopra a quella funzione ci son altre due funzioni che partono a determinati eventi, la prima viene attivata in caso vengano premuti la "x" in alto a destra o il bottone ok e la seconda quando viene cliccato al di fuori del popup, ed entrambi hanno lo stesso compito, chiudono, o riendono invisibile, la schermata mostrando di nuovo il login.

Problemi riscontrati e soluzioni adottate

Inizialmente ho avuto dei problemi nella query, perché prima provavo a fare tutti e tre i login nella stessa funzione quindi dovevo decidere la tabella da inserire nella query **"SELECT * FROM :table WHERE email LIKE :mail"** e doveva essere diversa per ogni tipo di utente, ma il metodo mostrato nella stringa, ovvero mettere anche la tabella come variabile, non è utilizzabile con PDO, che è la classe che ho deciso di utilizzare per la comunicazione con il database.

Per ovviare a questo problema ho optato per creare il metodo delle 3 funzioni spiegato nella sezione precedente, in questo modo ogni funzione ha una query che fa riferimento alla tabella dell'utente proprio.

Un altro problema che ho avuto è stato con l'inserimento del popup. Essendo che l'ho preso da un template, ho dovuto estrapolare solo ciò che mi serviva, e quindi ho dovuto anche riadattare tutti i file come è stato per la pagina di login, quindi questo mi ha preso un po' di tempo.

Un altro problema è stato l'adattamento del codice JavaScript, essendo che il template originale faceva apparire la schermata al click di un bottone il tutto era gestito all'evento "onclick" del link, ma essendo che non avevo un link ma doveva apparire quando si tornava alla pagina ho dovuto riadattarlo.

Inizialmente ho provato a creare un link nascosto che tramite php se viene settata la variabile di sessione viene cliccata, da codice con JavaScript con il codice

"document.getElementById('id').click()" i modo da scatenare l'evento di JQuery, ma questo non funzionava essendo che il click da programma non attiva JQuery, cosa di cui mi sono accorto a mie spese, e quindi ho dovuto gestirlo in un altro modo, e ho optato per il metodo mostrato nella sezione precedente.

Punto della situazione rispetto alla pianificazione

Sono in sempre un po' in ritardo, ho finito oggi il login.

Programma di massima per la prossima giornata di lavoro

Portare avanti la pagina di messa in vendita dei prodotti.