

Diario di lavoro

Luogo	SAMT
Data	04.02.2019

Lavori svolti

Questa mattina essendo che avevo solo queste 2 ore ho deciso di utilizzarle per portare a termine i controlli dei campi nel form della registrazione, in modo che se si scrivono campi errati viene segnalato colorando in rosso il testo errato come mostrato nell'immagine qui sotto. Le prime 2 immagini mostrano il controllo applicato ai campi che possono contenere solo lettere, come ad esempio nome e cognome.

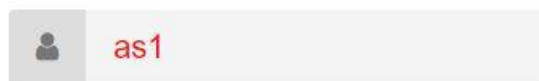


Figura 1 campo solo lettere errato

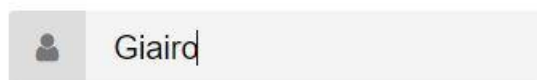


Figura 2 campo solo lettere corretto

Le foto che seguono mostrano il campo del numero di telefono, per questo vengono accettati i numeri gli spazi e il più solo all'inizio, se si dovesse mettere un più in un punto diverso viene segnalato l'errore come qualsiasi altro caso sbagliato.

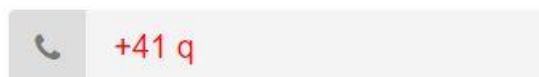


Figura 3 campo numero di telefono errato



Figura 4 campo numero di telefono corretto

I campi mostrati qui sotto mostrano il controllo della mail in cui bisogna inserire il seguente tipo di stringa nell'input: "Stringa + @ + stringa + . + stringa", all'interno di "stringa" può andarci qualsiasi cosa.

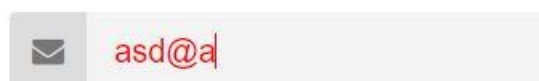


Figura 5 campo email errato



Figure 6 shows a light gray input field with an envelope icon on the left. The text 'asd@a.a' is entered in the field.

Figura 6 campo email corretto

L'ultimo campo da controllare è quello della password, per questo è un po' più complicato perché l'errore non viene segnalato colorando il testo, di conseguenza ho sviluppato un altro metodo, scrivere l'errore sotto ai campi, essendo che ci sono 2 campi per la password, una normale e una di verifica. I requisiti per la password sono sia lettere che numeri e da 8 a 25 caratteri e quelle mostrate qui sotto sono i messaggi di errore possibili che vengono mostrati.

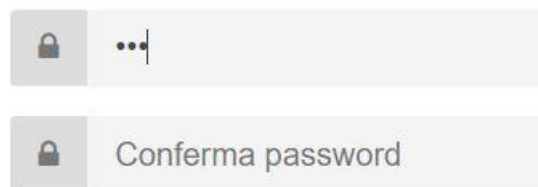


Figure 7 shows two password input fields. The first field has a lock icon and three dots. The second field has a lock icon and the text 'Conferma password'. Below the fields is a red error message: 'Password non valida (8-25 caratteri, sia lettere che numeri)'.

Figure 7 shows a red error message: 'Password non valida (8-25 caratteri, sia lettere che numeri)'.

Figura 7 password non valida



Figure 8 shows two password input fields. The first field has a lock icon and eight dots. The second field has a lock icon and three dots. Below the fields is a red error message: 'ERRORE: password diverse'.

Figure 8 shows a red error message: 'ERRORE: password diverse'.

Figura 8 password non coincidenti



Figure 9 shows two password input fields. The first field has a lock icon and eight dots. The second field has a lock icon and eight dots. Below the fields is a blue button with the text 'REGISTER'.

Figura 9 password corrette

Come si può anche notare dall'ultima immagine se i dati sono tutti corretti viene attivato il bottone che permette la registrazione, se invece ci sono segnalati dei messaggi di errore esso rimane disattivo. C'è la possibilità che il bottone sia attivo con dei campi vuoti, ma in quel caso se viene cliccato darà errore al campo vuoto perché sono tutti "required".

Tutti i controlli sono fatti con JavaScript grazie a delle funzioni. Nella funzione di convalida generale mostrata nel diario del 28.01.2019 si nota che se passa il controllo della mail si apre una nuova funzione, questa si occupa di fare una richiesta con Ajax ad una funzione che ritorna tutti i dati degli utenti nel database e compara la mail a quella scritta per controllare che non esista già, e se dovesse esistere segnala un errore alla fine del form.

```

if(id.name == "emailR"){
    //Colora il testo di nero
    id.style.color = "black";

    //Controlla la mail
    checkMail(value);
}

```


Figura 10 convalida email


```


66 function checkMail(mail){
67     var mailExists = document.getElementById( elementid: 'mailExists');
68     var type = document.getElementById( elementid: 'type').value;
69
70     xhttp.onreadystatechange = function() {
71         if (this.readyState === 4 && this.status === 200) {
72
73             //Prendo tutti i valori passati dalla pagina
74             var results = JSON.parse(xhttp.responseText);
75
76             //Controlla che l'email inserita non esista già nel database
77             for(var i = 0; i < results.length; i++){
78
79                 //Se la mail esiste già cambio la variabile di controllo e smetto di controllare segnalando il messaggio d'errore
80                 if(results[i]['email'] == mail){
81                     mailExists.hidden = false;
82                     checkMailE = false;
83                     break;
84                 }
85                 //Se la mail non esiste nasconde il messaggio d'errore e cambio la variabile di controllo.
86                 else{
87                     checkMailE = true;
88                     mailExists.hidden = true;
89                 }
90             }
91
92             //Se la mail non esiste va avanti
93             if(checkMailE){
94                 confirm();
95             }
96             //se la mail esiste
97             else{
98                 //
99             }
100         }
101     };
102     xhttp.open( method: "POST", url: "/MVC/registration/checkMail/", async: true);
103     xhttp.setRequestHeader( name: 'Content-type', value: 'application/x-www-form-urlencoded');
104     xhttp.send( body: "type="+ type);


```

Figura 11 funzione controllo unicità mail









ERRORE: mail già in uso

Figura 12 email già esistente

Le funzioni vengono prese da model e controller che si occupano solo di questo.

```
75 public function checkMail(){
76     if($_SERVER['REQUEST_METHOD'] == 'POST'){
77
78         $user = "";
79
80         switch ($_POST['type']){
81             case "customer":
82                 require_once 'application/models/customer.php';
83                 $user = new Customer();
84                 break;
85             case "dealer":
86                 require_once 'application/models/customer.php';
87                 $user = new Customer();
88                 break;
89             case "admin":
90                 require_once 'application/models/customer.php';
91                 $user = new Customer();
92                 break;
93         }
94
95         //Stampo con json tutti gli utenti
96         header( string: 'Content-Type: application/json');
97         echo json_encode($user->getUsers());
98     }else{
99         header( string: "location: javascript://history.back()");
100     }
101 }
```

Figura 13 controller checkMail

La funzione sopra mostrata è quella richiamata da Ajax e si occupa solo di creare un oggetto della classe richiesta in base alla registrazione che si vuole fare e poi richiamare la funzione nella determinata classe.

```

140 public function getUsers() {
141     //Connetto al database
142     $conn = $this->connection->sqlConnection();
143
144     //Prendo i dati dell'utente in base alla mail
145     $sql = $conn->prepare( statement: "SELECT * FROM cliente");
146     $sql->execute();
147
148     $dataArray = array();
149
150     //Se ci sono dei valori
151     if($sql->rowCount() > 0){
152
153         // while che prende tutti i dati
154         while($row = $sql->fetch()) {
155             array_push( &array: $dataArray, $row);
156         }
157     } else {
158         $conn = null;
159         return false;
160     }
161
162     $conn = null;
163     return $dataArray;
164 }

```

Figura 14 model getUsers

La funzione “getUsers” sopra mostrata è quella che si occupa di prendere tutti i dati degli utenti di una determinata tabella, per ogni tipo di utente i dati vengono presi dalla propria funzione, e quindi la funzione è pressoché la stessa cambia solo quel dettaglio.

Problemi riscontrati e soluzioni adottate

Ho avuto un problema all’inizio con il css, perché, come spiegato precedentemente nei diari, ho spostato le cartelle di css e JavaScript delle pagine principali in una cartella unica in modo che tutti facciano riferimento a quello, ma a quanto pare non si era salvato il cambiamento di “src” nel tag link e quindi non riusciva a caricare lo stile delle pagine.
Oltre a questo non ho avuto grandi problemi se non alcune modifiche fatte durante la scrittura dei codici per i controlli della registrazione, ma seguenti la sezione precedente tutto risulta funzionante.

Punto della situazione rispetto alla pianificazione

Oggi ho iniziato semplicemente completato la registrazione.

Programma di massima per la prossima giornata di lavoro

Portare avanti la pagina di messa in vendita dei prodotti.