



Documentazione del Progetto Carrello

Sommario

Documentazione del Progetto Carrello  	1
Sommario.....	1
Introduzione.....	2
Funzionalità Sviluppate.....	3
1. Registrazione.....	3
2. Login.....	4
3. Visualizzazione Prodotti.....	5
4. Carrello.....	6
5. Aggiunta di Prodotti da Parte di un Admin.....	7
Funzioni Extra.....	8
Password Dimenticata.....	8

Introduzione

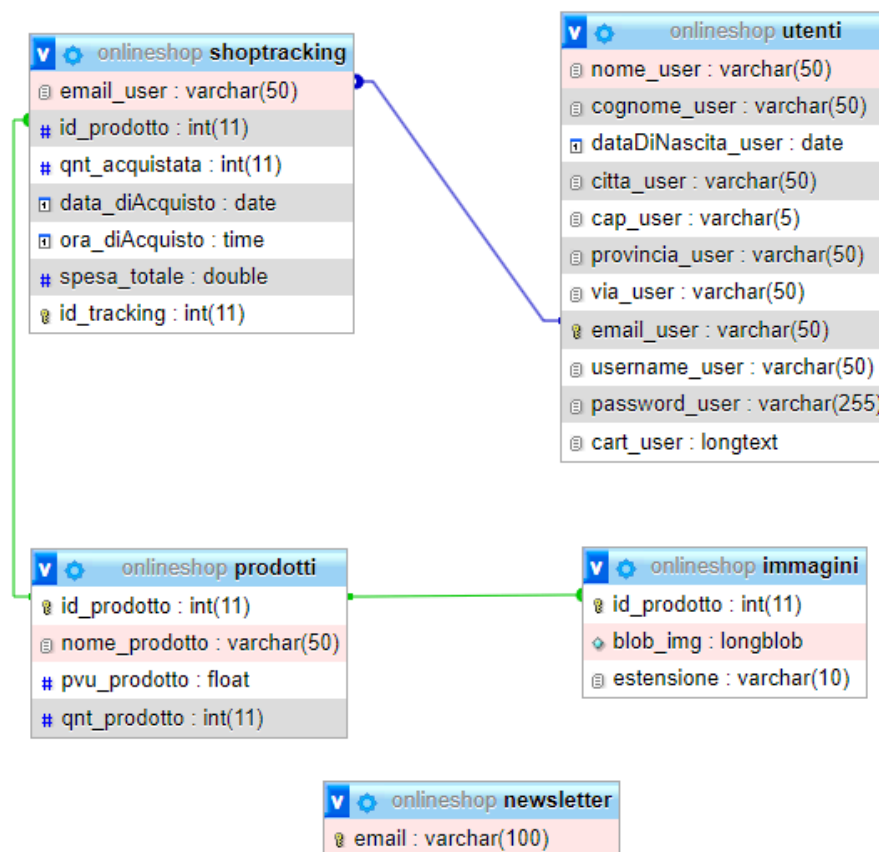
Il progetto “Carrello” è un’applicazione web che consente agli utenti di visualizzare una lista di prodotti e di effettuare acquisti all’interno di un ipotetico supermercato online.

L’applicazione offre funzionalità di registrazione, login, visualizzazione dei prodotti, aggiunta al carrello e svuotamento del carrello. Inoltre, sono previste funzionalità avanzate come il carrello avanzato, la traccia degli acquisti e l’aggiunta di prodotti da parte di un amministratore.

Passando alla parte di programmazione ogni funzione avrà una propria pagina php che utilizza elementi grafici della libreria di Bootstrap e metodi del file src.php.

Il file `src.php` contiene tutte le funzioni utili per le pagine PHP dell’applicazione dai metodi di registrazione, a quelli che si interfacciano con la classe SMTP per mandare email a quelli di acquisto prodotto.

Database Utilizzato



Funzionalità Sviluppate

1. Registrazione

- Gli utenti devono registrarsi per utilizzare i servizi dell'applicazione.
- I dati dell'utente registrato includono:
 - Indirizzo email (chiave)
 - Username
 - Password
 - Data di nascita
 - Cap
 - Cognome
 - Nome
 - Indirizzo
 - Città
 - Provincia
- Questo processo è dotato di un controllo presente nel file src.php che garantisce che tutti i dati siano stati inseriti e che la password sia corretta.
- La password viene cifrata utilizzando un algoritmo di hash per garantire che anche con l'accesso al database non si conoscano le password degli utenti registrati.
- Utilizzati dei metodi di mysql per evitare attacchi di SQL Injection.
- Il metodo "signup" fa parte delle funzioni fornite dal file src.php e esegue la creazione della query con i dati forniti, l'inserimento del database e il controllo delle informazioni con ulteriore funzione dataVerify che restituisce, in caso di errore, la lista di tutte le incongruenze riscontrate.

```
<?php

if(isset($_POST["signup"])){

    $username =
$db_connection->real_escape_string(stripslashes($_POST["username"]));

    $email = $db_connection->real_escape_string(stripslashes($_POST["email"]));

    $nome = $db_connection->real_escape_string(stripslashes($_POST["nome"]));

    $cognome =
$db_connection->real_escape_string(stripslashes($_POST["cognome"]));

    $dataNascita =
$db_connection->real_escape_string(stripslashes($_POST["dataNascita"]));

    $citta = $db_connection->real_escape_string(stripslashes($_POST["citta"]));

    $cap = $db_connection->real_escape_string(stripslashes($_POST["cap"]));

    $provincia =
$db_connection->real_escape_string(stripslashes($_POST["provincia"]));
```

```

        $via =
$db_connection->real_escape_string(stripslashes($_POST["via"]))."/". $db_connection
->real_escape_string(stripslashes($_POST["via2"]))."/". $db_connection->real_escape
_string(stripslashes($_POST["via3"]));

        $password =
$db_connection->real_escape_string(stripslashes($_POST["password"]));

        $confermaPassword =
$db_connection->real_escape_string(stripslashes($_POST["confermaPassword"]));

        $error =
signup($username,$email,$nome,$cognome,$dataNascita,$citta,$cap,$provincia,$via,$pa
ssword,$confermaPassword);

        if($error == "Errore/i: "){

            alert("Registrazione avvenuta con successo!");

            echo '<script> window.location.href = "login.php"; </script>';

        }else{

            alert("Registrazione fallita! ".$error);

        }

    }    ?>

```

2. Login

- Il login viene effettuato verificando la corrispondenza tra username e password.
- Anche qui vengono utilizzati metodi per evitare SQL Injection
- Viene utilizzata la funzione “login” di src.php che confronta i valori con quelli presenti nel database (password deve essere cifrata per essere controllata)
- Se tutto va a buon fine, viene creata una sessione con attributo username e come valore il nome/email dell’utente, e si viene trasportati alla pagine della visualizzazione prodotti

```

<?php

    if(isset($_POST["login"])){

        include "connessione.php";

        $username =
$db_connection->real_escape_string(stripslashes($_POST["username"]));

        $password =
$db_connection->real_escape_string(stripslashes($_POST["password"]));

```

```

if(login($username,$password)){

    alert("Utente loggato con successo! Trasferimento in corso...");

    $_SESSION['username'] = $username;

    echo '<script> window.location.href = "index.php"; </script>';

}else{

    alert("Errore durante il login. Username/email non esistente o password errata.");

}

} ?>

```

3. Visualizzazione Prodotti

- I prodotti o servizi vengono visualizzati in “card”.
- Ogni card include:
 - Nome del prodotto
 - Immagine
 - Prezzo
 - Campo per indicare la quantità da acquistare
 - Tasto per aggiungere al carrello
 - Tasto per acquistare subito la quantità scelta



Il tasto “Acquista ora” porterà ad una pagina dedicata all’acquisto (utilizzata anche dal carrello) con parametri: [id_prodotto:n;quantità:k]. Verrà chiesta una conferma e acquistato il prodotto, rimossa la quantità dal database (visto che viene eseguito un controllo prima di acquistare/aggiunge al carrello che la quantità desiderata sia minore o uguale a quella presente nel database) e registrato l’acquisto nell’apposita tabella “shopTracking”, definita più avanti nel documento.

Il tasto “Aggiungi al carrello” aggiunge il prodotto ad un array (composto dai seguenti parametri: \$ARRAY[(id:n,quantità:k),(id:n1,quantità:k1)]) aggiunto alla sessione dell’utente e poi utilizzato dal successivo componente.

4. Carrello



- **Acquisto Prodotti**
 - Dopo aver selezionato un prodotto e scelto la quantità, l'utente può aggiungerlo al carrello.
 - La tabella del carrello mostra:
 1. Nome del prodotto
 2. Prezzo unitario
 3. Quantità selezionata
 4. Totale costo del prodotto
 5. Totale costo
 6. Tasto per eliminare il singolo prodotto dal carrello
 - Il costo totale del carrello viene riportato a destra della tabella.
 - Il tasto "Acquista" porta alla pagina "acquista.php" utilizzato precedentemente per l'acquisto del singolo prodotto: in questo caso la quantità sarà nulla, mentre l'id = -1 (nessun prodotto potrà mai avere questo id quindi la pagina php capisce che l'utente vuole acquistare l'intero carrello). Per ogni prodotto presente nel carrello viene fatto un acquisto, scalata la quantità nel database e registrato all'interno dello shopTracking, come accade per l'acquisto ora.
- **Svuota Carrello**
 - L'utente può eliminare tutti i prodotti dal carrello tramite la pagina "clearCart.php" che imposta l'attributo array nella sessione a null.
- **Traccia degli Acquisti**
 - Riepiloga tutti gli acquisti dell'utente, mostrando:
 1. email_user
 2. id_prodotto
 3. qnt_acquistata
 4. data_diAcquisto
 5. ora_diAcquisto
 6. spesa_totale
 7. id_tracking

5. Aggiunta di Prodotti da Parte di un Admin

- L'amministratore può aggiungere prodotti al database, inclusa l'aggiunta di immagini.
- Per entrare in questa pagina è necessario accedere utilizzando questo [link](#), dove è presente un attributo password con valore "Ciao123!" che viene richiamato tramite metodo GET.



è possibile inserire un prodotto con tutti i suoi attributi, le immagini saranno caricate all'interno del database sotto forma di LONGBLOB, insieme alla propria estensione, utilizzata per codificare l'immagine nel formato utilizzato e visualizzarla all'interno del carrello.

```
<?php

include "connessione.php";

if (isset($_POST['submit'])) {

    $nome = $_POST["nome"];

    $descrizione = null;

    $quantita = $_POST["quantita"];

    $pvu = $_POST["pvu"];

    $immagine = null;

    $ImgData = null;

    if(isset($_POST['descrizione'])) {

        $descrizione = $_POST["descrizione"];

    }

}
```

```

$image = $_FILES['image']['tmp_name'];

$imgData = addslashes(file_get_contents($image));

$fileName = basename($_FILES["image"]["name"]);

    $estensione = pathinfo($fileName, PATHINFO_EXTENSION);
aggiungiAlMagazzino($nome,$descrizione,$quantita,$pvu,$estensione,$imgData);

}

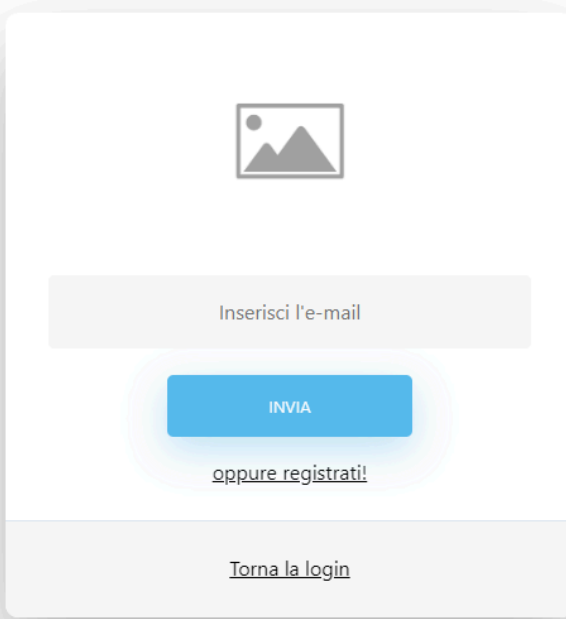
?>

```

>codice utilizzato: la funzione “aggiungiAlMagazzino” viene presa dal file src.php e consiste nella creazione della query con gli attributi passati e le varie aggiunti al database, sia del prodotto che dell’immagine.

Funzioni Extra

Password Dimenticata



Pagina utilizzata per richiedere il reset della password in caso di dimenticanza della stessa. Viene mandato un codice OTP (One Time Password) all’email inserita (tramite una libreria SMTP installata con l’utilizzo di composer), se il codice viene inserito correttamente allora sarà garantito l’accesso ad una pagina che prevede il reset della password all’interno del database.