Procedura di *login* e sessione

Costruire una pagina Web con un form di *login* per l'inserimento delle credenziali dell'utente (email, password).

Si consideri il database *db1* su server MySQL con la tabella *utenti* avente la seguente struttura:

utenti (ID, Cognome, Nome, Email, Telefono, Citta, Provincia, Password)

La pagina Web seguente realizza la procedura di **login** per l'acquisizione dei dati di un utente (*credenziali*) e per la sua identificazione.

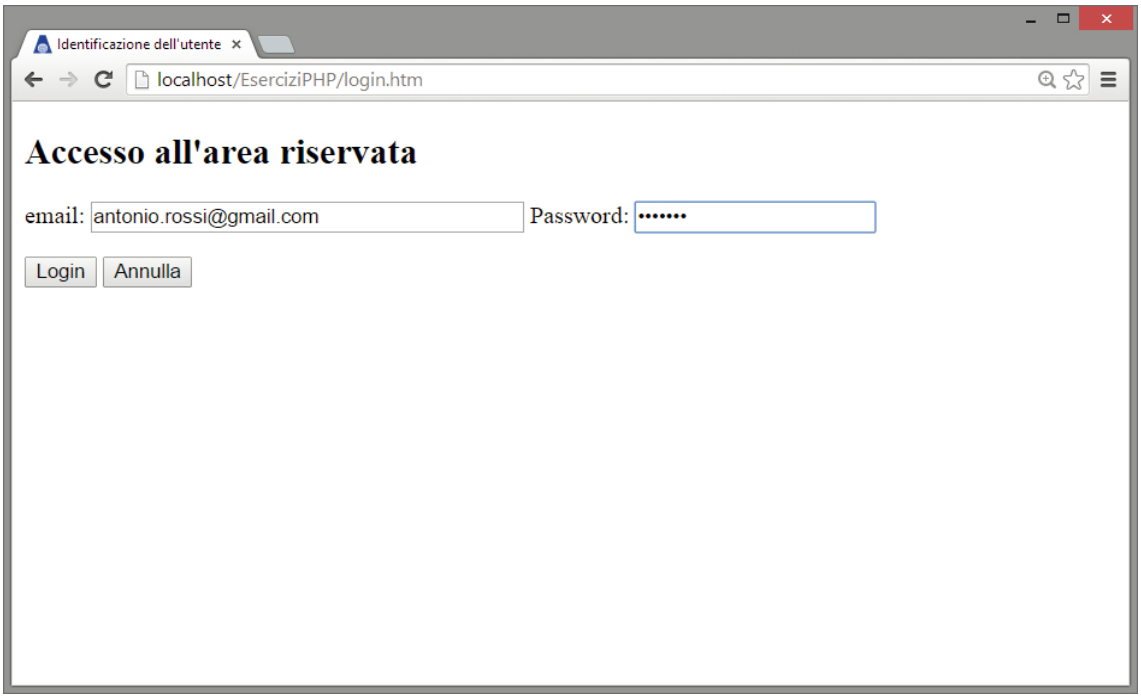
La pagina HTML presenta la maschera di *login* e richiama la pagina PHP per il controllo.

Lo script PHP controlla che l'utente sia registrato sul server, fornendo anche un messaggio con l'esito del controllo. In caso positivo fornisce l'accesso alla prima pagina del sito, contenente anche un link per consentire all'utente di disconnettersi (*logout*).

Pagina HTML *(login.htm)*

```
<!doctype html>
<html>
<head>
   <title>Identificazione dell'utente</title>
</head>
<body>
   <h2>Accesso all'area riservata</h2>
   <form action="controllalogin.php" method=”post”>
      <p>
         email: <input type="text" name="email" size="40" />
         Password: <input type="password" name="password" size="40" /><br/>
      </p>
      <p>
         <input type="submit" name="invio" value="Login" />
         <input type="reset" name="cancella" value="Annulla" />
      </p>
   </form>
</body>
</html>
```

Casella di input di tipo **password**: sostituisce i caratteri inseriti con i puntini durante l'inserimento della password.



Lo script seguente, richiamato dalla pagina di *login*, accede al database *db1* sul server e controlla se l'utente è registrato nella tabella *utenti*: in caso positivo apre la pagina di accesso al sito creando una nuova *sessione* Web, altrimenti visualizza un messaggio di identificazione non riuscita.

> Con il termine **sessione** si intende un insieme di accessi dello stesso utente a pagine Web diverse. La gestione della sessione consiste essenzialmente nel registrare le informazioni dell'utente, in modo da consentire la navigazione nelle altre pagine del sito senza richiedere ogni volta le credenziali dell'utente.

Ogni pagina del sito, al momento del caricamento, controlla che i dati dell'utente siano già registrati in una sessione: in caso negativo, rimanda alla pagina di *login*.
Una sessione termina nel momento in cui l'utente effettua un *logout* oppure chiude il browser.

Pagina PHP *(controllalogin.php)*

```php
<?php
$host="localhost";
$username="root";
$password="root";
$db_nome="db1";
$tab_nome="utenti";
mysql_connect($host, $username, $password) or die('Impossibile connettersi al
server: ' . mysql_error());
mysql_select_db($db_nome) or die ('Accesso al database non riuscito: '
. mysql_error());
// acquisizione dati dal form HTML
$email = $_POST["email"];
$password = $_POST["password"];
```

```php
// protezione per SQL injection
$email = stripslashes($email);
$password = stripslashes($password);
$email = mysql_real_escape_string($email);
$password = mysql_real_escape_string($password);
$passmd5 = md5($password);
```

Password crittografata con la funzione **md5**.

```php
// lettura della tabella utenti
$sql="SELECT * FROM $tab_nome WHERE Email='$email' AND Password='$passmd5'";
$result=mysql_query($sql);
$conta=mysql_num_rows($result);
if($conta==1){
    session_start();
    $_SESSION['email'] = $email;
    $_SESSION['password'] = $passmd5;
    header("Location: loginok.php");
}
else {
    echo "Identificazione non riuscita: nome utente o password errati <br />";
    echo "Torna a pagina di <a href=\"login.htm\">login</a>";
}
?>
```

Avvio della sessione.

Reindirizzamento a un'altra pagina (*loginok.php*).

Lo script utilizza due funzioni per la protezione dalla *SQL injection*, che è presentata nei *Contenuti digitali integrativi* di questo capitolo.

In particolare:

- la funzione **stripslashes** rimuove i backslash (\) da una stringa (\' diventa '). I doppi backslash (\\) sono ridotti ad un singolo backslash (\);
- la funzione **mysql_real_escape_string** aggiunge le sequenza di *escape* ai caratteri speciali in una stringa in modo che sia utilizzata in modo sicuro nei comandi *mysql_query*.

Inoltre la password viene crittografata con la funzione **md5**, applicando l'algoritmo MD5; in questo modo nella tabella del database la password viene registrata non in chiaro, ma come una stringa di 32 cifre esadecimali (128 bit).
Si osservi che, se si vuole crittografare il campo *password* nella tabella con la funzione MD5, occorre che la lunghezza del campo sia almeno di 32 caratteri.
Se la crittografia riguarda una tabella nella quale le password siano state già inserite in chiaro, si può usare la funzione MD5 per aggiornare i valori nel campo *password* con il seguente comando SQL:

```sql
UPDATE utenti SET password = MD5(password);
```

Il controllo sulla presenza dell'utente nel database viene eseguito tramite il numero delle righe restituite dalla query (**mysql_num_rows**) che deve essere uguale a 1:

```php
$conta=mysql_num_rows($result);
if($conta==1){
. . . . . .
}
```

Nel caso di esito positivo del controllo, lo script avvia una sessione (**session_start**) registrando le informazioni dell'utente nella variabile **$_SESSION** (array associativo).

Inoltre il comando **header** avente la sintassi generale:

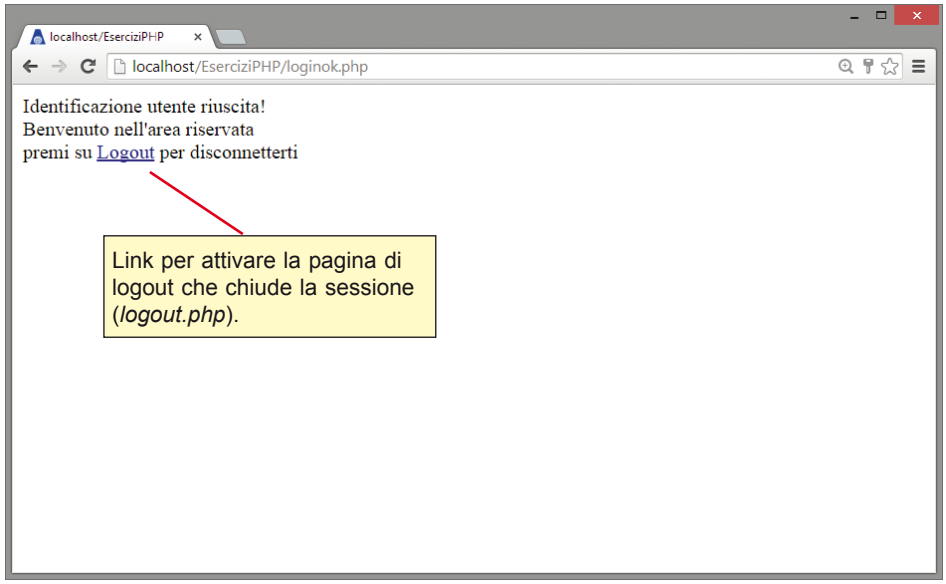
```
header("Location: URL pagina");
```

provoca il reindirizzamento a un'altra pagina *loginok.php*, che potrebbe essere la prima pagina dell'area riservata alla quale possono accedere gli utenti autenticati.
Nel caso di esito negativo del controllo sull'identificazione dell'utente, viene visualizzato un messaggio di identificazione non riuscita e l'utente deve tornare alla pagina di *login*.

Pagina PHP *(loginok.php)*

```
<?php
session_start();
if(!isset($_SESSION['email'])) {
    header("Location: login.htm");
}
?>
<html>
<body>
    Identificazione utente riuscita! <br />
    Benvenuto nell'area riservata <br />
    premi su <a href="logout.php">Logout</a> per disconnetterti
</body>
</html>
```

Se non sono state impostate le variabili di sessione, torna alla pagina di *login*.



Le pagine del sito con accesso riservato devono contenere la scelta per il *logout*, che attiva i comandi per eliminare le variabili di sessione:

```
$_SESSION = array();
```

e per eliminare la sessione stessa:

```
session_destroy();
```

Pagina PHP *(logout.php)*

```php
<?php
session_start();
// elimina le variabili di sessione impostate
$_SESSION = array();
// elimina la sessione
session_destroy();
echo "Disconnessione riuscita, arrivederci!"
?>
```