



ACCESSO A UNA BASE DI DATI IN LINGUAGGIO PHP

PROF. SANTANIELLO ASSUNTA

ACCESSO AD UNA BASE DI DATI IN LINGUAGGIO PHP: FUNZIONI FONDAMENTALI

Il linguaggio PHP dispone della libreria denominata *mysqli* per l'interfacciamento con DBMS MySQL. Questa libreria può essere usata sia utilizzando il paradigma di programmazione procedurale sia quello a oggetti. L'approccio procedurale è basato sulle seguenti funzioni fondamentali:

Funzione	Descrizione		
<code>mysqli_connect</code>	Connessione a un server DBMS specificando nome (o indirizzo IP) del computer su cui è in esecuzione, username, password e database. Restituisce un identificativo della connessione.	<code>mysqli_fetch_row</code>	Recupera una riga del risultato di una query in forma di array indicizzato. Restituisce un array, oppure NULL se non vi sono ulteriori righe nel risultato.
<code>mysqli_connect_errno</code>	Restituisce l'eventuale codice numerico di errore della connessione, il valore 0 se la connessione è stata stabilita correttamente.	<code>mysqli_fetch_assoc</code>	Recupera una riga del risultato di una query in forma di array associativo avente come chiavi i nomi delle colonne. Restituisce un array oppure NULL se non vi sono ulteriori righe nel risultato.
<code>mysqli_query</code>	Esecuzione di un comando SQL. Per i comandi DML restituisce true in caso di successo, oppure false in caso di errore. Per le query restituisce un riferimento al risultato, oppure false in caso di errore.	<code>mysqli_fetch_array</code>	Recupera una riga del risultato di una query in forma combinata di array indicizzato e associativo. Restituisce un array oppure NULL se non vi sono ulteriori righe nel risultato.
<code>mysqli_affected_rows</code>	Restituisce il numero di righe coinvolte dall'ultimo comando DML eseguito, il valore -1 in caso di errore.	<code>mysqli_free_result</code>	Rilascia la memoria del risultato di una query. Non restituisce nulla.
<code>mysqli_num_rows</code>	Restituisce il numero di righe del risultato di una query.	<code>mysqli_close</code>	Chiude la connessione al server DBMS. Restituisce true in caso di successo, oppure false in caso di errore.

ACCESSO AD UNA BASE DI DATI IN LINGUAGGIO PHP: REALIZZA IL DATABASE TRAMITE I COMANDI SQL

Per poter realizzare una base di dati in linguaggio php, bisogna:

- Creare il db:

```
CREATE DATABASE hogwarts;  
USE hogwarts;
```

-
- Creare due tabelle: personaggi e casehw:

```
CREATE TABLE Personaggi (  
id_personaggio VARCHAR(2) NOT NULL,  
nome VARCHAR(32) NOT NULL,  
id_casa CHAR(2) NOT NULL,  
CONSTRAINT chiave_primaria PRIMARY KEY (id_personaggio),  
CONSTRAINT chiave_esterna FOREIGN KEY (id_casa) REFERENCES Casehw (id_casa)  
);
```

```
CREATE TABLE Casehw (  
id_casa CHAR(2) NOT NULL,  
denominazione VARCHAR(16) NOT NULL,  
CONSTRAINT chiave_primaria PRIMARY KEY (id_casa)  
);
```

-
- Popolare le tabelle:

```
INSERT INTO Personaggi (id_personaggio, nome, id_casa) VALUES  
('HP', 'Harry Potter', 'GD'),  
('HG', 'Hermione Granger', 'GD'),  
('RW', 'Ron Weasley', 'GD'),  
('CD', 'Cedric Diggory', 'TR'),  
('TL', 'Ted Lupin', 'TR'),  
('DM', 'Draco Malfoy', 'SV'),  
('SP', 'Severus Piton', 'SV'),  
('LL', 'Luna Lovegood', 'CN'),  
('MM', 'Mirtilla Malcontenta', 'CN');
```

ACCESSO AD UNA BASE DI DATI IN LINGUAGGIO PHP: L'INTERFACCIA DEL LINGUAGGIO PHP - VISUALIZZAZIONE

Il seguente codice hogwarts.php genere la pagina **dinamica raffigurata**: una tabella che elenca i personaggi memorizzati nel database hogwarts e la relativa casa di appartenenza.

```
<html>
  <head>
    <title>DB & PHP test</title>
  </head>
  <body>
    <?php
      $connection = mysqli_connect("localhost", "root", "", "hogwarts");
      $query = "SELECT *
                  FROM Personaggi, Casehw
                 WHERE Personaggi.id_casa = Casehw.id_casa
                   ORDER BY denominazione, nome";
      $result = mysqli_query($connection, $query);
      if (mysqli_num_rows($result) != 0) {
        echo "<table border>";
        echo "<tr>";
        echo "<th>Id</th>";
        echo "<th>Personaggio</th>";
        echo "<th>Casa</th>";
        echo "</tr>";
      }
```

```
      while ($row = mysqli_fetch_array($result)) {
        echo "<tr>";
        echo "<td>$row[id_personaggio]</td>";
        // echo "<td>$row[0]</td>";
        echo "<td>$row[nome]</td>";
        // echo "<td>$row[1]</td>";
        echo "<td>$row[denominazione]</td>";
        // echo "<td>$row[1]</td>";
        echo "</tr>";
      }
      echo "</table>";
    }
    else
      echo "Nessun personaggio &egrave; presente nel database.";
      mysqli_close($connection);
    ?>
    <br>
    <a href = "http://localhost/hogwarts/add.php">
      Inserimento nuovo personaggio.
    </a>
    <br>
    <a href = "http://localhost/hogwarts/del.php">
      Eliminazione personaggio esistente.
    </a>
  </body>
</html>
</html>
```

Id	Personaggio	Casa
LL	Luna Lovegood	Corvonero
MM	Mirtilla Malcontenta	Corvonero
HP	Harry Potter	Grifondoro
HG	Hermione Granger	Grifondoro
RW	Ron Weasley	Grifondoro
DM	Draco Malfoy	SerpEVERDE
SP	Severus Piton	SerpEVERDE
CD	Cedric Diggory	TassOROSSO
TL	Ted Lupin	TassOROSSO

[Inserimento nuovo personaggio.](#)
[Eliminazione personaggio esistente.](#)

ACCESSO AD UNA BASE DI DATI IN LINGUAGGIO PHP: L'INTERFACCIA DEL LINGUAGGIO PHP - VISUALIZZAZIONE

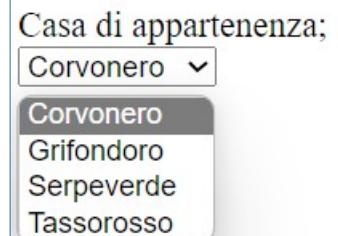
La funzione ***mysqli_connect*** comprende la selezione del database su cui operare; per verificare un eventuale errore di connessione al DBMS è necessario invocare la funzione ***mysqli_connect_errno***.

I due collegamenti ipertestuali inseriti nella pagina di visualizzazione dei personaggi invocano una pagina dinamica generata da uno script PHP contenente un form rispettivamente per l'aggiunta di un nuovo personaggio (add.php) o per l'eliminazione di un personaggio esistente (del.php)

ACCESSO AD UNA BASE DI DATI IN LINGUAGGIO PHP: L'INTERFACCIA DEL LINGUAGGIO PHP - INSERIMENTO

```
<html>
  <head>
    <title>DB & PHP test: INSERT</title>
  </head>
  <body>
    <form action = "insert.php" method = "GET"><br>
      Inserimento nuovo personaggio di Hogwarts<br><br>
      Id:  <input type = "text" name = "id_personaggio" size = '2'
      maxlength = '2'>&ampnbsp&ampnbsp
      Nome:  <input type = "text" name = "personaggio"><br><br>
      Casa di appartenenza;<br>
      <select name = "casa">
        <option value = "CN">Corvonero</option>
        <option value = "GD">Grifondoro</option>
        <option value = "SV">Serpeverde</option>
        <option value = "TR">Tassorosso</option>
      </select><br><br>
      <input type = "submit" value = "Inserisci">
    </form>
  </body>
</html>
```

add.php



Inserimento nuovo personaggio di Hogwarts

Id: Nome:

Casa di appartenenza;

Corvonero ▾

ACCESSO AD UNA BASE DI DATI IN LINGUAGGIO PHP: L'INTERFACCIA DEL LINGUAGGIO PHP - INSERIMENTO

Il codice `add.php` è solo HTML. L'elenco consente di selezionare la casa del personaggio è infatti istanziato staticamente con i nomi delle case di Hogwarts presenti nella tabella `Casehw` del DB `hogwarts`.

Casa di appartenenza;

Corvonero

Corvonero
Grifondoro
Serpeverde
Tassorosso

I dati del form della pagina `add.php` sono inviati utilizzando il metodo **GET** alla pagina `insert.php` costituita da uno script PHP che effettua l'inserimento nel database del nuovo personaggio utilizzando il comando **SQL insert**.

Inserimento nuovo personaggio di Hogwarts

Id: Nome:

Casa di appartenenza;

Corvonero

ACCESSO AD UNA BASE DI DATI IN LINGUAGGIO PHP: L'INTERFACCIA DEL LINGUAGGIO PHP - INSERIMENTO

```
<html>
    <head>
        <title>DB & PHP test: INSERT</title>
    </head>
    <body>
        <?php
            $id_personaggio = strtoupper($_GET["id_personaggio"]);
            $personaggio = $_GET["personaggio"];
            $id_casa = $_GET["casa"];
            $connection = mysqli_connect("localhost", "root", "", "hogwarts");
            $query = "SELECT *
                        FROM Personaggi
                        WHERE id_personaggio = '$id_personaggio'
                            OR nome = '$personaggio'";
            $result = mysqli_query($connection, $query);
            if (mysqli_num_rows($result) != 0)
                echo "Personaggio o id esistente!";
            else {
                $query = " INSERT INTO Personaggi VALUES
                    ('$id_personaggio', '$personaggio', '$id_casa')";
                $result = mysqli_query($connection,$query);
                if ($result > 0)
                    echo "Personaggio $personaggio inserito!";
                else
                    echo "Errore, personaggio $personaggio non inserito!";
            }
            mysqli_close($connection);
        ?>
        <br><br>
        <a href = "http://localhost/hogwarts/hogwarts.php">
        Visualizzazione elenco personaggi.
        </a>
    </body>
</html>
```

insert.php

Personaggio Albus Silente inserito!

[Visualizzazione elenco personaggi.](#)

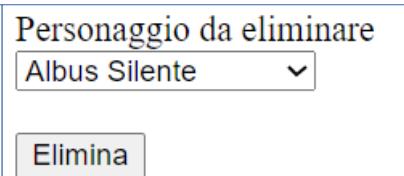
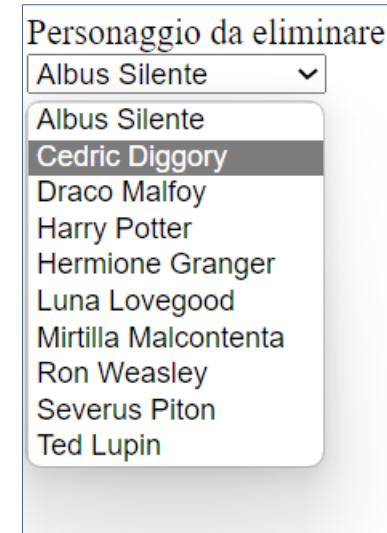
Personaggio o id esistente!

[Visualizzazione elenco personaggi.](#)

ACCESSO AD UNA BASE DI DATI IN LINGUAGGIO PHP: L'INTERFACCIA DEL LINGUAGGIO PHP - ELIMINAZIONE

```
<html>
<head>
    <title>DB & PHP test: DELETE</title>
</head>
<body>
    <?php
        $connection = mysqli_connect("localhost", "root", "", "hogwarts");
        $query = "SELECT id_personaggio, nome
                  FROM Personaggi
                  ORDER BY nome";
        $result = mysqli_query($connection, $query);
        if (mysqli_num_rows($result) != 0) {
    ?>
    <form action = "delete.php" method = "GET"><br>
        Personaggio da eliminare<br>
        <select name = "personaggio">
            <?php
                while ($row = mysqli_fetch_array($result))
                    echo "<option value = \"$row[0]-$row[1]\">$row[1]</option>";
                    // echo "<option value = \"$row[id_personaggio]-$row[nome]\">
                    // $row[nome] </option>";
            ?>
        </select><br><br>
        <input type = "submit" value = "Elimina">
    </form>
    <?php
    }
    else
        echo "Nessun personaggio presente nel database.";
    mysqli_close($connection);
    ?>
</body>
</html>
```

del .php



La riga di codice commentata rappresenta l'accesso alternativo per chiave associativa, anziché per indice, all'array che contiene una riga del risultato della query.

ACCESSO AD UNA BASE DI DATI IN LINGUAGGIO PHP: L'INTERFACCIA DEL LINGUAGGIO PHP - ELIMINAZIONE

Il seguente script PHP effettua l'eliminazione dal database utilizzando il comando SQL DELETE

```
<html>
  <head>
    <title>DB & PHP test: DELETE</title>
  </head>
  <body>
    <?php
      $personaggio = explode('-', $_GET["personaggio"]);
      $connection = mysqli_connect("localhost", "root", "", "hogwarts");
      $query = "DELETE
                  FROM personaggi
                 WHERE id_personaggio = '$personaggio[0]' ";
      $result = mysqli_query($connection, $query);
      echo "Personaggio $personaggio[1] eliminato!";
      mysqli_close($connection);
    ?><br><br>
    <a href = "http://localhost/hogwarts/hogwarts.php">
      Visualizza elenco personaggi.
    </a>
  </body>
</html>
```

delete.php

Personaggio Albus Silente eliminato!

[Visualizza elenco personaggi.](#)

GESTIONE DEGLI UTENTI E DELLE PASSWORD CON DBMS MYSQL E LINGUAGGIO PHP

PROF. SANTANIELLO ASSUNTA

GESTIONE DEGLI UTENTI E DELLE PASSWORD CON DBMS MYSQL E LINGUAGGIO PHP

La gestione degli utenti con le relative credenziali d'accesso è una delle problematiche più comuni nella realizzazione di un sito web. I dati relativi al singolo utente devono essere memorizzati in una specifica tabella di un database ospitato dal DBMS di un supporto del sito web stesso. A questo proposito al db `hogwarts` è stata aggiunta una nuova tabella utenti con il seguente comando sql:

```
USE hogwarts;

CREATE TABLE Utenti (
    username VARCHAR (16) NOT NULL,
    password VARCHAR(256) NOT NULL,
    DB_username VARCHAR (16) NOT NULL,
    DB_password VARCHAR(256) NOT NULL,
    CONSTRAINT chiave_primaria PRIMARY KEY (username)
);
```

Memorizzare le credenziali d'accesso al
sito web

Memorizzare le credenziali d'accesso al
dbms associate al singolo utente

GESTIONE DEGLI UTENTI E DELLE PASSWORD CON DBMS MYSQL E LINGUAGGIO PHP

Le password saranno memorizzate nel database in forma cifrata utilizzando la funzione PHP `crypt`; il loro riconoscimento avverrà cifrando la password digitata dall'utente e confrontando il risultato con quello presente nel db: l'accesso sarà consentito se saranno uguali, altrimenti verrà negato.

NB. al campo password è stata assegnata una dimensione di **256 caratteri** perché l'algoritmo di cifratura implementato dalla funzione `crypt` può generare stringhe più lunghe della stringa fornita come input, cioè della password scelta dall'utente.

PERCHE' USARE LA FUNZIONE PASSWORD_HASH()

Algoritmi di Hashing Moderni	password_hash() utilizza algoritmi di hashing moderni, come bcrypt, Argon2, o scrypt, che sono stati progettati specificamente per la sicurezza delle password. Questi algoritmi sono resistenti agli attacchi di blue force e alle tecniche di decriptazione.
Generazione Automatica del Salt	Quando utilizzi password_hash(), non devi preoccuparti di generare manualmente un salt. La funzione genera automaticamente un salt casuale e lo combina con la password prima di eseguire l'hashing. Questo rende più difficile per gli attaccanti prevedere il salt e rende ogni hash unico.
Costanti di Iterazione Configurabili	password_hash() consente di specificare il numero di iterazioni (o "costo") da utilizzare durante l'hashing. Maggiore è il numero di iterazioni, maggiore è la sicurezza, ma richiede anche più risorse computazionali. Questo ti permette di bilanciare la sicurezza con le prestazioni del tuo sistema.
Compatibilità con password_verify()	password_hash() genera hash che possono essere verificati utilizzando password_verify(). Questa funzione semplifica la verifica delle password senza dover gestire manualmente il salt.

La funzione password_hash() è generalmente considerata più sicura di crypt(). E' la scelta migliore per l'hashing delle password grazie alla sua sicurezza integrata, alla generazione automatica del salt e alla flessibilità nelle iterazioni. Si consiglia di utilizzare password_hash() anziché crypt() per le password nel tuo codice PHP.

DES VS HASH

Descrizione:	DES (Data Encryption Standard) USATO DALLA FUNZIONE CRYPT()	HASH USATO DALLA FUNZIONE PASSWORD_HASH()
Scopo:	DES è un algoritmo di cifratura utilizzato per proteggere i dati rendendoli segreti per tutti tranne che per il destinatario previsto. Usato per cifrare dati.	Hashing è utilizzato per verificare l'integrità dei dati e fornire prove che un messaggio non sia stato modificato. È anche utilizzato per autenticazione (ad esempio, firme digitali).
Funzionamento:	Prende una stringa di testo in chiaro (plaintext) e la trasforma in una stringa cifrata (ciphertext) utilizzando una chiave segreta. La stessa chiave viene utilizzata per la decifratura.	Prende un input (ad esempio, una stringa) e lo trasforma in un valore hash, che è una stringa di lunghezza fissa. L'hash è univoco per ogni input.
Chiave:	Utilizza una chiave simmetrica, il che significa che la stessa chiave viene utilizzata sia per la cifratura che per la decifratura.	Non viene utilizzata una chiave segreta. L'hashing è un processo unidirezionale
Esempi:	DES (obsoleto), 3DES (obsoleto), AES (più moderno).	SHA-256, MD5, SHA-1.

ACCESSO AD UNA BASE DI DATI IN LINGUAGGIO PHP: L'INTERFACCIA DEL LINGUAGGIO PHP - ELIMINAZIONE

```
<html>
    <head>
        <title>Nuovo utente</title>
    </head>
    <body>
        <?php
        if (!isset($_POST['username']) || !isset($_POST['password'])) {
        ?>
            <form method = "POST" action = "nuovo_utente.php">
                Username <input name = "username" type = "text"><br>
                Password &nbsp <input name = "password" type = "password"><br><br>
                <input type = "submit" value = "Registra utente">
            </form>
        <?php
        }else {
            $username = $_POST['username'];
            $password = $_POST['password'];
            if (strlen($username) != 0 && strlen($password) != 0) {
                $password = password_hash($password, PASSWORD_DEFAULT); // cifratura della password
                $connection =mysqli_connect("localhost", "root", "", "hogwarts");
                $query = "SELECT * FROM Utenti WHERE username = '$username'";
                $result = mysqli_query($connection, $query);
                if (mysqli_num_rows($result) != 0)
                    echo "Utente $username già presente nel database.";
                else {
                    $query = "INSERT INTO Utenti (username, password, DB_username, DB_password) VALUES ( '$username', '$password', 'root', '')";
                    mysqli_query($connection, $query);
                    echo "Utente $username aggiunto al database.";
                }
                mysqli_close($connection);
            }
            else
                echo "Username/password non validi.";
        }
        ?>
    </body>
</html>
```

nuovo_utente.php

Username

Password

	username	password	DB_username	DB_password
1	Assunta	*0		root
2	Diego	\$2y\$10\$Smngj4Q9tvKBC3OkdenkV.2cXdSgZ.XJgbWNX92Zz66...	root	
3	Marialuisa	*0		root

LA PERSISTENZA IN PHP

Il protocollo per il web – HTTP – è di tipo **stateless**, si tratta di un protocollo senza memoria **di stato**, in cui ciascuna coppia **request_client/response_server** è indipendente dalle altre. Questo è un problema che si è rivelato sin dall'inizio dell'espansione del web un forte limite. Per questo motivo esistono delle tecniche che simulano lo stato in una tipica connessione client/server.

Le tecniche che consentono di memorizzare le informazioni sul server possono essere suddivise:

- A carattere **temporaneo**: trasferimento temporaneo tra pagina e pagina oppure durante una sola sessione di lavoro;
- A carattere **definitivo**: salvataggio di informazioni per un tempo indefinito;

LA PERSISTENZA IN PHP – TECNICHE TEMPORANEE

A carattere **temporaneo**: trasferimento temporaneo tra pagina e pagina oppure durante una sola sessione di lavoro;

- Uso di variabili passate alla pagina successiva;

<http://www.sito.it/pagina.php?visite=1>

- Uso di campi nascosti(di tipo hidden);

```
<FORM METHOD="post" ACTION=" http://www.sito.it/pagina.php ">
...
<INPUT TYPE="hidden" NAME="visite" VALUE="1">
...
</FORM>
```

- Uso dei cookies: Nel caso di ricorso a questa tecnica, il server, oltre alla pagina richiesta, può includere nell'header della risposta la richiesta al client di ricevere un cookie. Il cookie è una variabile che ha un nome e un valore. Se abilitato a ricevere cookies, il client salva il nome e il valore della variabile sul proprio disco in un file che ha lo stesso nome del server che ha inviato il cookie. Il cookie scade dopo un certo tempo specificato dal server. I cookies, tuttavia, presentano una debolezza enorme: le informazioni che si vogliono conservare tra una pagina e l'altra vengono salvate in un file sul disco del client, e sono pertanto modificabili dall'utente;

- Uso di sessioni;

LA PERSISTENZA IN PHP – TECNICHE DEFINITIVE

A carattere **definitivo**: salvataggio di informazioni per un tempo indefinito;

- Uso dei file system: In questo caso è il server a memorizzare man mano le informazioni all'interno di file memorizzati nel proprio sistema, come per esempio sull'hard disk. Questo metodo viene utilizzato solo per memorizzare informazioni che non sono oggetto di continue interrogazioni o elaborazioni, situazioni per le quali risulta preferibile l'utilizzo di un data-base;
- Uso dei database: I database rappresentano la tecnica privilegiata per realizzare l'architettura Three-Tier, tipica della conversazione client/server con linguaggi server side. In questo caso, la quantità di dati da gestire deve giustificare l'uso: non avrebbe infatti senso utilizzare un database per memorizzare solo informazioni temporanee tra pagine, mentre è fondamentale nel caso di informazioni massive e permanenti.;

LA PERSISTENZA IN PHP – COOKIES

I cookies sono dei file di testo di piccole dimensioni che vengono salvate sul computer dell'utente e servono per salvare le preferenze su un sito.

I settaggi impostati rimarranno tali, finché non decidiate di modificarli o svuotare la cache e cancellare i cookies del proprio browser. In questo caso, sarà necessario reimpostare i settaggi scelti. Questo accade perché questa preferenza è salvata nei cookies e non su youtube perché non vuole occupare spazio sui propri server con le tue preferenze (soprattutto perché non riguarda dati sensibili).

```
//salvare cookie  
setcookie('username', 'luca rossi', time() + 60 * 60 * 24 * 30)
```

LA PERSISTENZA IN PHP – COOKIES

```
// salvare cookie  
<?php  
    setcookie("username", "luca rossi", time() + 60 * 60 * 24 * 30)  
?>
```

Parametro	Descrizione
name	Il nome del biscotto.
value	Il valore del cookie. Non archiviare informazioni sensibili poiché questo valore è archiviato sul computer dell'utente.
expires	La data di scadenza nel formato timestamp UNIX. Trascorso questo tempo i cookie diventeranno inaccessibili. Il valore predefinito è 0.
path	Specificare il percorso sul server per il quale il cookie sarà disponibile. Se impostato su /, il cookie sarà disponibile in tutto il dominio.
domain	Specificare il dominio per il quale il cookie è disponibile, ad esempio www.example.com.
secure	Questo campo, se presente, indica che il cookie deve essere inviato solo se esiste una connessione HTTPS sicura.

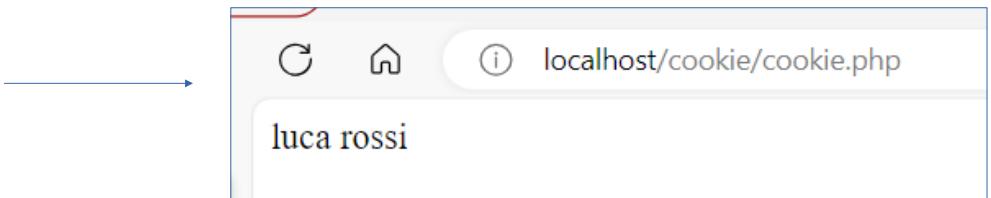
The screenshot shows the Chrome DevTools interface with the Application tab selected. A red circle highlights the 'Cookies' section under the Storage heading. A single cookie entry is listed:

Name	Value	Domain	Path	Expires ...	Size	Http
username	luca%20rossi	localhost	/cookie	2024-0...	20	

Below the table, a message reads: "Select a cookie to preview its value".

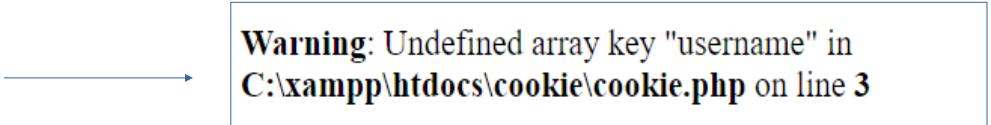
LA PERSISTENZA IN PHP – COOKIES

```
//accedere ai dati salvati $_COOKIE["name"] check isset cookie
<?php
    //setcookie("username", "luca rossi", time() + 60 * 60 * 24 * 30)
    echo $_COOKIE["username"];
?>
```

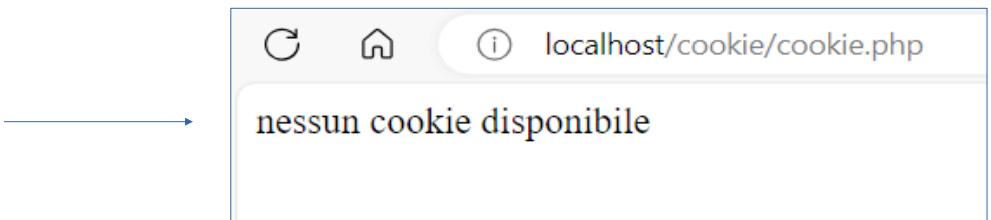


Se cancello i cookie visualizzerò questo warning,
è buona norma quindi inserire la funzione isset:

```
//accedere ai dati salvati $_COOKIE["name"] check isset cookie
<?php
//setcookie("username", "luca rossi", time() + 60 * 60 * 24 * 30)
if(isset($_COOKIE["username"])){
    echo $_COOKIE["username"];
} else{
    echo "nessun cookie disponibile";
}
?>
```



```
//per rimuovere un cookie
<?php
    setcookie("username","",time()-3600);
?>
```



LA PERSISTENZA IN PHP - SESSIONI

Le sessioni sono simili ai cookie, ma differiscono per alcune caratteristiche: le sessioni sono dei dati temporanei salvati sul server ed sono in grado di riconoscerci. In realtà possiamo dire che sono complementari.

Esempio: immaginiamo di avere un login con la possibilità di avere la spunta tienimi collegato. Nel momento in cui facciamo l'accesso noi stiamo aprendo una sessione. La sessione permette di restare collegato sul sito senza dover fare l'accesso ogni volta che cambio pagina finché non “distruggiamo” la sessione.

La possibilità di effettuare la spunta “tienimi collegato” è gestito dai cookie, se io ho questo cookie automaticamente quando mando la richiesta di mantenere collegata la sessione non viene distrutta, ma viene mantenuta e quindi quanto torniamo in un secondo momento rimarremo connessi.

La sessione è ciò che succede nel sito utente mentre il cookie è la nostra preferenza.

LA PERSISTENZA IN PHP - SESSIONI

La gestione dell'accesso degli utenti al sito viene regolata inizializzando una nuova sessione al momento del riconoscimento della validità della password e di conseguenza dell'identità dell'utente che l'ha fornita effettuando il login: nessuna pagina del sito deve risultare accessibile se non è stato effettuato un login valido.

Le pagine del sito prevedono per l'utente la possibilità di chiudere la sessione con un logout oppure trascorso un certo tempo dal momento del login la sessione sarà automaticamente invalidata.

Nel file di configurazione php.ini (che si trova in C:\xampp\php) tra le varie impostazioni troviamo anche la session maxlifetime che è la durata massima in secondi della sessione, questa è possibile gestirla.

```
; After this number of seconds, stored data will be  
; cleaned up by the garbage collection process.  
; https://php.net/session.gc-maxlifetime  
session.gc_maxlifetime=1440
```

LA PERSISTENZA IN PHP – SESSIONI

```
<?php  
session_start( );  
?>  
<HTML>  
***
```

Una sessione http è un insieme di request e response tra client e server che condividono uno stato, rappresentato da informazioni che persistono tra una connessione e le successive

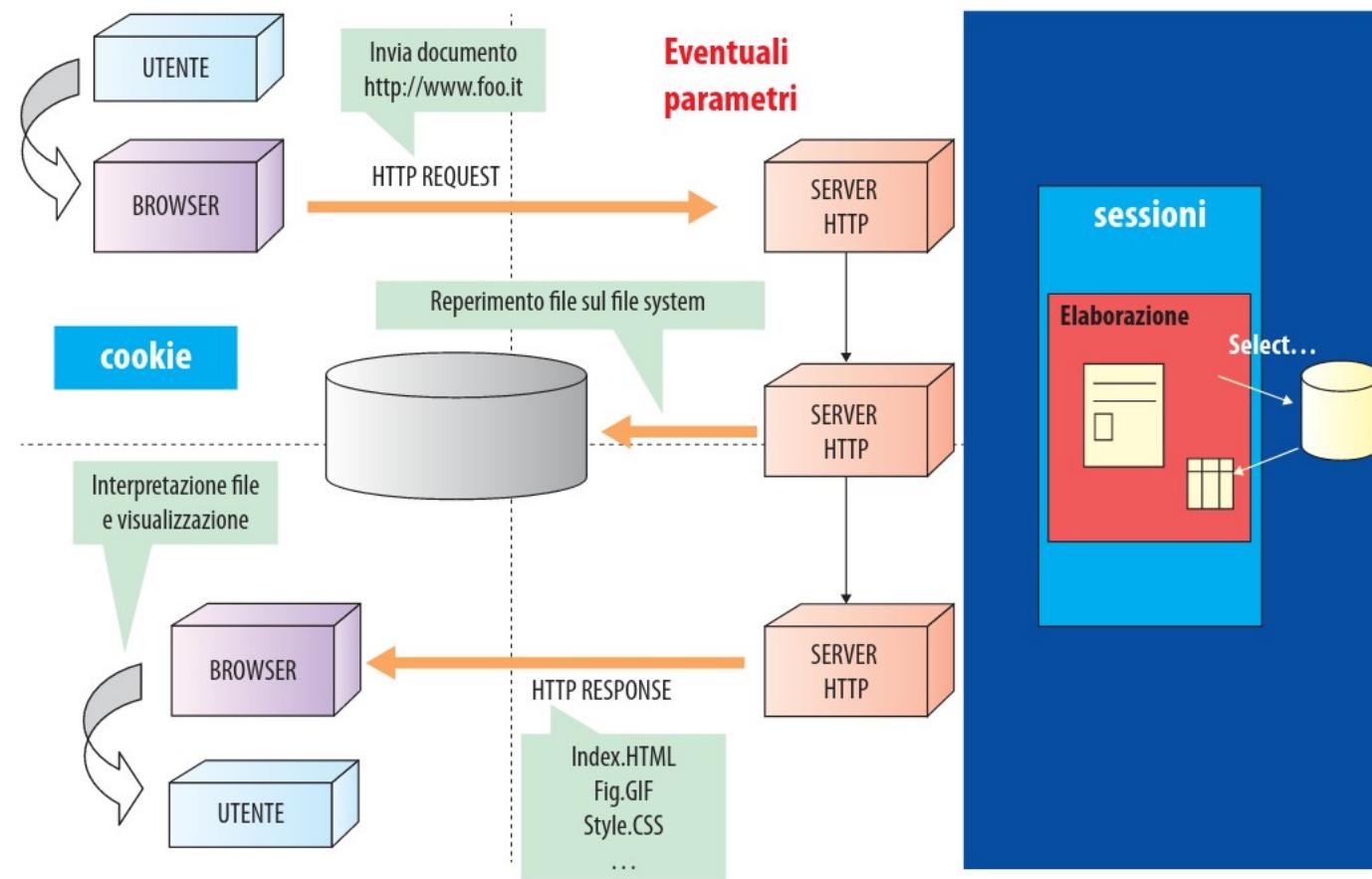
Le sessioni http vengono attivate, in php, dalla funzione `session_start()`, che deve essere scritta intesta alla pagina, prima di qualunque altra riga di codice HTML:

La funzione `session_start()` memorizza in un cookie (`PHPSESSID`) sul computer del client un identificativo di sessione (Session ID) casuale, l'unica informazione memorizzata sul client.

La tecnica delle sessioni ricorre contemporaneamente all'uso di tre strategie:

- viene utilizzato un valore di codice identificativo (PID) molto difficile da indovinare (per esempio: “H6K7DHIE737462BSTG37”);
- i dati sensibili vengono memorizzati sul disco del server, associandoli al codice identificativo dell’utente SID (Session Identifier);
- il cookie che contiene il codice identificativo viene impostato a durata temporale 0, inoltre viene eliminato chiudendo il browser

LA PERSISTENZA IN PHP - SESSIONI



LA PERSISTENZA IN PHP – SESSIONI

```
<?php
    //accedere ad una sessione
    session_Start();

    //salvare e accedere ai dati della sessione
    $_SESSION["user_id"] =23;
    echo $_SESSION["user_id"];

    //rimuovere dati di una sessione
    //unset($_SESSION["user_id"]);

    //distruggere una sessione
    session_destroy();
?>
```

LA PERSISTENZA IN PHP - SESSIONI (ALTRI ESEMPI)

CONTA VISITE DELLO STESSO UTENTE

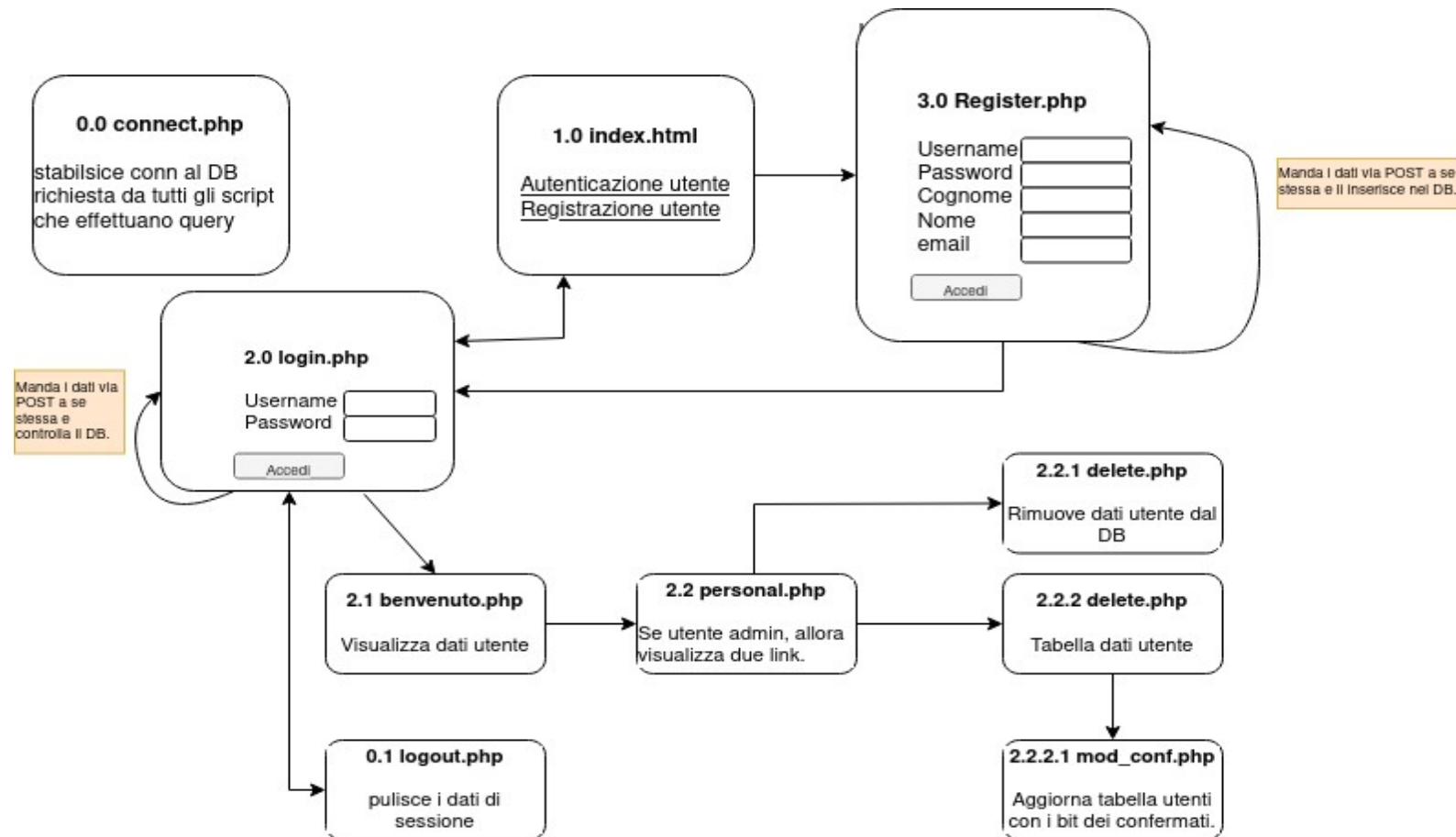
Lo script indicato di seguito esegue il calcolo delle visite effettuate alla pagina dallo stesso utente, utilizzando le sessioni per memorizzare il valore del conteggio. Se proviamo a chiudere e a riaprire il browser, il contatore di visite verrà azzerato, in quanto alla chiusura del browser vengono eliminate tutte le sessioni aperte. Lo script inizia con la funzione `session_start()`, che crea nel cookie una variabile chiamata `PHPSESSID` per attivare le sessioni. Ovviamente, affinché lo script funzioni è necessario avere i cookies attivati sul browser. Come possiamo notare, premendo più volte il tasto F5 otteniamo: Puoi trovare il codice di questo esempio nel file `Contavisite_utente.php`. Vediamo adesso un altro esempio, che mostra come usare le sessioni per effettuare l'autenticazione di un utente.

ESEMPIO AUTENTICAZIONE "STUPIDA" DI UN UTENTE In questo esempio viene effettuata una autenticazione definibile come "stupida", in quanto il nome dell'utente autorizzato è presente in una variabile all'interno del codice stesso. Un sistema più valido prevede invece che la lettura avvenga da un file, oppure da un database che contiene una tabella con i dati degli utenti registrati. Si compone di due pagine, la prima delle quali consente di effettuare il `login(Accedi_dummy.php)`, richiede i dati all'utente da autenticare, quindi, tramite la tecnica postback, richiama la pagina stessa passando come parametri i campi nome utente e password. Se i campi vengono riconosciuti, viene richiamata la pagina `Accedi_dummy_aut.php`. La prima parte del codice della pagina `Accedi_dummy.php` effettua il controllo sul settaggio della variabile `$_POST['utente']`, in grado di verificare se si tratta del primo accesso o del successivo. Nel caso di primo accesso viene mostrato il Form di login.

Ovviamente, affinché lo script funzioni è necessario avere i cookies attivati sul browser

```
<?php
//attivazione delle sessioni
//obbligatorio come prima riga in PHP ogni volta che si usano
le sessioni
session_start();
//se la variabile di sessione non esiste
if(!isset($_SESSION['numero_visite']))
{
    //settaggio variabile sessione a 1
    $_SESSION['numero_visite']=1;
}
else
{
    //se esiste già incremento variabile sessione
    $_SESSION['numero_visite']++;
}
?>
<HTML>
    <BODY>
        il numero di volte che ti sei connesso a questa
pagina e':
        <B>
            //visualizzazione variabile di sessione
            <?php echo $_SESSION['numero_visite'];?>
        </B>
    </BODY>
</HTML>
```

LA PERSISTENZA IN PHP – SESSIONI (ESERCIZIO LOGIN)



LA PERSISTENZA IN PHP – SESSIONI (ESERCIZIO LOGIN)

Dopo aver progettato il mio sito web, creo il DB che chiamerò utenti ed avrà due tabelle:

utenti gruppo	
id_gruppo	int(11)
descrizione	varchar(20)

utenti utente	
username	varchar(20)
password	varchar(32)
cognome	varchar(80)
nome	varchar(80)
email	varchar(255)
id_gruppo	int(11)
confermato	tinyint(1)
codice_conferma	varchar(32)

Farò poi qualche inserimento:

		username	password	cognome	nome	email	id_gruppo	confermato	codice_conferma
<input type="checkbox"/>	Modifica Copia Elimina	Assunta	1234	Santaniello	Assunta	a.santaniello@itismonza.it	2	0	NULL
<input type="checkbox"/>	Modifica Copia Elimina	doc0	DoC0	Bassani	Claudia	c.bassani@hensemberger.it	1	0	dOc0
<input type="checkbox"/>	Modifica Copia Elimina	doc1	DoC1	Ferrari	Marco	m.ferrari@hensemberger.it	1	0	dOc1

		id_gruppo	descrizione
<input type="checkbox"/>	Modifica Copia Elimina	1	docente
<input type="checkbox"/>	Modifica Copia Elimina	2	amministratore
<input type="checkbox"/>	Modifica Copia Elimina	3	studente

LA PERSISTENZA IN PHP – SESSIONI (ESERCIZIO LOGIN)

Realizzo delle pagine che utilizzerò spesso all'interno di altre
come l'apertura di un file log:

Log.php

```
<?php
    $f = fopen("log.txt", "a");
?>
```

<https://guidaphp.it/base/gestione-file>

Apre il file in scrittura posizionando
il puntatore alla fine del file. Se il
file non esiste lo crea.

2. La connessione al db:

connect.php

```
<?php
require "log.php";
$connection = mysqli_connect("localhost", "root", "", "utenti");
fwrite($f, "Connessione al database\n");

if (mysqli_connect_errno()) {
    echo "Connessione al DB fallita: " . mysqli_connect_error();
    fwrite($f, "Connessione al DB fallita: ". mysqli_connect_error() . "\n");
    exit();
}

?>
```

LA PERSISTENZA IN PHP – SESSIONI (ESERCIZIO LOGIN)

Il logout:

```
<?php
    require 'log.php';

    if(isset($_POST['errore'])){
        echo "$errore <br>";
        fwrite($f, "Errore nel logout\n");
    }

    // distruzione sessione corrente

    fwrite($f, "LOGOUT UTENTE \n\n");
    session_start();
    session_unset();
    session_destroy();
    $_SESSION = array();

?>
<html>
<head>
    <title>Logout</title>
</head>
<body>
Sessione conclusa o scaduta:
    <a href="login.php">Login</a>
</body>
</html>
```

Logout.php

LA PERSISTENZA IN PHP – SESSIONI (ESERCIZIO LOGIN)

Index.html

Creo la pagina index che con un collegamento ipertestuale sarà possibile autenticarsi o registrarsi:

```
<html>
  <head><title>Homepage SimpleAuth</title></head>
<body>
<a href="login.php">Autenticazione Utente</a><br>
<a href="register.php">Registrazione Utente</a><br>
</body>
</html>
```

LA PERSISTENZA IN PHP – SESSIONI (ESERCIZIO LOGIN)

```
<html>
  <head>
    <title>Login</title>
  </head>
  <body>
<?php
    if (!isset($_POST['username']) || !isset($_POST['password'])) {
?>
  <body>
    <form method="POST" action="register.php">
      Username <input name="username" type="text"><br>
      Password <input name="password" type="password"><br><br>
      Cognome <input name="cognome" type="text"><br>
      Nome <input name="nome" type="text"><br>
      email <input name="email" type="text"><br>
      <input type="submit" value="Accedi">
    </form>
  </body>
</html>
<?php
    }
    else {
        require 'connect.php'; //connetto al DB e stabilisce la variabile $connection
        require 'log.php';

        $username = $_POST['username'];
        $password = $_POST['password'];
        $nome = $_POST['nome'];
        $cognome = $_POST['cognome'];
        $email = $_POST['email'];
        if (strlen($username) != 0 && strlen($password) != 0){
            $query ="INSERT INTO `utente`(`username`, `password`, `cognome`, `nome`, `email`, `id_gruppo`, `confermato`, `codice_conferma`)
VALUES
('{$username}', '{$password}', '{$cognome}', '{$nome}', '{$email}', 3, 1, 'NULL')";
```

register.php

```
if (!mysqli_query($connection,$query)) {
    echo "Registrazione utente $username fallita: ";
    echo " <a href=\"index.html\"> riprova.</a><br>";
    fwrite($f, "Registrazione utente ".$username ." fallita\n");
}
else {
    echo "Registrazione riuscita, effettua il ";
    echo " <a href=\"login.php\"> login.</a>";

    fwrite($f, "Registrazione utente ".$username ." riuscita\n");
}
mysqli_close($connection); //chiudere la connessione al server dbms
}
else {
    echo "Username/password non validi: ";
    echo " <a href=\"index.html\"> riprova.</a>";
}
?>
</body>
</html>
<?php
}
?>
```

Username

Password

Cognome

Nome

email

Accedi

Username/password non validi: [riprova](#).

Registrazione riuscita, effettua il [login](#).

Utente Assunta collegato
Connessione al database
Errore login
Connessione al database

Utente Assunta collegato
Connessione al database
Connessione al database
Registrazione utente DocenteProva riuscita

LA PERSISTENZA IN PHP – SESSIONI

```
<html>
  <head>
    <title>Login</title>
  </head>
  <body>
<?php
require 'log.php';
    if (!isset($_POST['username']) || !isset($_POST['password'])) {
?>
  <body>
    <form method="POST" action="login.php">
      Username <input name="username" type="text"><br>
      Password
      <input name="password" type="password"><br><br>
      <input type="submit" value="Accedi">
    </form>
  </body>
</html>
<?php
}
else {
  require 'connect.php'; //connetto al DB e stabilisce la variabile $connection

  $username = $_POST['username'];
  $password = $_POST['password'];
  if (strlen($username) != 0 && strlen($password) != 0){
    $query ="SELECT * FROM utente WHERE username = '$username'";
    $result = mysqli_query($connection,$query); // $connection stabilita in
connect.php
    if (mysqli_num_rows($result) == 0) {
      echo "Utente $username sconosciuto: ";
      echo " <a href=\"login.php\"> Riprova</a><br>";
    }
    else {
      $user_row = mysqli_fetch_array($result);
    }
  }
}
```

login.php

The screenshot shows a simple login interface. It has two text input fields: one for 'Username' and one for 'Password'. Below these is a grey rectangular button with the text 'Accedi' in white.

```
if ($password == $user_row['password']) {
  echo "Password corretta: ";
  echo " <a href=\"benvenuto.php\"> Accedi</a><br>";
  // distruzione eventuale sessione precedente
  session_start();
  session_unset();
  session_destroy();
  // inizializzazione nuova sessione
  session_start();
  $_SESSION['username'] = $username;
  $_SESSION['start_time'] = time();
  $_SESSION['DB_username'] = $user_row['username'];
  $_SESSION['DB_password'] = $user_row['password'];
  $_SESSION['gruppo'] = $user_row['id_gruppo'];

  echo " <a href=\"logout.php\"> [$username logout]</a>";

  fwrite($f, "\n\nUtente ".$username." collegato\n");
}
else {
  echo "Password errata: ";
  fwrite($f, "Errore login\n");
  echo " <a href=\"login.php\"> Riprova</a>";
}

mysqli_close($connection); // chiudere la connessione al server dbms
else {
  echo "Username/password non validi: ";
  echo " <a href=\"index.html\"> riprova.</a>";

  fwrite($f, "Errore login\n");
} ?>
</body>
</html>
<?php
?>
```

LA PERSISTENZA IN PHP – SESSIONI

```
<?php //connessione al DB e controllo
    require 'connect.php';
    require 'log.php';
?>
<html>
<head><title>Landing Utente Registrato</title></head>
<body>
<?php
    session_start();
    if(isset($_SESSION['username'])) {
        $username=$_SESSION['username'];
        echo "Ciao ".$username."!<br>";
        echo "Ti sei collegato ".date("Y-m-d H:i:s",
$_SESSION['start_time'])."<br>";
        echo "e la tua password è ".
$_SESSION['DB_password']."<br>

        $query ="SELECT id_gruppo,
gruppo.descrizione FROM utente natural join gruppo
        WHERE username = '$username'";
        $result = mysqli_query($connection,
$query); // $connection stabilita connect.php
//        print_r($result);
        if (mysqli_num_rows($result) > 0) {
            $riga = mysqli_fetch_array($result);
//            print_r($riga);
            $gruppo = $riga['id_gruppo'];
            $descrizione = $riga['descrizione'];
            $_SESSION['gruppo']=$gruppo;
            $_SESSION['descrizione']=$descrizione;
            echo "<br> Sei un $descrizione, accedi
alla tua area ";
            echo " <a href=\"personal.php\">
privata</a><br>";
            echo " <a href=\"logout.php\">
[$username logout]</a>";
        }
    }
?>
```

benvenuto.php

Ciao Assunta!
Ti sei collegato 2024-02-08 10:17:41
e la tua password è 1234

Sei un amministratore, accedi alla tua area [privata](#)
[\[Assunta logout\]](#)

```
        else {
            $errormsg="Gruppo dell'utente non trovato.";
            $errormsg=htmlentities($errormsg, ENT_HTML5, 'ISO-8859-1');
            $_POST['errore']=$errormsg;
            header("Location: logout.php");
        }
    }
    else{
        $errormsg="Utente non impostato.";
        $errormsg=htmlentities($errormsg, ENT_HTML5, 'ISO-8859-1');
        $_POST['errore']=$errormsg;
        header("Location: logout.php");
    }
?>

</form>
</body>
</html>
```

Sessione conclusa o scaduta: [Login](#)

LA PERSISTENZA IN PHP – SESSIONI

```
<?php //connessione al DB e controllo
    require 'connect.php';
?>
<html><head><title>Homepage Utente Registrato</title></head>
<body>
<?php
    session_start();
    if(isset($_SESSION['username']) && isset($_SESSION['gruppo'])) {
        $username=$_SESSION['username'];
        $gruppo=$_SESSION['gruppo'];
        $descrizione=$_SESSION['descrizione'];
        echo "Ciao ".$username."!  
";
        echo "Ti sei collegato ".date("Y-m-d H:i:s", $_SESSION['start_time'])."  
";
        echo "e la tua password è ".$_SESSION['DB_password']."  
";

        if($gruppo==1 || $gruppo==0){
            echo "<br> Visto che sei un $descrizione qui troverai tutte le tue informazioni.";

        }
        if($gruppo==2){
            echo "<br> Sei un $descrizione, puoi effettuare le seguenti operazioni:<br>";
            echo " <a href=\"delete.php\">Rimuovi utente</a><br>";
            echo " <a href=\"convalida.php\">Convalida utente</a><br>";
        }
    }
    else{
        $errormsg="Utente o gruppo non impostato.";
        $errormsg=htmlentities($errormsg, ENT_HTML5, 'ISO-8859-1');
        $_POST['errore']=$errormsg;
        header("Location: logout.php");
    }
?>
</form>
</body>
</html>
```

personal.php

Ciao Marialuisa!
Ti sei collegato 2024-02-08 11:47:27
e la tua password è Santillo

Sei un amministratore, puoi effettuare le seguenti operazioni:
[Rimuovi utente](#)
[Convalida utente](#)

LA PERSISTENZA IN PHP – SESSIONI

```
<?php
session_start();
$gruppo = $_SESSION["gruppo"];
if($gruppo != 2)
{
    echo("<p>Non hai i permessi per eliminare gli utenti.</p><a href='benvenuto.php'>
        Torna alla home</a>");
    fwrite($f, "Tentato accesso non autorizzato a convalida.php\n");
}
else
{

require "connect.php";
$query = "SELECT * FROM utente";
$result = mysqli_query($connection, $query);

$txt = "<table><tr><th>Username</th><th>Nome</th><th>Cognome</th>
<th>Email</th><th>Ruolo</th><th>Conferma</th></tr>";

while($row = mysqli_fetch_array($result))
{
    $user = $row["username"];
    $txt = $txt . "<tr><th>". $user . "</th><th>". $row["nome"] . "</th>
    <th>". $row["cognome"] . "</th><th>". $row["email"] . "</th><th>";

    $ruolo = $row["id_gruppo"];
    if($ruolo == 1) {$ruolo = "Insegnante";}
    else if($ruolo == 2) {$ruolo = "Amministratore";}
    else if($ruolo == 3) {$ruolo = "Studente";}

    $txt = $txt . $ruolo . "</th><th>";
```

convalida.php

Username	Nome	Cognome	Email	Ruolo	Conferma
Assunta	Assunta	Santaniello	a.santaniello@itismonza.it	0	Convalidato
doc0	Claudia	Bassani	c.bassani@hensemberger.it	Insegnante	Convalidato
doc1	Marco	Ferrari	m.ferrari@hensemberger.it	Insegnante	Convalidato
doc2	Giorgio	Castoldi	g.castoldi@hensemberger.it	Insegnante	Convalidato
DocenteProva	Prova	Docente	D.prova@gmail.com	Studente	Convalida utente
gadd	Tony	Gaddis	tony@gaddis.com	Amministratore	Convalidato
Marialuisa	Santillo	Marialuisa	m.santillo@itismonza.it	Amministratore	Convalidato
prova0	Provino	Provetti	p@prova.com	Amministratore	Convalidato
root				Studente	Convalida utente
stud0	Emanuele	Villa	e.villa@hensemberger.it	0	Convalidato
stud1	Valter	Pirovano	v.pirovano@hensemberger.it	0	Convalidato

[Torna alla home page](#)

```
if($row["confermato"] == 1) {
    $conf = "Non confermato";
    $txt = $txt . "<a href='mod_conf.php?user=". $user ."'">
        Convalida utente</a>";
}
else $txt = $txt . "Convalidato";
$txt = $txt."</th></tr>";
}
$txt = $txt . "</table>";
echo($txt);
}

?>
<head>
    <style>
        table, th, td {
            border: 1px solid black;
        }
    </style>

</head>
<body>
    <br><br>
    <a href="benvenuto.php">Torna alla home page </a>
</body>
```

LA PERSISTENZA IN PHP – SESSIONI

```
<?php
require 'connect.php';
require 'log.php';
if(isset($_POST["cancUser"]))
{
    $query= 'DELETE FROM utente WHERE username = "' . $_POST["cancUser"] . 
'";
    $res = mysqli_query($connection, $query);
    if($res == false) {
        echo("C'è stato un errore. Verifica se hai scritto alcuni parametri
errati oppure ritenta");
        fwrite($f, "Errore cancellazione utente\n");
    }
    echo("La cancellazione dell'utente " . $_POST["cancUser"] . " è avvenuta
con successo, se esistente nel database.<br><br>");
    fwrite($f, "Utente ".$_POST["cancUser"]." cancellato\n");
    $_POST["cancUser"] = "";
}
?>
<html>
<head><title>Cancellazione utente</title></head>
<body>
<?php
    session_start();
    $gruppo = $_SESSION["gruppo"]; //per sicurezza controlliamo il tipo di
utente
    if($gruppo == 2) //amministratore
{
?>
```

delete.php

Inserisci lo username dell'utente che vuoi eliminare:

Username

[Torna all'area privata](#)

```
<body>
    Inserisci lo username dell'utente che vuoi eliminare:
    <form method="POST" action="delete.php">
        Username <input name="cancUser" type="text"><br>
        <input type="submit" value="Elimina">
    </form>
    <a href="benvenuto.php">Torna all'area privata</a>
</body>
<?php
}
else
{
    echo("<p>Non hai i permessi per eliminare gli utenti.</p><a
href='benvenuto.php'>Torna alla home</a>");
    fwrite($f, "Tentato accesso non autorizzato a delete.php\n");
}
?>
```

La cancellazione dell'utente Assunta è avvenuta con successo, se esistente nel database.

Inserisci lo username dell'utente che vuoi eliminare:

Username

[Torna all'area privata](#)

LA PERSISTENZA IN PHP – SESSIONI

mod_conf.php

```
<?php
require "log.php";
session_start();
$gruppo = $_SESSION["gruppo"];
if($gruppo != 2)
{
    echo("<p>Non hai i permessi per eliminare gli utenti.</p><a href='benvenuto.php'>Torna alla home</a>");
    fwrite($f, "Accesso non autorizzato a mod_conf.php\n");
}
else
{

require "connect.php";
$user = $_GET["user"];

$query = "UPDATE utente SET confermato = 0 WHERE username = '" . $user . "';";
mysqli_query($connection, $query);

fwrite($f, "Utente ".$user." ha ora i permessi\n");

header("location: convalida.php");
}
?>
```