

# Laboratorio 6

☀️ Appunti	Done
☀️ Comprensione	Tutto
☰ Giorni	07/12/23

## 1) Scaricare file

- Scaricare la cartella del moodle in “Laboratori di PHP” e mettere il suo contenuto all'interno della cartella del nostro laboratorio. (Nel mio caso la cartella si chiama Laboratorio\_6)
- Possiamo già adesso sostituire il nome utente all'interno del database “tecweb2023.sql” (si trova sul moodle) nella prima riga, dopo `USE`, mettendo quello del nostro account di laboratorio (altrimenti dovremo comunque farlo dopo). Posso anche farlo con il comando `NANO tecweb2023.sql`
- Stringa per accedere tramite ssh da casa (dal laboratorio non serve)
- `ssh stbressa@sshpaolotti.studenti.math.unipd.it -L8080:tecweb:80 -L8022:tecweb:22`

## 2) Caricare la cartella locale su ssh Paolotti

- Vogliamo caricare la cartella locale “Laboratorio\_6” su ssh Paolotti
- Vado nella cartella padre della cartella che voglio caricare, nel mio caso:
  - `C:\Users\... \Laboratori`
- Eseguo questo comando per caricare la cartella sul server
- `scp -r Laboratorio_6 stbressa@sshpaolotti.studenti.math.unipd.it:~`
  - `scp` è il comando secure copy
  - Serve `-r` perché stiamo caricando una cartella, altrimenti possiamo caricare solo file
  - Dopo inseriamo l'indirizzo di destinazione (composto da nomeUtenteLab@macchina:percorso) che nel mio caso è `stbressa@sshpaolotti.studenti.math.unipd.it:~`
    - “~” indica che carichiamo nella home
  - Verrà chiesta la password del nostro account di laboratorio
- `ssh stbressa@sshpaolotti.studenti.math.unipd.it`
  - ssh (serve per accedere al server del Paolotti)
  - La prima volta che si accede chiederà la password
- `ls`
  - Con `ls` controlliamo che la cartella Laboratorio\_6 che abbiamo caricato è effettivamente presente sul server
- Sul moodle c'è tutorial su come fare il tunnel da casa dai vari S.O.

## 3) Passaggio cartella da ssh Paolotti a Tecnologie Web

- `scp -r Laboratorio_6/ stbressa@tecweb.studenti.math.unipd.it:~/public_html`
  - Perché il nostro sito sia visibile da browser dobbiamo copiarlo all'interno di `public_html`, perché il modulo Apache può accedere solo a quella cartella. Tutto quello che è al di fuori da essa non sarà accessibile via web

- Verifica:
  - Da browser scriviamo accediamo al link:
  - `tecweb.studenti.math.unipd.it/stbressa/Laboratorio_6`
  - e dovremmo vedere il nostro sito
- `ssh stbressa@tecweb.studenti.math.unipd.it`
  - ssh (serve per accedere al server di Tecnologie Web):
  - Chiede la password

#### 4) Database

- `ls`
  - Con ls vediamo la cartella "public\_html" e il file "pwd\_db\_2023-24.txt"
  - Il database ha una password diversa da quella degli account, otteniamola facendo un cat del file e copiamocela:
  - `cat pwd_db_2023-24.txt` (oppure `more pwd_db_2023-24.txt`)
- Entriamo in public\_html e nella cartella che abbiamo caricato (ora vedremo come accedere al db e come eseguire un file)
  - `cd public_html/Laboratorio_6/`
- Per aprire il database facciamo `mysql -u stbressa -p`
  - inserire password del database
  - siamo all'interno del database (dovremmo avere "MariaDB[(none)]>" a sinistra
- Per eseguire il database dalla console di mysql usiamo `source tecweb2023.sql`

#### 5) PHP

- Ora che abbiamo installato il sito sul server di Tecnologie Web, vediamo come scrivere codice php che permette al nostro sito di interfacciarsi con il Database.
- Ora lavoriamo in locale, poi il file lo caricheremo sul server.
- Creiamo il file file "DBAccess.php"

```
<?php
namespace DB;
class DBAccess {
    private const HOST_DB = "localhost";
    private const DATABASE_NAME = "utenteLabMio";
    private const USERNAME = "utenteLabMio";
    private const PASSWORD = "passwordDelMioDB";

    private $connection;

    public function openDBConnection(){
        $this -> connection = mysqli_connect(
            self::HOST_DB,
            self::USERNAME,
            self::PASSWORD,
            self::DATABASE_NAME,
        );
        return mysqli_connect_errno() == 0;
    }

    public function getListaAlbum(){
        $query = "SELECT ID, Titolo, Copertina, idCss FROM Album ORDER BY DataPubblicazione DESC";
        $queryResult = mysqli_query($this -> connection, $query) or die("Errore in DBAccess" . mysqli_error($this -> connection));
```

```

        if (mysqli_num_rows($queryResult) != 0){
            $result = array();
            while($row = mysqli_fetch_assoc($queryResult)){
                $result[] = $row; //append
            }
            $queryResult -> free(); //liberiamo memoria, potrebbe essere pesante
            return $result;
        }else{
            return null;
        }
    }
}
}
?>

```

- Creiamo una classe DBAccess che gestisce la parte di codice relativo al database.
  - Per il progetto non è richiesto che il codice sia ad oggetti, adesso lo facciamo così per la semplicità di avere tutto in un unico posto.
- Creiamo 4 variabili che useremo spesso (hostDB, DBName, User, Pass), senza doverle scrivere tante volte
- DATABASE\_NAME e USERNAME sono le stesse solo in questo caso, in altri casi potrebbero essere diverse.
- `private $connection` è una variabile che rappresenta la connessione
  - Le variabili vanno col dollaro \$, invece nelle variabili costanti non si mette
- La prima cosa che dobbiamo fare è aprire una connessione col nostro DB.
  - Creiamo una funzione `openDBConnection()` che usa la libreria mysqli (N.B.: la libreria mysql (senza 'i') è deprecata)
  - Usiamo la funzione `mysqli_connect` che chiede come parametri le nostre 4 variabili costanti
    - il `"self::"` indica che si riferisce alle variabili all'interno della classe, è simile al `this`
    - Assegniamo il suo valore alla variabile connection
  - Dato che effettuando la connessione ad un DB potrebbero esserci problemi, usiamo `mysqli_connect_errno() = 0;`
    - Questa funzione ritorna zero se tutto è andato bene
- Ci serve anche la funzione per la chiusura della connessione: `closeConnection()`
  - È importante chiudere la connessione perché se non lo facciamo la variabile rimane lì, "zombie", occupandoci delle risorse e potendoci creare problemi in futuro senza che ce ne accorgiamo
- Funzione `getListaAlbum()` che fa una query al DB e ritorna una lista di album. Questa lista è una cosa che cambia nel tempo, perciò è sensato che poi la pagina venga generata dinamicamente andando a leggere il DB.
  - Andiamo a definire la query in SQL che prende Titolo, Copertina e un ID (che serve al codice CSS) dalla tabella album, ordinando i risultati in ordine decrescente per la data di pubblicazione dell'album.
  - Ci salviamo il risultato della query nella variabile `queryResult`
  - Le query si fanno con la funzione `mysqli_query` a cui diamo la connessione e la query.
  - Nella query `or die("Errore in DBAccess" .mysqli_error($this -> connection))`
    - Serve per farci stampare l'eventuale errore (lo possiamo fare dato che siamo ancora in fase di sviluppo)
  - **La prof si aspetta di trovare i try e catch nei nostri progetti.** Adesso non li mettiamo perché stiamo facendo in velocità, senza soffermarci sui dettagli.
  - Ci sono più soluzioni possibili, nel nostro caso scegliamo di ritornare null nel caso la query sia vuota. Contiamo il numero di righe della query con la funzione `mysqli_num_rows($queryResult)`
  - Il ciclo while prende ogni riga della query con `mysqli_fetch_assoc($queryResult)` e la inserisce nell'array facendo un append (inserimento alla fine).

- queryResult non ci serve più e potrebbe essere molto pesante, perciò liberiamo la memoria con `$queryResult -> free();`

- Riassunto: quello che abbiamo fatto fino adesso è stato costruire un file con delle funzioni per accedere al database. 1) accesso e 2) funzione che restituisce la lista degli album
- Il file "discografiaTemplate.html" che ci hanno dato è un file in cui c'è quello che abbiamo fatto nei lab precedenti (dentro c'è ul della discografia `<ul id="discografia">` e un segnaposto `{album}` ← è solo una stringa che dobbiamo riconoscere))

## Che cosa si manca? Compito per casa

- Ci manca un file che prende questo, si connette al DB, lancia la funzione getListaAlbum() per ogni album fa un `li`, lo mette in una stringa e con un string replace rimpiazza quell'album con il risultato della nostra computazione.
- Collegandosi da casa a [tecweb.studenti.math.unipd.it/stbressa/Laboratorio\\_6](http://tecweb.studenti.math.unipd.it/stbressa/Laboratorio_6) deve uscirci la discografia.php dei pinguini tattici nucleari

## Prossime volte

La prossima volta faremo da soli gli album

Giovedì inseriremo un brano nuovo