

Info

[Nome progetto] Nome e Cognome|Data| |---|---|---|---| |Gestione Whitelist proxy|Carlo Pezzotti|03.10.2019|

Lavoro svolto

Siccome negli scorsi ho riscoperto il mondo MVC, ho pensato che fosse meglio se il sito venisse rifatto con il seguente linguaggio. Per farlo ho usato un template modificato da un mio compagno di classe e creato da Massimo Sartori. Sono riuscito a convertire gran parte del progetto. Mi manca sistemare giusto i link e poi dovrei aver spostato l'intero progetto su MVC.

Per gestire la connessione al database uso il seguente codice Database.php :

```
<?php
namespace Libs;

class Database
{
    private static $connection = null;

    public static function get()
    {
        if (self::$connection === null) {
            if (SQLITE === null) {
                self::$connection = new \PDO("mysql:host=".DB_HOST.";dbname=".DB_NAME.
            } else {
                self::$connection = new \PDO("sqlite:".SQLITE);
            }
            self::$connection->setAttribute(\PDO::ATTR_ERRMODE, \PDO::ERRMODE_EXCEPTION);
        }
        return self::$connection;
    }
}
```

Il file di configurazione è il seguente config.php :

```
<?php
error_reporting(E_ALL);
ini_set("display_errors", 1);

define('URL', 'http://127.0.0.1:8080/');

define('DB_USER', 'root');
define('DB_PASS', 'Password&1');
define('DB_HOST', '10.20.143.220');
define('DB_NAME', 'gestioneWhitelist');
define('DB_PORT', 3307);
```

```
define('SQLITE', null);

$autoload_directories = array(
    "application/controllers/",
    "application/libs/",
    "application/models/"
);
```

Per effettuare il login uso il controller `home.php` la view `login.php` e il model `users.php`. La view non è cambiata. `Users.php`:

```
<?php
namespace Models;

use Libs\Database as Database;

class Users
{
    const USERNAME_NOT_EXIST = 0;
    const PASSWORD_WRONG = 1;
    const LOGIN_SUCCESS = 2;

    ...

    public static function exist($username, $password)
    {
        $query = Database::get()->prepare("select password from user where username=:u");
        $query->bindParam(":username", $username);
        $query->execute();
        $result = $query->fetchAll(PDO::FETCH_ASSOC);
        if(count($result)==0){
            return self::USERNAME_NOT_EXIST;
        }else if(password_verify($password, $result[0]["password"])){
            return self::LOGIN_SUCCESS;
        }else{
            return self::PASSWORD_WRONG;
        }
    }
}
```

Contiene tutti i metodi per collegarsi al database e fare richieste. Quindi rimane pressoché invariato dal codice precedente tranne per il metodo mostrato qui sopra. `exist()` va a controllare se un utente esiste oppure no e in caso ritorna `USERNAME_NOT_EXIST` se non esiste `PASSWORD_WRONG` se la password è sbagliata oppure `LOGIN_SUCCESS` se l'utente e la password sono giuste.

Oltre al login ho modificato la parte amministrativa. Ovvero siccome utilizzo MVC non è più necessario creare dei file a parte per la cancellazione dei dati o la modifica ciò l'ho gestito nel seguente modo:

```
public function modify($ID){
    if(!isset($_POST["username"])){
        $info = Admin::getInfo($ID);
        ViewLoader::load("adminpanel/index", array(
            'setID'=>$info["id"],
            'setUsername'=>$info["username"],
            'setName'=>$info["name"],
            'setSurname'=>$info["surname"],
            'setType'=>$info["type"],
            'setEmail'=>$info["email"]
        ));
    }else{
        $values = array(
            'username' => htmlspecialchars($_POST["username"]),
            'name' => htmlspecialchars($_POST["name"]),
            'surname' => htmlspecialchars($_POST["surname"]),
            'email' => htmlspecialchars($_POST["email"]),
            'type' => (htmlspecialchars($_POST["type"]) == "Admin") ? 1 : 2
        );
        $password = htmlspecialchars($_POST["password"]);
        if (!empty($password))
            $values["password"]=password_hash($password, PASSWORD_BCRYPT);

        Admin::modify($ID,$values);
        ViewLoader::load("adminpanel/index");
    }
}

public function delete($ID)
{
    Admin::delete($ID);
    ViewLoader::load('adminpanel/index', $_SESSION);
}
```

Errori riscontrati

Nessuno

Lavoro prossima lezione

Cotinuare con la conversione in MVC