# Lucrare Laborator 10 – PSD

## Obiective

1. Php. Clase.
2. Php Framework. CodeIgniter.

## Php.Clase.

Exemplu 1: Urmariti clasa DAL implementata pentru a realiza operatiile de adaugare, preluare, actualizare a datelor despre un utilizator in baza de date. Observati:

* Constantele utilizate pentru a memora detaliile de conectare la baza de date;
* Constructorul/destructorul care realizeaza initializarea/incheierea conexiunii cu baza de date.

<? php

/\*\*

 \* DAL - Data Access Layer

 \*

 \* The DAL class performs basic operations with database.

 \* This class implements the independence relative to a specific DBMS.

 \*

 \*/

class DAL

{

 /\*\*

 \* connection object

 \*/

 private $connection;

 const DB\_HOST = "localhost";

 const DB\_NAME = "PrimaBD";

 const DB\_USER = "root";

 const DB\_PASSWORD = "root";

 /\*\*

 \* Constructor

 \*

 \* @throws PDOException if operation fails

 \*/

 public function \_\_construct()

 {

 $connectionString = "mysql:host=".DAL::DB\_HOST.";dbname=".DAL::DB\_NAME;

 $this->connection = new PDO($connectionString, DAL::DB\_USER, DAL::DB\_PASSWORD);

 $this->connection->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

 }

 /\*\*

 \* add a new user

 \*

 \* @return void

 \* @throws PDOException if operation fails

 \*/

 public function addUser($username, $password)

 {

 $sql = "INSERT INTO Utilizatori(Utilizator, Parola)

 VALUES('".$username."', '".md5($password)."'); ";

 $this->connection->exec($sql);

 }

 /\*\*

 \* Get user by username

 \*

 \* @return an user

 \* @throws PDOException if operation fails

 \*/

 public function getUser($username)

 {

 $sql = "SELECT Id, Utilizator, Parola FROM Utilizatori

 WHERE Utilizator = '".$username."'; ";

 return $this->connection->query($sql)->fetch();

 }

 /\*\*

 \* update user

 \*

 \* @return void

 \* @throws PDOException if operation fails

 \*/

 public function updateUser($Id, $username, $password)

 {

 $sql = "Update Utilizatori set Utilizator = '".$username."', Parola = '".md5($password)."' WHERE Id=".$Id.";";

 $this->connection->exec($sql);

 }

 /\*\*

 \* Get all users

 \*

 \* @return array, list of users

 \* @throws PDOException if operation fails

 \*/

 public function getUsers()

 {

 $sql = "SELECT Id, Utilizator FROM Utilizatori";

 return $this->connection->query($sql);

 }

 /\*\*

 \* close the database connection

 \*/

 public function \_\_destruct()

 {

 // close the database connection

 $this->connection = null;

 }

}

?>

Exemplu 2: Utilizarea clasei DAL.

|  |
| --- |
| <!DOCTYPE html><html><body> <?php require\_once 'DAL.php'; try { $dal = new DAL(); $result = $dal->getUsers();if ($result->rowCount() > 0) { echo "<table><tr><th>ID</th><th>Utilizator</th><th></th></tr>"; // output data of each row foreach($result as $row) { echo "<tr><td>".$row["Id"]."</td><td>".$row["Utilizator"]."</td><td><a href='modificaUtilizator.php?Id=".$row["Id"]."' target='\_blank'>Update</a></tr>"; } echo "</table>"; } else { echo "Nici un utilizator"; } echo "<br><a href='adaugaUtilizator.php' target='\_blank'>Adauga utilizator</a>"; } catch(PDOException $e) { echo "Eroare: " . $e->getMessage(); } ?></body></html> |

## Php Framework.CodeIgniter.

Urmariti tutorialul de la adresa <https://www.tutorialspoint.com/codeigniter/>.

### Exercitii

1. Modificati jocul din laboratorul 9 astfel incat la finalizarea lui numele utilizatorului, numarul de intrebari corecte si numarul total de intrebari sa fie salvate intr-o baza de date. Proiectati si utilizati o clasa DAL asemanatoare cu cea din laborator pentru salvarea si extragerea datelor in/din baza de date.
2. Adaugati o sectiune BestScores care sa afiseze cele mai bune 10 scoruri obtinute.
3. Modificati view-ul si controller-ul prezentate la curs astfel incat sa fie afisat/adaugat si numarul matricol.
4. Implementati operatiile de stergere si actualizare urmand exemplul de la curs.

# Bibliografie:

1. <https://www.tutorialspoint.com/codeigniter/>