

# Dizajni dhe Zhvillimi i Web PHP lidhja me BDH (CRUD-BDH)

---

Greta Ahma  
greta.ahma@ubt-uni.net  
2021/2022

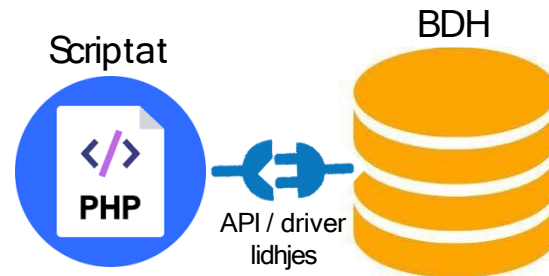
# Lidhja me baze të të dhënave

---

PJESA PARË

# qasja në Bazat e të Dhënave në PHP

- Për tu **qasur** dhe shtuar kontenit në **një bazë të dhënash** (p.sh; MySQL), fillimisht duhet të krijoni një lidhje midis bazës së të dhënave dhe një **skripte PHP**.



- Ekzistojnë **tre lloje metodash** në PHP *për të lidhur bazën e të dhënave* (p.sh MySQL) përmes backend:
  - **MySQL**
  - **MySQLi** (*vetem per MySQL*) (*OOP*)
  - **PDO** (*PHP Data Objects*) (*OOP*)

# qasja në Bazat e të Dhënave në PHP....

## MySQLi

- **MySQLi është një API** që përdoret si *funksion lidhës* për të *lidhur* pjesën e **server-side** (skriptat) të *aplikacionit* në **PHP** me bazën e të dhënave **MySQL**.

```
<?php
$servername = "localhost"; // 'localhost'
$db = "dbname"; //p.sh L12_CRUD
$username = "username"; //p.sh root
$password = "password"; //p.sh ' ' nuk ka

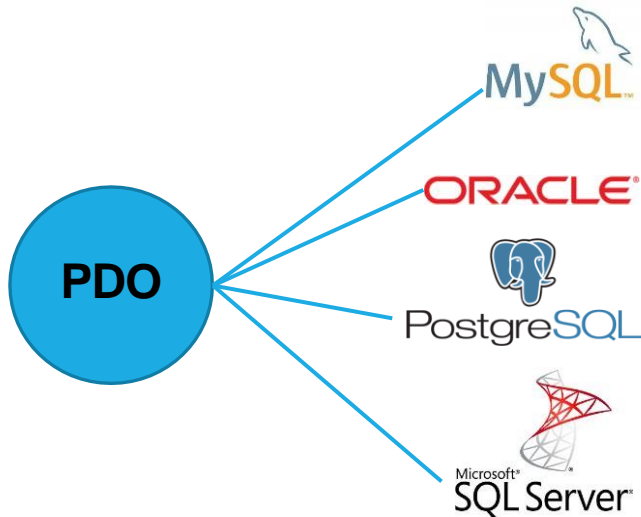
// Krijohet lidhja
$conn = mysqli_connect($servername, $username, $password,$db);

// Kontrolloni lidhjen
if (!$conn) {
    die("Lidhja dështoi : " . mysqli_connect_error());
}
echo "Është lidhur me sukses";
?>
```

# qasja në Bazat e të Dhënave në PHP....

□ **PDO extension** (PHP Data Objects), është një *shtresë abstraksioni* i Bazave të të Dhënave.

- Është si një **interface** për backend që të *ndërveprojë* me bazën e të dhënave.
- Gjithashtu ju **mundeson** për të *punuar* me baza të të dhënave të **ndryshme**.



```
<?php
    $servername = "localhost"; // 'localhost'
    $db = "dbname"; //p.sh L12_CRUD
    $username = "username"; //p.sh root
    $password = "password"; //p.sh ' ' nuk ka
    try {
        // Krijohet lidhja
        $conn = new PDO("mysql:host=$servername;dbname=myDB", $username, $password, $db);
        // set the PDO error mode to exception
        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        echo "Është lidhur me sukses";
    }
    catch(PDOException $e)
    {
        echo " Lidhja dështoi: " . $e->getMessage();
    }
    ?>
```

# Implementimi i operacionit: CRUD

---

PJESA DYTË

# maipulimi me dhena (DB) dhe shfaqja në Uebfaqe

❑ **Opercaioni bazë** i një Webfaqe/Webaplikacioni është *ralizimi* **operacionit CRUD**.

❑ Lidhja me BDH & Operacioni **CRUD**:

- **lidheni** dhe **krijoni** bazën e të dhënave
- **futni** të dhënat në bazën e të dhënave
- **lexoni** të dhënat nga baza e të dhënave
- **përditësoni** të dhënat në bazën e të dhënave
- **fshini** të dhënat në bazën e të dhënave

## krijimi dhe lidhja me bazën e të dhënave

---

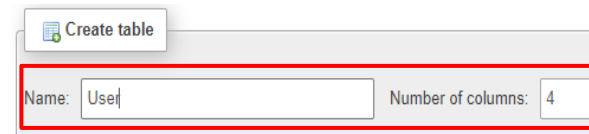
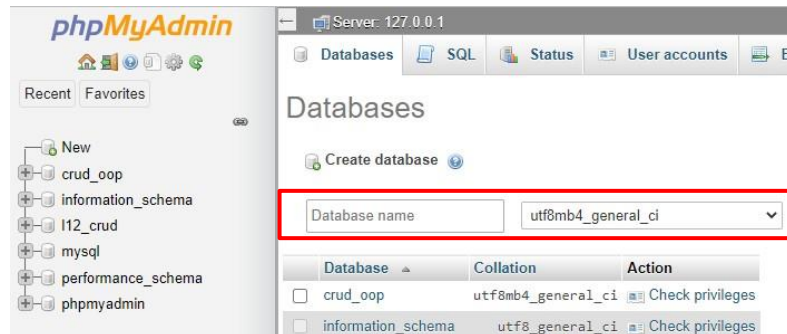
- **PDO (PHP Data Objects)** është një *extension* **PHP** përmes së cilës ne mund të *qasemi* dhe të *punojmë* me **bazat e të dhënave**.
  - Është më mirë i *mbrojtur* nga **hakerat**.
  - Është **konsistente** në të gjitha bazat e të dhënave, kështu që mund të punojë me **MySQL**, si dhe me lloje të tjera të **bazave të të dhënave** (**MS SQL**, **SQLite**, **Oracle**, **PostgreSQL**, etj.)
  - Në thelb është i **orientuar nga objekti**.



# krijimi i bazës së të dhënave

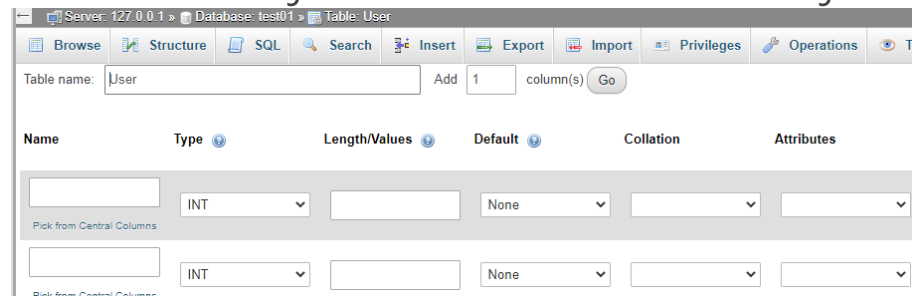
## □ Krijimi i BDH, p.sh; L12\_CRUD:

- Duke përdorur XAMPP; Open a web browser, enter a URL <http://localhost/phpmyadmin/>
- Në fushën database name – *emerto* p.sh: **l12\_crud**



Emerto-> tabelen dhe numrin e kolonave

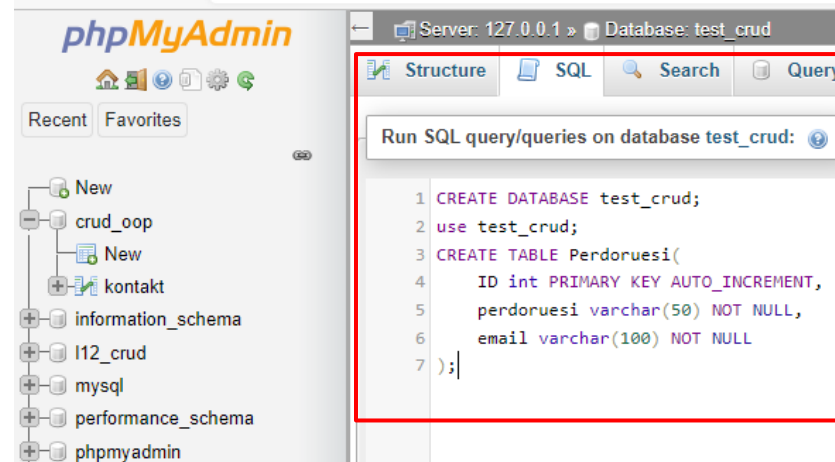
- Përcaktimi i lloji të të dhënave dhe vetit tjera të attributeve



# krijimi i bazës së të dhënave...

## □ Krijimi i BDH, p.sh; L12\_CRUD:

- Duke përdorur panelin e XAMPP'it



- Ose duke përdorur **shell console** në admin/xampp etj.

# lidhja me bazën të të Dhënave

- Për t'u *lidhur* me bazën e të dhënave, do të na duhet **emri i bazës së të dhënave**, emri i **përdoruesit** dhe **fjalëkalimi**.

```
//Kredencialet e DB-së.  
define('DB_HOST','localhost');  
define('DB_USER','your user name');  
define('DB_PASS','your user password');  
define('DB_NAME','your database name');
```

```
private $host = 'localhost';  
private $dbname = 'l12_crud';  
private $username = 'root';  
private $password = '';
```

- Vendosnet **lidhjen** e bazës së të dhënave ose *shfaqni* një mesazh **ERROR**.

```
$this->conn = new PDO("mysql:host=$this->host;dbname=$this->dbname",  
    $this->username, $this->password);
```

- **PDO::ATTR\_DEFAULT\_FETCH\_MODE, PDO::FETCH\_ASSOC**: kthen një grup të indeksuar sipas emrit të kolonës siç është kthyer në grupin tuaj të rezultateve.
- **PDO::FETCH\_BOTH**: kthen një grup të indeksuar si nga emri i kolonës ashtu edhe nga numri i kolonës me 0 të indeksuar siç është kthyer në grupin tuaj të rezultateve

```
$this->conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
$this->conn->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);
```

PDO::ERRMODE\_WARNING: Ngritni E\_WARNING.  
PDO::ERRMODE\_EXCEPTION: Throw exceptions.

# Implementimi i Klases për t'u lidhur me bazën e të dhënave duk përdorur metoden PDO.

```
class dbConnect{
    private $conn =null;
    private $host = 'localhost';
    private $dbname = 'l12_crud';
    private $username = 'root';
    private $password = '';

    public function connectDB(){ try {
        $this->conn = new PDO("mysql:host=$this->host;dbname=$this->dbname",
                               $this->username, $this->password);
        $this->conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION) . "<br/>";
        $this->conn->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC) . "<br/>";
        // echo "Lidhja me: {$this->dbname} - {$this->host}: eshte ralizuar me sukses!";
    } catch (PDOException $pdoe) {
        die("Nuk mund të lidhej me bazën e të dhënave {$this->dbname} :" . $pdoe->getMessage());
    }
    return $this->conn;
}
}
```

# Insertimi i të dhënave në bazën e të dhënave

- ❑ Fillimisht **krijo query** të zakonshem në SQL për **insertim** të dhënave

```
$sql="INSERT INTO Studenti (emri,mbiemri,departamenti,adresa) VALUES (?, ?, ?, ?)";
```

- ❑ Krijon një variable **\$stm** dhe pregadit **deklaraten e SQL'it** për **insertim**

```
$stm=$this->dbconn->prepare($sql);
```

- ❑ Ekzekuton të dhënat e shtuar

```
$stm->execute([$this->emri,$this->mbiemri,$this->departamenti,$this->adresa]);
```

Nëse deklarata është ekzekutuar me sukses shfaq mesazhin dhe ri-direkton të dokumenti **displayDhenat.php**.

```
public function insertoDhenat(){

    $sql="INSERT INTO Studenti (emri,mbiemri,departamenti,adresa) VALUES (?, ?, ?, ?)";

    $stm=$this->dbconn->prepare($sql);
    $stm->execute([$this->emri,$this->mbiemri,$this->departamenti,$this->adresa]);

    echo "<script>
        alert('dhenat jane ruajtur me sukses');
        document.location='displayDhenat.php';
    </script>";

}
```

# Insertimi i të dhënave në bazën e të dhënave...

□ **Metoda e insertimi** *inicohet* nëse user *shtype* buttonin **RUAJ** (save) në HTML;

- **Testohe** shtypja e buttonit **ruaj(save)** me metoden: `if (isset($_POST['save'])),`
- Krijohet **objekti** (instancë) regj: `$regj= new Studenti();`
- **Thirret** metoda p.sh; `setEmri($_POST['emri']);` nga klasa Studenti për të i shtuar dhënat e attributeve përkatse.

- **Thirret** metoda `insertoDhenat();` perkatesisht *inicohet* **ekzekutimi** i saj.

```
require_once('modeliStudenti.php');

if (isset($_POST['save'])) {

    $regj= new Studenti();
    $regj->setEmri($_POST['emri']);
    $regj->setMbiemri($_POST['mbiemri']);
    $regj->setDept($_POST['departamenti']);
    $regj->setAdresa($_POST['adresa']);

    $regj->insertoDhenat();
}
```

# Leximi i të dhënave nga baza e të dhënave

- Fillimisht **krijo query** të zakonshme në SQL për **lexim** të të gjitha dhënave

```
$sql='SELECT * FROM Studenti';
```

- Krijon një variabël **\$stm** dhe pregadit **deklaraten e SQL'it** për **lexim** të të dhënave

```
$stm=$this->dbconn->prepare($sql);
```

- Ekzekuton deklarata e query'it perkatesishte

```
$stm->execute();
```

- Të dhënat e nxjerrura nga baza e të dhënave

ruhen në variablen **\$rezultati** e cila është e llojit array asociativ

```
$rezultati =$stm->fetchAll(PDO::FETCH_ASSOC);  
return $rezultati;
```

```
public function lexoDhenat(){  
    $sql='SELECT * FROM Studenti';  
    $stm=$this->dbconn->prepare($sql);  
    $stm->execute();  
    $rezultati =$stm->fetchAll(PDO::FETCH_ASSOC);  
    return $rezultati;  
}
```

```
fetchAll(PDO::FETCH_ASSOC);
```

# Leximi i të dhënave nga baza e të Dhënave...

- Metoda e leximit *inicohet* nëse **user e incion** me ndonjë *veprim që* ka të bëjë me *shfaqje* e të dhënave, p.sh:

```
echo "<script>document.location='displayDhenat.php';</script>";
ose
header("Location:displayDhenat.php");
```

- Sipas **veprimi** paraprak dokumenti që *shfaq të dhënat* është emruar **displayDhenat.php**, d.m.th në këtë HTML dokument fillimisht *nevojitet* të **instanciohet Objekti .psh Studenti** dhe të thirret metoda **lexoDhenat()**, dhe ruhen në një variable **\$all** array.

- Krijohet cikli **foreach** për të iteruar në *array'in associative* të *ruajtur* në variable **\$all** dhe shfaq dhënat e *tërhequara* nga DB.
- P.sh `<?php echo $value['emri'];>` nevojitet pastaj që vlerën e *tërhequr* nga DB të shfaqen në tabel.

```
<?php
require_once('modeliStudenti.php');
$dhenat=new Studenti();
$all=$dhenat->lexoDhenat();
?>
```

```
<tr>
  <?php foreach($all as $key=>$value){ ?>
    <td><?php echo $value['emri'];></td>
    <td><?php echo $value['mbiemri'];></td>
    <td><?=$value['departamenti'];></td>
    <td><?=$value['adresa'];></td>
  } ?>
</tr>
```



## Përditëson të dhënat në bazën e të Dhënave

- Fillimisht **krijo query** të zakonshem në SQL për **përditsimi** të dhënave për entitet të kushtzuar sipas **id**. `where id=?`;

```
$sql='UPDATE Studenti SET emri=?, mbiemri=?, departamenti=?, adresa=? where id=?';
```

- Krijon një variable **\$stm** dhe pregadit **deklaraten e SQL'it** për **lexim** të të dhënave `$stm=$this->dbconn->prepare($sql);`

- Ekzekuton deklarata e query'it përkatesisht

```
$stm->execute([$this->emri, $this->mbiemri,$this->departamenti,$this->adresa, $this->id]);
```

# Përditëson të dhënat në bazën e të Dhënave...

- Metoda e përditsimi *inicohet* nëse **user** e *incion veprim duke shtup butonin Edit.*

```
<a href="edit.php?id=php echo $value['id'];?&gt;"&gt;&lt;button id='e'&gt;EDIT&lt;/button&gt;&lt;/td&gt;&lt;/a&gt;</pre
```

- Sipas **veprimi** paraprak dokumenti që *aktivizohet* është `href="edit.php?"`. Në dokmenti HTML që shfaq rezultatin nevojitet të *instancet Objekti psh Studenti*, ku metoda `$_GET` `$myId=$_GET['id'];` kthen IDn e të **dhënës** që *refrencohet* në URL. Pastaj *thirret* metoda `lexoDhenatSipasIDs()` e cila kthen *reshtin e të dhënave* nga **DB** dhe *ruhen* në një variable e Objektivit `$records`.

```
require_once('modeliStudenti.php');  
$dhenat= new Studenti();  
$myId=$_GET['id'];  
$record=$dhenat->lexoDhenatSipasIDs($myId);
```

- Të dhënat pasta janë *shfaqur* ne formen e HTMLs.

Perditso te dhenat ne Formular

Emri	<input type="text" value="Medina"/>
Mbiemri	<input type="text" value="Shamulli"/>
Departamenti	<input type="text" value="MEK"/>
Adresa	<input type="text" value="Prishtine"/>
	<input type="button" value="RUAJ"/>

# Përditëson të dhënat në bazën e të Dhënave...

- ❑ **krijë query** të zakonshem në SQL për **lexim** të të gjitha dhënave për *objektin të kushtzuar*

sipas **id**: `$sql='SELECT * FROM Studenti where id=:id';`

- ❑ Pregadit **deklaraten e SQL'it** për **leximin** e të dhënave duke *siguruar lidhjen me BDH*, krijohet variabla **\$stm** që të dhënat e deklarates SQL të ruhen si object:

, `$stm=$this->dbconn->prepare($sql);`

- ❑ **Ekzekuton** deklarata e query'it sipas e kushtzuar për **rekordin e IDs përkatës**

`$stm->execute([':id'=>$this->id=$id]);`

- ❑ **Caktoni** një **variabël** për të **rekordin** që keni *nxjerrë* nga BDH (në hapin e mëparshëm) .

`$rezultati = $stm->fetch(PDO::FETCH_ASSOC);`

- ❑ **Metoda kthen** variable: `return $rezultati;`

```
public function lexoDhenatSipasIDs($id){  
    $sql='SELECT * FROM Studenti where id=:id';  
  
    $stm=$this->dbconn->prepare($sql);  
    $stm->execute([':id'=>$this->id=$id]);  
    $rezultati = $stm->fetch(PDO::FETCH_ASSOC); return  
    $rezultati;  
}
```

## Përditëson të dhënat në bazën e të Dhënave...

- Pas *shfaqjes* së të **dhënave** në formularin e **editim**, përdoruesi **editon të dhënën** përkatese dhe me rastin e *shtypjes* së buttonit **save (edit) inicohet** funksioni `updateDhenat();`

- p.sh *përditsohet* departamenti: `$dhenat->setDept($_POST['departamenti']);`

- Për të *siguruar* që përdoruesi ka *shtypur* butoni **edit** `if (isset($_POST['edit']))` dhe e dhëna **përkon** me **id** e *incuar* `if($myId==$dhenat->getId())` për *përditsim*, nëse është e **saktë** vazhdon *dërgimi* i të dhënave të *përditsuar* në **sistem (DB)**.

```
if (isset($_POST['edit'])){\n    if ($myId==$dhenat->getId()){\n\n        $dhenat->setEmri($_POST['emri']);\n        $dhenat->setMbiemri($_POST['mbiemri']);\n        $dhenat->setDept($_POST['departamenti']);\n        $dhenat->setAdresa($_POST['adresa']);\n\n        echo $dhenat->updateDhenat();\n    }\n    echo\n    "<script>\n        alert('dhenat jane PERDITSUAR me sukses');\n        document.location='displayDhenat.php'\n    </script>";\n}
```

## delete të dhënat në bazën e të dhënave...

- ❑ Fillimisht **krijo query** të zakonshem në SQL për **fshirje** e të dhënave për rekordin e percaktuar nga kushti sipas **id**. `$sql="DELETE FROM Studenti WHERE id=:id";`
- ❑ *Caktohet* një variable **\$stm** dhe *pregadit deklarata e SQL* për **fshirje** të të dhënave dhe duke *siguruar lidhjen me BDH*; `$stm=$this->dbconn->prepare($sql);`

- *Lidh parametrat*: `$stm->bindParam(':id',$this->id=$id);`
- *Ekzekuton deklarata e query* të sipas *ekushti* për rekordin e **IDs** përkatës; `$stm->execute();`

```
public function deleteDhenat($id){
    $sql="DELETE FROM Studenti WHERE id=:id";

    $stm=$this->dbconn->prepare($sql);
    $stm->bindParam(':id',$id);
    $stm->execute();
    if ($stm==true){
        echo "<script>
            alert('dhenat jane DELETE me sukses');
            document.location='displayDhenat.php';
        </script>";
    }
    else {
        return false;
    }
}
```

## delete të dhënat në bazën e të dhënave...

□ Metoda delete inicohet nëse user e incion veprim duke shtup butonin

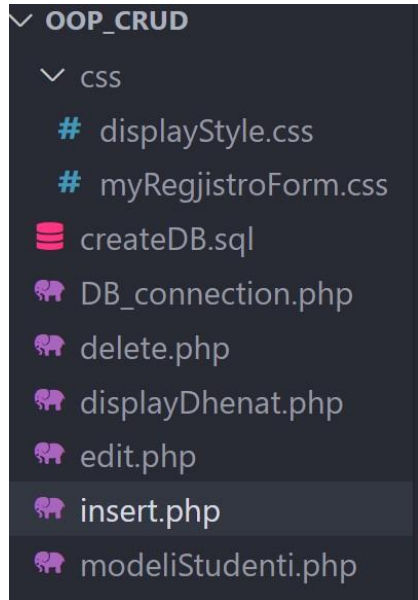
```
<a href="delete.php?id=<?php echo $value['id']?>"><button id="d">DELETE</button></a>
```

- Sipas **veprimi** paraprak dokumenti që **aktivizohet** është `href=delete.php?`. Më që rast URL përmban reference e **delete.php** dhe **ID** e rekordit të përzgjedhur për **delete dergohet** në BDH për të e realizuar veprim **DELETE**.

- Fillimishte **testohet** nëse është **terhequr** ID e rekordit me: `if(isset($_GET['id']))`
- Pastaj **caktohet** variabla **\$myID** që e merr atë të **dëne (ld)** `$myID=$_GET['id'];` që përcakton rekordin për t'u fshine në BDH duke e thirrur metoden: `$dhena->deleteDhenat($myID);`

```
require_once('modeliStudenti.php');  
$dhena = new Studenti();  
  
if(isset($_GET['id'])) {  
    $myID=$_GET['id'];  
    $dhena->deleteDhenat($myID);  
}
```

# Rasti: Implementimi i operacionit CRUD: PDO - MySQL



```
create DATABASE L12_CRUD; USE L12_CRUD
CREATE TABLE Studenti(
    id int Primary key auto_increment,
    emri varchar(50) not null,
    mbiemri varchar(50) not null,
    departamenti char(6)not null,
    adresa varchar(20)
);
```

Krijimi i **BDH** dhe tabeles **Studenti**

```
?php
class dbConnect{
    private $conn =null;
    private $host = 'localhost';
    private $dbname = 'l12_crud';
    private $username = 'root';
    private $password = '';

    public function connectDB(){

        try {
            $this->conn = new PDO("mysql:host=$this->host;dbname=$this->dbname",
                                $this->username, $this->password);
            $this->conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION) . "<br/>";
            $this->conn->setAttribute(PDO::FETCH_BOUND, PDO::FETCH_BOTH);
            // $this->conn->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC) . "<br/>";

            // echo "Lidhja me: {$this->dbname} - {$this->host}: eshte ralizuar me sukses!";
        } catch (PDOException $pdoe) {
            die("Nuk mund të lidhej me bazën e të dhënave {$this->dbname} : " . $pdoe->getMessage());
        }
        return $this->conn;
    }
}
```

Krijimi i klases **dbConnect** për të e ralizuar lidhjen me **BDH** sipas **PDO**

# Rasti: Implementimi i operacionit CRUD: PDO –MySQL...

```
<?php
require_once('DB_connection.php');

class Studenti extends dbConnect {

    private $id;
    private $emri;
    private $mbiemri;
    private $departamenti;
    private $adresa;
    protected $dbconn;

    public function __construct($id='', $emri='',
        $mbiemri='', $departamenti='', $adresa='')
    {
        $this->id=$id;
        $this->emri=$emri;
        $this->mbiemri=$mbiemri;
        $this->departamenti=$departamenti;
        $this->adresa=$adresa;

        $this->dbconn=$this->connectDB();
    }
}
```

```
public function setId($id){
    $this->id=$id;
}

public function getId(){
    return $this->id;
}

public function setEmri($emri){
    $this->emri=$emri;
}

public function getEmri(){
    return $this->emri;
}

public function setMbiemri($mbiemri){
    $this->mbiemri=$mbiemri;
}

public function getMbiemri(){
    $this->mbiemri;
}

public function setDept($departamenti){
    $this->departamenti=$departamenti;
}

public function getDept(){
    return $this->departamenti;
}

public function setAdresa($adresa){
    $this->adresa=$adresa;
}

public function getAdresa(){
    return $this->adresa;
}
}
```

Implementimi Klases **studenti** përkatesisht dokumentit **modeliStudenti**.



# Rasti: Implementimi i operacionit CRUD: PDO –MySQL...

```
public function insertoDhenat(){  
  
    $sql="INSERT INTO Studenti (emri,mbiemri,departamenti,adresa)  
                                VALUES (?, ?, ?, ?)";  
  
    $stm=$this->dbconn->prepare($sql);  
  
    $stm->execute([$this->emri,$this->mbiemri,$this->departamenti,$this->adresa]);  
  
    echo "<script>  
        alert('dhenat jane REGJISTRUAR me sukses');  
        document.location='displayDhenat.php';  
    </script>";  
  
}  
  
public function lexoDhenat(){  
    $sql='SELECT * FROM Studenti';  
  
    $stm=$this->dbconn->prepare($sql);  
    $stm->execute();  
    $rezultati =$stm->fetchAll(PDO::FETCH_ASSOC);  
    return $rezultati;  
}
```

```
public function lexoDhenatSipasIDs($id){  
    $sql='SELECT * FROM Studenti where id=:id';  
  
    $stm=$this->dbconn->prepare($sql);  
    $stm->execute([':id'=>$this->id=$id]);  
    $rezultati =$stm->fetch(PDO::FETCH_ASSOC);  
    return $rezultati;  
}
```

Implementimi clases **Studenti** përkatesisht dokumentit **modeliStudenti.php**.

## Rasti: Implementimi i operacionit CRUD: PDO –MySQL...

```
public function updateDhenat(){
    $sql='UPDATE Studenti SET emri=?, mbiemri=?, departamenti=?, adresa=? where id=?';

    $stm=$this->dbconn->prepare($sql);
    $stm->execute([$this->emri, $this->mbiemri,$this->departamenti,$this->adresa, $this->id]);

}

public function deleteDhenat($id){
    $sql="DELETE FROM Studenti WHERE id=:id";

    $stm=$this->dbconn->prepare($sql);
    $stm->bindParam(':id',$id);
    $stm->execute();
    if ($stm==true){
        echo "<script>
            alert('dhenat jane DELETE me sukses');
            document.location='displayDhenat.php';
        </script>";
    }
    else {
        return false;
    }
}

}
```

Implementimi Klases **studenti** përkatesisht dokumentit **modeliStudenti**.

# Implementimi dokumentit për t'i shfaqur të dhenat.

🐘 displayDhenat.php >

```
<?php
    require_once('modeliStudenti.php');
    $dhenat=new Studenti();
    $all=$dhenat->lexoDhenat();
?>
<!DOCTYPE html>
<html>
    <head>
    <link rel="stylesheet" href ="css/displayStyle.css" />
    <title>Shfaq dhenat</title>
    </head>
    <body>
        <div id="a1">
            <header>
                <h3>Ju lutem shtype per te regjistruar te dhenat ne Sistem</h3>
                <a href="insert.php"><Button id='r'>Regjistrohu</Button></a>
            </header>
            <table>
            <tr>
            <th>Emri</th>
            <th>Mbiemri</th>
            <th>Departamenti</th>
            <th>Adresa</th>
            <th>Action</th>
            </tr>
```

# Implementimi dokumentit për t'i shfaqur të dhenat.

```
<tr>

    <?php
        foreach($all as $key=>$value){
            ?>
            <td><?php echo $value['emri']?></td>
            <td><?php echo $value['mbiemri']?></td>
            <td><?php echo $value['departamenti']?></td>
            <td><?php echo $value['adresa']?></td>
            <td id='de'><a href="delete.php?id=<?php echo $value['id']?>"><button id="d">DELETE</button></a>
            <a href="edit.php?id=<?php echo $value['id']?>"><button id='e'>EDIT</button></td></td></a>
        }
    </tr>
    <?php
        ?>
    </table>
</div>
</body>
</html>
```

localhost/UEB2122/OOP\_CRUD/displayDhenat.php


Ju lutem shtype per te regjistruar te dhenat ne Sistem

Regjistrohu

Lista e te dhenave:

Emri	Mbiemri	Departamenti	Adresa	Action	
Yllka	Gashi	MBE	Prishtine	DELETE	EDIT
Kroi	Hoxha	SHKI	Ferizaj	DELETE	EDIT

# Implementimi dokumentit për t'i insertuar të dhenat.

 insert.php >

Shto te dhenat ne Formularin e Regjistrimit

**Emri**

**Mbiemri**

**Departamenti**

**Adresa**

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="css/mysingUPstyle.css" />
    <title>Formulari i Regjistrimit</title>
  </head>
  <body>
    <div id="formulari">
      <h3>Shto te dhenat ne Formularin e Regjistrimit</h3>
      <form action="" method="POST">
        <label>Emri</label>
        <input type="text" class="inp" name="emri" placeholder="shto emrin..."/>
        <label>Mbiemri</label>
        <input type="text" class="inp" name="mbiemri" placeholder="shto mbiemrin..."/>
        <label>Departamenti</label>
        <input type="text" class="inp" name="departamenti" placeholder="shto departamentin..."/>
        <label>Adresa</label>
        <input type="text" class="inp" name="adresa" placeholder="shto adresen"/>
        <button name='save'>RUAJ</button>
      </form>
    </div>
  </body>
```

```
<?php
require_once('modeliStudenti.php');

if (isset($_POST['save'])) {

    $regj= new Studenti();
    $regj->setEmri($_POST['emri']);
    $regj->setMbiemri($_POST['mbiemri']);
    $regj->setDept($_POST['departamenti']);
    $regj->setAdresa($_POST['adresa']);

    $regj->insertoDhenat();

}
?>
```

# Implementimi dokumentit për t'i përditsuar të dhënat.

Perditso te dhenat ne Formular

Emri

Mbiemri

Departementi

Adresa

edit.php >

```
<?php
require_once('modeliStudenti.php');
$dhenat= new Studenti();
$myId=$_GET['id'];
$record=$dhenat->lexoDhenatSipasIDs($myId);

if (isset($_POST['edit'])){
    if ($myId==$dhenat->getId()){

        $dhenat->setEmri($_POST['emri']);
        $dhenat->setMbiemri($_POST['mbiemri']);
        $dhenat->setDept($_POST['departamenti']);
        $dhenat->setAdresa($_POST['adresa']);
        echo $dhenat->updateDhenat();
    }
    echo
    "<script>
        alert('dhenat jane PERDITSUAR me sukses');
        document.location='displayDhenat.php'
    </script>";
}
?>
```

```
<!DOCTYPE html>
<html>
    <head>
        <link rel="stylesheet" href ="css/mysingUPstyle.css" />
        <title>Formulari i Regjistrimit</title>
    </head>
    <body>
        <div id="formulari">
            <h3>Perditso te dhenat ne Formular</h3>
            <form action='' method="POST">
                <label>Emri</label>
                <input type="text" class="inp" name="emri"
                    value ="<?php echo $record['emri'];?> "/>
                <label>Mbiemri</label>
                <input type="text" class="inp" name="mbiemri"
                    value ="<?php echo $record['mbiemri'];?> "/>
                <label>Departementi</label>
                <input type="text" class="inp" name="departamenti"
                    value ="<?php echo $record['departamenti'];?> "/>
                <label>Adresa</label>
                <input type="text" class="inp" name="adresa"
                    value ="<?php echo $record['adresa'];?> "/>
                <button name='edit'>RUAJ</button>
            </form>
        </div>
    </body>
</html>
```

## Implementimi dokumentit për t'i **DELETE** të dhënat.

```
<?php
    require_once('modeliStudenti.php');
    $dhena = new Studenti();

    if(isset($_GET['id'])){
        $myID=$_GET['id'];
        $dhena->deleteDhenat($myID);
    }
?>
```

```

* {
  margin: 0;
  padding: 0;
}

body {
  font-family: 'Segoe UI light', Tahoma, Geneva, Verdana, sans-serif;
}

#a1 {
  display: flex;
  flex-direction: column;
  width: 60%;
  margin: 0 auto;
  background-color: rgb(215, 216, 228);
}

hr {
  width: 99%;
  height: 3px;
  background: cornflowerblue;
  margin: 0 auto;
}

header,
table {
  display: flex;
  flex-direction: column;
  width: 100%;
  min-height: 100px;
  margin: 0 auto;
  justify-content: center;
  align-items: center;
}

p {
  text-align: center;
  font-size: 14pt;
  font-weight: bolder;
  padding: 5px 0px;
}

```

```

button {
  background-color: cornflowerblue;
  width: 100px;
  height: 30px;
  border-radius: 5px;
}

#r {
  margin: 10px;
}

td {
  width: 100px;
  padding: 5px;
}

th {
  background-color: rgb(56, 21, 212);
  color: rgb(202, 214, 214);
  padding: 5px;
  font-size: 14pt;
}

#de {
  width: 80px;
  display: flex;
  justify-content: center;
  padding-right: 5px;
  width: 100%;
}

#d {
  background-color: red;
}

#e {
  background-color: orange;
}

tr:nth-child(odd) {
  background-color: rgb(148, 145, 148);
}

```



css > # myRegjistroForm.css >

```
#formulari {
  display: flex;
  flex-direction: column;
  justify-content: center;
  width: 40%;
  min-height: 300px;
  border: 1px solid blue;
  margin: 0 auto;
  color: blue;
  font-family: 'Segoe UI lighte', Tahoma, Geneva, Verdana, sans-serif;
}

form {
  display: flex;
  flex-direction: column;
  background-color: bisque;
}

h3 {
  text-align: center;
}

input,
label {
  width: 200px;
  padding: 5px;
  margin-left: 15px;
}

button {
  margin: 15px;
  width: 60px;
}
```

# Pyetje

---



# Referencat

---

- ❑ The essentials of Object-Oriented PHP by Joseph Benharosh.

