


APLICAȚIE NIVEL MEDIU

Fișă de lucru – Aplicație CRUD (Agendă de contacte)

Obiectiv

Realizarea unei aplicații web simple care permite gestionarea contactelor (Create, Read, Update, Delete) folosind:

- PHP (procedural sau OOP simplu)
- MySQL (baza de date: dbx)
- PDO pentru conexiune securizată

 **Structura proiectului.** Se realizează un folder cu numele **agenda**, respectând următoarele:

```
agenda/  
├─init_table  
├─ config.php  
├─ index.php  
├─ create.php  
├─ edit.php  
├─ delete.php  
└─ db.sql (opțional pentru referință)
```

0. **config.php** Conexiunea la baza de date

```
<?php  
$host = 'localhost';  
$db   = 'linndows_cms';  
$user = 'linndows_eduard';  
$pass = '3duard@123';  
$charset = 'utf8mb4'; // nu accepta diacritice  
  
$dsn = "mysql:host=$host;dbname=$db;charset=$charset";  
  
$options = [ //Acesta este un vector de opțiuni trimis către server la conexiune  
    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,  
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,  
];  
  
try {$pdo = new PDO($dsn, $user, $pass, $options);}  
catch (PDOException $e) {die("Eroare conexiune: " . $e->getMessage());}
```

1. `init_table.php` – realizarea tabelii în baza de date

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

// Introducere inițialele nume X, prenume Y
$initiale = "xy";

// Validare
if (!preg_match('/^[a-zA-Z]{2,5}$/', $initiale)) {
    die("Inițiale invalide!");
}

// Numele tabelii
$tableName = "contacts_" . strtolower($initiale);

try {

    // Creează tabela dacă nu există
    $createSql = "CREATE TABLE IF NOT EXISTS `{$tableName}` (
        id INT AUTO_INCREMENT PRIMARY KEY,
        nume VARCHAR(100) NOT NULL,
        prenume VARCHAR(100) NOT NULL,
        oras VARCHAR(100),
        adresa VARCHAR(255),
        email VARCHAR(100),
        telefon VARCHAR(20),
        created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4";

    $pdo->exec($createSql);

    // Golește tabela (șterge toate înregistrările)
    $truncateSql = "TRUNCATE TABLE `{$tableName}`";
    $pdo->exec($truncateSql);

    echo "✅ Tabela <b>{$tableName}</b> este pregătită (creată + golită)!"

} catch (PDOException $e) {
    echo "❌ Eroare: " . $e->getMessage();
}


// Presupunem că elevul se numește Avramescu Catalin, tabela se va numi contacts_ac
```

Ce face acum scriptul

La fiecare accesare:

1. Creează tabela dacă nu există
 2. Șterge toate datele din ea
 3. Resetează AUTO_INCREMENT la 1
-

Atenție

 Toate datele vor fi șterse la fiecare rulare!

2. Listare contacte – index.php

```
<?php require 'config.php'; ?>

<!DOCTYPE html>
<html>
<head>
    <title>Agenda contacte</title>
</head>
<body>

<h2>Lista contacte</h2>
<a href="create.php">Adaugă un contact NOU</a>

<table border="1">
<tr>
    <th>Nume</th>
    <th>Prenume</th>
    <th>Oraș</th>
    <th>Adresă</th>
    <th>Email</th>
    <th>Telefon</th>
    <th>Acțiuni</th>
</tr>

<?php
$stmt = $pdo->query("SELECT * FROM contacts_ac"); //ac=Avramescu Catalin
while ($row = $stmt->fetch()) {
    echo "<tr>
        <td>{$row['nume']}</td>
        <td>{$row['prenume']}</td>
        <td>{$row['oras']}</td>
        <td>{$row['adresa']}</td>
        <td>{$row['email']}</td>
        <td>{$row['telefon']}</td>
        <td>
            <a href='edit.php?id={$row['id']}'>Editare</a>
            <a href='delete.php?id={$row['id']}'>Stergere</a>
        </td>
    </tr>";
}
?>

</table>
</body>
</html>
```

+ 3. Adăugare contact – create.php

```
<!DOCTYPE html>
<html>
<head>
  <title>Agenda contacte</title>
</head>
<body>

<h3>Adaugare contact nou</h3>

<?php
require 'config.php'; //ac=Avramescu Catalin

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $sql = "INSERT INTO contacts_ac (nume, prenume, oras, adresa, email, telefon)
          VALUES (?, ?, ?, ?, ?, ?)";

    $stmt = $pdo->prepare($sql);
    $stmt->execute([
        $_POST['nume'],
        $_POST['prenume'],
        $_POST['oras'],
        $_POST['adresa'],
        $_POST['email'],
        $_POST['telefon']
    ]);

    header("Location: index.php");
}
?>

<form method="post">
  Nume: <input type="text" name="nume"><br>
  Prenume: <input type="text" name="prenume"><br>
  Oraș: <input type="text" name="oras"><br>
  Adresă: <input type="text" name="adresa"><br>
  Email: <input type="email" name="email"><br>
  Telefon: <input type="text" name="telefon"><br>
  <button type="submit">Salvează</button>
</form>
<br><hr>
</table>
</body>
</html>
```

4. Editare contact – edit.php

```
<?php
require 'config.php'; //ac=Avramescu Catalin

$id = $_GET['id'];

$stmt = $pdo->prepare("SELECT * FROM contacts_ac WHERE id = ?");
$stmt->execute([$id]);
$contact = $stmt->fetch();

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $sql = "UPDATE contacts_ac SET
        nume = ?, prenume = ?, oras = ?, adresa = ?, email = ?, telefon = ?
        WHERE id = ?";

    $stmt = $pdo->prepare($sql);
    $stmt->execute([
        $_POST['nume'],
        $_POST['prenume'],
        $_POST['oras'],
        $_POST['adresa'],
        $_POST['email'],
        $_POST['telefon'],
        $id
    ]);

    header("Location: index.php");
}
?>

<form method="post">
    Nume: <input type="text" name="nume" value="<?= $contact['nume'] ?>"><br>
    Prenume: <input type="text" name="prenume" value="<?= $contact['prenume']
?>"><br>
    Oraș: <input type="text" name="oras" value="<?= $contact['oras'] ?>"><br>
    Adresă: <input type="text" name="adresa" value="<?= $contact['adresa']
?>"><br>
    Email: <input type="email" name="email" value="<?= $contact['email']
?>"><br>
    Telefon: <input type="text" name="telefon" value="<?= $contact['telefon']
?>"><br>
    <button type="submit">Actualizează</button>
</form>
```

✘ 5. Ștergere contact – delete.php

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

$id = $_GET['id'];

$stmt = $pdo->prepare("DELETE FROM contacts_ac WHERE id = ?");
$stmt->execute([$id]);

header("Location: index.php");
```



Sarcini pentru elevi

- ✓ Implementați aplicația
 - ✓ Adăugați minim 10 contacte cu date total diferite, fără a utiliza diacritice în formularele de înregistrare (Florică, București, Guță)
 - ✓ Testați fiecare operație CRUD
 - 🎨 Îmbunătățiți interfața (CSS sau Bootstrap)
 - 🔍 Adăugați funcționalitate de căutare după nume
-