

Zajęcia 3

Wykorzystanie bazy danych MySQL w PHP

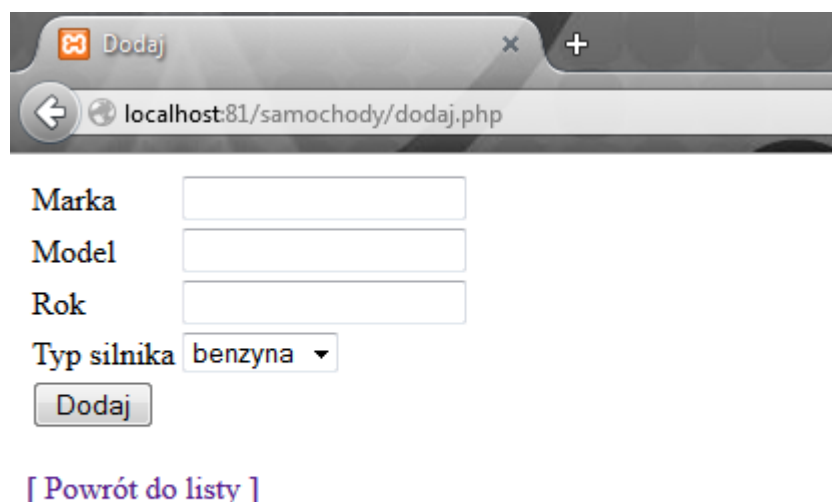
Cel zajęć

Na zajęciach zaprezentowana zostanie mini aplikacja służąca do zarządzania danymi zapisanymi w tabeli *samochody* w bazie danych MySQL.

Dostęp do aplikacji będzie możliwy po zalogowaniu się. Dane użytkowników przechowywane będą w tabeli *uzytkownicy* w bazie danych.

Dodawanie samochodu

Stworzenie funkcjonalności odpowiedzialnej za dodawanie samochodu wymaga zbudowania formularza, a następnie przekazania z niego danych do zapytania SQL *INSERT*, które wstawi rekord do tabeli *samochody*.



The screenshot shows a web browser window with a single tab titled 'Dodaj'. The address bar displays 'localhost:81/samochody/dodaj.php'. The page content includes a form with the following elements:

- Text input field for 'Marka'.
- Text input field for 'Model'.
- Text input field for 'Rok'.
- Dropdown menu for 'Typ silnika' with 'benzyna' selected.
- A 'Dodaj' button.
- A link '[Powrót do listy]' below the button.

Kod podstrony:

```
<?php

session_start();

if (empty($_SESSION['zalogowany'])) {
    header("Location: logowanie.php");
    exit();
}

if (isset($_POST['dodaj'])) {
    $pdo = new PDO('mysql:host=localhost;dbname=test', 'root', '');

    $stmt = $pdo->prepare("INSERT INTO samochody (marka, model, rok, typ_silnika) VALUES
(:marka, :model, :rok, :typ_silnika)");
    $wynik = $stmt->execute([
        'marka' => $_POST['marka'],
        'model' => $_POST['model'],
        'rok' => $_POST['rok'],
        'typ_silnika' => $_POST['typ_silnika'],
    ]);

    if ($wynik == true) {
        header("Location: index.php?komunikat=1");
        exit();
    } else {
        die("Operacja się nie powiodła: " . $pdo->errorInfo());
    }
}
?>

<!DOCTYPE html>
<html>

<head>
    <title>Dodaj</title>
    <meta charset="utf-8">
</head>
```

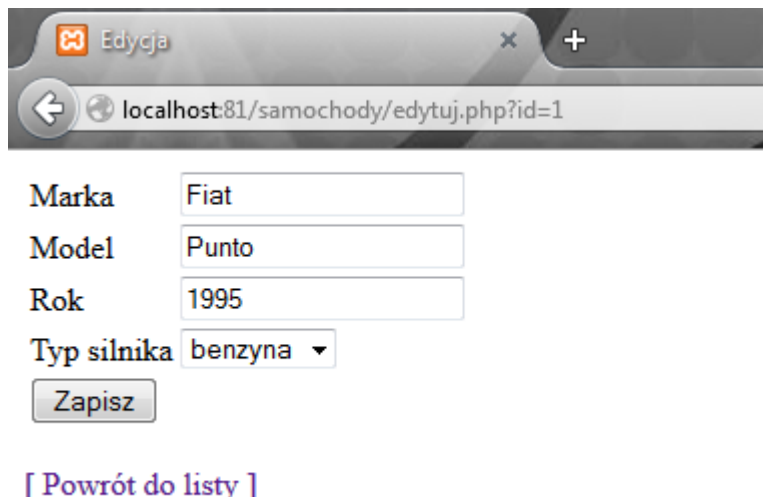
```
<body>
  <form method="post" action="">
    <table>
      <tr>
        <td>Marka</td>
        <td><input type="text" name="marka" /></td>
      </tr>
      <tr>
        <td>Model</td>
        <td><input type="text" name="model" /></td>
      </tr>
      <tr>
        <td>Rok</td>
        <td><input type="text" name="rok" /></td>
      </tr>
      <tr>
        <td>Typ silnika</td>
        <td>
          <select name="typ_silnika">
            <option value="benzyna">benzyna</option>
            <option value="diesel">diesel</option>
          </select>
        </td>
      </tr>
      <tr>
        <td colspan="2"><input type="submit" name="dodaj" value="Dodaj" /></td>
      </tr>
    </table>
  </form>

  <p>
    <a href="index.php">[ Powrót do listy ]</a>
  </p>
</body>

</html>
```

Edycja samochodu

Edycja rekordu jest najbardziej skomplikowaną operacją z zaprezentowanych, gdyż wymaga najpierw wczytania rekordu o wybranym ID (podobnie jak w szczegółach samochodu), a następnie zaktualizowania rekordu istniejącego w bazie przy pomocy zapytania SQL *UPDATE*.



Edycja

localhost:81/samochody/edytuj.php?id=1

Marka

Model

Rok

Typ silnika

[\[Powrót do listy \]](#)

Kod podstrony:

```
<?php
```

```
session_start();
```

```
if (empty($_SESSION['zalogowany'])) {  
    header("Location: logowanie.php");  
    exit();  
}
```

```
if (isset($_GET['id'])) {  
    $id = (int) $_GET['id'];  
    $pdo = new PDO('mysql:host=localhost;dbname=test', 'root', '');  
  
    if (isset($_POST['zapisz'])) {  
        // obsługa edycji rekordu
```

```
$stmt = $pdo->prepare("UPDATE samochody SET marka = :marka, model = :model, rok = :rok,
typ_silnika = :typ_silnika WHERE id = :id");
$wynik = $stmt->execute([
    'marka' => $_POST['marka'],
    'model' => $_POST['model'],
    'rok' => $_POST['rok'],
    'typ_silnika' => $_POST['typ_silnika'],
    'id' => $id,
]);

if ($wynik == true) {
    header("Location: edytuj.php?id=$id&komunikat=1");
    exit();
} else {
    die("Operacja się nie powiodła: " . $pdo->errorInfo());
}
} else {
    // wczytanie danych samochodu z bazy
    $stmt = $pdo->prepare("SELECT * FROM samochody WHERE id = :id");
    $stmt->execute(['id' => $id]);
    $wiersz = $stmt->fetch();

    if (!$wiersz) {
        die("Nie znaleziono podanego samochodu!");
    }
}
} else {
    die("Nie podano ID");
}
?>
<!DOCTYPE html>
<html>

<head>
    <title>Edycja</title>
    <meta charset="utf-8">
</head>
```

```
<body>
  <?php if (($_GET['komunikat'] ?? null) == 1) : ?>
    <p style='color:red; font-weight:bold;'>Dane zostały zapisane.</p>
  <?php endif; ?>

  <form method="post" action="">
    <table>
      <tr>
        <td>Marka</td>
        <td><input type="text" name="marka" value="<?= $wiersz['marka'] ?? " ?>" /></td>
      </tr>
      <tr>
        <td>Model</td>
        <td><input type="text" name="model" value="<?= $wiersz['model'] ?? " ?>" /></td>
      </tr>
      <tr>
        <td>Rok</td>
        <td><input type="text" name="rok" value="<?= $wiersz['rok'] ?? " ?>" /></td>
      </tr>
      <tr>
        <td>Typ silnika</td>
        <td>
          <select name="typ_silnika">
            <option value="benzyna" <?= ($wiersz['typ_silnika'] ?? ") == 'benzyna' ? 'selected' : "
?>>benzyna</option>
            <option value="diesel" <?= ($wiersz['typ_silnika'] ?? ") == 'diesel' ? 'selected' : "
?>>diesel</option>
          </select>
        </td>
      </tr>
      <tr>
        <td colspan="2"><input type="submit" name="zapisz" value="Zapisz" /></td>
      </tr>
    </table>
  </form>
```

```
<p>
  <a href="index.php">[ Powrót do listy ]</a>
</p>
</body>

</html>
```

Usuwanie samochodu

Usuwanie jest najprostszą operacją, gdyż wymaga jedynie wywołania zapytania SQL *DELETE* (nie trzeba generować żadnego kodu wynikowego HTML).

```
<?php

session_start();

if(empty($_SESSION['zalogowany'])) {
    header("Location: logowanie.php");
    exit();
}
if(isset($_GET['id'])) {
    $pdo = new PDO('mysql:host=localhost;dbname=test', 'root', '');
    $stmt = $pdo->prepare("DELETE FROM samochody WHERE id = :id");
    $wynik = $stmt->execute(['id' => $_GET['id']]);

    if($wynik == true) {
        header("Location: index.php?komunikat=2");
    } else {
        echo "<p style='color:red; font-weight:bold;'>Usuwanie nie powiodło się.</p>";
    }
}
}
```

Refaktoryzacja

Jak widać w powyższych przykładach, we wszystkich plikach występują powtarzające się fragmenty kodu. Pierwszy z nich to start sesji i weryfikacji, czy użytkownik jest zalogowany. Drugi z nich to utworzenie instancji klasy PDO z parametrami dostępu,

Obie funkcjonalności można wyłączyć z plików w formie funkcji i umieścić je w zewnętrznym pliku, a następnie zaimportować w dotychczasowych plikach.

Plik *funkcje.php*:

```
<?php
```

```
define('DB_HOST', 'localhost');  
define('DB_NAME', 'test');  
define('DB_USERNAME', 'root');  
define('DB_PASSWORD', '');
```

```
session_start();
```

```
function polacz(): PDO
```

```
{  
    return new PDO(  
        sprintf('mysql:host=%s;dbname=%s', DB_HOST, DB_NAME),  
        DB_USERNAME,  
        DB_PASSWORD  
    );  
}
```

```
function sprawdzLogowanie(): void
```

```
{  
    if (empty($_SESSION['zalogowany'])) {  
        header("Location: logowanie.php");  
        exit();  
    }  
}
```


Import pliku funkcje w innych plikach i przykład wywołania (z *index.php*):

```
<?php

require_once 'funkcje.php';

sprawdzLogowanie();

$dane = [];
$pdo = polacz();
foreach ($pdo->query('SELECT * FROM samochody ORDER BY marka, model') as $wiersz) {
    $dane[] = $wiersz;
}
?>
```

Zadania

1. Proszę o uzupełnienie formularza dodawania samochodu (oraz obsługę w kodzie PHP) polami:
 - a. Pojemność (pole tekstowe),
 - b. liczba poduszek (lista rozwijalna z wartościami 1, 2, 4, 6, 8),
 - c. abs (pole typu radio),
 - d. esp (pole typu radio)
2. Proszę o wykonanie analogicznej czynności w formularzu edycji samochodu (należy pamiętać o poprawnym przepisaniu/zaznaczeniu wartości w polach formularza).
3. Proszę o dodanie potwierdzenia (np. w Javascript) przed operacją usunięcia rekordu.