SQL: Създаване на база данни и таблица

CREATE DATABASE test\_db;

USE test\_db;

CREATE TABLE registrations (

id INT AUTO\_INCREMENT PRIMARY KEY, -- Уникален идентификатор за всяка регистрация

name VARCHAR(255) NOT NULL, -- Име на потребителя (задължително поле)

email VARCHAR(255) NOT NULL, -- Имейл на потребителя (задължително и уникално поле)

message TEXT NOT NULL, -- Съобщение от потребителя (задължително поле)

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP -- Време на създаване на записа (по подразбиране текущото време)

);

Обяснение на SQL:

CREATE DATABASE test\_db; - Създава нова база данни с име test\_db.

USE test\_db; - Задава активната база данни test\_db, в която ще се създава таблицата.

Таблица registrations: Създава таблица за съхранение на регистрации с полета:

id: Уникален идентификатор за всяка регистрация, който се увеличава автоматично.

name: Име на потребителя, което е задължително.

email: Имейл на потребителя, който също е задължителен и трябва да бъде уникален.

message: Съобщение от потребителя.

created\_at: Поле за времето на създаване на записа, което се попълва автоматично с текущото време.

HTML: Формуляр за регистрация

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Регистрация</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<div class="form-container">

<!-- Формулярът изпраща данни към submit.php -->

<form action="submit.php" method="post">

<h1>Регистрация</h1>

<!-- Показване на съобщението, ако е зададено чрез GET параметър -->

<?php if (isset($\_GET['message'])): ?>

<p class="message"><?php echo htmlspecialchars($\_GET['message']); ?></p>

<?php endif; ?>

<label for="name">Име:</label>

<input type="text" id="name" name="name" required>

<label for="email">Имейл:</label>

<input type="email" id="email" name="email" required>

<label for="message">Съобщение:</label>

<textarea id="message" name="message" rows="4" required></textarea>

<button type="submit">Изпрати</button>

</form>

</div>

</body>

</html>

Обяснение на HTML:

<form action="submit.php" method="post"> - Формулярът изпраща данни чрез POST метода към submit.php.

<h1>Регистрация</h1> - Заглавие на формуляра.

<?php if (isset($\_GET['message'])): ?> - Условие за показване на съобщение, ако е зададено чрез GET параметър.

Полетата:

<label for="name">: Таг за етикет за името.

<input type="text" id="name" name="name" required>: Текстово поле за името, което е задължително.

<input type="email" id="email" name="email" required>: Поле за имейл, което трябва да бъде в правилен формат и също задължително.

<textarea id="message" name="message" rows="4" required></textarea>: Текстово поле за съобщение.

<button type="submit">: Бутон за изпращане на формуляра.

PHP: Обработка на данните от формата

<?php

// Конфигурация за връзка с базата данни

$host = "localhost";

$dbname = "test\_db";

$username = "root";

$password = "";

// Променливи за съобщения

$message = ""; // По подразбиране празно

try {

// Създаване на PDO връзка

$conn = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8", $username, $password);

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

// Проверка дали формата е изпратена

if ($\_SERVER["REQUEST\_METHOD"] === "POST") {

// Уверяваме се, че данните са подадени чрез формата

if (isset($\_POST['name'], $\_POST['email'], $\_POST['message'])) {

$name = htmlspecialchars($\_POST['name']);

$email = htmlspecialchars($\_POST['email']);

$user\_message = htmlspecialchars($\_POST['message']);

// SQL заявка за вмъкване на данните

$sql = "INSERT INTO registrations (name, email, message) VALUES (:name, :email, :message)";

$stmt = $conn->prepare($sql);

// Свързване на параметри

$stmt->bindParam(':name', $name);

$stmt->bindParam(':email', $email);

$stmt->bindParam(':message', $user\_message);

// Изпълнение на заявката

$stmt->execute();

$message = "Вашите данни бяха успешно запазени!";

}

}

} catch (PDOException $e) {

$message = "Грешка: " . $e->getMessage();

}

?>

Обяснение на PHP:

Конфигурация на базата данни: Определят се параметрите за връзка с MySQL сървъра.

PDO връзка: Използва се PDO за свързване с базата данни и изпълнение на заявки.

Обработка на POST данни: Проверява се дали данните са подадени от формата.

SQL заявка: Подготвя и изпълнява заявка за вмъкване на данни в таблицата.

Обработка на грешки: Ако има грешка, тя се хваща и се показва на потребителя.

CSS: Стилове за формуляра

body {

font-family: Arial, sans-serif;

background-color: #f4f4f9;

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

.form-container {

background: #fff;

padding: 20px;

border-radius: 8px;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

width: 300px;

}

form {

display: flex;

flex-direction: column;

}

form h1 {

text-align: center;

margin-bottom: 20px;

}

label {

margin-bottom: 5px;

font-weight: bold;

}

textarea {

resize: none;

height: 100px;

}

input, textarea {

margin-bottom: 15px;

padding: 10px;

border: 1px solid #ccc;

border-radius: 4px;

}

button {

background-color: #28a745;

color: #fff;

border: none;

padding: 10px;

border-radius: 4px;

cursor: pointer;

transition: background-color 0.3s;

}

button:hover {

background-color: #218838;

}

.message {

padding: 10px;

background-color: #d4edda;

color: #155724;

border: 1px solid #c3e6cb;

border-radius: 4px;

margin-bottom: 15px;

text-align: center;

}

Обяснение на CSS:

body - Целият фон на страницата е светъл, с центрирано съдържание.

.form-container - Стил за контейнера на формуляра с бяла основа, заоблени ръбове и сянка.

.message - Стил за съобщението, което се показва при успешна регистрация или грешка.