

Запись данных в базу

Создадим проект Магазин цветов.

Структура проекта

/flowershop

 /config

 config.php

 /public

 register.php

 submit_review.php

 contact.php

 index.php

База данных проекта

База состоит из трех таблиц: Отзывы, Пользователи и Сообщения

```
✓ CREATE TABLE reviews (  
    |     id INT AUTO_INCREMENT PRIMARY KEY,  
    |     reviewer_name VARCHAR(100) NOT NULL,  
    |     review TEXT NOT NULL,  
    |     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
    | );  
  
✓ CREATE TABLE users (  
    |     id INT AUTO_INCREMENT PRIMARY KEY,  
    |     username VARCHAR(50) NOT NULL,  
    |     email VARCHAR(100) NOT NULL UNIQUE,  
    |     password VARCHAR(255) NOT NULL  
    | );  
  
✓ CREATE TABLE messages (  
    |     id INT AUTO_INCREMENT PRIMARY KEY,  
    |     contact_name VARCHAR(255) NOT NULL,  
    |     message TEXT NOT NULL,  
    |     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
    | );
```

Главная страница

На главной странице самостоятельно подготовьте стандартную структуру с адаптивным меню, для доступа к страницам сайта.

Страница регистрация

- Страница содержит простую форму для регистрации.
- Валидность полей проверяется встроенными средствами Bootstrap.

```
<div class="container mt-5">
  <h2>Регистрация пользователя</h2>
  <form action="register.php" method="POST">
    <div class="mb-3">
      <label for="username" class="form-label">Имя пользователя</label>
      <input type="text" class="form-control" id="username" name="username" required>
    </div>
    <div class="mb-3">
      <label for="email" class="form-label">Электронная почта</label>
      <input type="email" class="form-control" id="email" name="email" required>
    </div>
    <div class="mb-3">
      <label for="password" class="form-label">Пароль</label>
      <input type="password" class="form-control" id="password" name="password" required>
    </div>
    <button type="submit" class="btn btn-primary">Зарегистрироваться</button>
  </form>
</div>
```

- Обработчик формы находится в этом же файле.

```
<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    include '../config/config.php';

    $username = $_POST['username'];
    $email = $_POST['email'];
    $password = password_hash($_POST['password'], PASSWORD_DEFAULT); // Хеширование пароля

    $sql = "INSERT INTO users (username, email, password) VALUES ('$username', '$email', '$password')";

    if (mysqli_query($connection, $sql)) {
        echo "<div class='alert alert-success mt-3'>Регистрация успешна!</div>";
    } else {
        echo "<div class='alert alert-danger mt-3'>Ошибка: " . mysqli_error($connection) . "</div>";
    }

    mysqli_close($connection);
}
?>
```

Построчное описание кода обработчика

<?php

- Открытие блока PHP для обработки данных формы.

if (\$_SERVER['REQUEST_METHOD'] == 'POST') {

- Проверка, был ли запрос методом POST (т.е. отправлена ли форма).

include '../config/config.php';

- Подключение файла конфигурации, содержащего параметры подключения к базе данных.

\$username = \$_POST['username'];

- Получение введенного имени пользователя из POST-запроса.

\$email = \$_POST['email'];

- Получение введенной электронной почты из POST-запроса.

```
$password = password_hash($_POST['password'], PASSWORD_DEFAULT); // Хеширование пароля
```

- Хеширование пароля с использованием функции password_hash, чтобы безопасно хранить его в базе данных.

```
$sql = "INSERT INTO users (username, email, password) VALUES ('$username', '$email', '$password')";
```

- Формирование SQL-запроса для вставки новых данных пользователя (имя, электронная почта, хешированный пароль) в таблицу users.

```
if (mysqli_query($connection, $sql)) {
```

- Выполнение SQL-запроса и проверка, успешно ли выполнен запрос.

```
echo "<div class='alert alert-success mt-3'>Регистрация успешна!</div>";
```

- Если запрос выполнен успешно, выводится сообщение об успешной регистрации.

```
} else {
```

- Начало блока else, который выполняется, если запрос не удался.

```
echo "<div class='alert alert-danger mt-3'>Ошибка: " . mysqli_error($connection) . "</div>";
```

- Вывод сообщения об ошибке с текстом ошибки, полученным с помощью mysqli_error.

```
}
```

- Заккрытие блока if/else.

```
mysqli_close($connection);
```

- Заккрытие подключения к базе данных после выполнения запроса.

```
}
```

```
?>
```

- Заккрытие блока PHP.

Страница отправки сообщения

- Страница содержит простую форму для отправки сообщения.
- Валидность полей проверяется встроенными средствами Bootstrap.

```

<div class="container mt-5">
  <h2>Контактная форма</h2>
  <form action="contact.php" method="POST">
    <div class="mb-3">
      <label for="contact_name" class="form-label">Ваше имя</label>
      <input type="text" class="form-control" id="contact_name" name="contact_name" required>
    </div>
    <div class="mb-3">
      <label for="message" class="form-label">Ваше сообщение</label>
      <textarea class="form-control" id="message" name="message" rows="3" required></textarea>
    </div>
    <button type="submit" class="btn btn-primary">Отправить сообщение</button>
  </form>
</div>

```

- Обработчик формы находится в этом же файле.

```

<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    include '../config/config.php';

    $contact_name = $_POST['contact_name'];
    $message = $_POST['message'];

    $sql = "INSERT INTO messages (contact_name, message) VALUES ('$contact_name', '$message')";

    if (mysqli_query($connection, $sql)) {
        echo "<div class='alert alert-success mt-3'>Сообщение успешно отправлено!</div>";
    } else {
        echo "<div class='alert alert-danger mt-3'>Ошибка: " . mysqli_error($connection) . "</div>";
    }

    mysqli_close($connection);
}
?>

```

Построчное описание кода обработчика

<?php

- Открывающий тег для PHP-кода. Всё, что находится между этим тегом и закрывающим (?>), будет интерпретироваться как PHP.

if (\$_SERVER['REQUEST_METHOD'] == 'POST') {

- Условие, проверяющее, был ли запрос к странице выполнен методом POST. Это означает, что данные формы были отправлены. Если условие истинно, выполняется содержимое блока.

include '../config/config.php';

- Подключение файла конфигурации базы данных, расположенного по указанному пути. В этом файле, вероятно, содержатся настройки для соединения с базой данных (например, хост, имя пользователя, пароль и имя базы данных).

\$contact_name = \$_POST['contact_name'];

- Извлечение значения поля contact_name из массива \$_POST, который содержит данные, отправленные через форму. Это значение будет сохранено в переменной \$contact_name.

\$message = \$_POST['message'];

- Аналогично, извлечение значения поля message и сохранение его в переменной \$message.

```
$sql = "INSERT INTO messages (contact_name, message) VALUES ('$contact_name', '$message')";
```

- Формирование SQL запроса для вставки данных в таблицу messages. Данные, полученные из формы, подставляются в запрос. Однако важно отметить, что данный вариант уязвим для SQL-инъекций, если не использовать проверку и экранирование вводимых данных.

```
if (mysqli_query($connection, $sql)) {
```

- Выполнение SQL-запроса с помощью функции mysqli_query(). Если запрос выполнен успешно (т.е. сообщение добавлено в базу данных), тогда выполняется следующий блок кода.

```
echo "<div class='alert alert-success mt-3'>Сообщение успешно отправлено!</div>";
```

- Если запрос был успешным, выводится сообщение об успешной отправке сообщения, обернутое в HTML-класс Bootstrap для стилизации (сообщение об успехе).

```
} else {
```

- Если выполнение SQL-запроса завершилось ошибкой, выполняется следующий блок кода.

```
echo "<div class='alert alert-danger mt-3'>Ошибка: " . mysqli_error($connection) . "</div>";
```

- Выводится сообщение об ошибке, которое включает текст самой ошибки, возвращаемой функцией mysqli_error(). Это может помочь в отладке, показывая, что именно пошло не так.

```
}
```

- Заккрытие блока условия для обработки результата запроса.

```
mysqli_close($connection);
```

- Заккрытие соединения с базой данных после завершения всех операций с ней. Это важно для освобождения ресурсов.

```
}
```

```
?>
```

- Заккрытие блока условия для метода POST, а также закрывающий тег PHP.

Страница отзыва

- Страница содержит простую форму для отправки отзыва. Форма практически идентична предыдущей.
- Валидность полей проверяется встроенными средствами Bootstrap.

```

<div class="container mt-5">
  <h2>Добавить отзыв</h2>
  <form action="submit_review.php" method="POST">
    <div class="mb-3">
      <label for="reviewer_name" class="form-label">Ваше имя</label>
      <input type="text" class="form-control" id="reviewer_name" name="reviewer_name" required>
    </div>
    <div class="mb-3">
      <label for="review" class="form-label">Ваш отзыв</label>
      <textarea class="form-control" id="review" name="review" rows="3" required></textarea>
    </div>
    <button type="submit" class="btn btn-primary">Отправить отзыв</button>
  </form>
</div>

```

- Обработчик формы находится в этом же файле.

```

<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    include '../config/config.php';

    $reviewer_name = $_POST['reviewer_name'];
    $review = $_POST['review'];

    $sql = "INSERT INTO reviews (reviewer_name, review) VALUES ('$reviewer_name', '$review')";

    if (mysqli_query($connection, $sql)) {
        echo "<div class='alert alert-success mt-3'>Отзыв успешно добавлен!</div>";
    } else {
        echo "<div class='alert alert-danger mt-3'>Ошибка: " . mysqli_error($connection) . "</div>";
    }

    mysqli_close($connection);
}
?>

```