

Университет ИТМО

Факультет «Инфокоммуникационных технологий»

Лабораторная работа №2
«Разработка web-форм»

Выполнил:

Кузнецов Никита Сергеевич

Группа №K33212

Проверил:

Марченко Елена Вадимовна

Санкт-Петербург
2023

Цель работы:

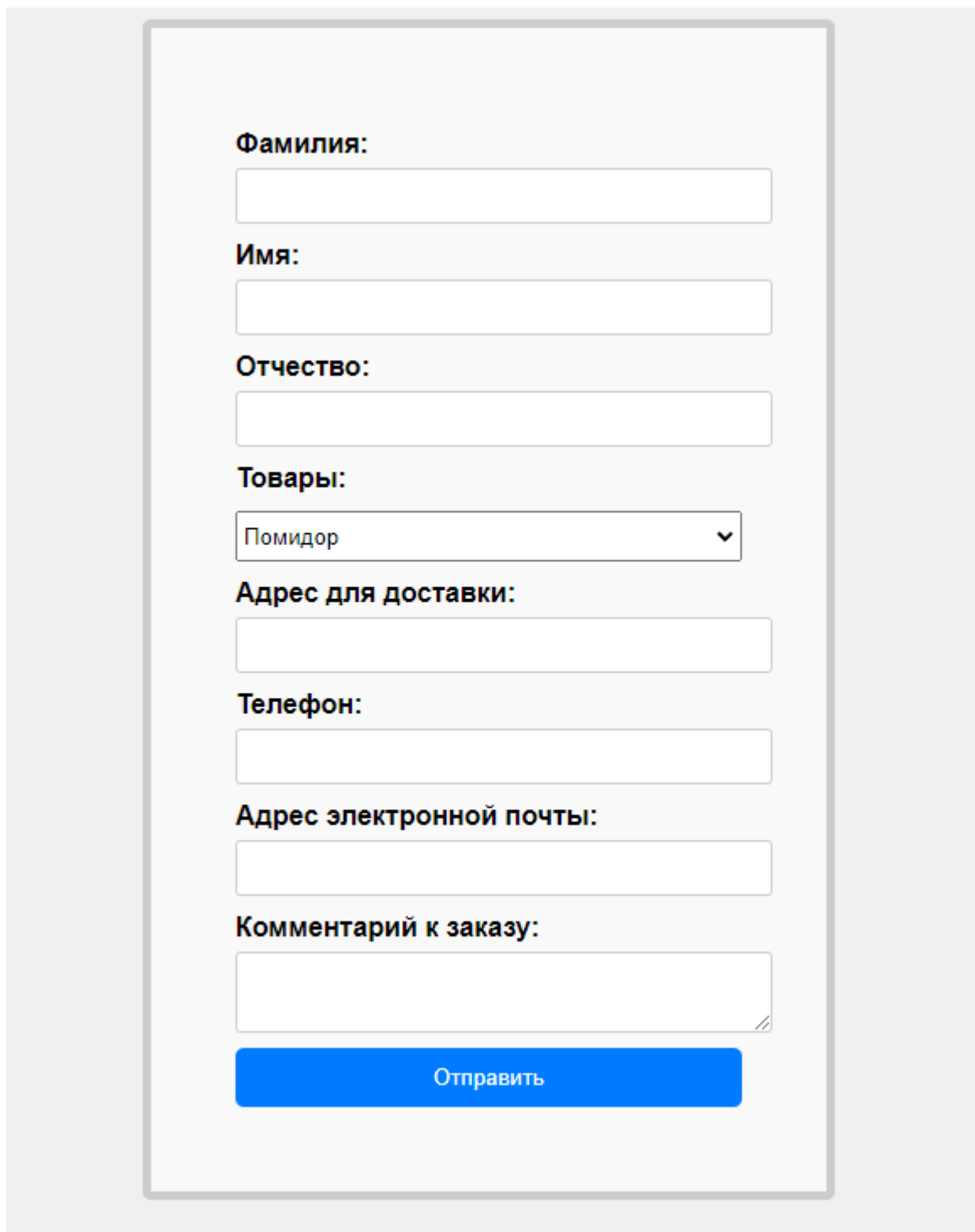
С помощью средств языка PHP разработать две страницы HTML с вводом логина и пароля и оформлением заказа, введенную пользователем информацию в формах сохранить в базу данных MySQL, а также запустить локальный сервер для получившихся страниц.

Ход работы:

1 задание: разработать веб-страницу, на которой пользователь может оставить данные о себе – фамилия, имя, отчество, адрес для физической доставки товара, телефон, адрес электронной почты. Товары покупатель выбирает из списка (реализовать меню), предусмотреть отдельную позицию, в которой пользователь может оставить комментарий по заказу. Данные, введенные в форму, обрабатываются php скриптом и заносятся в таблицу БД MySQL Server. Разработать структуру таблицы, создать таблицу и скрипт php.

Для решения данного задания было представлено три файла: страница HTML с кодом формы (приложение 1), стили CSS для всего проекта (приложение 2) и файл PHP (приложение 3), выполняемый при отправке формы.

В файле order.html (приложение 1) расписан тэг <form>, отправляющий POST-запрос к файлу save_data.php, которые обрабатывает данные формы и сохраняет их в базу данных как новый заказ. Конечный интерфейс, представляемый пользователю, выглядел как окно с полями для ввода различных данных (текстовых, выбора из списка) и кнопкой «Отправить», которая отправляла форму на исполнение PHP (рис. 1).



The image shows a web form for sending an order, enclosed in a light gray border. The form contains several input fields and a submit button. The labels for the fields are in Russian. The 'Товары' (Goods) field is a dropdown menu with 'Помидор' (Tomato) selected. The 'Отправить' (Send) button is blue with white text.

Фамилия:

Имя:

Отчество:

Товары:

Помидор ▼

Адрес для доставки:

Телефон:

Адрес электронной почты:

Комментарий к заказу:

Отправить

Рисунок 1 – Внешний вид формы отправки заказа

В файле `save_data.php` принимаются данные из формы, создается соединение с базой данных, создается таблица заказов (если её еще не

существует), заказ сохраняется в базу, а пользователю показывается сообщение об успешности или неуспешности операции, а также кнопка, которая может вернуть его назад (рис. 2).

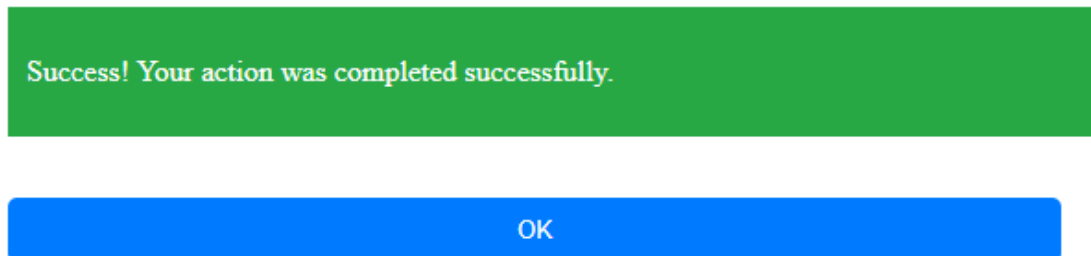


Рисунок 2 – Интерфейс, показывающийся пользователю при успешной операции добавления заказа.

2 задание: в движке wordpress используется форма для авторизации пользователей – пользователь вводит логин и пароль. Логин и хэш от пароля записываются в таблицу БД MySQL. Необходимо модифицировать php скрипт, чтобы дополнительно логин и пароль записывались в отдельную новую таблицу в БД MySQL. Предусмотреть два варианта записи: а) пароль записывается в исходном виде, б) инвертируются биты байт пароля (ноль меняется на единицу, единица меняется на ноль).

Для решения данного задания необходимо было создать дополнительный HTML (приложение 4) и PHP файл (приложение 5), отвечающих за обработку логина и пароля пользователя, и сохранения их в базу данных.

Для пользовательского интерфейса ввода логина и пароля был написан файл index.html (приложение 4) в корневой директории, который и будет отправной точкой для пользователей, вошедших на сайт. Он выглядит как

обычная форма с вводом логина и пароля (рис. 3), которые потом отправляются на обработку в файл login.php.

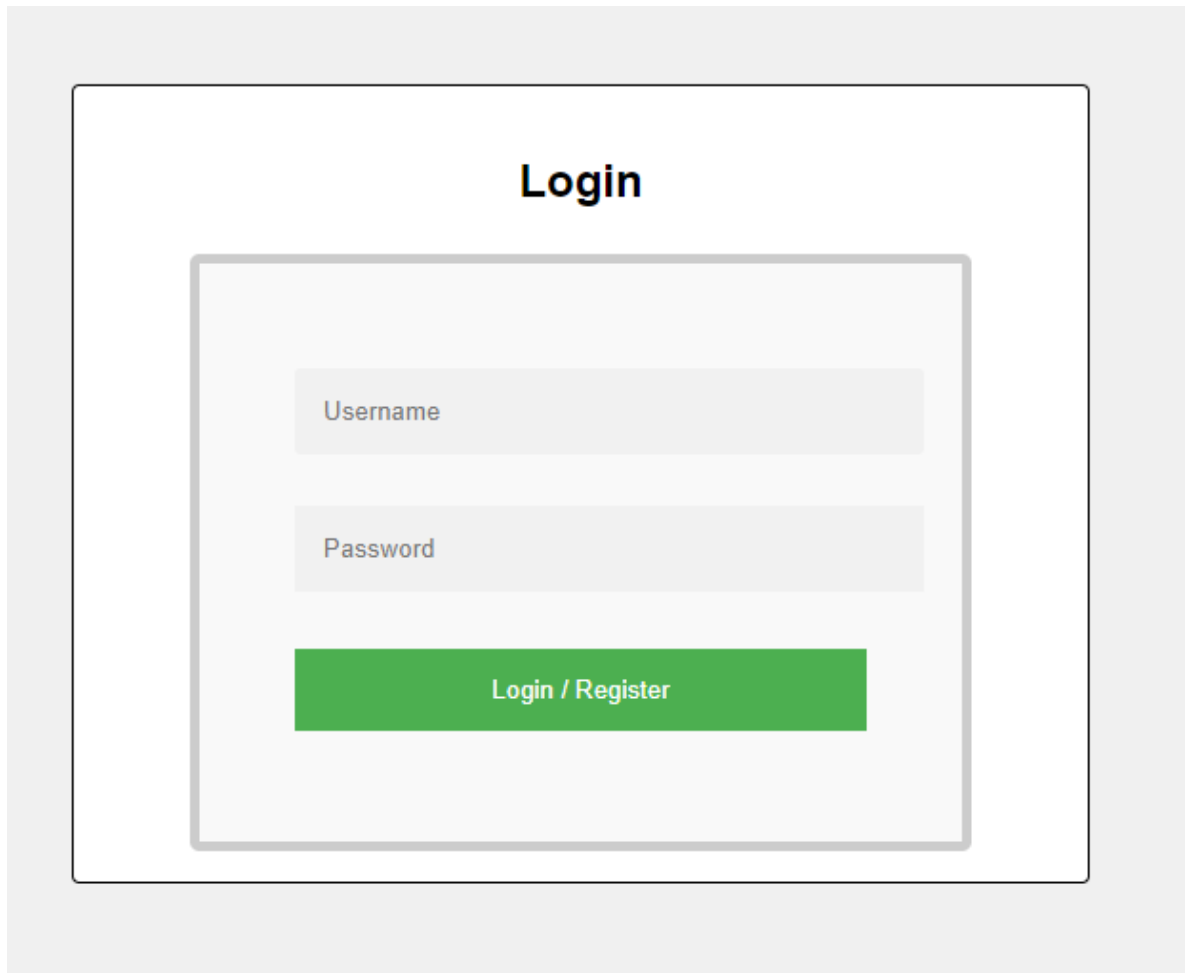
The image shows a web form titled "Login" in a bold, black font. Below the title is a light gray rectangular box containing two input fields. The first field is labeled "Username" in a light gray font. The second field is labeled "Password" in a light gray font. Below these fields is a green rectangular button with the text "Login / Register" in white. The entire form is centered within a light gray background.

Рисунок 3 – Внешний вид формы для входа/регистрации пользователя.

Для обработки пользовательского ввода в форме был разработан скрипт для PHP (приложение 5), который выполняет следующие действия:

- 1) Соединяется с базой данных;
- 2) Создает таблицу пользователей с логином, хешированным, обычным и инвертированным паролем;
- 3) Получает информацию из формы;

- 4) Проверяет существует ли уже пользователь с таким логином и паролем;
- 5) Если пользователь существует – его перебрасывает на страницу с заказом с всплывающим окном «Login success!» (рис. 4);
- 6) Если пользователя не существует – он создается в системе, а его пароль хэшируется и его биты инвертируются.
- 7) Затем пользователя перебрасывает на страницу с заказом с всплывающим окном «Register success!» (рис. 5);

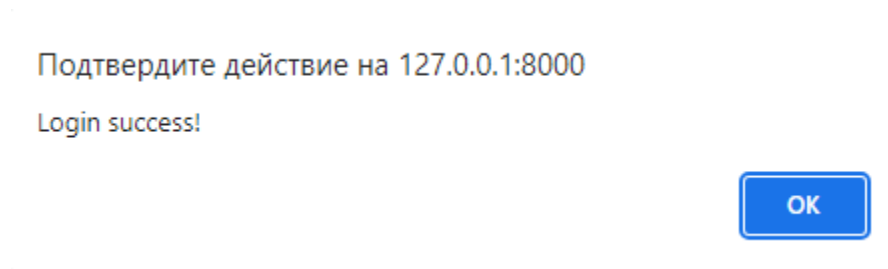


Рисунок 4 – Успешный вход пользователя в систему.

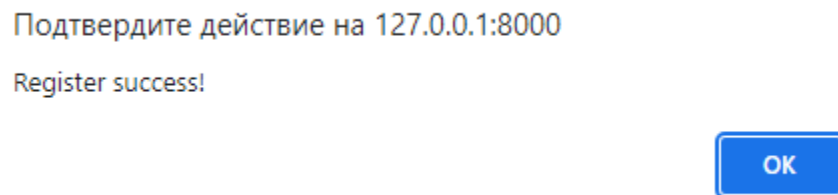


Рисунок 5 – Успешная регистрация пользователя в системе.

Задание 3: выбрать среду разработки (язык программирования) на ваше усмотрение, используя его написать web-сервер. В программе предусмотреть возможность указать порт, на котором будет работать сервер. При обращении вида `http://127.0.0.1:888/` где 888 – заданный порт, сервер должен вернуть содержимое файла `index.html`, находящегося в том же каталоге, что и исполняемый файл написанной программы-сервера.

Для решения данной задачи был выбран стандартный способ развертывания веб-сервера с помощью PHP, а именно команды `php -S 127.0.0.1:8000`, где 127.0.0.1 – это локальный IP-адрес, а 8000 – порт, на котором будет веб-сервер.

Также, для правильного отображения страницы входа по корневому адресу, а также появления внятной структуры веб-приложения файл `index.html` был перенесен в корневую директорию, а файл `order.html` был перенесен в директорию `static`. Аналогично, все скрипты PHP и файл MySQL базы данных были перенесены в директорию `scripts` (рис. 6).

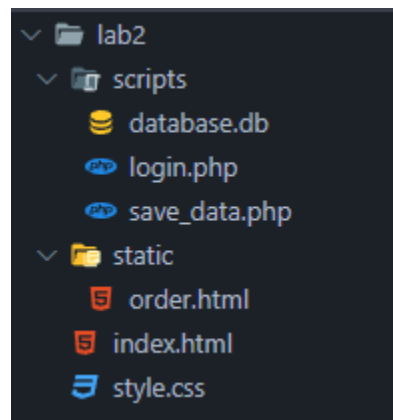


Рисунок 6 – Финальная структура веб-приложения.

После всех изменений и рефакторинга кода была запущена команда `php -S 127.0.0.1:8000` и запущен PHP Development Server, который обрабатывал и принимал все запросы, приходящие со страницы от пользователя (рис. 7).

```
$ php -S 127.0.0.1:8000
[Wed Oct 18 00:39:35 2023] PHP 8.2.11 Development Server (http://127.0.0.1:8000) started
[Wed Oct 18 00:39:41 2023] 127.0.0.1:65217 Accepted
[Wed Oct 18 00:39:41 2023] 127.0.0.1:65217 [200]: GET /
[Wed Oct 18 00:39:41 2023] 127.0.0.1:65218 Accepted
[Wed Oct 18 00:39:41 2023] 127.0.0.1:65217 Closing
[Wed Oct 18 00:39:41 2023] 127.0.0.1:65218 [200]: GET /style.css
[Wed Oct 18 00:39:41 2023] 127.0.0.1:65218 Closing
[Wed Oct 18 00:39:48 2023] 127.0.0.1:65220 Accepted
[Wed Oct 18 00:39:48 2023] 127.0.0.1:65220 [200]: POST /scripts/login.php
[Wed Oct 18 00:39:48 2023] 127.0.0.1:65220 Closing
[Wed Oct 18 00:39:48 2023] 127.0.0.1:65221 Accepted
[Wed Oct 18 00:39:49 2023] 127.0.0.1:65221 [200]: GET /static/order.html
[Wed Oct 18 00:39:49 2023] 127.0.0.1:65221 Closing
[Wed Oct 18 00:39:49 2023] 127.0.0.1:65222 Accepted
[Wed Oct 18 00:39:49 2023] 127.0.0.1:65223 Accepted
[Wed Oct 18 00:39:49 2023] 127.0.0.1:65222 [200]: GET /style.css
[Wed Oct 18 00:39:49 2023] 127.0.0.1:65222 Closing
[Wed Oct 18 00:39:57 2023] 127.0.0.1:65223 [200]: POST /scripts/save_data.php
[Wed Oct 18 00:39:57 2023] 127.0.0.1:65223 Closing
[Wed Oct 18 00:39:57 2023] 127.0.0.1:65225 Accepted
[Wed Oct 18 00:39:58 2023] 127.0.0.1:65226 Accepted
```

Рисунок 7 – Запущенный веб-сервер на PHP.

Вывод:

В ходе работы было разработано небольшое веб-приложение на PHP и HTML + CSS, реализующее логику входа и регистрации пользователей, а также оставления заказов на продукты. Все эти сущности были сохранены в локальную базу данных MySQL в файле с расширением .db. В конце работы был запущен веб-сервер через PHP и реализована базовая структура веб-сервиса, который может отправлять и принимать HTTP-запросы от клиента к серверу и от сервера к клиенту.

Приложение 1. HTML файл с кодом формы заказа продуктов (order.html).

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Форма</title>
  <link rel="stylesheet" href="../style.css">
</head>
<body>
  <form action="../scripts/save_data.php" method="post">

    <label for="surname">Фамилия:</label>
    <input type="text" id="surname" name="surname">
    <label for="name">Имя:</label>
    <input type="text" id="name" name="name">
    <label for="patronymic">Отчество:</label>
    <input type="text" id="patronymic" name="patronymic">

    <label for="items">Товары:</label>
    <select id="items" name="items">
      <option value="item1">Помидор</option>
      <option value="item2">Огурец</option>
      <option value="item3">Перец</option>
      <option value="item4">Брокколи</option>
      <option value="item5">Картофель</option>
      <option value="item6">Морковь</option>
      <option value="item7">Лук</option>
    </select>

    <label for="delivery_address">Адрес для доставки:</label>
    <input type="text" id="delivery_address" name="delivery_address">
    <label for="phone">Телефон:</label>
    <input type="text" id="phone" name="phone">
    <label for="email">Адрес электронной почты:</label>
    <input type="text" id="email" name="email">
    <label for="comment">Комментарий к заказу:</label>
    <textarea id="comment" name="comment"></textarea>
    <input type="submit" value="Отправить">

  </form>
</body>
</html>
```

Приложение 2. CSS файл с кодом для стилей всего веб-приложения (style.css)

```
body {
    font-family: Arial, sans-serif;
    background-color: #f0f0f0;
}

form {
    width: 300px;
    margin: 0 auto;
    padding: 50px;
    border: 5px solid #ccc;
    border-radius: 5px;
    background-color: #f9f9f9;
}

label {
    display: block;
    margin-top: 10px;
    font-weight: bold;
}

select {
    width: 100%;
    margin-top: 10px;
    height: 30px;
}

input[type="text"], textarea {
    width: 100%;
    padding: 8px;
    border: 1px solid #ccc;
    border-radius: 3px;
    margin-top: 5px;
}

input[type="submit"] {
    display: block;
    width: 100%;
    padding: 10px;
    color: white;
    background-color: #007BFF;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    margin-top: 5px;
}

input[type="submit"]:hover {
    background-color: #0056b3;
}
```

```
}

.login-container {
  width: 500px;
  padding: 16px;
  background: white;
  margin: 0 auto;
  margin-top: 100px;
  border: 1px solid black;
  border-radius: 4px;
}

.login-container h2 {
  text-align: center;
  margin-bottom: 24px;
}

.login-container input {
  width: 100%;
  padding: 15px;
  margin: 5px 0 22px;
  border: none;
  background: #f1f1f1;
}

.login-container button {
  background-color: #4CAF50;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
}

.login-container button:hover {
  opacity: 0.8;
}
```

Приложение 3. PHP файл для обработки запроса со страницы с формой заказа (save_data.php)

```
<?php
$db = new SQLite3('database.db');

$db->exec("CREATE TABLE IF NOT EXISTS orders (
    id INTEGER PRIMARY KEY,
    surname TEXT,
    name TEXT,
    patronymic TEXT,
    items TEXT,
    delivery_address TEXT,
    phone TEXT,
    email TEXT,
    comment TEXT
)");

$surname = $_POST['surname'];
$name = $_POST['name'];
$patronymic = $_POST['patronymic'];
$items = $_POST['items'];
$delivery_address = $_POST['delivery_address'];
$phone = $_POST['phone'];
$email = $_POST['email'];
$comment = $_POST['comment'];

$query = "INSERT INTO orders (
    surname,
    name,
    patronymic,
    items,
    delivery_address,
    phone,
    email,
    comment
) VALUES (
    '$surname',
    '$name',
    '$patronymic',
    '$items',
    '$delivery_address',
    '$phone',
    '$email',
    '$comment'
)";
$result = $db->exec($query);

$results = $db->query("SELECT * FROM orders");

$db->close();
?>
```

```
<?php
if ($result) {
    ?>
    <div style="
        background-color: #28a745;
        color: white;
        padding: 10px;
        margin-bottom: 10px;
        width: 30%;
    ">
        <p>Success! Your action was completed successfully.</p>
    </div>

    <?php
} else {
    ?>
    <div style="
        background-color: #dc3545;
        color: white;
        padding: 10px;
        margin-bottom: 10px;
        width: 30%;
    ">
        <p>Error! Something went wrong. Please try again.</p>
    </div>
    <?php
}
?>

<br>
<button style="
    display: block;
    width: 30%;
    padding: 10px;
    color: white;
    background-color: #007bff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    margin-top: 5px;"
    onclick="history.go(-1);"
>
OK
</button>
```

Приложение 4. HTML файл для страницы входа/регистрации пользователя (index.html)

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css" href="style.css">
  <title>Login Page</title>
</head>
<body>
  <div class="login-container">
    <h2>Login</h2>
    <form action="scripts/login.php" method="post">
      <input type="text" name="username" placeholder="Username" required>
      <input type="password" name="password" placeholder="Password" required>
      <button type="submit">Login / Register</button>
    </form>
  </div>
</body>
</html>
```

Приложение 5. PHP файл для регистрации или подтверждения входа пользователя в веб-приложение (login.php)

```
<?php

$db = new SQLite3('database.db');

$db->exec("CREATE TABLE IF NOT EXISTS users (
    id INTEGER PRIMARY KEY,
    login TEXT,
    hashed_password TEXT,
    plain_password TEXT,
    inverse_password TEXT
)");

// функция меняющая 1 и 0 местами
function bitnot($bin) {
    $not = "";
    for ($i = 0; $i < strlen($bin); $i++)
    {
        if($bin[$i] == 0) { $not .= '1'; }
        if($bin[$i] == 1) { $not .= '0'; }
    }
    return $not;
}

function invert_password($pass) {
    // получаем биты
    $bytes = unpack('C*', $pass);
    $binary = '';
    foreach ($bytes as $byte) {
        $binary .= str_pad(decbin($byte), 8, '0', STR_PAD_LEFT);
    }
    // меняем 1 и 0 местами
    $inverted_binary = bitnot($binary);
    // декодируем обратно в строку
    $length = strlen($inverted_binary);
    $inverted_password = '';
    for ($i = 0; $i < $length; $i += 8) {
        $byte = bindec(substr($inverted_binary, $i, 8));
        $inverted_password .= chr($byte);
    }
    return $inverted_password;
}

function check_user($login, $password, $db) {
    $result = $db->query("SELECT * FROM users WHERE login = '$login'");
    if (!$result) return false;

    if ($row = $result->fetchArray()) {
        $hashed_password = $row["hashed_password"];
        $correct = password_verify($password, $hashed_password);
    }
}
```

```

        if ($correct) {
            return true;
        }
        return false;
    }
    return false;
}

function redirect_next($message, $next_page) {
    echo "<script type='text/javascript'>alert('$message');
window.location.href='$next_page';</script>";
}

$success_message = "Error! Something went wrong.";
$login = $_POST['username'];
$password = $_POST['password'];

$user_exists = check_user($login, $password, $db);
echo $user_exists;
if ($user_exists) {
    redirect_next("Login success!", "../static/order.html");
} else {
    $inverse_password = invert_password($password);
    $hashed_password = password_hash($password, PASSWORD_BCRYPT);

    $query = "INSERT INTO users (
        login, plain_password, inverse_password, hashed_password
    ) VALUES (
        '$login', '$password', '$inverse_password', '$hashed_password'
    )";

    $result = $db->query($query);
    redirect_next("Register success!", "../static/order.html");
}
$db->close();

?>

```