XTКафедра Информатики

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 100 | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|  | 90 |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 80 |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 70 |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 60 |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 50 |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 40 |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 30 |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 20 |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 10 |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 0 |  |  |  |  |  |  |  |  |  |  |  |  |

**ОТЧЕТ**

|  |
| --- |
| по лабораторной работе №4 |
| «Авторизация пользователя» |
|  |

|  |
| --- |
| по дисциплине **Сетевые сервисы обработки информации в ОТС** |
|  |

|  |
| --- |
| 1306.547408.000 ПЗ |
| (обозначение документа) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Группа |  |  | Фамилия И.О. | Подпись | Дата | Оценка |
| СТС-407 |  |
|  |  |
| Студент | | | Гараев Д.Н. |  |  |  |
| Консультант | | | Минасов Ш. М. |  |  |  |
| Принял | | |  |  |  |  |

Уфа – 2021 г.

**Содержание**

[Введение 3](#_Toc69327022)

[Ход работы 4](#_Toc69327023)

[Заключение 7](#_Toc69327024)

[Список литературы 8](#_Toc69327025)

[Приложение 1 9](#_Toc69327026)

[Приложение 2 11](#_Toc69327027)

[Приложение 3 12](#_Toc69327028)

[Приложение 4 15](#_Toc69327029)

# Введение

Целью лабораторной работы является изучения порядка создания сайта с авторизацией и регистрацией пользователей.

В рамках данной работы необходимо разработать процедуру авторизации для доступа к прописанной базе данных.

При авторизации необходимо предусмотреть добавление новых пользователей, также ошибки такого характера, как: незаполненные поля обязательных данных, ошибки при регистрации, переходы между страницами.

# Ход работы

Для начала необходимо создать главную страницу авторизации, где пользователю нужно будет ввести логин и пароль для того, чтобы получить доступ к базе данных о телескопах. Также если новый пользователь хочет зарегистрироваться в системе, то необходимо нажать на ссылку “Регистрация”.

Ниже представлена страница авторизации (Рисунок 1).

Изображение выглядит как текст, монитор, ноутбук, компьютер

Автоматически созданное описание

Рисунок 1 - Главная страница авторизации

После того, как уже зарегистрированный пользователь введет все данные, он должен нажать на кнопку «Войти», после чего откроется страница с базой данных о телескопах которая была выполнена в лабораторной работе №3.

Если же какие-то поля останутся незаполненными, то выйдет предупреждение «Заполните это поле».

Также если данные будут введены неверно, то будет выведено соответствующее предупреждение.

Далее рассмотрим страницу для регистрации новых пользователей. Эта страница открывается при нажатии на ссылку «Регистрация». Ниже представлено открывшееся окно.

Изображение выглядит как текст, монитор, ноутбук, внутренний

Автоматически созданное описание

Рисунок 2 - Регистрация новых пользователей.

Если какие-то из полей останутся незаполненными, то регистрация не завершится и покажется ошибка пользователю.

После прохождения авторизации пользователю будут доступны данные о полумягких телескопах(Рисунок 3).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3 - База данных

Также для того, чтобы зайти в систему, можно ввести из уже введенных в скрипте учетных записей, а именно: login:Admin password:Admin, login:sts07, password:sts407-077.

# Заключение

В результате выполнения лабораторной работы №4 был проделан процесс по созданию сайта с авторизацией и регистрацией пользователей.

В рамках данной работы разработаны процедуры авторизации для доступа к прописанной базе данных о телескопах.

При авторизации предусмотрено добавление новых пользователей, также ошибки такого характера, как: незаполненные поля обязательных данных, ошибки при регистрации, переходы между страницами.

Авторизацию может проделать любой посетитель написанного сайта.

# Список литературы

1. РНР: настольная книга программиста /Александр Мазуркевич, Дмитрий Еловой. – Мн.: Новое знание, 2003. – 480 с.: ил.
2. Разработка Web-приложений на РНР и MySQL: Пер. с

англ./Лаура Томсон, Люк Веллинг. - 2-е изд., испр. – СПб: ООО

«ДиаСофтЮП», 2003. – 672 с.

1. Дейтел Х.М. Как программировать для Internet & WWW /

Х.М. Дейтел, П.Дж. Дейтел, Т.Р. Нието; Пер. с англ. Е.Е. Клименкова,

Ю.А. Левчука, Я.А. Урджумова и др.; Под ред. А.В. Козлова. – М.:

БИНОМ, 2002. – 1184 с.

1. Вейтман В. Программирование для WEB. Руководство разработчика: Учеб.пособие. – М.; СПб.; Киев: Диалектика, 2000. – 368с.
2. Козырев А.А. Информационные технологии в экономике и

управлении: Учебник / А.А. Козырев. – СПб: Издательство Михайлова В.А., 2001. – 360 с.

1. Грир Т. Сети Интранет / Пер. с англ. – М.: Русская редакция,

2000. – 368с.

1. Петров В.Н. Информационные системы: Учеб. пособие для

вузов / В.Н. Петров. – СПб: ПИТЕР, 2003. – 688 с.

1. Вин Дж. Искусство web - дизайна: Самоучитель / Пер. с англ.

О. Кузнецовой. – СПб.: Питер, 2002. – 224 с.

# Приложение 1

SignIn.php

<!DOCTYPE html>  
<lang="ru">  
<head>  
 <meta charset="utf-8">  
 <meta name="viewport" content="with-device-width, initial-scale = 1.0">  
 <title>Авторизация</title>  
</head>  
<body >  
  
<?php require "../../../parts/header.php"?>  
  
<?php  
  
session\_start();  
  
require\_once '../../../Script/database\_connection.php';  
  
$error\_message = $\_REQUEST['error\_message'];  
  
// Если пользователь уже зашел, то кидаем на главную страницу  
if (!isset($\_SESSION['user\_id'])) {  
  
// Если уже отправлен пост запрос с именем пользователя, то смотрим  
if (isset($\_POST['username'])) {  
 // получаем логин и пароль  
 $username = $link->real\_escape\_string(trim($\_REQUEST['username']));  
 $password = $link->real\_escape\_string(trim($\_REQUEST['password']));  
  
 // Ищем пользователя в бд  
 $query = sprintf("SELECT id ,login, password FROM users " .  
 " WHERE login = '%s' AND password = '%s';",  
 $username, crypt($password, $username));  
  
 $results = mysqli\_query($link, $query);  
 //Если он есть то сохраняем в сессии его ид, иначе выводим ошибку  
 if (mysqli\_num\_rows($results) == 1) {  
 $result = mysqli\_fetch\_array($results);  
 $user\_id = $result['id'];  
 $\_SESSION['user\_id'] = $user\_id;  
 $\_SESSION['login'] = $username;  
  
 header("Location: ../main.php");  
 exit();  
 } else {  
 $error\_message = "Комбинация логин/пароль не правильна. Проверьте ваши введенные данные";  
 }  
} ?>  
  
<!-- BODY -->  
<body>  
<link rel="stylesheet" href="auth.css" type="text/css">  
<div class="body-content">  
 <div class="module">  
 <h1>Войти в аккаунт</h1>  
 <form class="form" action="signIn.php" method="post" enctype="multipart/form-data" autocomplete="off">  
 <div class="alert alert-error"><?php if (isset($error\_message)) echo $error\_message; ?></div>  
 <input type="text" placeholder="Логин пользователя" name="username" required/>  
 <input type="password" placeholder="Пароль" name="password" required/>  
 </br>  
 <a href="signUp.php"> <p class="">Регистрация</p></a>  
 </br>  
 <input type="submit" value="Войти" name="register" class="btn btn-block btn-primary"/>  
 </form>  
 </div></div>  
<?php  
} else {  
 // Now handle the case where they're logged in  
 // redirect to another page, most likely show\_user.php  
 header("Location: ../main.php");  
}  
?>  
  
<?php require "../../../parts/footer.php"?>  
</body>  
</html>

# Приложение 2

signOut.php

<?php  
  
session\_start();  
  
$\_SESSION['user\_id'] = null;  
session\_commit();  
  
header("Location: signIn.php");

# Приложение 3

signUp.php

<?php session\_start();?>  
<!DOCTYPE html>  
<html lang="ru">  
<head>  
 <meta charset="utf-8">  
 <meta name="viewport" content="with-device-width, initial-scale = 1.0">  
 <title>Регистрация</title>  
</head>  
<body >  
  
<?php require "../../../parts/header.php"?>  
<?php  
  
require\_once "../../../Script/database\_connection.php";  
//создаем пользователя если есть хотя бы логин и пароль  
if (isset($\_REQUEST['username']) | isset($\_REQUEST['password'])) {  
 $username = trim($\_REQUEST['username']);  
 $password = trim($\_REQUEST['password']);  
 $email = trim($\_REQUEST['email']);  
 $first\_name = trim($\_REQUEST['first\_name']);  
 $last\_name = trim($\_REQUEST['second\_name']);  
 $middle\_name = trim($\_REQUEST['middle\_name']);  
 $address = trim($\_REQUEST['address']);  
 $confirm\_password = trim($\_REQUEST['confirm\_password']);  
  
 $checkUserQuery = sprintf("SELECT login FROM users WHERE login = '%s'", $link->real\_escape\_string($username));  
 $result = $link->query($checkUserQuery);  
  
 if ($password != $confirm\_password) {  
 $error\_message = "Пароли не совпадают";  
 } else if ($result->num\_rows >= 1) {  
 $error\_message = "Пользователь с таким логином уже существует";  
 } else {  
 $insert\_sql = sprintf("INSERT INTO users " .  
 "( login, password, email, first\_name, last\_name, middle\_name)" .  
 "VALUES ('%s', '%s', '%s', '%s', '%s', '%s');",  
 $link->real\_escape\_string($username),  
 $link->real\_escape\_string(crypt($password, $username)),  
 $link->real\_escape\_string($email),  
 $link->real\_escape\_string($first\_name),  
 $link->real\_escape\_string($last\_name),  
 $link->real\_escape\_string($middle\_name),);  
 // Insert the user into the database  
 $result = $link->query($insert\_sql);  
  
 if ($result) {  
 $\_SESSION['user\_id'] = $link->insert\_id;  
 $\_SESSION['login'] = $username;  
 header("Location: ../3/main.php");  
 exit();  
 } else {  
 $error\_message = "Произошла непредвиденная ошибка";  
 }  
 }  
}  
  
?>  
<link rel="stylesheet" href="auth.css" type="text/css">  
<div class="body-content">  
 <div class="module">  
 <h1>Зарегистрируйте новый аккаунт</h1>  
 <form class="form" action="signUp.php" method="post" enctype="multipart/form-data" autocomplete="off">  
 <div class="alert alert-error"><?php if (isset($error\_message)) echo $error\_message; ?></div>  
 <input type="text" placeholder="Логин пользователя" name="username" required/>  
 <input type="email" placeholder="Email" name="email" required/>  
 <input type="password" placeholder="Пароль" name="password" autocomplete="new-password"  
 onkeydown="clearClicked(***event***)" onkeypress="pressedButton(***event***)" required/>  
 <input type="password" placeholder="Повторите пароль" name="confirm\_password" autocomplete="new-password"  
 required/>  
 <input type="text" placeholder="Имя"onkeypress="return ( ***event***.charCode === 32 || (***event***.keyCode > 1071 && ***event***.keyCode < 1104 ) ||  
 (***event***.keyCode > 1039 && ***event***.keyCode < 1104));" name="first\_name" required/>  
 <input type="text" placeholder="Фамилия" onkeypress="return ( ***event***.charCode === 32 || (***event***.keyCode > 1071 && ***event***.keyCode < 1104 ) ||  
 (***event***.keyCode > 1039 && ***event***.keyCode < 1104));" name="second\_name" required/>  
 <input type="text" placeholder="Отчество" onkeypress="return ( ***event***.charCode === 32 || (***event***.keyCode > 1071 && ***event***.keyCode < 1104 ) ||  
 (***event***.keyCode > 1039 && ***event***.keyCode < 1104));" name="middle\_name" required/>  
 <input type="submit" value="Зарегистрироваться" id="register" name="register"  
 class="btn btn-block btn-primary"/>  
 </form>  
 </div>  
</div>  
<div id="footer"></div>  
</body>  
<script>  
 let ***clicked*** = {};  
  
 function clearClicked(event) {  
 if (event.keyCode === 8) {  
 ***clicked*** = {};  
 }  
 }  
  
 function pressedButton(event) {  
  
 ***clicked***[event.keyCode] = isNaN(***clicked***[event.keyCode]) ? 1 : ***clicked***[event.keyCode] + 1;  
 let flag = false;  
 for (key in ***clicked***) {  
 if (***clicked***[key] > 3) {  
 flag = true;  
 }  
 }  
  
 if (flag) {  
 $(".alert").text("Более трёх повторяющихся символов");  
 $("#register")[0].disabled = true;  
 } else {  
 $(".alert").text("");  
 $("#register")[0].disabled = false;  
 }  
 }  
</script>  
</html>  
<?php require "../../../parts/footer.php"?>  
</body>  
</html>

# Приложение 4

Auth.css

\*{  
 margin:0;  
 padding:0;  
}  
  
h1 {  
 font-size: 2em;  
 font-family: "Core Sans N W01 35 Light";  
 font-weight: normal;  
 margin: .67em 0;  
 display: block;  
}  
  
.avatar {  
 margin: 10px 0 20px 0;  
}  
  
.module{  
 position:relative;  
 top:10%;  
 height:65%;  
 width:450px;  
 margin-left:auto;  
 margin-right:auto;  
}  
  
::-moz-selection {  
 background: #19547c;  
}  
::selection {  
 background: #19547c;  
}  
input::-moz-selection {  
 background: #037db6;  
}  
input::selection {  
 background: #037db6;  
}  
  
body{  
 color: #fff;  
 background-color:#f0f0f0;  
 font-family:helvetica;  
 background:url('http://clevertechie.com/img/bnet-bg.jpg') #0f2439 no-repeat center top;  
}  
  
.body-content{  
 position:relative;  
 top:20px;  
 height:700px;  
 width:800px;  
 margin-left:auto;  
 margin-right:auto;  
 background: transparent;  
}  
  
input[type="text"],  
input[type="password"],  
input[type="email"]  
{  
 height:34px;  
 width:100%;;  
 display: inline-block;  
 vertical-align: middle;  
 padding-left: 10px **!important**;  
 margin-top: 3px ;  
 margin-bottom: 10px;  
 font-size: 15px;  
 line-height: 20px;  
 border: 1px solid rgba(255, 255, 255, 0.3);  
 background-color: rgba(0, 0, 0, 0.5);  
 color: rgba(255, 255, 255, 0.7);  
 -moz-box-sizing: border-box;  
 box-sizing: border-box;  
 border-radius: 2px;  
 offset: 10px;  
}  
  
input[type="text"],  
input[type="password"],  
input[type="email"] {  
 -webkit-appearance: none;  
 -moz-appearance: none;  
 -ms-appearance: none;  
 appearance: none;  
 -webkit-transition: background-position 0.2s, background-color 0.2s, border-color 0.2s, box-shadow 0.2s;  
 transition: background-position 0.2s, background-color 0.2s, border-color 0.2s, box-shadow 0.2s;  
}  
input[type="text"]:hover,  
input[type="password"]:hover,  
input[type="email"]:hover {  
 border-color: rgba(255, 255, 255, 0.5);  
 background-color: rgba(0, 0, 0, 0.5);  
 color: rgba(255, 255, 255, 0.7);  
}  
input[type="text"]:focus,  
input[type="password"]:focus,  
input[type="email"]:focus {  
 border: 2px solid;  
 border-color: #1e5f99;  
 background-color: rgba(0, 0, 0, 0.5);  
 color: #ffffff;  
}  
.btn {  
 text-overflow: ellipsis;  
 white-space: nowrap;  
 overflow: hidden;  
 margin: 3px 0;  
 padding: 6px 20px;  
 font-size: 15px;  
 line-height: 20px;  
 height: 34px;  
 background-color: rgba(0, 0, 0, 0.15);  
 color: #00aeff;  
 border: 1px solid rgba(255, 255, 255, 0.15);  
 box-shadow: 0 0 rgba(0, 0, 0, 0);  
 border-radius: 2px;  
 -webkit-transition: background-color 0.2s, box-shadow 0.2s, background-color 0.2s, border-color 0.2s, color 0.2s;  
 transition: background-color 0.2s, box-shadow 0.2s, background-color 0.2s, border-color 0.2s, color 0.2s;  
}  
.btn:active {  
 padding: 7px 19px 5px 21px;  
}  
.btn:hover,  
.btn:focus {  
 background-color: rgba(0, 0, 0, 0.25);  
 color: #ffffff;  
 border-color: rgba(255, 255, 255, 0.3);  
 box-shadow: 0 0 rgba(0, 0, 0, 0);  
}  
.btn:active {  
 background-color: rgba(0, 0, 0, 0.15);  
 color: rgba(255, 255, 255, 0.8);  
 border-color: rgba(255, 255, 255, 0.07);  
 box-shadow: inset 1.5px 1.5px 3px rgba(0, 0, 0, 0.5);  
}  
.btn-primary {  
 background-color: #098cc8;  
 color: #ffffff;  
 border: 1px solid transparent;  
 box-shadow: 0 0 rgba(0, 0, 0, 0);  
 border-radius: 2px;  
 -webkit-transition: background-color 0.2s, box-shadow 0.2s, background-color 0.2s, border-color 0.2s, color 0.2s;  
 transition: background-color 0.2s, box-shadow 0.2s, background-color 0.2s, border-color 0.2s, color 0.2s;  
 background-image: -webkit-linear-gradient(top, #0f9ada, #0076ad);  
 background-image: linear-gradient(to bottom, #0f9ada, #0076ad);  
 border: 0;  
 box-shadow: 0 1px 1px rgba(0, 0, 0, 0.3), 0 0 0 1px rgba(255, 255, 255, 0.15) inset;  
}  
.btn-primary:hover,  
.btn-primary:focus {  
 background-color: #21b0f1;  
 color: #ffffff;  
 border-color: transparent;  
 box-shadow: 0 0 rgba(0, 0, 0, 0);  
}  
.btn-primary:active {  
 background-color: #006899;  
 color: rgba(255, 255, 255, 0.7);  
 border-color: transparent;  
 box-shadow: inset 1.5px 1.5px 3px rgba(0, 0, 0, 0.5);  
}  
.btn-primary:hover,  
.btn-primary:focus {  
 background-image: -webkit-linear-gradient(top, #37c0ff, #0097dd);  
 background-image: linear-gradient(to bottom, #37c0ff, #0097dd);  
}  
.btn-primary:active {  
 background-image: -webkit-linear-gradient(top, #006ea1, #00608d);  
 background-image: linear-gradient(to bottom, #006ea1, #00608d);  
 box-shadow: 1px 1px 2px rgba(0, 0, 0, 0.6) inset, 0 0 0 1px rgba(255, 255, 255, 0.07) inset;  
}  
.btn-block {  
 display: block;  
 width: 100%;  
 padding-left: 0;  
 padding-right: 0;  
}  
  
.alert {  
 -moz-box-sizing: border-box;  
 box-sizing: border-box;  
 padding: 4px 20px 4px 20px;  
 font-size: 13px;  
 line-height: 20px;  
 margin-bottom: 20px;  
 text-shadow: none;  
 position: relative;  
 background-color: #272e3b;  
 color: rgba(255, 255, 255, 0.7);  
 border: 1px solid #000;  
 box-shadow: 0 0 0 1px #363d49 inset, 0 5px 10px rgba(0, 0, 0, 0.75);  
}  
.alert-error {  
 color: #f00;  
 background-color: #360e10;  
 box-shadow: 0 0 0 1px #551e21 inset, 0 5px 10px rgba(0, 0, 0, 0.75);  
}  
.alert:empty{  
 display: none;  
}