МИНИСТЕРСТВО ОБРАЗОВАНИЯ и науки РЕСПУБЛИКИ БАШКОРТОСТАН

Государственное автономное ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

УФИМСКИЙ КОЛЛЕДЖ СТАТИСТИКИ, ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Руководство программиста

По МДК.09.02. Оптимизация веб-приложений

Выполнил студент группы 21Веб-1

Карпов Д. В

Проверил(а) преподаватель информатики

и программирования

Дмитриева Е.К

2024

Руководство программиста для Веб-Приложения «Список дел» (Todo list).

1. Введение

Это веб-приложение построено на основе клиент-серверной архитектуры, где серверная часть реализована с использованием PHP, а фронтенд использует JavaScript и jQuery для отправки AJAX-запросов. Приложение также использует Bootstrap для стилизации интерфейса.

Todo list – это веб-приложение для создания заметок, в нем пользователь может создавать свои заметки и по мере выполнения отмечать как выполненные.

Основные функции:

* регистрация и авторизация пользователей;
* создание, редактирование и удаление заметок;
* Сортировка, фильрация и поиск на клиентской стороне
* отметки выполненных заметок.

Технологический стек:

Фронтенд: HTML, CSS, Bootstrap.

Бэкенд: PHP 8.1, MySQL, JavaScript, jQuery.

Сервер: Apache\_2.4-PHP\_8.0 - 8.1

2. Архитектура приложения

2.1 Общая схема

TodoList использует классическую архитектуру клиент-сервер.

MySQL— база данных для хранения информации о пользователях, постах и комментариях.

2.2 Взаимодействие компонентов

* Пользователь взаимодействует с интерфейсом через браузер, отправляя запросы к базе данных.
* Серверная часть обрабатывает запросы и взаимодействует с MySQL для получения или записи данных.
* Ответ возвращается в формате HTML

2.3 Основные модули данных

Connect.php модуль для подключения к базе данных MySQL

reg-db.php обработка запросов регистрации.

auth-db.php обработка запросов авторизации

delete-note.php обработка запросов для удаления заметки

edit-note.php управление статусом заметки

update\_data.php обработка запросов для редактирования

1. Разработка фронтенда
   1. Установка окружения

Для разработки фронтенда вам потребуется запустить локальный сервер OpenServer и подключить необходимые библиотеки:

<!--bootstrap-->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"integrity="sha384QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"

integrity="sha384YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"crossorigin="anonymous"></script>

<!-- jQuery -->

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

* 1. Структура проекта

LISTDELS/

index.php страница для регистрация, главная страница

login.php страница для входа

logout.php страница для выхода

search.php страница для поиска

database/

db.php - страница для подключения к БД

reg-db.php - страница для обработки запросов регистрации.

auth-db.php - страница для обработки запросов авторизации

* 1. Стилизация

Стилизация компонентов реализована с помощью Bootstrap и файла style.css.

LISTDELS/design/css/main.css

*.*navbar *{*

    background-color*:* *#6c63ff;*

    color*:* black*;*

    width*:* 100%*;*

    transition*:* all 0.6s ease-in-out*;*

*}*

*.*navbar-dark *.*navbar-nav *.*nav-link *{*

    color*:* white !important*;*

*}*

body *{*

    height*:* 750px*;*

    display*:* flex*;*

    flex-direction*:* column*;*

    align-items*:* center*;*

    gap*:* 18px*;*

*}*

1. Разработка бэкенда

4.1 Подключение к базе данных(db.php)

Файл `db.php`, содержащий функции подключения к базе данных MySQL:

<?php

$host = “localhost”;

$host = “root”;

$host = “”;

$host = “todolist”;

$connect = mysqli\_connest($host, $root, $pass\_db, $db\_name);

4.2 Обработка данных на сервере

Пример обработки запроса на добавление заметки в `tasks/new-note.php`:

<?php

session\_start();

require “../database/db.php”;

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

$title = trim($\_POST["title"]);

$desc = trim($\_POST["desc"]);

$user = $\_SESSION["user"];

if(empty($title) || empty($desc)) {

echo “<script>

alert(‘Пожалуйста, заполните все поля.’);

location.href=’../’;

</script>;

exit;

}

$sql = "INSERT INTO `tasks`(

`user\_id`,

`title`, `description`)

VALUES (

'$user\_id',

'$title',

'$desc')";

$result = mysqli\_query($connect, $sql);

if ($result) {

echo “<script>

alert(‘Заметка добавлена!’);

location.href=’../’;

</script>;

} else {

echo “<script>

alert(‘Произошла ошибка. Попробуйте ещё раз.’);

location.href=’../’;

</script>;

}

}

4.3 Аутентификация (auth-db.php)

При аутентификации происходит пароля:

<?php

session\_start();

require\_once "db.php";

$login = $\_POST["login"];

$password = $\_POST["password\_hash"];

if(empty($login) || empty($password)){

echo “<script>alert(‘Заполните все поля!’); location.href=’../login.php’;</script>

exit;

} else {

$sql = “SELECT \* FROM users WHERE username = ‘$login’ AND password\_hash = ‘$password’;

$result = mysqli\_query($connect, $sql);

if(mysqli\_num\_rows($result)>0){

$res = mysqli\_fetch\_assoc($result);

$\_SESSION[“user”] = $res[‘id’];

echo “<script>alert(‘Вы вошли как пользователь $login’); location.href=’../index.php’;</script>”;

} else {

echo “<script>alert(‘Неверный логин или пароль’); location.href=’../login.php’;</script>”;

}

}

4.4 Структура базы данных

Пример таблиц базы данных:

CREATE TABLE User(

Id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(60)

password\_hash VARCHAR(40)

)

CREATE TABLE tasks(

Id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id: INT

title VARCHAR(100)

description VARCHAR(150)

is\_completed TINYINT

created\_at TIMESTAMP

updated\_at TIMESTAMP

)

5. Тестирование

5.1 Тестирование на клиенте.

1. Убедитесь, что сервер корректно возвращает HTML ответы, которые обрабатываются на клиенте.

5.2 Тестирование на сервере

1. Проверьте работу скриптов PHP с помощью тестовых данных.

2. Убедитесь, что обработка ошибок на сервере работает корректно.

6. Развертывание

6.1 Настройка сервера

1. Убедитесь, что сервер поддерживает PHP 8.1 и MySQL.

2. Настройте виртуальный хост в Apache\_2.4-PHP\_8.0 - 8.1 для работы с проектом.

6.2 Развертывание на хостинге

1. Скопируйте файлы проекта на сервер.

2. Настройте файл конфигурации базы данных (`db.php`) с корректными данными подключения.

3. Импортируйте структуру базы данных (SQL-файлы) на сервер.

7. Безопасность

7.1 Защита пароля происходит с помощью хеширования (password\_hash()):

$password\_hash = password\_hash($pass, PASSWORD\_DEFAULT);

$sql = mysqli\_query($con, "INSERT INTO users (username, password\_hash) VALUES ('$login', '$password\_hash')");