Руководство программиста для веб-приложения "ToDoList"

1. Введение

ToDoList — это веб-приложение предназначено для управления задачами пользователей. Оно должно быть интуитивно понятным, простым в использовании и адаптивным для разных устройств.

Основные функции:

* регистрация и авторизация пользователей;
* создание, редактирование и удаление задач;
* отметка выполненных задач с возможностью изменения;
* отображение списка задач на главной странице с возможностью сортировки и фильтрации;
* реализация поиска без перезагрузки страницы.

Технологический стек:

* Фронтенд: HTML, CSS, JavaScript, jQuery, Bootstrap;
* Бэкенд: PHP 8+, MySQL;
* Запросы: AJAX для взаимодействия клиента с сервером.

2. Архитектура приложения

2.1 Общая схема

ToDoList использует PHP для реализация серверной логики, а также для регистрации, авторизации, добавления, редактирования, удаления и получения задач. Обработка запросов и взаимодействие с базой данных SQL также реализуется с помощью PHP.

2.2 Взаимодействие компонентов

* пользователь взаимодействует с интерфейсом через браузер, отправляя запросы к базе данных;
* серверная часть обрабатывает запросы и взаимодействует с sql (phpmyadmin) для получения или записи данных;
* результаты операций возвращаются в виде ресурсов.

2.3 Основные модели данных

- users: представляет зарегистрированного пользователя.

- `id`: строка, уникальный идентификатор пользователя.

- `passwordHash`: строка, хешированный пароль пользователя.

- `username`: строка, публичное имя пользователя.

- Post: представляет запись в блоге.

- `id`: строка, уникальный идентификатор задачи.

- `user\_id`: ссылка на автора (связь с моделью Users).

- `title`: заголовок поста.

- `description`: основной текст задачи.

- `is\_complete`: информация о выполнении задачи.

- `createdAt`: дата публикации.

- `updateAt`: дата редактирования.

3. Разработка фронтенда

3.1 Установка окружения

Необходимо запустить локальный сервер и подключить библиотеки (рисунок 1).

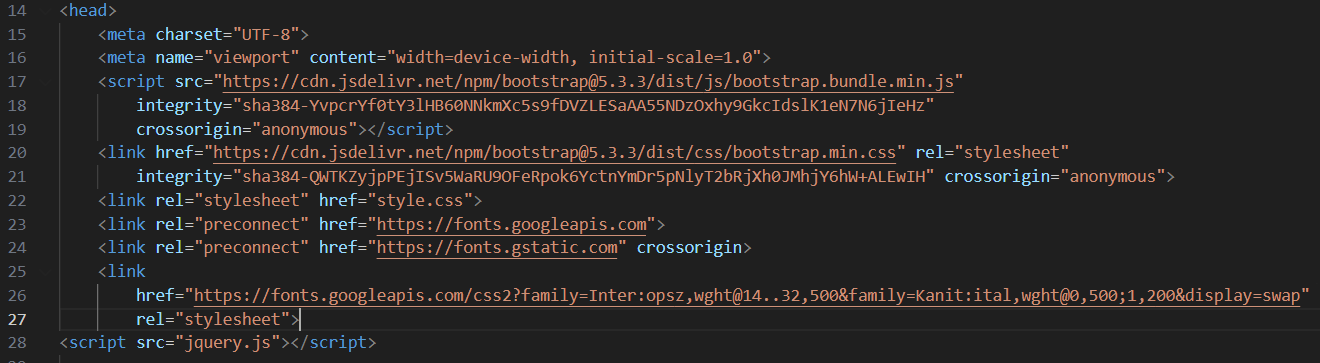


Рисунок 1 – Подключение библиотек.

3.2 Структура проекта

```plaintext

/db

signin\_db.php связь с базой данных для регистрации

signup\_db.php связь с базой данных для авторизации

create-tsak.php компонент для добавления задач

delete.php компонент для удаления задач

edit.php компонент для редактирования задач

update.php компонент для изменения статуса задачи

//

connect.php связь с базой данных sql

header.php файл с шапкой сайта

index.php главная страница сайта

sign-out.php выход из личного кабинета

personal.php страница личного кабинета

sign-up.php страница регистрации

sign-in.php страница авторизации

/css

style.css стили страниц

/js

jquery.js библиотека jquery

```

3.3 Взаимодействие через AJAX

Пример отправки AJAX-запроса при смене статуса задачи (рисунок 2).

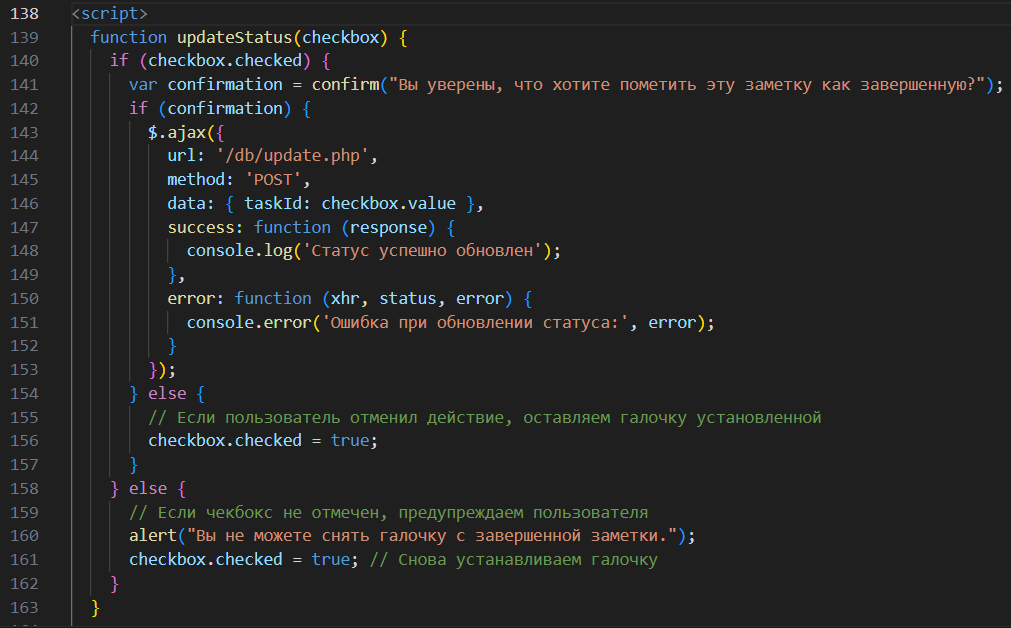


Рисунок 2 – Пример отправки AJAX-запроса.

3.4 Стилизация

Стилизация компонентов реализована с помощью Bootstrap и кастомных CSS-модулей. В проекте подключен Bootstrap, но для дополнительных правок используются CSS:

```css

style.css

.menu-bar {

    display: flex;

    gap: 18px;

    width: 80%;

    justify-content: center;

}

```

4. Разработка бэкенда

4.1 Подключение к базе данных

Подключение к базе данных в файле connect.php:

$con = mysqli\_connect("localhost","root","","todolist");

4.2 Обработка данных на сервере

Пример обработки добавления новой заметки (рисунок 3).



Рисунок 3 – Добавление заметки.

4.3 Авторизация пользователя

Авторизация пользователя (рисунок 4).

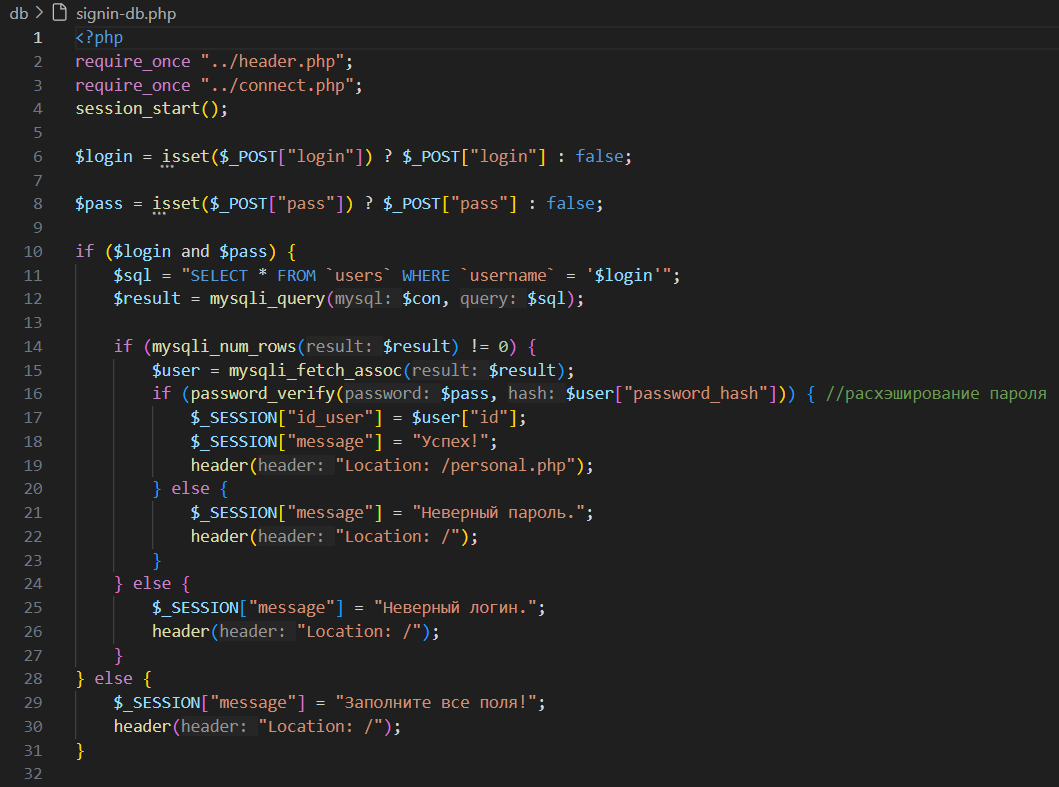


Рисунок 4 – Авторизация.

4.4 Структура базы данных

Структура таблиц базы данных (рисунок 5).

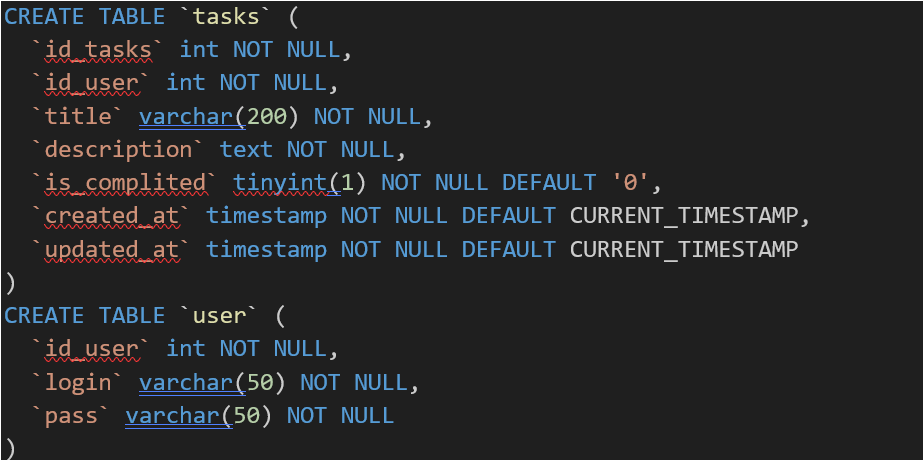


Рисунок 5 – Структура базы данных.

5. Тестирование

5.1 Тестирование на клиенте

* Проверьте корректность работы форм и их отправку через AJAX;
* Убедитесь, что сервер корректно возвращает JSON-ответы, которые обрабатываются на клиенте.

5.2 Тестирование на сервере

* Проверьте работу скриптов PHP с помощью тестовых данных;
* Убедитесь, что обработка ошибок на сервере работает корректно (например, при отсутствии обязательных полей).

6. Развертывание

6.1 Настройка сервера

Убедитесь, что сервер поддерживает PHP 8+ и MySQL.

6.2 Развертывание на хостинге

* Скопируйте файлы проекта на сервер;
* Настройте файл конфигурации базы данных (`connect.php`) с корректными данными подключения;
* Импортируйте структуру базы данных (SQL-файлы) на сервер.