РЕФЕРАТ

Курсовая работа по предмету базы данных, на тему “Создание WEB- приложения с использованием СУБД”, студента 3-го курса Эрдынеевой Номин Дамдиновны.

Курсовая работа изложена на 23 страницах машинописного текста. Состоит из: Введения, 4-х глав, заключения и списка использованных источников. Содержит вставки кода на языках HTML, CSS, PHP, JavaScript.

Цель работы: Приобретение навыков разработки WEB-приложений с использованием СУБД.

Ключевые слова: Сайт, WEB-приложение, база данных, разработка.

СОДЕРЖАНИЕ

Оглавление

[ВВЕДЕНИЕ 3](#_Toc153402286)

[1.1 Описание предметной области 5](#_Toc153402287)

[1.2 Постановка задачи 5](#_Toc153402288)

[2.1 Промежуточные данные (таблицы БД) 6](#_Toc153402289)

[ГЛАВА 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ 8](#_Toc153402290)

[ГЛАВА 4. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 22](#_Toc153402291)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 23](#_Toc153402292)

ВВЕДЕНИЕ

По данным на август 2023 года, в интернете насчитывается 1,88 млрд сайтов, и их количество продолжает расти. Несмотря на возможность создания WEB-сайта с помощью различных онлайн-конструкторов, спрос на специалистов в области WEB-разработки только повышается. Чаще всего для малого бизнеса достаточно готовых решений, но автоматизация бизнес- процессов и сложные WEB-приложения требуют участия WEB-программиста. Благодаря существующим технологиям, порог входа в профессию не так высок. С помощью различных фреймворков, даже начинающий специалист может создавать качественные макеты, а опытный программист экономит время и упрощает процесс разработки. Кроме того, навыки WEB-программирования полезны WEB-дизайнерам, поскольку повышают эффективность их взаимодействия с разработчиками и расширяют профессиональные возможности. Все вышесказанное обуславливает актуальность данной работы.

Объект исследования курсовой работы - разработка WEB-сайта. Предмет исследования - разработка Web-приложения для работы с базами данных.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* Проанализировать возможные методы разработки WEB-сайтов;
* Создать дизайн-макет страницы;
* Разработать функционал WEB-сайта;
* Разработать базу данных для хранения записей;

Практическая значимость работы заключается в возможности использования результатов в личной профессиональной деятельности.

Теоретическая значимость курсовой работы заключается в закреплении теоретических знаний, полученных в ходе обучения.

Структура курсовой работы обусловлена целью и задачами исследования и включает в себя: введение, четыре главы, заключение и список источников.

Во введении описывается тема курсовой работы, цель, её актуальность, поставленные задачи, предмет и объект исследования.

В первой главе представлен анализ задачи, включающий описание предметной области и постановку задачи

Во второй главе представлен анализ данных, описывающий базовые понятия.

В третьей главе описывается программная реализация и представляется итоговый дизайн и функционал WEB-сайта.

В четвёртой главе представлено руководство пользователя.

**ГЛАВА 1. АНАЛИЗ ПОСТАВЛЕННОЙ ЗАДАЧИ**

* 1. Описание предметной области

Предметная область данной курсовой работы заключается в создании сайта для фирмы по продаже запчастей, позволяющего работать с базой данных этой конторы. Определимся с функционалом сайта.

Функционал сайта - это то, что делает WEB-ресурс полезным, что превращает его из обычного источника информации в полезную программу.

Программный модуль, который планируется сделать для будущего WEB- приложения: таблица с информацией из базы данных, операции Create, Update, Read, Delete (создать, обновить, прочитать, удалить. Далее CRUD), поиск по столбцу и значению, сортировка по возрастанию/убыванию, валидация данных авторизации в базу данных.

После того, как был определён дизайн WEB-приложения и его функциональные возможности, можно приступить к его проектированию.

* 1. Постановка задачи

Задачей данного проекта является разработка приложения для сети интернет с использованием соответствующих инструментов реализации.

Деятельность сайта заключается в том, что пользователи могут подключиться к определенной базе данных PostgreSQL, выполнить операции CRUD, выполнить поиск по ключевым словам и сортировать данные по возрастанию/убыванию.

В процессе реализации приложения необходимо обратить внимание на наличие гибкого дизайна, удобной системой реализации CRUD, простота в освоении программы.

**ГЛАВА 2. АНАЛИЗ ДАННЫХ**

* 1. Промежуточные данные (таблицы БД)

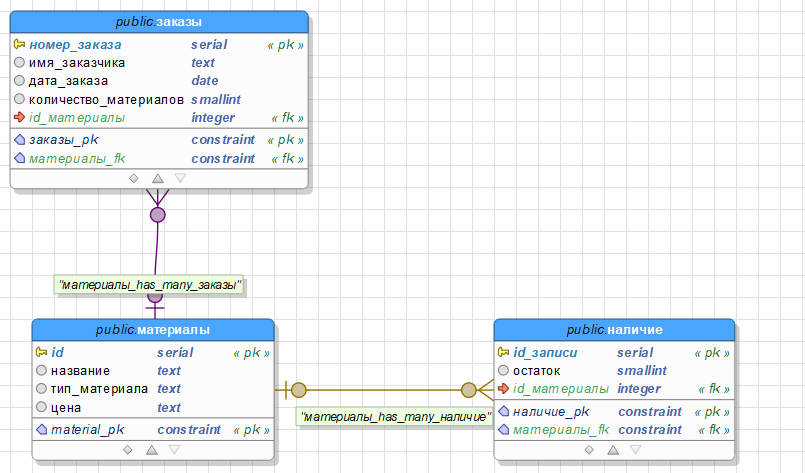
Проектирование БД является очень важным этапом, от которого зависят последующие этапы разработки приложения.

К базовым понятиями модели БД «сущность - связь» относятся: сущности, связи между ними и их атрибуты (свойства).

Сущность - любой конкретный или абстрактный объект в рассматриваемой предметной области (в реляционной БД каждой сущности назначается таблица).

Атрибут - это свойство сущности в предметной области. Его наименование должно быть уникальным для конкретного типа сущности. В реляционной БД атрибуты хранятся в полях таблиц.

Связь - взаимосвязь между сущностями в предметной области. Связи представляют собой соединения между частями БД (в реляционной БД - это соединение между записями таблиц).



1. Сущность materials имеет 4 атрибута:
   1. id – первичный ключ;
   2. Название материала;
   3. Тип материала;
   4. Цена;
2. Сущность stock имеет 2 атрибута:
   1. Id\_записи – первичный ключ;
   2. Остаток;
3. Сущность orders имеет 4 атрибутов:
   1. Id\_order – первичный ключ;
   2. Имя заказчика;
   3. Дата заказа;
   4. Количество материалов;

ГЛАВА 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Созданная программа состоит из следующих страниц:

* index.php
* dashboard.php
* search.php

При создании сайта использовался язык PHP и JavaScript, язык SQL для запросов в базу данных.

Каждая задача приложения была вынесена в отдельный скрипт, названный по сути своего действия.

Особое внимание уделялось универсальности работы функций интерфейсной стороны – для динамической загрузки данных и отказа от перезагрузок страницы, что негативно влияет на пользовательский опыт. Такой же подход применялся и к написанию кода серверной части - для повышения «гибкости» проекта и в целом эффективности его масштабирования.

Опишем детально каждый из скриптов серверной части и функций интерфейсной стороны, что используются в программе.

Скрипт **login.php** производит подключение к базе данных, сохраняет в памяти нужную для подключения информацию:

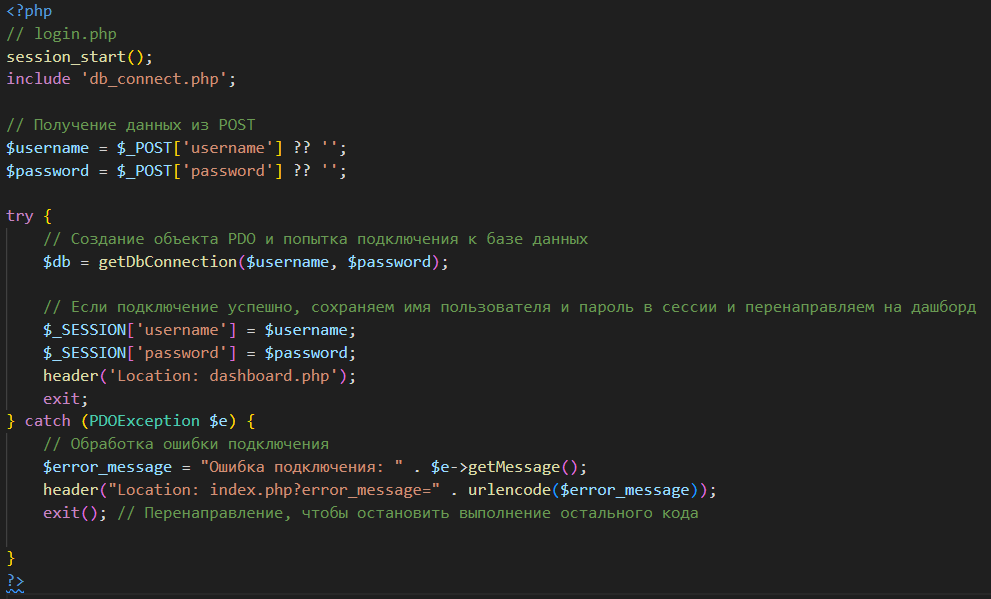


Рис. 1. login.php

На рис. 2 представлена главная страница созданного веб-приложения (index.html), что представляется пользователю после запуска приложения.

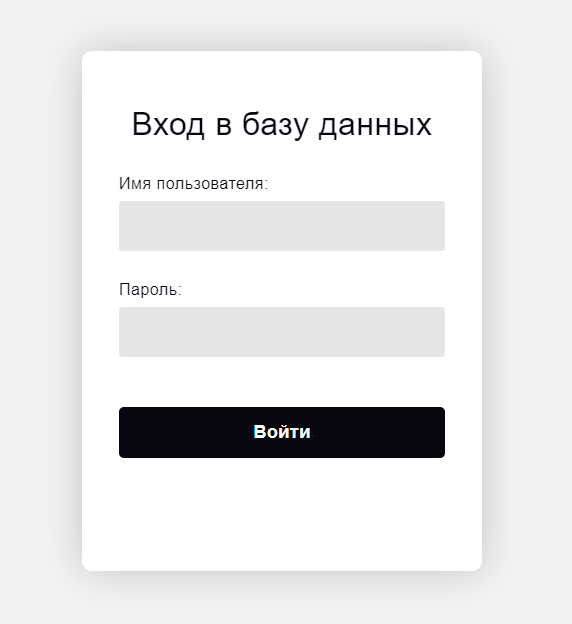


Рис. 2. Страница логина

На данной странице пользователь имеет возможность прописать заранее известные ему данные для подключения к базе данных, затем нажать на кнопку «Войти», и начать работу с базой данных. Валидация логина и пароля происходит в скрипте login.php.

Скрипт **base.php** объединяет стили и скрипты сайта – это позволяет без проблем добавить навигационную панель для каждой страницы сайта и в целом соблюдать общий дизайн сайта. Так же, в нем определены служебные (необходимые для корректной работы) переменные.

  
Рис. 3. base.php

На рисунке 4 показана домашняя страница, таблица с записями.



Рис. 4. Страница-dashboard.

На домашней странице мы видим все таблицы нашей базы данных. Они находятся во вкладках.

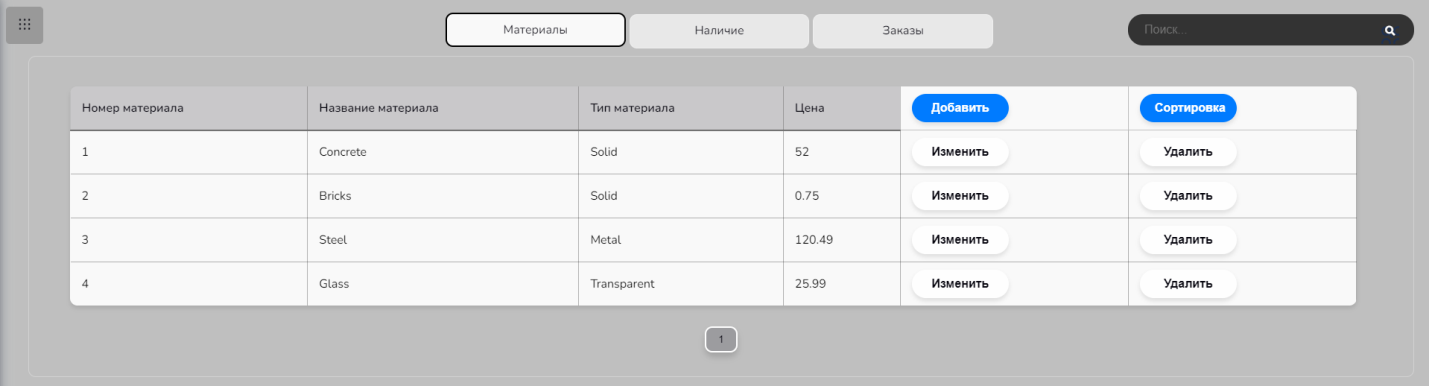
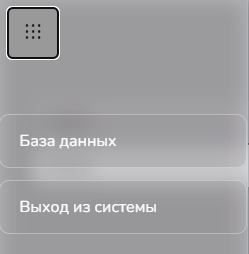


Рис. 5. Таблицы на dashboard.

Для удобства работы реализован постраничный вывод, сортировка по ключу таблицы, добавление, изменение, удаление кортежей данных, поиск по ключевым словам. При добавлении или изменении записей таблицы реализовано динамическое обновление и валидация введенных данных.

В верхнем левом углу находится кнопка вызова навигационной панели, благодаря которой мы можем перемещаться по сайту.

  
Рис. 6. Навигационная панель.

В верхнем правом углу находится поле поиска, который позволяет найти данные в текущей таблице по ключевым словам.



Рис. 7. Поле поиска.

После того как мы нажмем кнопку поиска, мы перейдем на страницу результатов поиска:

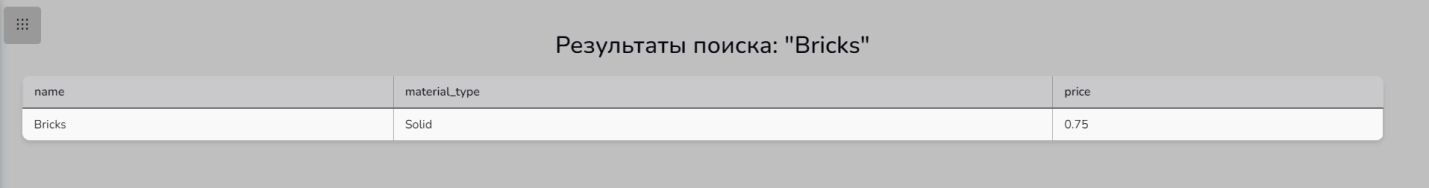


Рис. 8. Страница результатов поиска.

Скрипт **add\_record.php** отвечающий за добавление записей в таблицу:

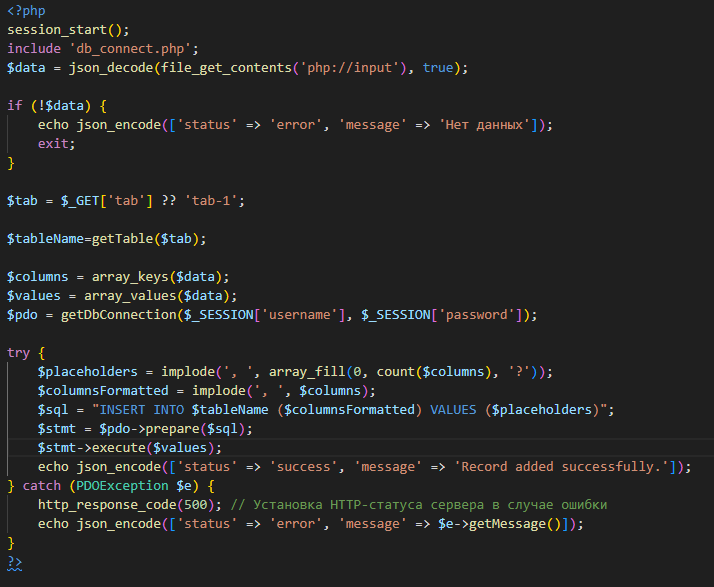


Рис. 9. add\_record.php.

Скрипт **db\_connect.php** отвечающий за создание объекта подключения к базе данных, определяет служебные переменные и функции для корректной работы серверной части Web-приложения:

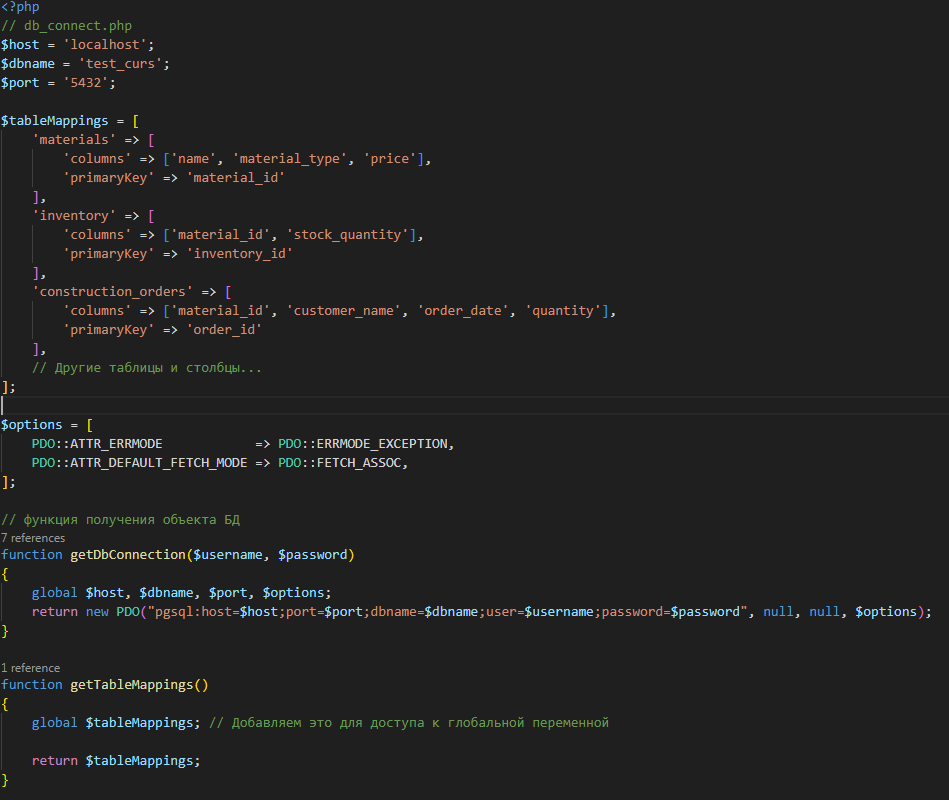
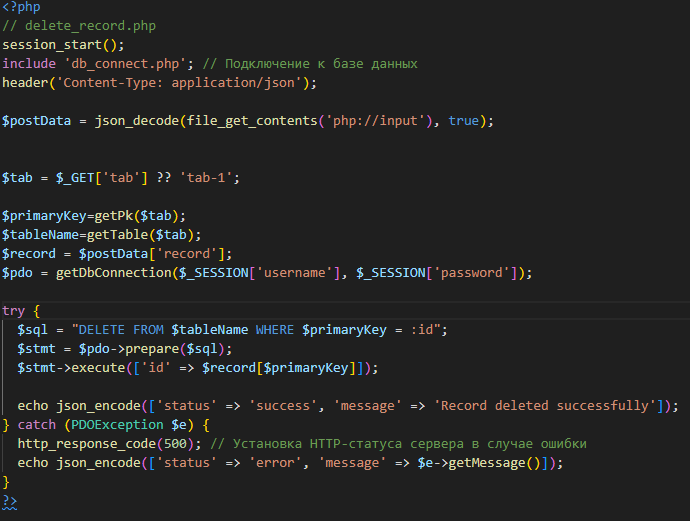


Рис. 10. db\_connect.php.

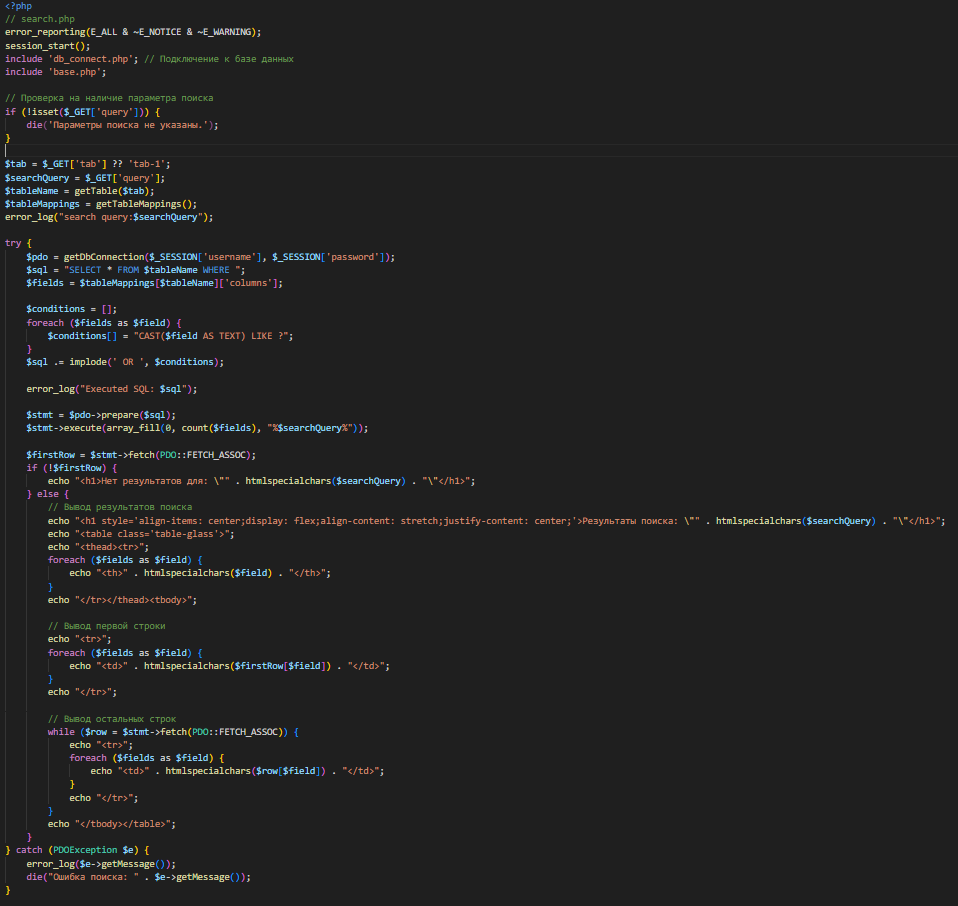
Скрипт **delete\_record.php** отвечающий за удаление записей в таблице:

  
Рис. 11. delete\_record.php.

Скрипт **edit\_data.php** отвечающий за редактирование записей в таблице:

  
Рис. 12. edit\_data.php.

Скрипт поиска и страница его результатов **search.php**:

****Рис. 13. search.php.

Отдельного внимания достойно то, что весь основной контент dashboard генерируется динамически (таблицы во вкладках, поля форм редактирования/добавления данных), что позволяет эффективно масштабировать данное Web-приложение (например, добавить новые таблицы), все JS скрипты и функции находятся в файле **main.js**, что позволяет эффективно использовать его во всем сайте через связующий скрипт **base.php**

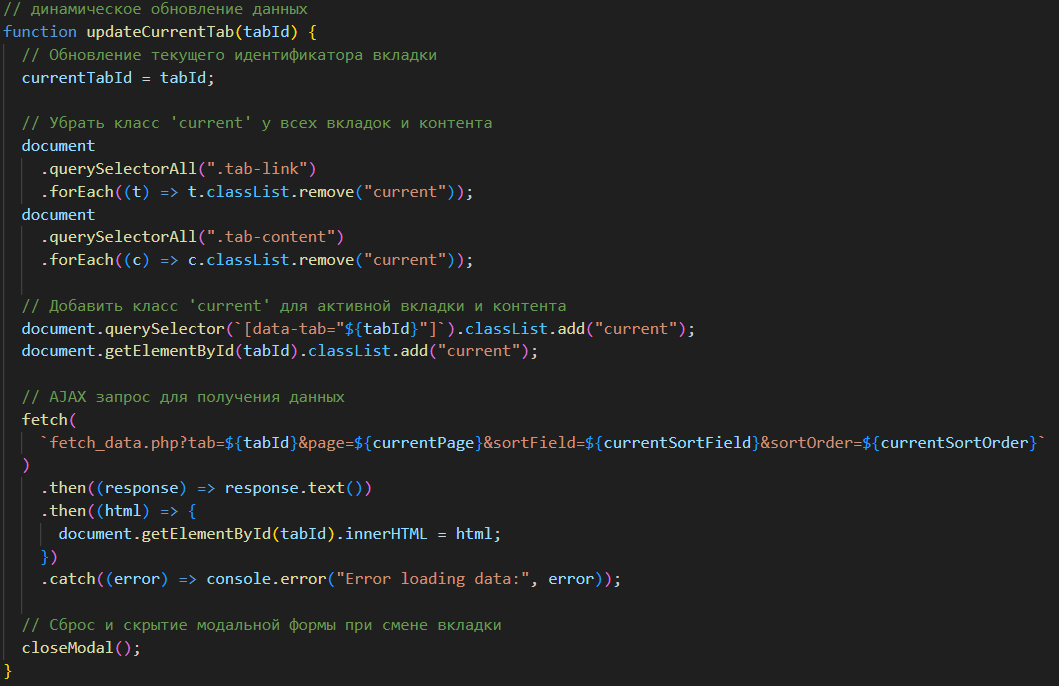


Рис. 14. JS-скрипт для динамической подгрузки таблиц.

****Рис. 15. JS-скрипт для динамического создания полей формы редактирования записи.

Для упрощения разработки и масштабирования в будущем, был создан отдельный файл стилей **base.css,** в котором находится дизайн-концепция, которому должны следовать все страницы Web-приложения:

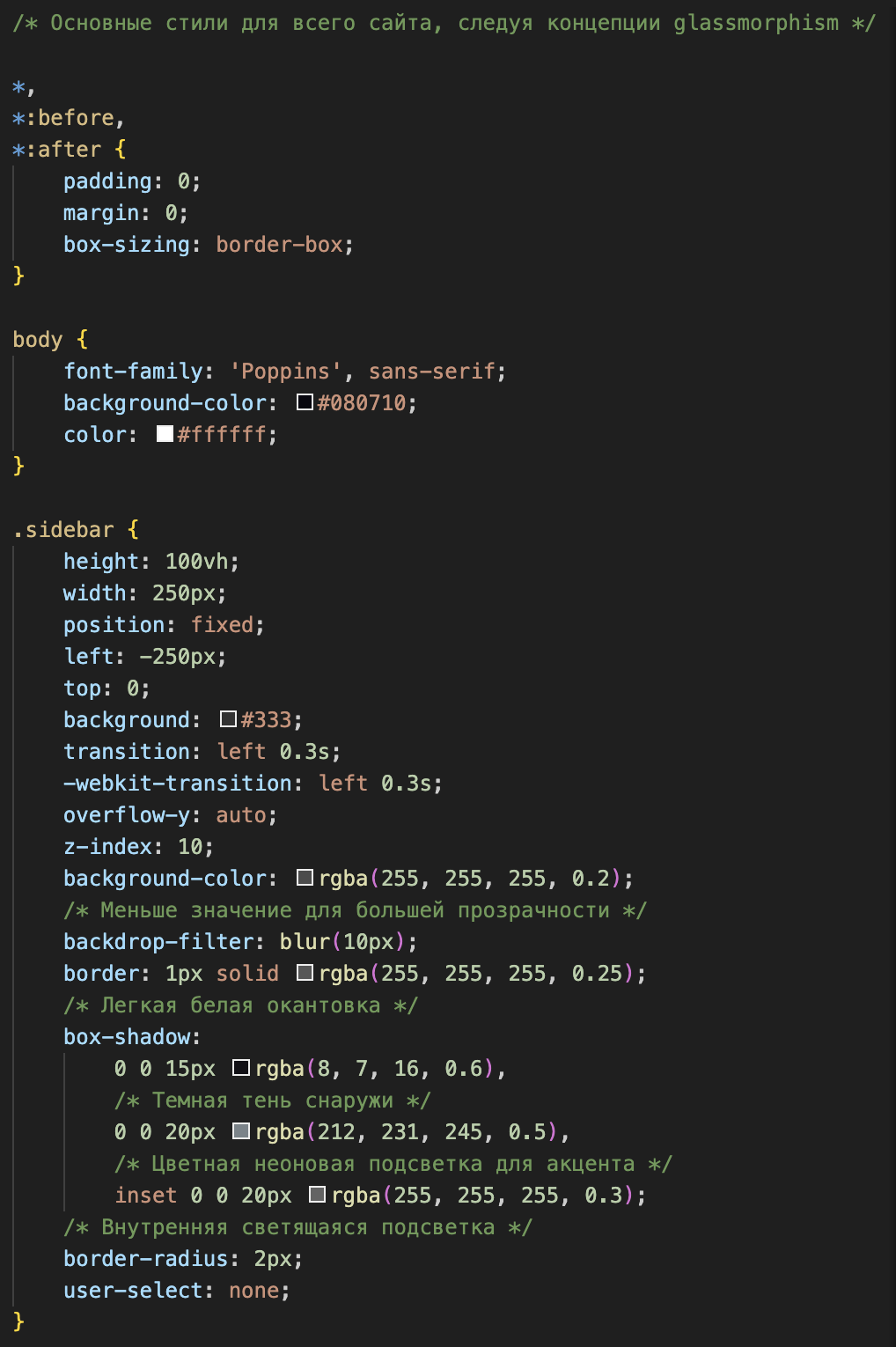


Рис. 16. Часть стилей из base.css

ГЛАВА 4. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

При первом посещении сайта пользователь сталкивается с страницей авторизации, где необходимо ввести учетные данные для установления подключения к системе. После успешной валидации данных и нажатия соответствующей кнопки, пользователь получает доступ к таблицам, которые могут быть редактированы.

На главной странице сайта пользователь имеет возможность управлять всеми записями, хранящимися в базе данных. При нажатии кнопки "Добавить" появляется модальная форма для создания новой записи. Если пользователь нажимает на кнопку "Изменить", открывается форма для редактирования существующей записи, а в случае выбора опции "Удалить", появляется форма для подтверждения удаления записи. Присутствует кнопка «Сортировка», позволяющая сортировать данные в таблице по первичному ключу.

Кроме того, в верхней части экрана, в правом верхнем углу, находится поле поиска. Путем ввода запроса в формате "ключевое слово", и нажатия на кнопку (иконка лупы), пользователь может выполнять поиск по текущей таблице и перейти на страницу с соответствующими результатами запроса. Это обеспечивает удобный механизм фильтрации данных.

В левой части экрана, в правом левом углу, находится кнопка навигации по сайту. Через нее можно попасть на главную страницу, или выйти из системы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. PHP documentation - https://www.php.net/docs.php
2. JavaScript documentation - [https://devdocs.io/javascript/](https://djangodoc.ru/3.2/)
3. HTML Book - [http: //htmlbook.ru/](http://htmlbook.ru/)
4. HTML5 Book -<https://html5book.ru/>
5. CSS documentation -<https://www.w3schools.com/css/>