РЕФЕРАТ

Курсовая работа по предмету базы данных, на тему “Создание WEB- приложения с использованием СУБД”, студента 3-го курса Рычкова Родиона Викторовича

Курсовая работа изложена на 25 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 6](#_Toc153402288)

[2.1 Промежуточные данные (таблицы БД) 8](#_Toc153402289)

[ГЛАВА 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ 10](#_Toc153402290)

[ГЛАВА 4. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 24](#_Toc153402291)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 25](#_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

В рамках данной курсовой работы предметной областью является разработка веб-сайта для интернет-магазина, специализирующейся на продаже видеоигр, с использованием базы данных. Основной упор делается на определение функционала веб-сайта, который придает ему функциональную ценность, превращая его из простого информационного ресурса в полноценное приложение.

Функционал веб-сайта представляет собой набор операций и возможностей, предоставляемых пользователю. В рамках данного веб-приложения планируется разработать следующие программные модули:

1. Таблица с данными из базы данных: Веб-сайт будет отображать информацию, хранящуюся в базе данных, позволяя пользователям просматривать доступные видеоигры и связанную с ними информацию.
2. Операции CRUD (Create, Update, Read, Delete): Пользователи смогут создавать новые записи, обновлять существующие, читать информацию из базы данных и удалять записи, обеспечивая управление данными на веб-сайте.
3. Поиск по столбцу и значению: Будет предоставлена возможность осуществлять поиск по заданному столбцу и значению, что позволит пользователям быстро находить нужную видеоигру или информацию о них.
4. Валидация данных авторизации в базе данных: Веб-сайт будет проверять и аутентифицировать пользователей, обеспечивая безопасность доступа к данным в базе данных.

После того, как был определен дизайн и функциональность веб-приложения, следующим шагом будет проектирование самого приложения, включая разработку архитектуры, базы данных, пользовательского интерфейса и программных компонентов.

* 1. Постановка задачи

Целью данного проекта является разработка веб-приложения, предназначенного для использования в интернете, с использованием соответствующих инструментов разработки. Основной задачей данного веб-приложения является обеспечение доступа пользователей к базе данных PostgreSQL с возможностью выполнения операций CRUD (Create, Read, Update, Delete), а также выполнения поиска данных по ключевым словам и сортировки данных по различным критериям.

1. При разработке данного приложения следует уделить особое внимание следующим аспектам:
2. Гибкий дизайн: Дизайн приложения должен быть адаптивным и удовлетворять потребностям пользователей, обеспечивая удобство и приятный внешний вид интерфейса.
3. Система реализации CRUD: Веб-приложение должно обладать эффективной и гибкой системой для выполнения операций CRUD, что позволит пользователям легко управлять данными в базе данных.
4. Простота в освоении: Важным аспектом разработки является создание интуитивно понятного и легкого в использовании интерфейса, что облегчит процесс освоения программы для пользователей.
5. Безопасность: Разработка должна учитывать вопросы безопасности, включая защиту данных в базе данных, аутентификацию пользователей и предотвращение несанкционированного доступа.
6. Эффективность и производительность: Приложение должно обеспечивать высокую производительность при выполнении операций с базой данных, чтобы пользователи могли быстро получать необходимую информацию.

Реализация проекта будет включать в себя этапы проектирования, разработки, тестирования и деплоймента приложения, с учетом вышеперечисленных критических аспектов.

**ГЛАВА 2. АНАЛИЗ ДАННЫХ**

* 1. Промежуточные данные (таблицы БД)

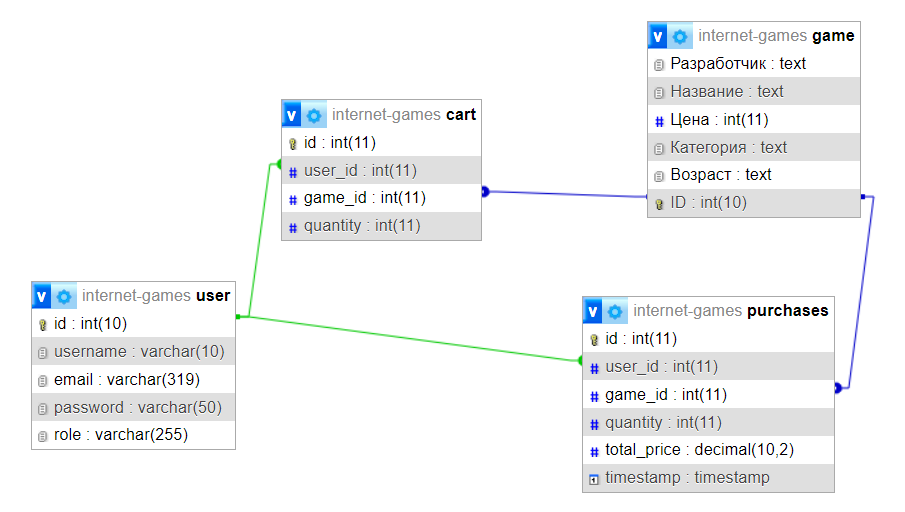
Проектирование БД является очень важным этапом, от которого зависят последующие этапы разработки приложения.

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 cart имеет 4 атрибута:
   1. id– первичный ключ;
   2. user\_id–id клиента;
   3. game\_id–id игры;
   4. quantity – объем заказа;
2. Сущность game имеет 6 атрибутов:
   1. Разработчик – имя разработчика игры;
   2. Название– название товара;
   3. Цена– цена;
   4. Категория– жанр товара.
   5. Возраст – игра для возрастной категории;
   6. Id – id игры;
3. Сущность purchases имеет 5 атрибутов
   1. Id – первичный ключ;
   2. user\_id– id пользователя
   3. game\_id– id игры;
   4. quantity – количество товара;
   5. total-price – итоговая цена;
4. Сущность user имеет 5 атрибутов:
   1. Id – первичный ключ;
   2. Username – имя пользователя ;
   3. Email – email адрес пользователя;
   4. Password- пароль пользователя;
   5. Role – роль пользователя;

ГЛАВА 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Созданная программа состоит из следующих страниц:

* index.php
* cart.php
* authorize.php
* login.html
* profile.php

При создании сайта применялись языки программирования PHP для серверной части и JavaScript (в частности, AJAX) для обеспечения динамической загрузки данных без необходимости перезагрузки страницы. Для взаимодействия с базой данных использовался язык SQL.

Каждая основная задача приложения была вынесена в отдельный скрипт, что способствует модульности и удобству обслуживания. Особое внимание уделялось универсальности функций интерфейсной части, обеспечивая динамическую загрузку данных и повышая пользовательский опыт. Аналогичный принцип был применен и к серверной части приложения с целью обеспечения гибкости и эффективности масштабирования проекта.

Давайте подробно рассмотрим каждый из скриптов серверной части и функций интерфейсной стороны, которые используются в рамках данной программы.

Скрипт authorize.phpпроизводит подключение к базе данных, сохраняет в памяти нужную для подключения информацию:

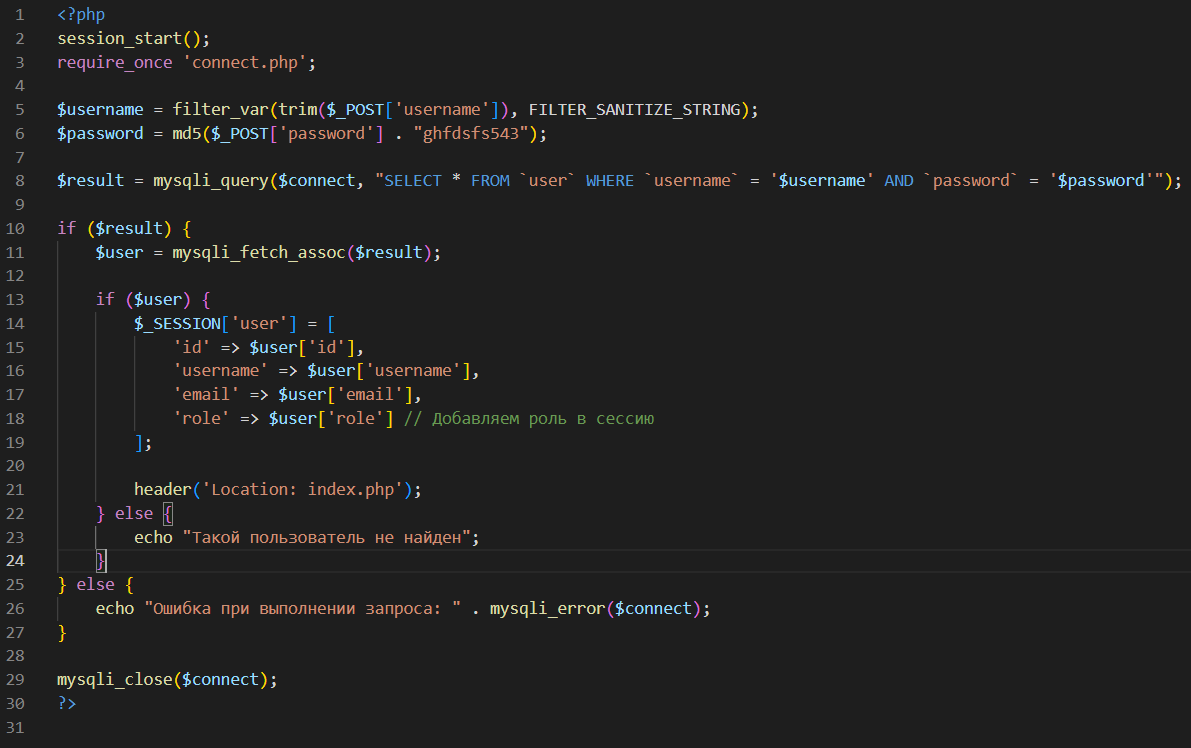


Рис. 1. authorize.php

На изображении 2 изображена стартовая страница веб-приложения (login.html), которую пользователь видит если не находится в сессии

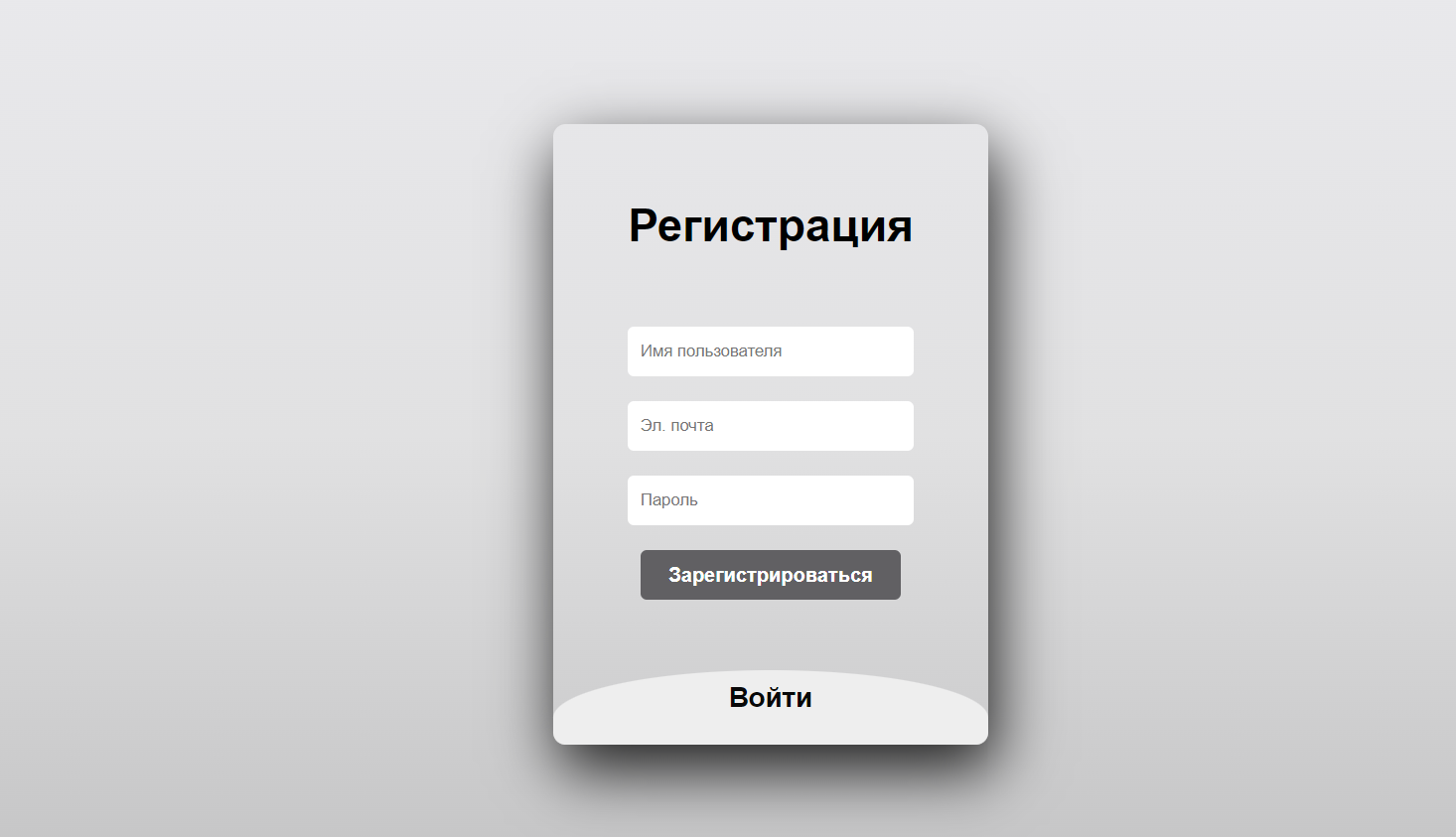


Рис. 2. Страница логина

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 authorize.php.

Скрипт **cart.php** позволяет пользователю добавить выбранную игру в корзину и приобрести ее, либо удалить из корзины

  
Рис. 3 cart.php

На рисунке 4 показана страница, таблица с записями заказов от клиентов.



Рис. 4. Страница-покупок пользователей.

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



Рис. 5. Таблица на главной странице.

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

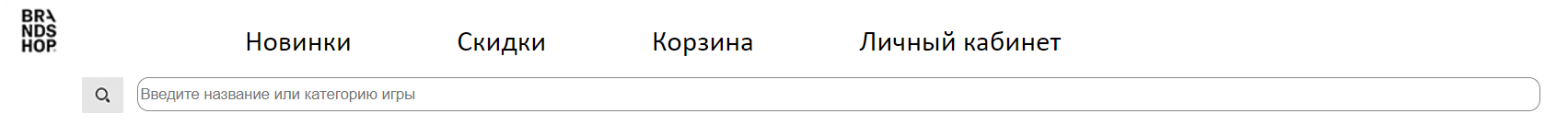
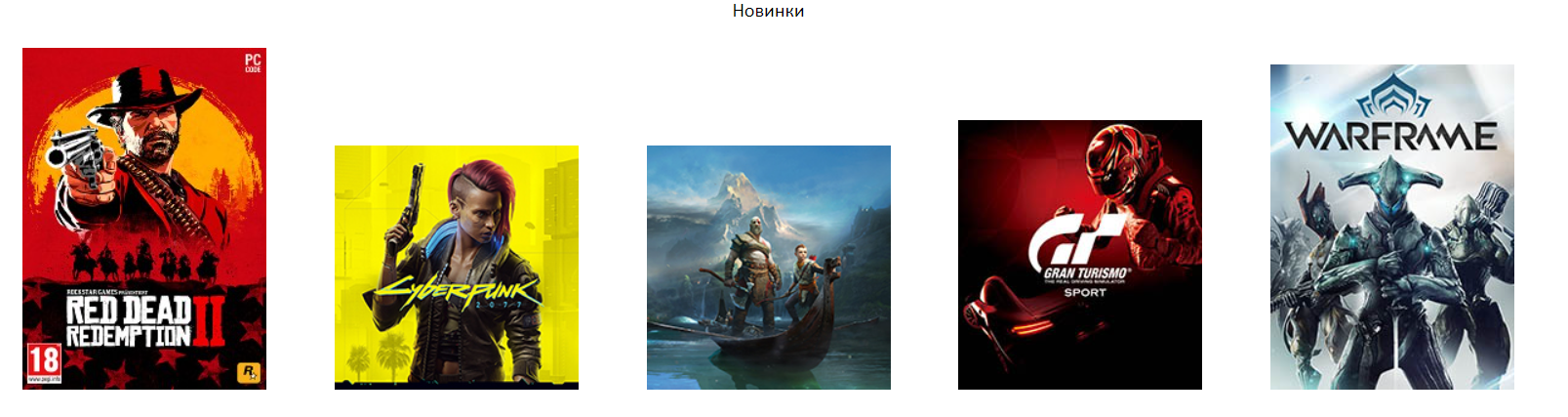


Рис. 6.1 Навигационная панель - вкладки.

На странице «Новинки» мы можем посмотреть какой товар у нас есть.



Ниже от навигационной панели находится поле поиска, который позволяет найти данные в текущей таблице по ключевым словам.



Рис. 8. Поле поиска.

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

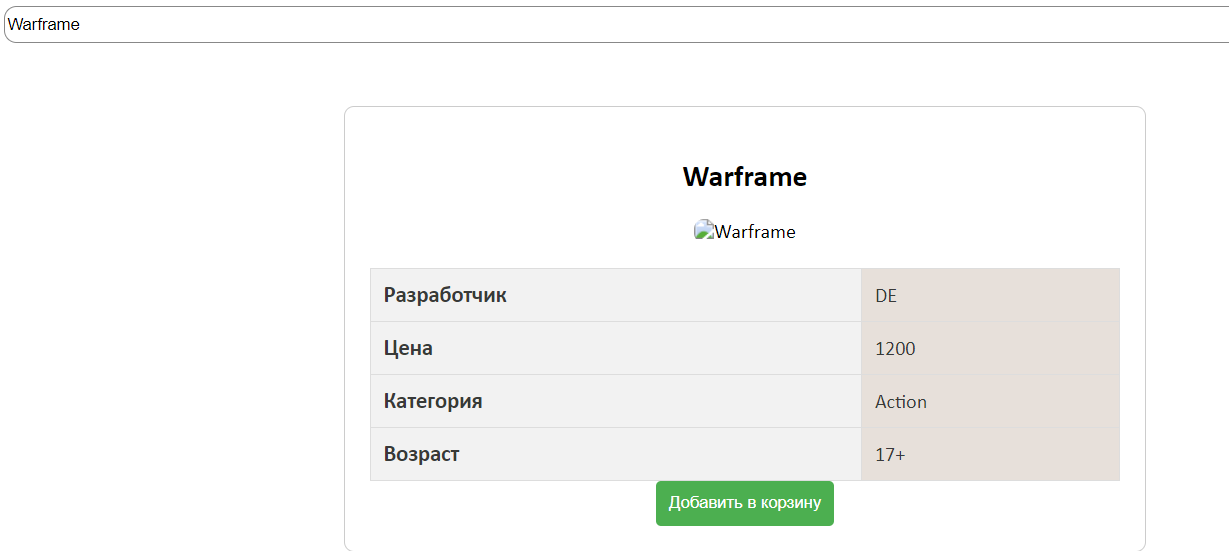


Рис. 9. Страница результатов поиска.

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

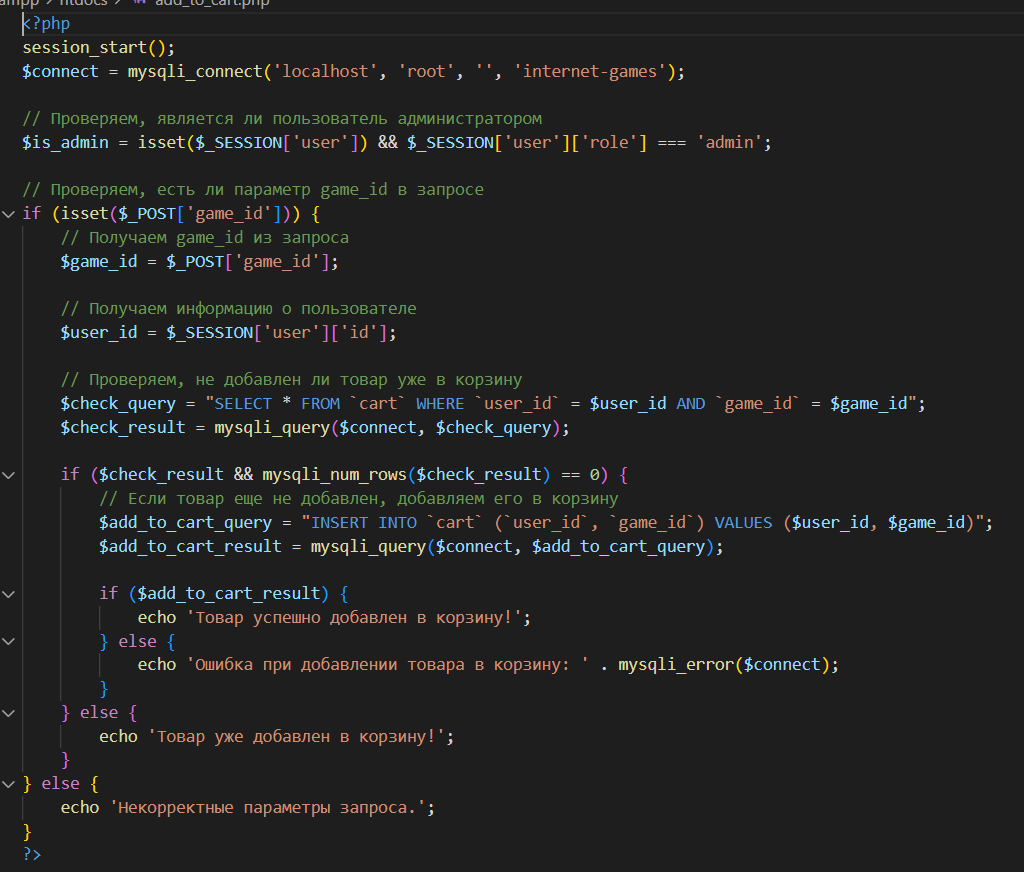


Рис. 10. add\_to\_cart.php.

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

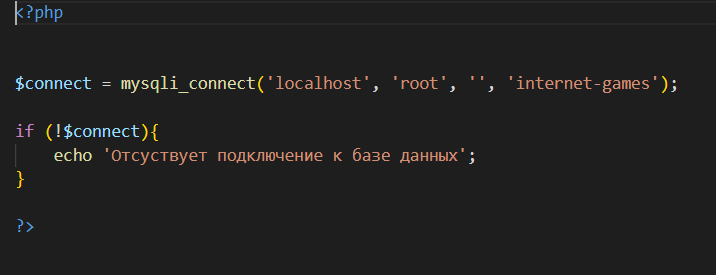
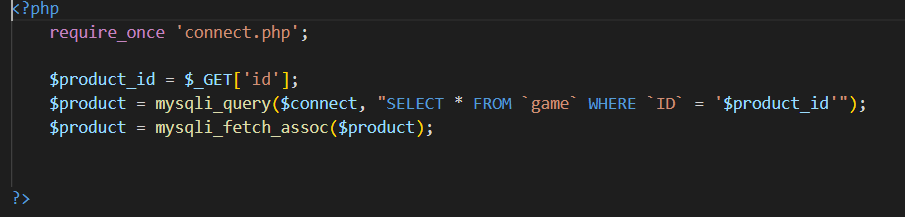


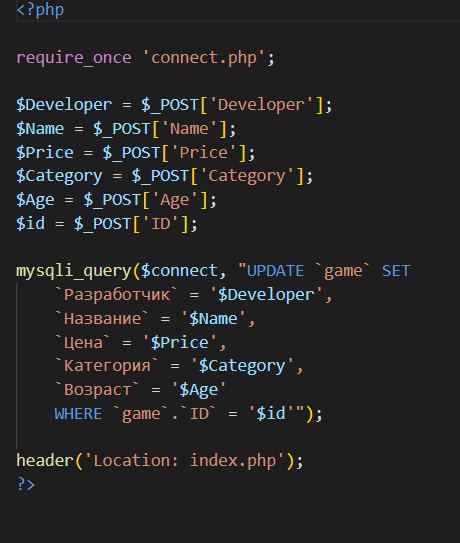
Рис. 11. connect.php.

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

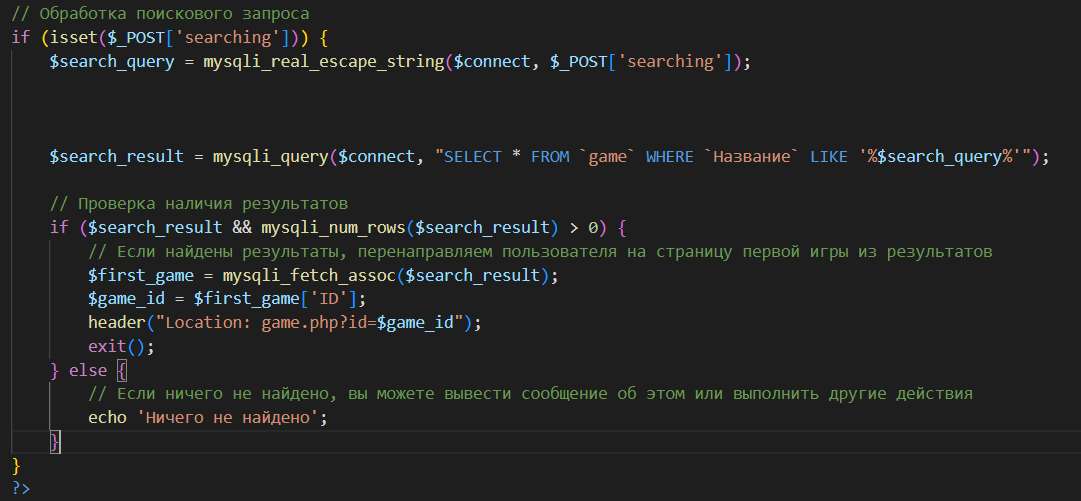
  
Рис. 12. delete.php.

Скрипт **update.php** отвечающий за редактирование записей в таблице:



  
Рис. 13.update.php.

Скрипт поиска и страница его результатов в **index.php**:

****Рис. 14 index.php.

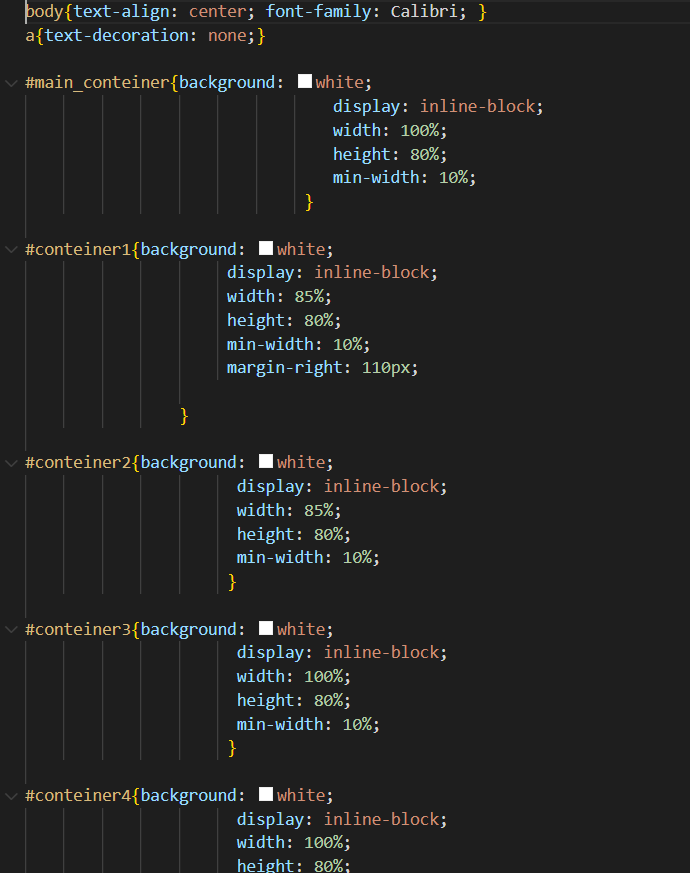


Рис. 16. Часть стилей из style1.css

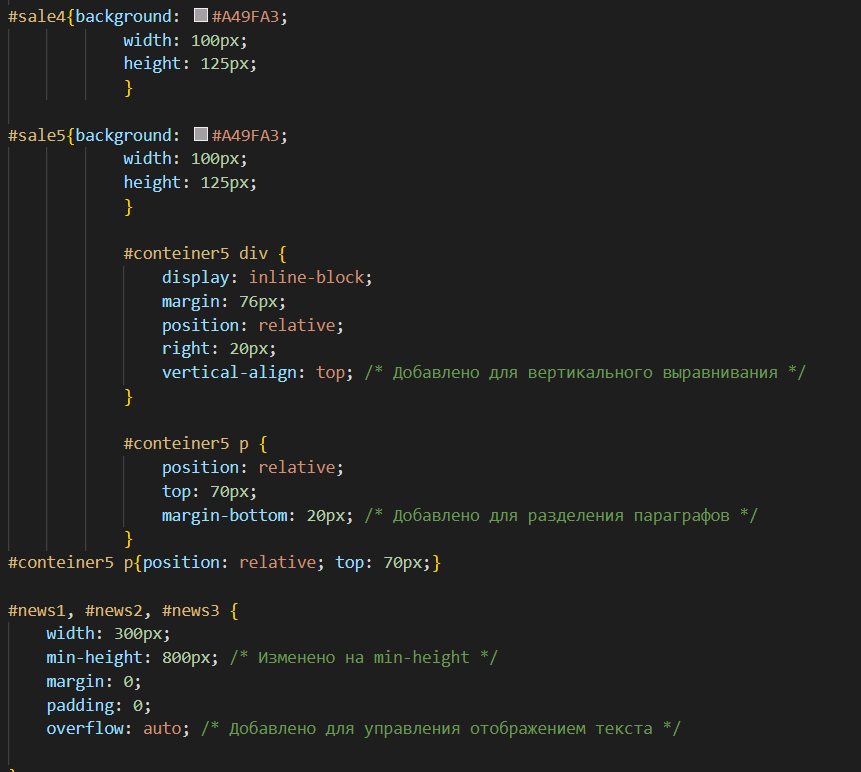


Рис. 17. Часть стилей из style1.css

ГЛАВА 4. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

При первом посещении веб-сайта пользователю предоставляется страница авторизации, где требуется вводить учетные данные для установления соединения с системой. После успешной проверки данных и активации соответствующей кнопки, пользователю предоставляется доступ к таблицам, которые могут быть изменены.

На главной странице веб-сайта пользователь может управлять всеми записями, хранящимися в базе данных. При выборе опции "Добавить", появляется модальное окно для создания новой записи. Если пользователь выбирает "Изменить", открывается форма для редактирования существующей записи, а при выборе опции "Удалить", открывается окно для подтверждения удаления записи.

В верхней части экрана, справа, располагается поле поиска. Путем ввода запроса в формате "ключевое слово" и нажатия на кнопку (иконку лупы), пользователь может выполнять поиск по текущей таблице и переходить на страницу с соответствующими результатами запроса. Это предоставляет удобный механизм фильтрации данных.

В левой части экрана, слева сверху, находится кнопка навигации по сайту. С ее помощью пользователь может вернуться на главную страницу, просмотреть состояние склада 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/>