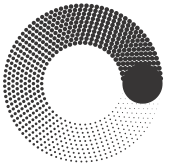
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ**

***Институт Принтмедиа и информационных технологий***

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»,**

**КУРСОВОЙ ПРОЕКТ**

**Дисциплина:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Тема:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Выполнили: студенты группы \_\_211-729\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_Файзуллин Альберт Ринатович**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Мавлютов Марат Ринатович**

(Фамилия И.О.)

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Проверил: \_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

(Фамилия И.О., степень, звание)

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва**

**2023**

Оглавление

[ВВЕДЕНИЕ 3](#_Toc75394621)

[ГЛАВА 1. Проектирование 4](#_Toc75394622)

[1.1 Описание предметной области 4](#_Toc75394623)

[1.2 Выбор инструментов 5](#_Toc75394624)

[ГЛАВА 2. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ «Кинотека» 8](#_Toc75394625)

[2.1 Проектирование и физическая реализация базы данных на СУБД 8](#_Toc75394626)

[2.2 Разработка программного продукта 12](#_Toc75394627)

[ЗАКЛЮЧЕНИЕ 25](#_Toc75394628)

[Библиографический список 26](#_Toc75394629)

ВВЕДЕНИЕ

Тема проекта – каталог компьютерных игр (веб-приложение).

Необходимость данного веб-приложения обусловлена тем, что:

1. Ресурсов для покупки игр не так уж и много
2. Все те, что есть на сегодняшний день, неудобны в использовании или часто имеют большое количество рекламы, которая порой не дает правильно загрузить игру с первого раза

Данный проект позиционируется как специализирующийся каталог игр. Здесь у вас будет возможность найти интересную игру не только по интересным жанрам, но и по разработчикам, по популярности.

Цель – создание веб-приложения для упрощения поиска новых игр, интересных пользователю.

Исходя из поставленной цели, были сформированы следующие задачи:

1. Изучение предметной области;
2. Выбор и обоснование инструментов разработки;
3. Проектирование базы данных
4. Реализация физической модели данных.

ГЛАВА 1. Проектирование

1.1 Описание предметной области

Многие из нас наслаждаются проведением времени перед компьютером, занимаясь различными играми, чтобы расслабиться после тяжелого дня и отвлечься от повседневных забот. Вместе с тем, игровая индустрия стремительно развивается, способствуя постоянному совершенствованию производства мощных компьютеров. Каждый год мы получаем все более мощные устройства, благодаря которым игровой процесс становится ещё более захватывающим и реалистичным.

С ростом индустрии и количество игр неуклонно увеличивается. Однако возникает вопрос: где можно найти эти игры, чтобы не приходилось бесконечно искать по просторам интернета и беспокоиться о возможности загрузки нежелательных программ вместе с ожидаемой игрой?

Мы разработали веб-приложение, чтобы предоставить потенциальным пользователям возможность воспользоваться нашим сервисом в любое время безопасно и без риска загрузки вредоносных программ, которые могут забивать память и использовать дополнительные ресурсы компьютера.

Наше веб-приложение предназначено для работы с разными типами пользователей: посетителями и авторизованными пользователями.

Для посетителей наше веб-приложение предоставляет возможность просматривать игры и добавлять их в корзину. Однако, для загрузки и покупки игр в дальнейшем потребуется пройти регистрацию и получить статус авторизованного пользователя.

После того, как пользователь выберет понравившуюся игру, он будет перенаправлен на страницу, где будет представлена следующая информация:

* Название игры
* Описание
* Название разработчика
* Рейтинг
* Платформы, на которых поддерживается данная игра
* Цена
* Трейлер игры

После того, как посетитель просмотрит страницы с описанием игр, у него будет возможность выбрать несколько игр, которые ему понравились, и перейти к их покупке. После успешной покупки, посетитель сможет скачать выбранные игры и наслаждаться игровым процессом.

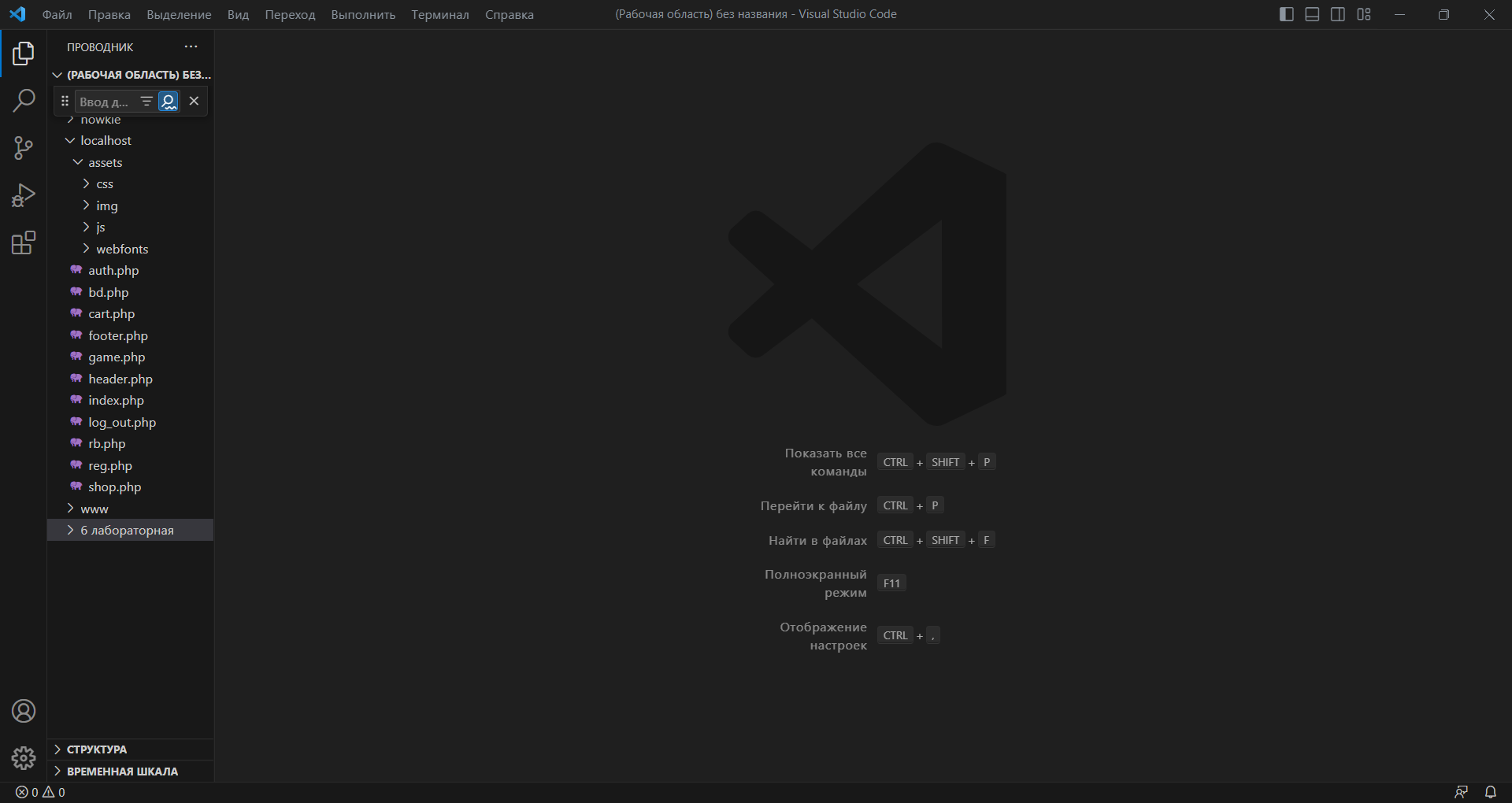
1.2 Выбор инструментов

Для реализации проекта был выбран формат веб-сайта. Этот выбор был обусловлен его доступностью, поскольку для доступа к сайту требуется только устройство с подключением к интернету, и поэтому любой желающий сможет воспользоваться ресурсом.

На этапе разработки был выбран редактор кода Visual Studio Code для кроссплатформенной разработки веб- и облачных приложений. Разметка страниц была написана на HTML, стили страниц создавались с использованием CSS, а для написания скриптов, необходимых для функционирования сайта, был использован PHP и JavaScript.

Visual Studio Code - это редактор исходного кода, разработанный Microsoft для платформ Windows, Linux и macOS. Он представляет собой легкий и гибкий инструмент для разработки веб- и облачных приложений. Включает в себя полезные функции, такие как отладчик, интеграцию с Git, подсветку синтаксиса, автодополнение IntelliSense и средства для рефакторинга кода. Редактор предлагает широкие возможности для настройки, включая пользовательские темы, настраиваемые комбинации клавиш и файлы конфигурации.

Visual Studio Code распространяется бесплатно и разрабатывается как программное обеспечение с открытым исходным кодом. Однако, готовые сборки распространяются под проприетарной лицензией.



*Рисунок 1. Оболочка Visual Studio Code*

HTML (HyperText Markup Language) - это стандартизированный язык разметки документов. Большинство веб-страниц содержат разметку на языке HTML (или XHTML). Браузеры интерпретируют язык HTML, а полученный от него отформатированный текст отображается на экране монитора компьютера или мобильного устройства. Спецификации HTML5 формулируются в терминах DOM (объектной модели документа). XHTML - это более строгий вариант HTML, который следует синтаксису XML и используется для разметки гипертекстовых документов.

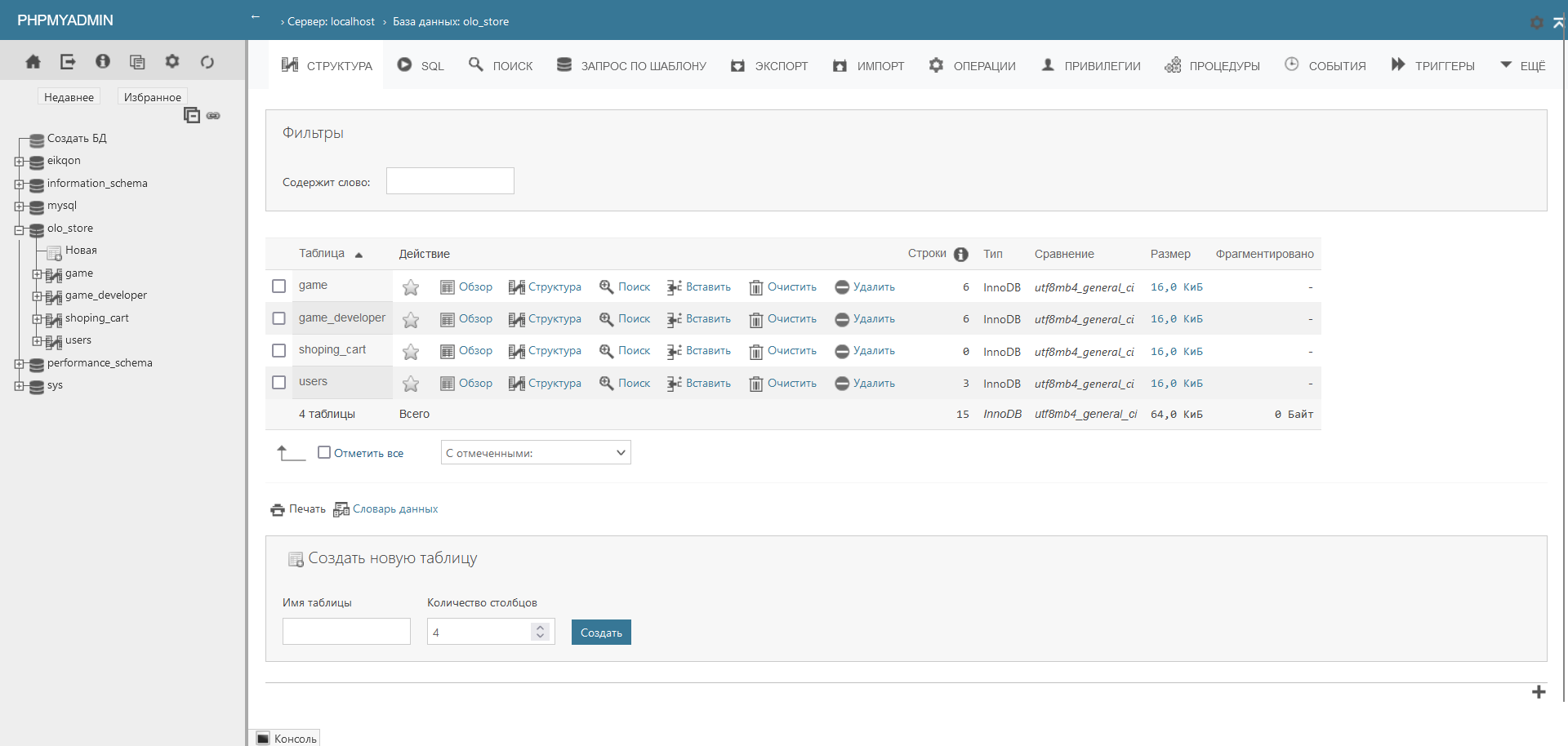
CSS (Cascading Style Sheets) - это формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Он применяется в основном для оформления внешнего вида веб-страниц, написанных на языках разметки HTML и XHTML, но также может применяться к любым XML-документам, таким как SVG или XUL.

PHP - это скриптовый язык общего назначения, широко применяемый для разработки веб-приложений. В настоящее время он поддерживается большинством хостинг-провайдеров и является одним из лидеров среди языков, используемых для создания динамических веб-сайтов.

В качестве локального сервера мы выбрали AMPPS.

Веб-сервер AMPPS — это одна из самых мощных, удобных и простых сборок для веб-разработчика, на которой можно создавать сайты, а затем их настраивать, тестировать и наполнять контентом перед тем, как сайт окажется в сети Интернет.

Для разработки баз данных было предпочтено использовать веб-приложение phpMyAdmin.



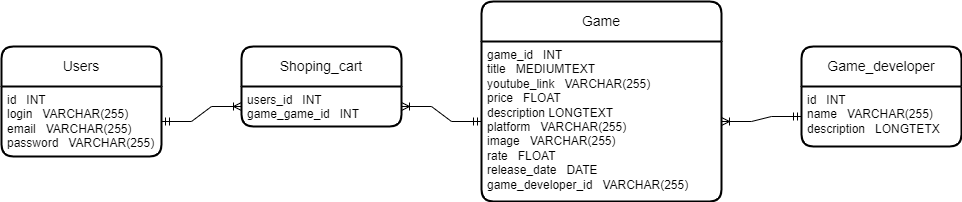
*Рисунок 2. Оболочка phpMyAdmin*

phpMyAdmin - это веб-приложение с открытым исходным кодом, написанное на языке PHP, которое представляет собой веб-интерфейс для администрирования СУБД MySQL. Оно позволяет администрировать сервер MySQL, запускать SQL-команды и просматривать содержимое таблиц и баз данных через веб-браузер. phpMyAdmin пользуется широкой популярностью среди веб-разработчиков, так как предоставляет удобный интерфейс для управления СУБД MySQL без необходимости вводить SQL-команды напрямую. Оно облегчает выполнение различных задач, связанных с управлением баз данных, и является незаменимым инструментом при разработке веб-приложений.

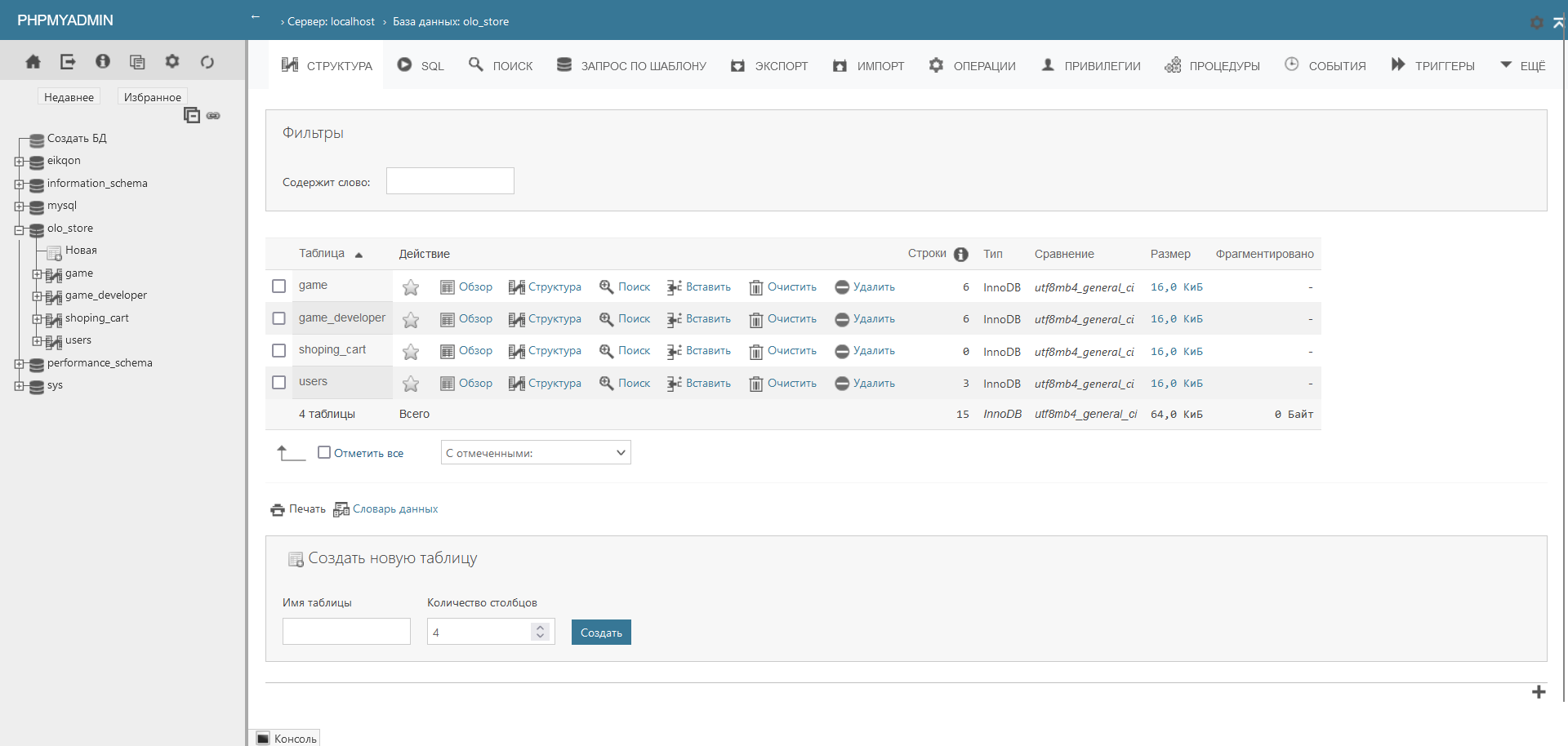
ГЛАВА 2. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ «Кинотека»

2.1 Проектирование и физическая реализация базы данных на СУБД

Для осуществления данного проекта была разработана база данных, которая включает четыре таблицы. Таблица в реляционной базе данных представляет собой набор связанных данных, хранящихся в структурированном формате в базе данных. Структура и взаимосвязи этой базы данных представлены на схематических диаграммах, представленных ниже.



*Рисунок 3. Структура базы данных*



*Рисунок 4. Структура базы данных в phpMyAdmin*

Ниже представлены структуры всех таблиц, включая пояснения к каждой из них:

Таблица game – таблица, в которой содержится информация об играх. Данная таблица содержит:

Game\_id – присваивается автоматически.

Title – название игры.

Youtube\_link – ссылка на страницу игры.

Description – описание игры.

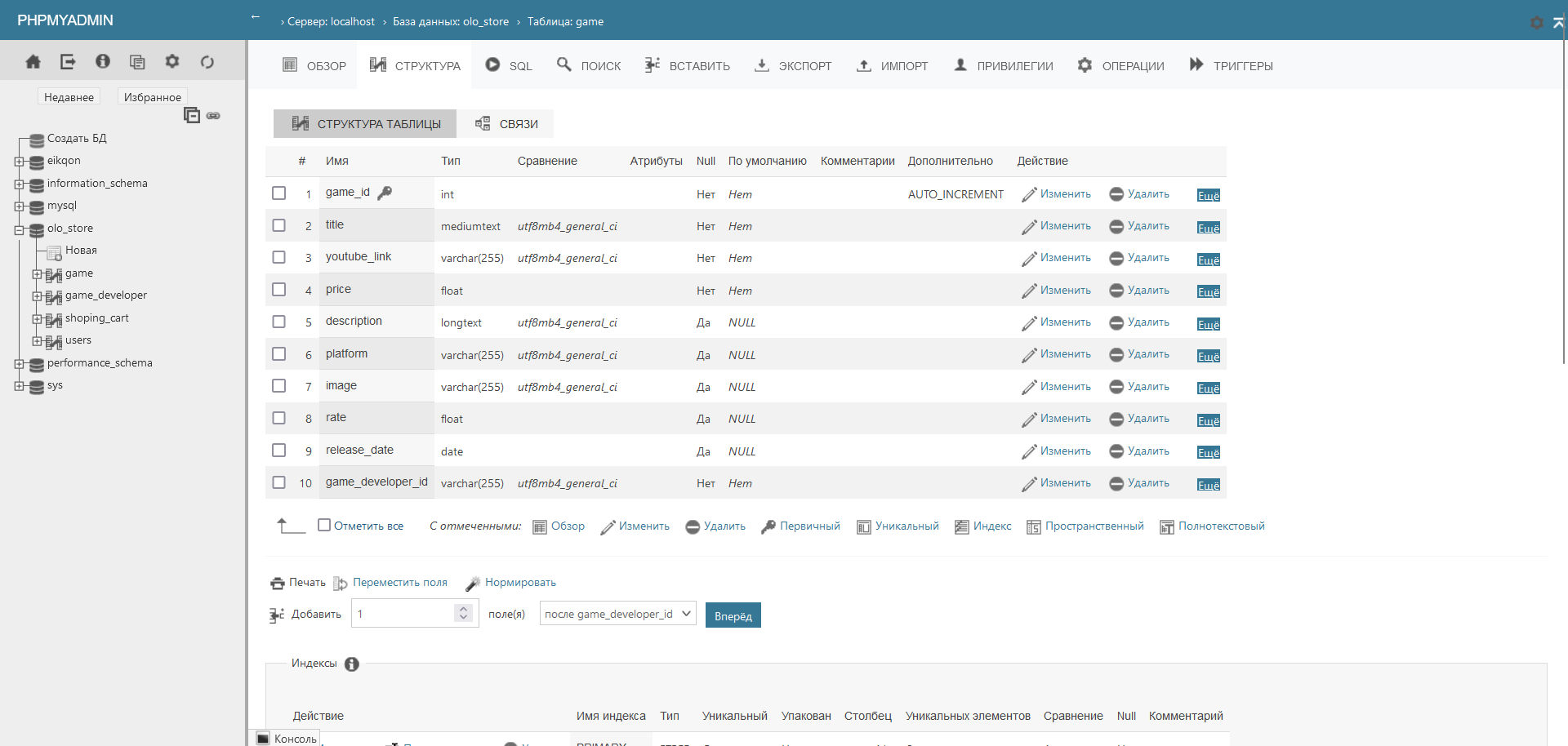
Price – цена.

Platform – платформа для которой была реализована игра или имеет поддержку.

Image – постеры игры.

Developer – разработчик.

Rate – рейтинг.



*Рисунок 5. Таблица game*

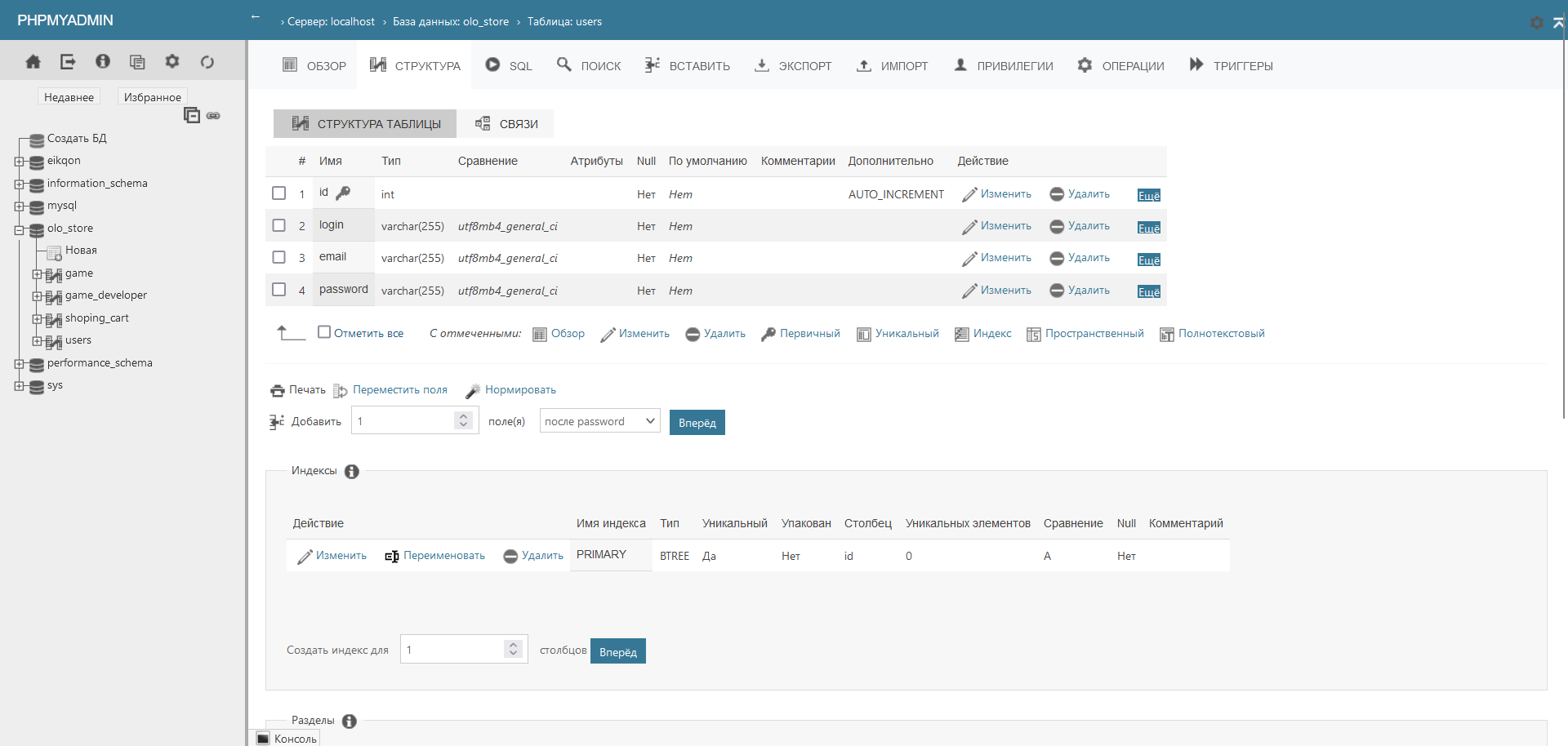
Таблица users – таблица, в которой содержится информация о зарегистрированных пользователях. Данная таблица содержит:

Id – задается автоматически.

Login – имя, под которым был зарегистрирован пользователь.

Email – почта, которая была использована для создания аккаунта.

Password - пароль пользователя.

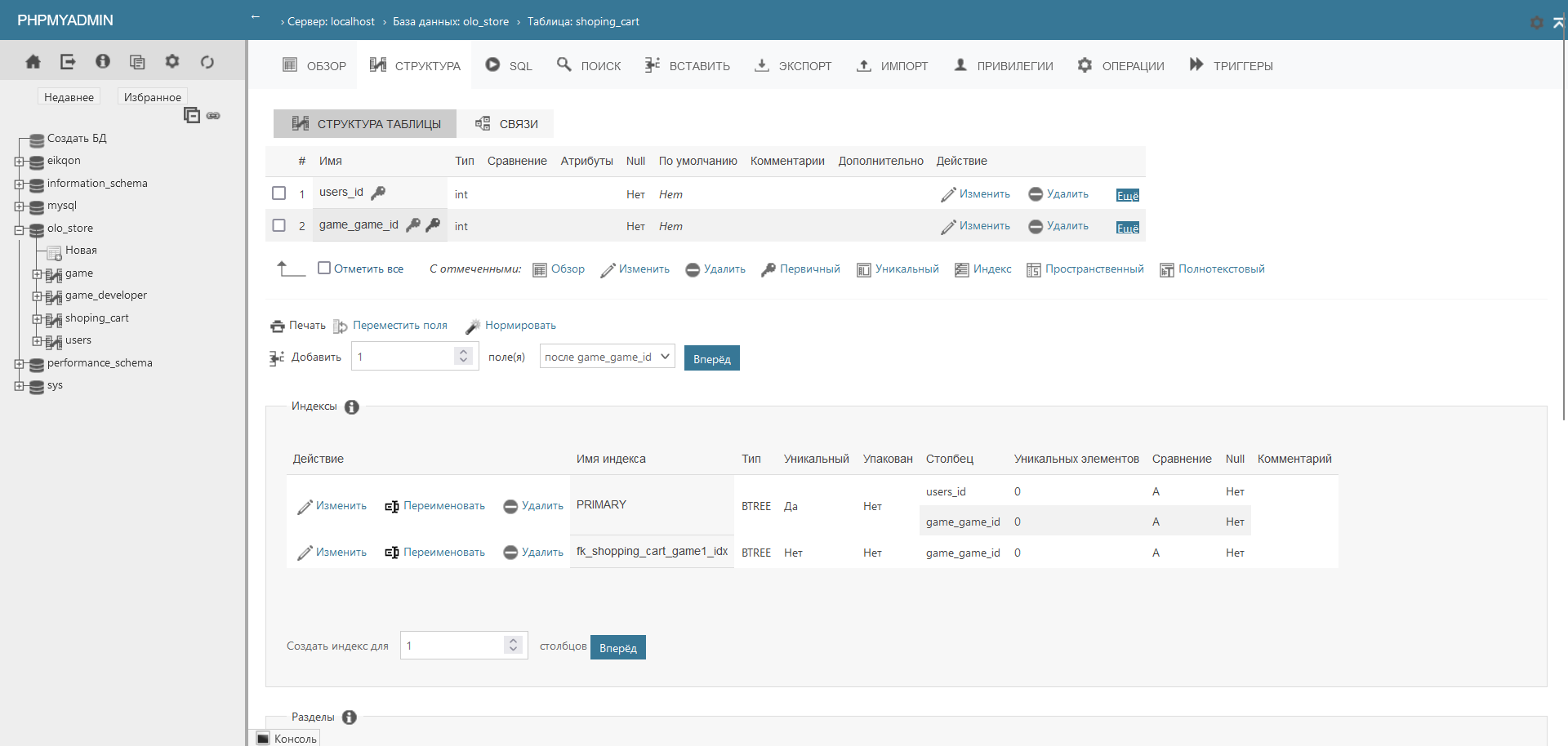


*Рисунок 6. Таблица users*

Таблица shopping\_cart – таблица (корзина), в которой содержится информация о выбранных пользователем игр. Данная таблица содержит:

User\_id – идентификационный номер пользователя (вызывается из таблицы users).

Game\_game\_id – идентификационный номер игры (вызывается из таблицы game).

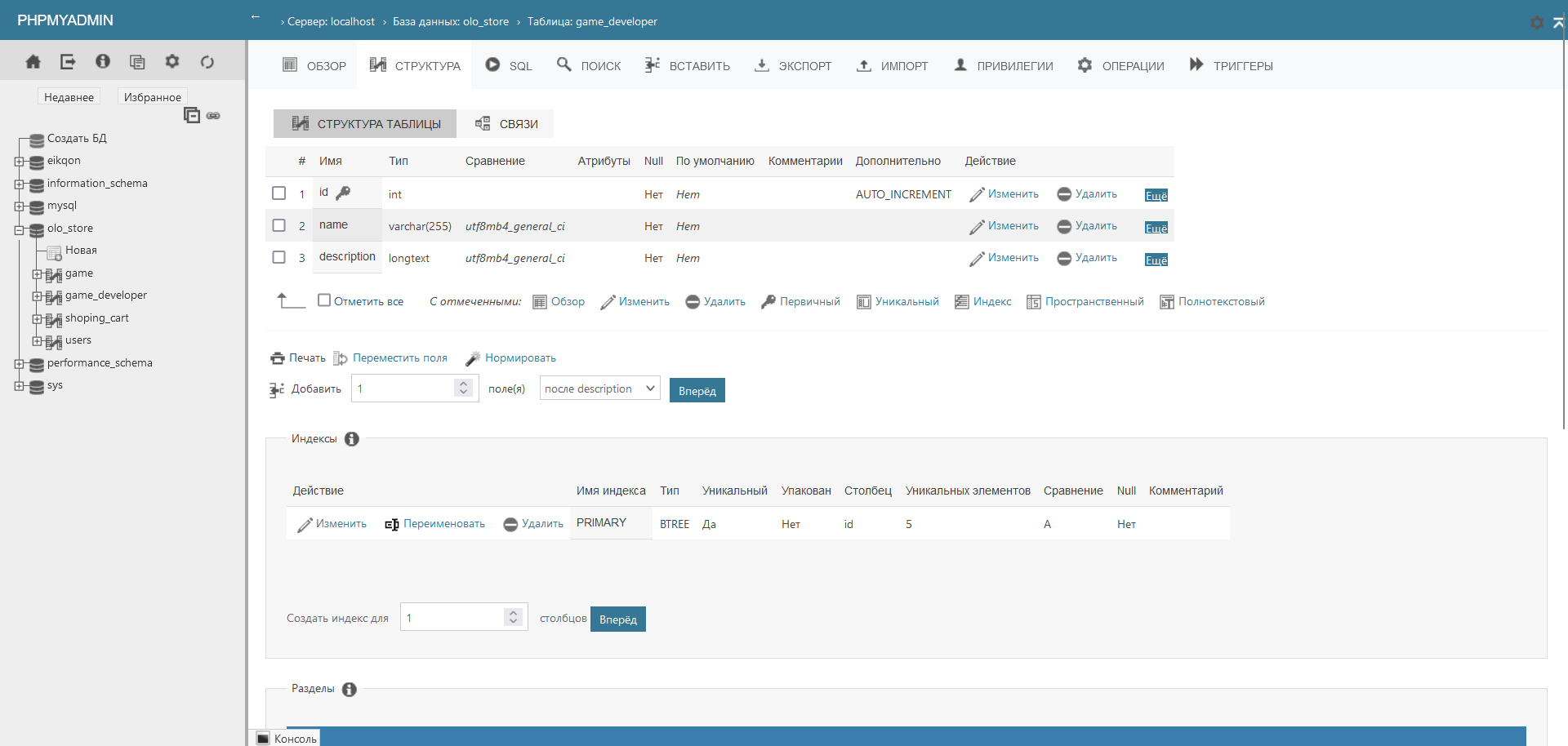


*Рисунок 7. Таблица shopping\_cart*

Таблица game\_developer – таблица, в которой содержится информация о разработчиках игр. Таблица содержит:

Name – название разработчика.

Description – описание деятельности компании (разработчика).



*Рисунок 8. Таблица game\_developer*

Таким образом, с использованием веб-приложения phpMyAdmin была успешно спроектирована и реализована база данных для информационной системы "OLO Store". Эта база данных предоставляет структурированное хранение данных, необходимых для функционирования системы, включая информацию о пользователях, играх, корзинах и разработчиках. phpMyAdmin предоставил удобный интерфейс для администрирования и управления базой данных, что позволило эффективно разработать и настроить информационную систему "OLO Store".

2.2 Разработка программного продукта

Для того чтобы реализовать данный проект на языке программирования PHP, необходимо создать файл "bd.php", в котором будет храниться подключение к базе данных. Это позволит нам повторно использовать подключение в различных файлах, вместо того чтобы копировать один и тот же код в каждый файл.

В файле "bd.php" мы создадим переменные и напишем функцию, которая будет устанавливать соединение с базой данных. Это позволит нам легко подключаться к базе данных в других частях приложения.

$servername = "localhost";

$database = "OLO\_Store";

$username = "root";

$password = "mysql";

*Листинг 1. Создание переменных*

$conn = mysqli\_connect($servername, $username, $password, $database);

*Листинг 2. Создание подключения к базе данных*

функция разрыва соединения:

function close() {

    global $conn;

    mysqli\_close($conn);

    }

*Листинг 3. Функция разрыва соединения с базой данных*

Теперь создадим функцию, которая будет выполнять поиск игры по уникальному идентификационному номеру при её добавлении в корзину и генерации странички с информации об игре:

function get\_game\_by\_id($game\_id){

    global $conn;

    $sql = mysqli\_query($conn, "SELECT \* FROM game WHERE game\_id = ".$game\_id);

    $game = mysqli\_fetch\_assoc($sql);

    return $game;

}

*Листинг 4. Функция поиска игры по идентификатору*

Также создадим функцию, которая будет возвращать массив ошибок, связанных с возможными ошибками, допущенными пользователем при регистрации и авторизации:

function showError($errors){

    return array\_shift($errors);

}

*Листинг 5. Функция возвращающая массив ошибок*

Для выполнения операций регистрации и авторизации пользователей объявим переменные, которые будут использоваться:

$data = $\_POST;

$showError = False;

*Листинг 6. Объявление переменных*

Для упрощения реализации системы регистрации и авторизации подключим файл "rb.php", который содержит соответствующие функции. Также создадим подключение к нашей базе данных:

require "rb.php";

R::setup('mysql:host=localhost;dbname=OLO\_Store', 'root', 'mysql');

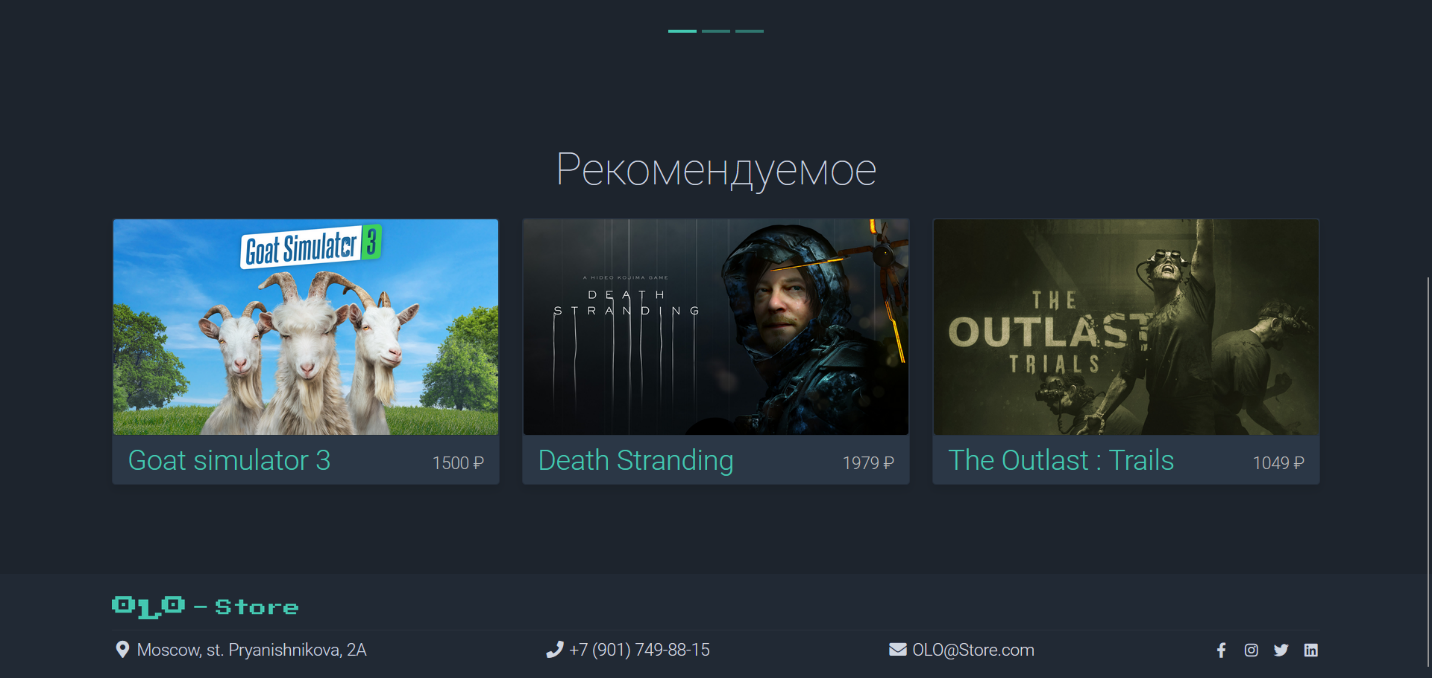
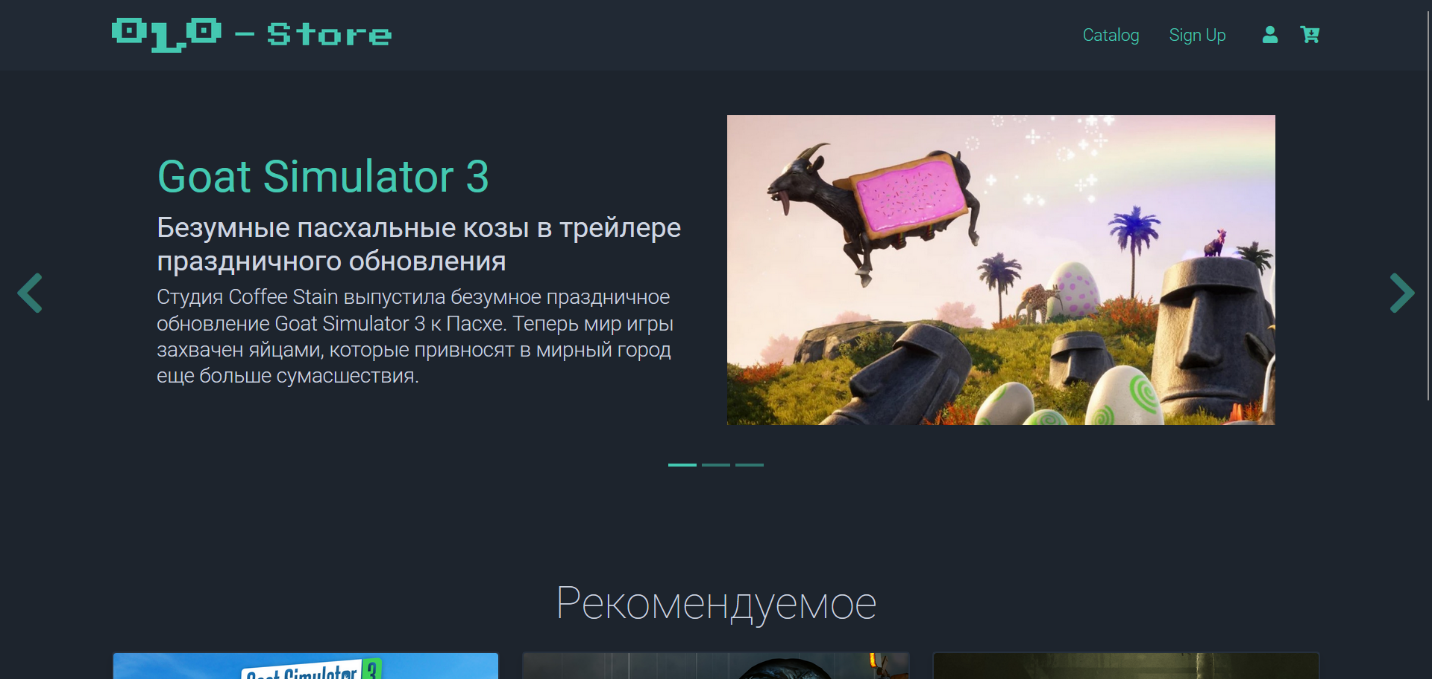
*Листинг 7. Подключение библиотеки rb.php*

Вызовем функции, которая поможет нам сохранить сессию одного пользователя, при переходе между страницами:

session\_start();

*Листинг 8. Функция создания сессии*

Теперь перейдем к созданию файла, который будет являться главной страницей нашего будущего веб-приложения.



*Рисунок 10. Главная страница*

Назовем его “index.php” и первым делом подключим к нему файл, сформированный ранее:

<?php

require "bd.php";

?>

*Листинг 9. Подключение файла bd.php*

Чтобы корректно отображать навигацию на главной странице, а также на всех остальных страницах веб-приложения, мы будем использовать функцию, которая будет проверять авторизацию пользователя. Если пользователь авторизован, то в меню навигации будут отображаться кнопки для доступа к каталогу и выхода из сессии. В противном случае будут отображаться кнопки для регистрации и авторизации :

<?php if (isset($\_SESSION['logged\_user'])) :?>

                        <a class="nav-link " href="shop.php"> Catalog </a>

                        <a class="nav-link text-success" href="log\_out.php">Log Out</a>

                    <?php else: ?>

                        <a class="nav-link" href="shop.php"> Catalog </a>

                        <a class="nav-link" href="reg.php"> Sign Up </a>

                        <a class="nav-icon position-top text-decoration-none" href="auth.php">

                        <i class="fa fa-fw fa-user mr-3"></i>

                        </a>

                        <?php endif; ?>

*Листинг 10. Проверка текущей сессии*

Следующим важным шагом в проектировании веб-приложения является отображение игр и связанной с ними информации на главной странице, которая будет получена из базы данных.

<?php

                    $sql = mysqli\_query($conn, 'SELECT \* FROM `game`');

                    $count = 0;

                    while (($game = mysqli\_fetch\_assoc($sql)) && $count < 3) {

                        $count++;

                    ?>

                        <div class="col-12 col-md-4 mb-4">

                            <div class="card mb-4 product-wap">

                                <div class="card h-75 rounded-0">

                                    <img class="card-img-top rounded-1 img-fluid h-100" src="assets/img/<?php print\_r($game['image']) ?>.jpg">

                                    <div class="card-img-overlay rounded-0 product-overlay d-flex align-items-center justify-content-center">

                                        <ul class="list-unstyled">

                                            <li><a class="btn btn-success text-white mt-2" href="game.php?game\_id=<?php echo $game['game\_id'] ?>"><i class="far fa-eye"></i></a></li>

                                            <li><a class="btn btn-success text-white mt-2" href="cart.php?game\_id=<?php echo $game['game\_id'] ?>"><i class="fas fa-cart-plus"></i></a></li>

                                        </ul>

                                    </div>

                                </div>

                                <div class="d-flex justify-content-between align-items-center mt-2 mx-3">

                                    <a href="<?php echo $game['link'] ?>.php" class="h2 text-success text-decoration-none text-dark"><?php echo $game['title']; ?></a>

                                    <div class="text-muted"><?php echo $game['price']; ?> ₽</div>

                                </div>

                            </div>

                        </div>

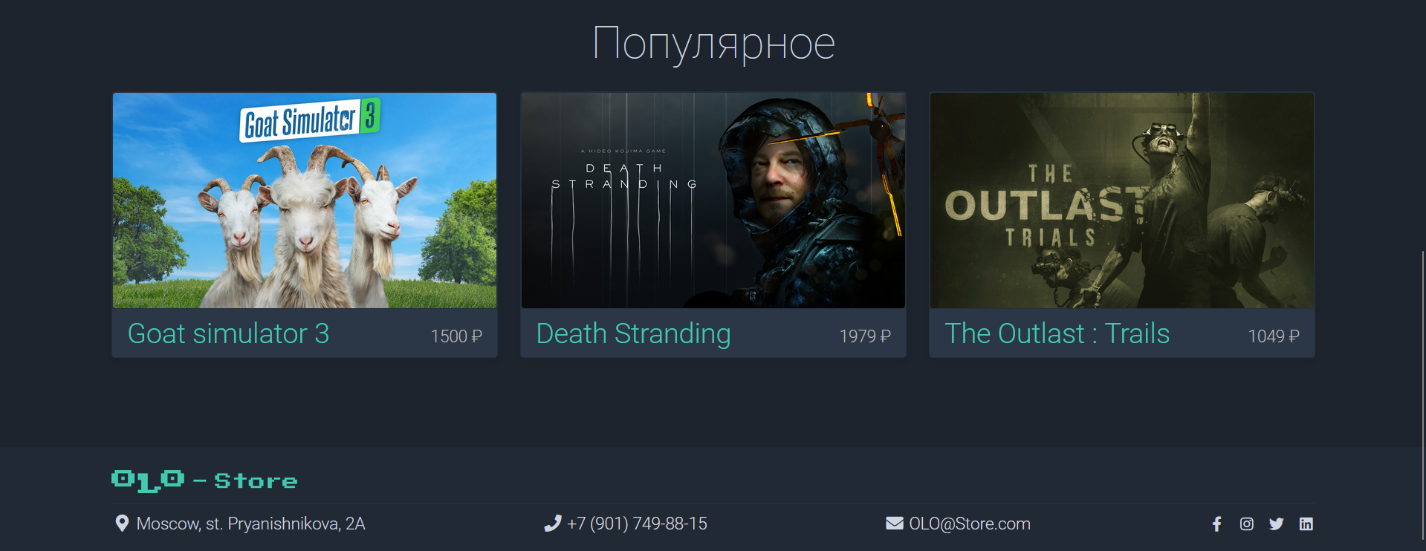
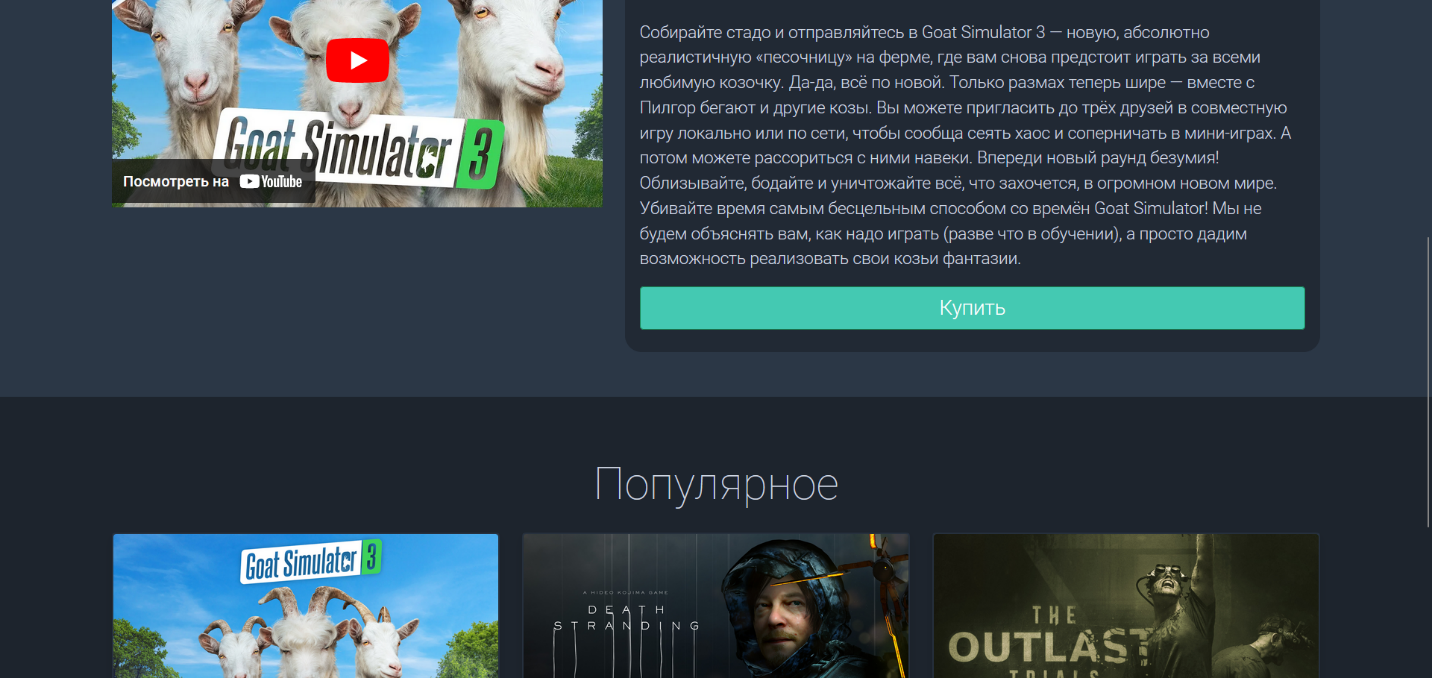
                    <?php

                    }

                    ?>

*Листинг 11. Подключение файла bd.php*

Рассмотрим страницу игры на примере Goat Simulator 3:



*Рисунок 11. Страница игры*

Также подключим файл «bd.php» и осуществим проверку текущей сессии.

Рассмотрим механизм получения информации об игре из базы данных:

if (isset($\_GET['game\_id']) && !empty($\_GET['game\_id'])) {

    $game = get\_game\_by\_id($\_GET['game\_id']);

}

Произведем вывод основной информации об игре:

<section class="bg-gray">

        <div class=" container pb-5">

            <div class="row">

                <div class="col-lg-5 mt-5">

                    <div class=" mb-3">

                        <img src="assets/img/<?php print\_r($game['image']) ?>.jpg" class="card-img-top">

                    </div>

                    <div class="row">

                        <iframe width="560" height="315" src="<?php echo ($game['youtube\_link']) ?>" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

                    </div>

                </div>

                <!-- col end -->

                <div class=" col-lg-7 mt-5">

                    <div>

                        <div class=" text-light bg-dark card-body">

                            <h1 class="h1"><?php echo $game['title'] ?></h1>

                            <p class="h2 text-right btn btn-success "><?php echo $game['price'] ?> ₽ </p>

                            <p class="py-2">

                                <span class="list-inline-item "><strong>Рейтинг: </strong> <?php echo $game['rate'] ?></span>

                            </p>

                            <p class="py-1">

                                <span class="list-inline-item "><strong>Дата выхода: </strong> <?php echo $game['release\_date'] ?></span>

                            </p>

                            <ul class="list-inline">

                                <li class="list-inline-item">

                                    <h6>Разработчик: </h6>

                                </li>

                                <li class="list-inline-item">

                                    <p class="text-muted"><strong><?php

                                                                    $game\_developer\_id = $game['game\_developer\_id'];

                                                                    $sql\_dev = mysqli\_query($conn, "SELECT \* FROM `game\_developer` WHERE id = " . $game\_developer\_id);

                                                                    $game\_developer = mysqli\_fetch\_assoc($sql\_dev);

                                                                    echo $game\_developer['name'] ?></strong></p>

                                </li>

                                <li class="list-inline-item">

                                    <h6>Платформа: </h6>

                                </li>

                                <li class="list-inline-item">

                                    <p class="text-muted"><strong><?php echo $game['platform'] ?></strong></p>

                                </li>

                            </ul>

                            <h6>Описание:</h6>

                            <p><?php echo $game['description'] ?></p>

                            <div class="col d-grid">

                                <a class="h3 text-decoration-none text-white btn btn-success" href="cart.php?game\_id=<?php echo $game['game\_id'] ?>">Купить</a>

                            </div>

                        </div>

                    </div>

                </div>

            </div>

        </div>

*Листинг 12. Вывод основной информации об игре*

Ниже на странице отображаются варианты самых популярных игр:

<?php

                    $game\_id = $game['game\_id'];

                    $sql = mysqli\_query($conn, 'SELECT \* FROM `game`');

                    $count = 0;

                    while (($game = mysqli\_fetch\_assoc($sql)) && $count < 3) {

                        $count++;

                    ?>

                        <div class="col-12 col-md-4 mb-4">

                            <div class="card mb-4 product-wap">

                                <div class="card h-75 rounded-0">

                                    <img class="card-img-top rounded-1 img-fluid h-100" src="assets/img/<?php print\_r($game['image']) ?>.jpg">

                                    <div class="card-img-overlay rounded-0 product-overlay d-flex align-items-center justify-content-center">

                                        <ul class="list-unstyled">

                                            <li><a class="btn btn-success text-white mt-2" href="game.php?game\_id=<?php echo $game['game\_id'] ?>"><i class="far fa-eye"></i></a></li>

                                            <li><a class="btn btn-success text-white mt-2" href="cart.php?game\_id=<?php echo $game['game\_id'] ?>"><i class="fas fa-cart-plus"></i></a></li>

                                        </ul>

                                    </div>

                                </div>

                                <div class="d-flex justify-content-between align-items-center mt-2 mx-3">

                                    <a href="<?php echo $game['link'] ?>.php" class="h2 text-success text-decoration-none text-dark"><?php echo $game['title']; ?></a>

                                    <div class="text-muted"><?php echo $game['price']; ?> ₽</div>

                                </div>

                            </div>

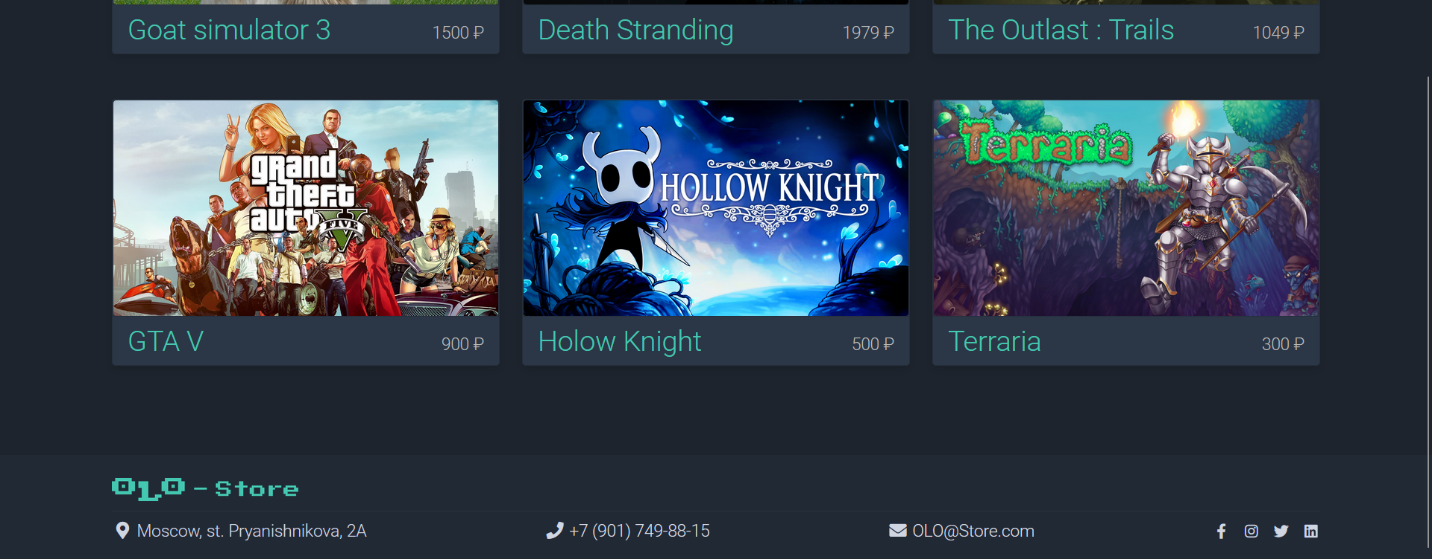
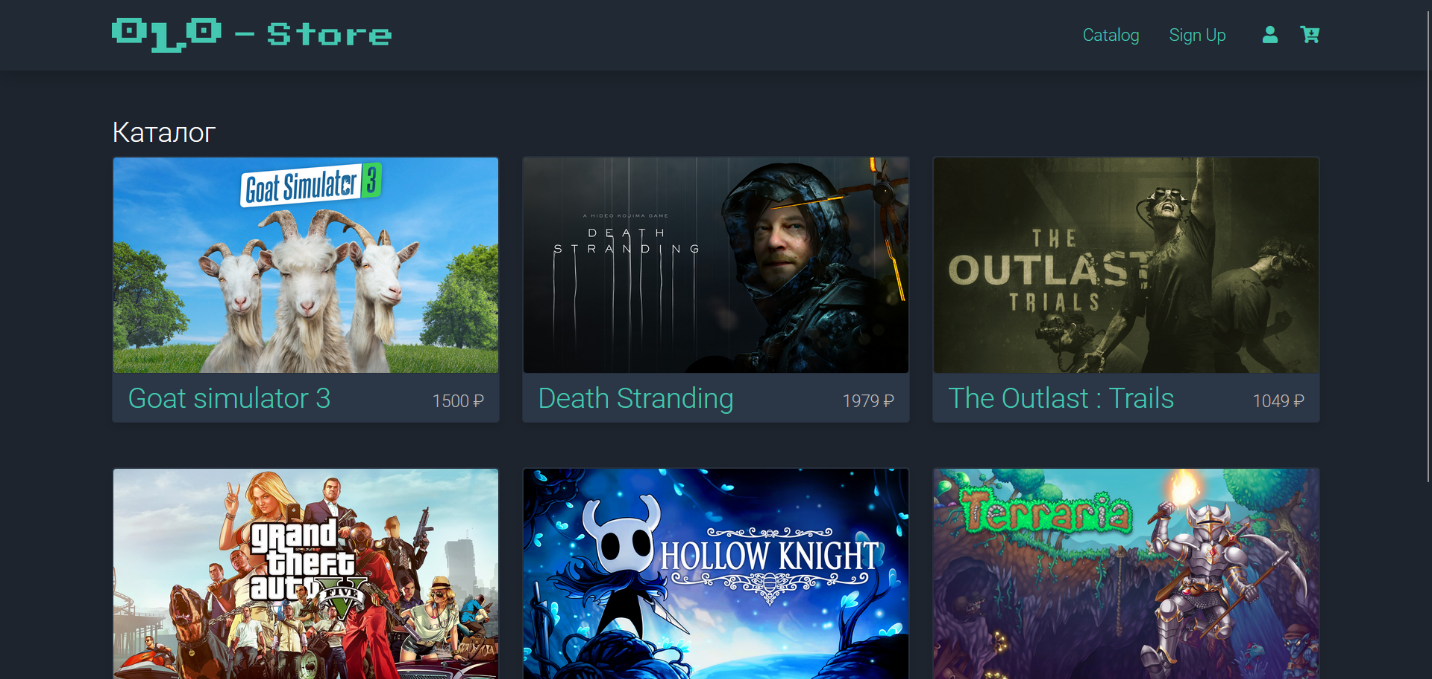
                        </div>

                    <?php

                    }

                    ?>

*Листинг 13. Подключение файла bd.php*

Рассмотрим страницу каталога (также подключим файл «bd.php» и осуществим проверку текущей сессии.) 

*Рисунок 12. Страница каталога*

Произведем вывод всех игр из базы данных (используется цикл «while»):

<?php

                    $sql = mysqli\_query($conn, 'SELECT \* FROM `game`');

                    while ($game = mysqli\_fetch\_assoc($sql)) {

                    ?>

                        <div class="col-12 col-md-4 mb-4">

                            <div class="card mb-4 product-wap">

                                <div class="card h-75 rounded-0">

                                    <img class="card-img-top rounded-1 img-fluid h-100" src="assets/img/<?php print\_r($game['image']) ?>.jpg">

                                    <div class="card-img-overlay rounded-0 product-overlay d-flex align-items-center justify-content-center">

                                        <ul class="list-unstyled">

                                            <li><a class="btn btn-success text-white mt-2" href="game.php?game\_id=<?php echo $game['game\_id'] ?>"><i class="far fa-eye"></i></a></li>

                                            <li><a class="btn btn-success text-white mt-2" href="cart.php?game\_id=<?php echo $game['game\_id'] ?>"><i class="fas fa-cart-plus"></i></a></li>

                                        </ul>

                                    </div>

                                </div>

                                <div class="d-flex justify-content-between align-items-center mt-2 mx-3">

                                    <a href="<?php echo $game['link'] ?>.php" class="h2 text-success text-decoration-none text-dark"><?php echo $game['title']; ?></a>

                                    <div class="text-muted"><?php echo $game['price']; ?> ₽</div>

                                </div>

                            </div>

                        </div>

                    <?php

                    }

                    ?>

*Листинг 14. Вывод всех игр из базы данных*

Дальше перейдем к системе регистрации. Также к ней подключим файл «bd.php». На странице мы расположим окно регистрации:



*Рисунок 13. Страница регистрации*

Регистрация будет осуществляться при помощи данной функции, которая вносит данные, введенные пользователем, в базу данных, а также осуществляет проверку на корректность (в случае, если данные не прошли одну из типовых проверок, ошибке присваивается имя, которое затем заносится в массив ошибок «errors» :

if (isset($\_POST['signup'])) {

    $errors = array();

    $showError = True;

    if (trim($\_POST['login']) == '') {

        $errors[] = 'Set Login';

    }

    if (trim($\_POST['email']) == '') {

        $errors[] = 'Set Email';

    }

    if (trim($\_POST['password']) == '') {

        $errors[] = 'Set Password';

    }

    if (trim($\_POST['password']) != trim($\_POST['password\_2'])) {

        $errors[] = 'Incorrect Password';

    }

    if (R::count('users', 'email = ?', array($\_POST['email'])) > 0) {

        $errors[] = 'Existing email';

    }

    if (empty($errors)) {

        $user = R::dispense('users');

        $user->login = $\_POST['login'];

        $user->email = $\_POST['email'];

        $user->password = password\_hash($\_POST['password'], PASSWORD\_DEFAULT);

        R::store($user);

    }

}

?>

*Листинг 15. Регисрация*

Окно регистрации реализовано следующим образом: при наличии ошибки, она будет отображаться на экране пользователя :

<div class="container-u">

        <div class="footer\_user bg-gray">

            <div class="bg-gray items">

                <form action="reg.php" method="post" >

                    <h1 class="text-success">Регистрация </h1>

                    <input class="search" type="text" name="login" placeholder="login"> <br>

                    <input class="search" type="text" name="email" placeholder="email"> <br>

                    <input class="search" type="text" name="password" placeholder="password"> <br>

                    <input class="search" type="text" name="password\_2" placeholder="password confirm"> <br>

                    <button class="btn btn-success" type="submit" name="signup"> Зрегистрироваться</button>

                </form>

                <p class="text-light pt-3">

                    <?php if ($showError) {

                        echo showError($errors);

                    } ?>

                </p>

            </div>

        </div>

    </div>

*Листинг 16. Окно регистрации*

Рассмотрим страницу авторизации. Также к ней подключим файл «bd.php». На странице мы расположим окно авторизации:



*Рисунок 14. Страница авторизации*

Для проверки соответствия данных, введенных пользователем, с базой данных, мы используем данную функцию. В случае несоответствия и возникновения ошибки, ей присваивается имя, которое затем добавляется в массив ошибок :

if (isset($data['signin'])) {

    $errors = array();

    $showError = True;

    $user = R::findOne('users', 'email = ?', array($data['email']));

    if (trim($data['email']) == '') {

        $errors[] = 'Set Email';

    }

    if (trim($data['password']) == '') {

        $errors[] = 'Set Password';

    }

    if ($user) {

        if (password\_verify($data['password'], $user->password)) {

            $\_SESSION['logged\_user'] = $user;

        } else {

            $errors[] = 'Wrong password';

        }

    } else {

        $errors[] = 'User does not exist';

    }

}

}

*Листинг 17. Авторизация*

Окно авторизации реализовано с учетом следующих особенностей. В случае возникновения ошибки, она отображается непосредственно на экране для пользователя:

<div class="container-u">

        <div class="footer\_user bg-gray">

            <div class="bg-gray items">

                <form action="auth.php" method="post">

                    <h1 class="text-success">Авторизация</h1>

                    <input class="search" type="text" name="email" placeholder="email"> <br>

                    <input class="search" type="text" name="password" placeholder="password"> <br>

                    <button class="btn btn-success" type="submit" name="signin"> Войти</button>

                </form>

                <p class="text-light pt-3">

                    <?php if ($showError) {

                        echo showError($errors);

                    } ?></p>

            </div>

        </div>

    </div>

*Листинг 18. Окно авторизации*

Перейдем к следующему шагу - корзине. Давайте рассмотрим процесс добавления игры в корзину:

if (isset($\_GET['game\_id']) && !empty($\_GET['game\_id'])) {

    $current\_added\_game = get\_game\_by\_id($\_GET['game\_id']);

    if (!empty($current\_added\_game)) {

        if (!isset($\_SESSION['cart\_list'])) {

            $\_SESSION['cart\_list'][] = $current\_added\_game;

        }

        $game\_check = false;

        if (isset($\_SESSION['cart\_list'])) {

            foreach ($\_SESSION['cart\_list'] as $value) {

                if ($value['game\_id'] == $current\_added\_game['game\_id']) {

                    $game\_check = true;

                }

            }

        }

        if (!$game\_check) {

            $\_SESSION['cart\_list'][] = $current\_added\_game;

        }

    }

}

*Листинг 18. Добавление игры в корзину*

Механизм удаления игры из корзины реализован следующим образом:

if (isset($\_GET['delete\_game\_id']) && isset($\_SESSION['cart\_list'])) {

    if (isset($\_SESSION['cart\_list'])) {

        foreach ($\_SESSION['cart\_list'] as $key => $value) {

            if ($value['game\_id'] == $\_GET['delete\_game\_id']) {

                unset($\_SESSION['cart\_list'][$key]);

            }

        }

    }

}

*Листинг 19. Удаление игры из корзину*

Для вывода списка игр, добавленных пользователем, используется цикл "foreach". Вот пример кода, демонстрирующий этот механизм:

<?php

            if (isset($\_SESSION['cart\_list']) && count($\_SESSION['cart\_list']) != 0) : ?>

                <?php

                foreach ($\_SESSION['cart\_list'] as $game) : ?>

                    <div class="col-md-4 ">

                        <div class="card mb-4 product-wap">

                            <div class="card-body ">

                                <div class="d-flex justify-content-between align-items-center">

                                    <a href="<?php echo $game['link'] ?>.php" class="h2 text-success text-decoration-none text-dark"><?php echo $game['title']; ?></a>

                                    <div class="text-muted"><?php echo $game['price']; ?> ₽</div>

                                </div>

                                <div class="col d-grid">

                                    <a class="h3 text-decoration-none text-white btn btn-red" href="cart.php?delete\_game\_id=<?php echo $game['game\_id']; ?>">Удалить</a>

                                </div>

                            </div>

                        </div>

                    </div>

                <?php endforeach;

            else : ?>

                <p class="text-light">'Корзина пустая'</p>

            <?php endif; ?>

*Листинг 20. Удаление игры из корзину*

Когда все функции и страницы разработаны, наше веб-приложение готово к работе. При первом посещении главной страницы (рисунок 8) пользователь видит интуитивно понятный интерфейс, что позволяет даже неопытному пользователю легко разобраться в использовании приложения. На главной странице можно перейти к отдельной странице любой выбранной игры или перейти в каталог всех игр. Кроме того, на каждой странице веб-приложения есть ссылки на каталог и корзину, а если пользователь не авторизован, есть ссылки на регистрацию и авторизацию.

При переходе на страницу выбранной игры (рисунок 9), пользователь видит основную информацию о ней, а также ссылки на самые популярные игры на текущий момент.

При переходе на страницу каталога (рисунок 10), пользователь видит список всех игр, доступных в базе данных.

При посещении страницы корзины (рисунок 13), пользователю отображается список всех добавленных игр в текущей сессии. Также есть возможность удаления ненужных игр из корзины.

На странице регистрации (рисунок 11) пользователь может ввести свои данные, которые будут добавлены в базу данных.

На странице авторизации (рисунок 12) пользователь вводит данные, и при успешной проверке происходит вход в аккаунт.

Таким образом, разработанное веб-приложение действительно является удобным и простым в использовании для любого пользователя.

ЗАКЛЮЧЕНИЕ

Цель данного проекта была успешно достигнута. Информационная система была реализована с использованием языка SQL для запросов к базе данных, а также с помощью набора дистрибутивов Denwer, которые обеспечили создание локальной серверной среды. В результате работы над проектом были приобретены следующие навыки:

Проектирование баз данных: был разработан структурированный и эффективный дизайн базы данных, учитывающий основные требования информационной системы.

Создание веб-приложения: было разработано веб-приложение с использованием языка программирования PHP, которое обеспечивает пользователю удобный интерфейс для поиска новых игр и их безопасной установки.

Подключение собственной базы данных к веб-приложению: было осуществлено подключение созданной базы данных к веб-приложению, что позволило эффективно хранить и получать данные, необходимые для функционирования системы.

В результате выполнения проекта было создано функциональное веб-приложение, которое значительно облегчает поиск новых игр пользователем и обеспечивает их безопасную установку.

Библиографический список

1. Попов Д.И., Попова Е.Д. Информационные технологии. Базы данных [Текст]: учеб. пособие для студ. вузов /Д.И. Попов. – М: гос. ун-т, печати, 2009. – 117 с.
2. Курсы в ЛМС по дисциплине «Базы данных» за 3, 4 семестры.
3. Руководство по PHP [Электронный ресурс]. - http://php.net/manual/ru/ - статья в интернете.
4. Руководство по MySQL [Электронный ресурс]. - https://dev.mysql.com/doc/ - статья в интернете.
5. Энциклопедия [Электронный ресурс]. - <https://ru.wikipedia.org>
6. Видео курсы - YouTube