БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Международный институт дистанционного образования

Кафедра «Информационные технологии в управлении»

Контрольная работа

по дисциплине: “Разработка веб-приложений”

Выполнил:

Прутенская Е.А.

группы 41702119

Проверил:

Хвитько Е.А.

**ОГЛАВЛЕНИЕ**

[Введение 3](#_gjdgxs)

[Планирование 4](#_rl6bhsg818rf)

[Реализация 7](#_oiitfmob5uuo)

[Tестирование 6](#_3znysh7)

[Листинг Кода 10](#_2et92p0)

[Вывод 15](#_3dy6vkm)

# Введение

Задание: Разработка веб-приложения для интернет-магазина одежды с разделением ролей «Администратор/Пользователь».

В ходе выполнения проекта были определены функциональные и нефункциональные требования к приложению, спроектирована его архитектура, реализованы основные функции и проведено тестирование.  
Данное приложение предоставляет удобный и эффективного интерфейса для пользователей, позволяющего осуществлять поиск, просмотр и покупку одежды, а также управление каталогом товаров. Разработка приложения осуществлялась с учетом современных технологий и требований безопасности.

Приложение написано на PHP с использованием Bootstrap, CSS и HTML, для управления базой данных использовалась СУБД MySQL и phpMyAdmin для администрирования СУБД, также использовался Apache HTTP-сервер.

# Планирование

Цель разработки веб-приложения

Целью проекта является предоставление удобной и интуитивно понятной интернет-платформы для онлайн-покупок. Создание функциональной системы управления каталогом товаров, позволяющей пользователям эффективно искать, фильтровать и совершать покупки в соответствии с их потребностями. А также предоставление инструментов администрирования для эффективного управления каталогом товаров, категориями товаров и пользователями.

Целевая аудитория

Веб-приложение предназначено для широкой аудитории: от покупателей, которые ищут удобный и безопасный способ осуществления покупок онлайн, до менеджеров и администраторов магазинов, которые управляют онлайн-магазинами и используют приложение для удобного управления каталогом товаров, категориями товаров и пользователями.

Анализ требований

Для успешной разработки веб-приложения интернет-магазина были выявлены и проанализированы функциональные и нефункциональные требования.

Функциональные требования включают в себя следующие основные элементы:

1. Регистрация и авторизация: необходимость предоставить пользователям функционал для создания учетной записи, а также для входа в систему с использованием логина и пароля.

2. Поиск и фильтрация товаров: пользователям должна быть предоставлена возможность осуществлять поиск товаров по различным критериям, таким как название, состав, категория и т.д.

3. Функционал для администрирования: администратору должна быть предоставлена возможность добавлять, изменять и удалять товары из каталога, управлять описаниями, изображениями и ценами. А также возможность управления каталогами товаров и пользователями.

Нефункциональные требования включают в себя:

1. Безопасность: гарантировать защищенность данных пользователей.

2. Производительность: обеспечить высокую скорость загрузки страниц, быструю обработку запросов и отзывчивый интерфейс.

3. Пользовательский интерфейс: разработка привлекательного, интуитивно понятного и удобного интерфейса для пользователей.

Структура

Физическая структура веб-приложения обеспечивает легкий и интуитивно понятный пользовательский интерфейс, позволяющий удобно навигироваться по сайту и находить необходимые товары. Логическая структура оптимизирована для быстрого доступа к информации и удовлетворения потребностей разнообразной аудитории.

Архитектура приложения

Для разработки интернет-магазина была выбрана модель клиент-сервер с применением многоуровневой архитектуры. На серверной стороне используется язык программирования PHP для обработки запросов и взаимодействия с базой данных MySQL. На клиентской стороне используется HTML, CSS и JavaScript для создания интерактивного пользовательского интерфейса.

Структура базы данных

База данных была спроектирована с учетом хранения информации о пользователях, продуктах, каталогах и других необходимых данных. Были созданы соответствующие таблицы и определены связи между ними, чтобы обеспечить эффективное хранение и доступ к информации.

Дизайн пользовательского интерфейса

При разработке интерфейса уделено особое внимание удобству использования и привлекательности дизайна. Использован фреймворк Bootstrap для создания адаптивного и современного пользовательского интерфейса. Также были продуманы интерактивные элементы, удобная навигация по сайту, отображение товаров и админ-панель для администрирования.

# 

# Реализация

1. Разработка серверной части

На серверной части приложения были реализованы основные функции, включая обработку запросов пользователей, аутентификацию и авторизацию, управление товарами и т.д. Для этого были созданы соответствующие PHP скрипты, обеспечивающие взаимодействие с базой данных MySQL.

2. Разработка клиентской части

Для клиентской части приложения были разработаны веб-страницы с использованием HTML для структуры, CSS для стилей и JavaScript для интерактивности. Страницы были адаптированы под различные устройства с помощью фреймворка Bootstrap, что обеспечило плавное и удобное взаимодействие с пользователем.

3. Безопасность

Для обеспечения безопасности приложения были применены современные методы защиты, включая фильтрацию пользовательского ввода, шифрование паролей пользователей. Это позволяет предотвратить возможные атаки и утечки данных.

# Tестирование

После завершения разработки приложения было проведено тестирование, включающее функциональное тестирование, тестирование безопасности, а также тестирование совместимости. Это позволило выявить и устранить возможные ошибки и недочеты в работе приложения.

1. Функциональное тестирование

В процессе тестирования проводилась проверка основных функциональных возможностей приложения, таких как регистрация и вход пользователей, поиск и добавление товаров, управление каталогом товаров, а также административные функции.

2. Тестирование безопасности

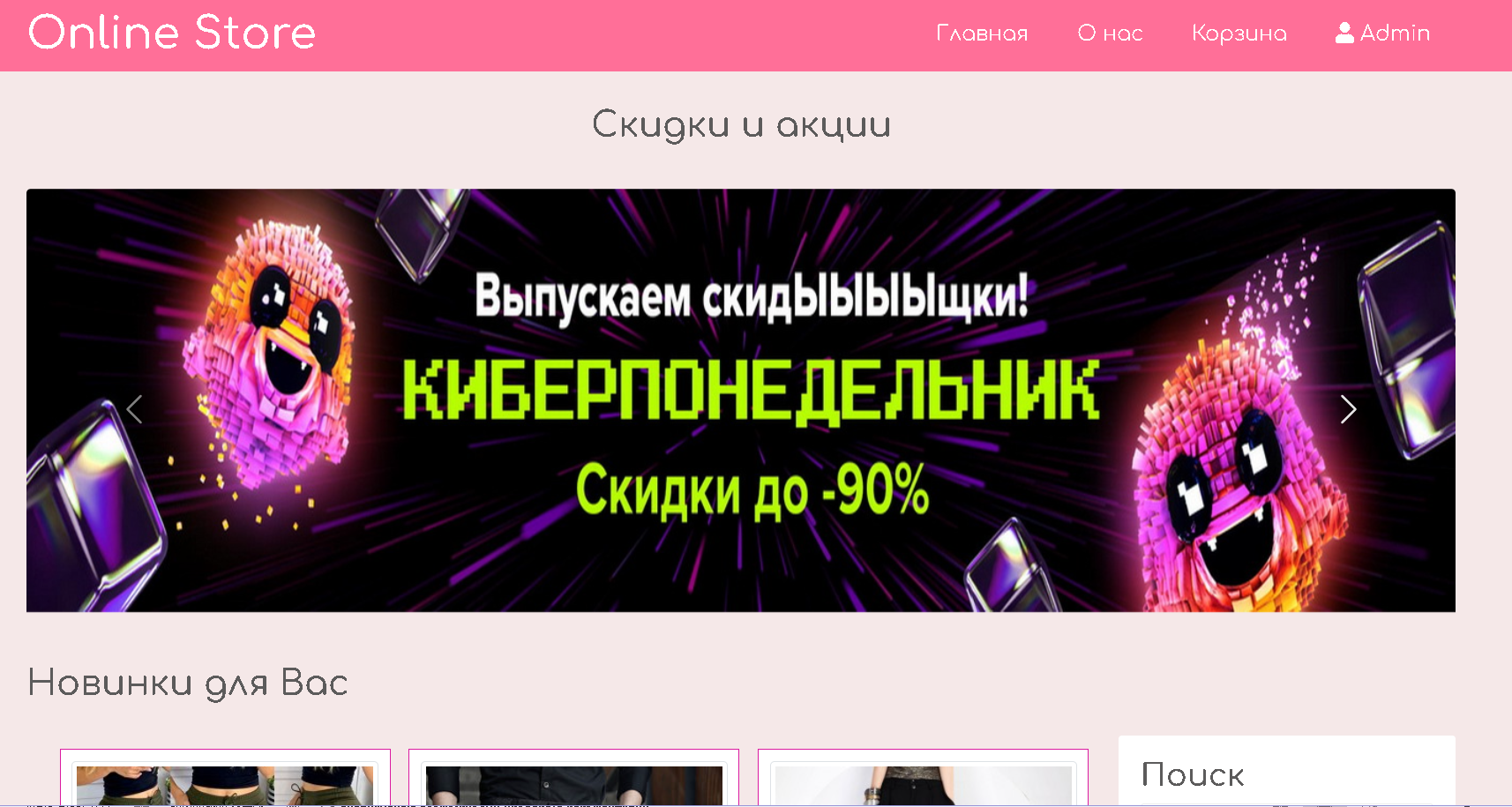
Тестированием приложения были проверены механизмы аутентификации и авторизации для обеспечения безопасности пользовательских данных.

4. Тестирование совместимости

Приложение было протестировано в различных веб-браузерах и с различными разрешениями экранов.

5. Тестирование пользовательского интерфейса

Было проведено тестирование удобства использования и навигации по интерфейсу приложения, а также проверка его соответствия установленным стандартам дизайна.

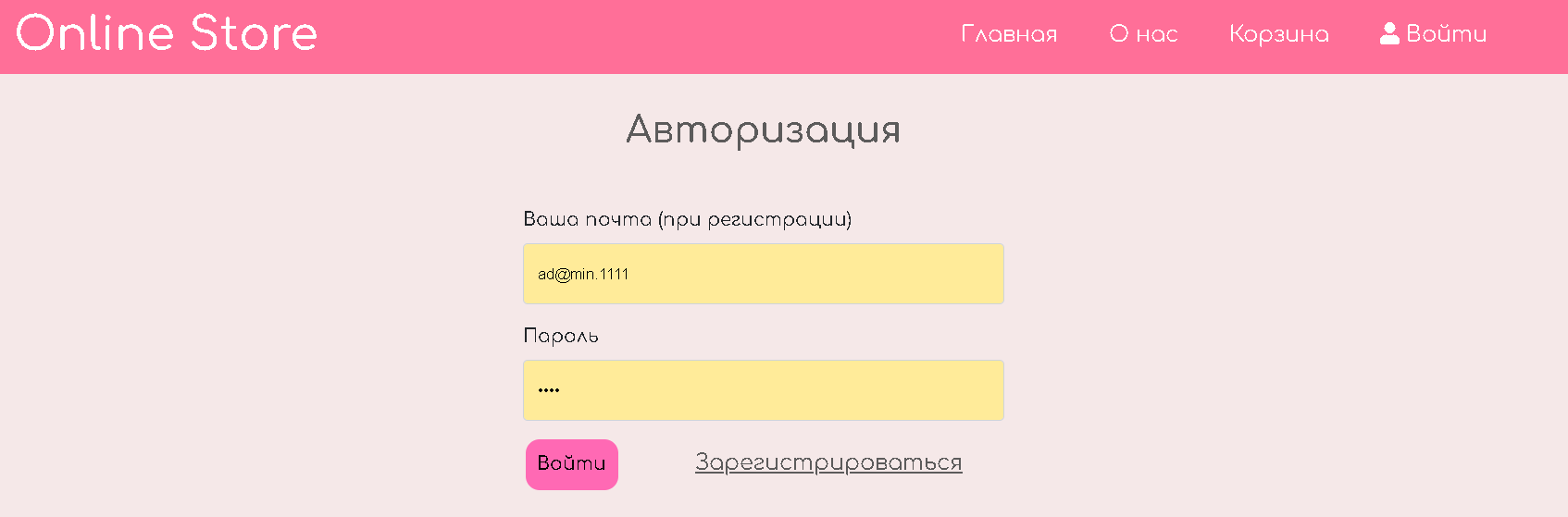


Главная страница

Тест кейс:

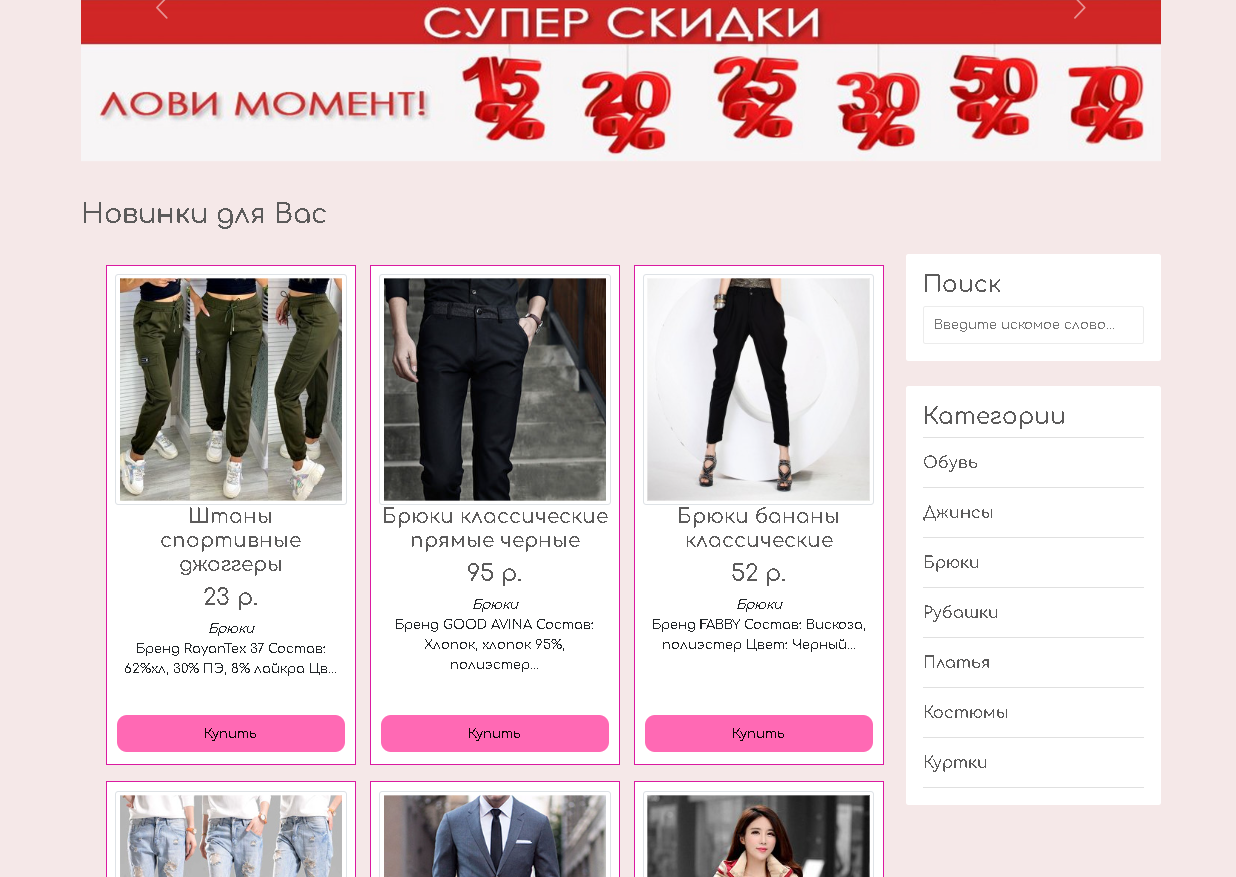
1. нажать кнопку “login” и перейти на страницу авторизации
2. ввести данные для авторизации
3. перейти на страницу магазина

Нажав на кнопку “login” происходит переход на страницу авторизации:



Страница авторизации

Затем вводим данные для авторизации, происходит перенаправление на страницу каталога магазина:



Страница каталога магазина

тест кейс был выполнен успешно.

# Листинг Кода

**Подключение к базе данных:**

<?php

$driver = 'mysql';

$host = 'localhost';

$db\_name = 'onlinestore';

$db\_user = 'root';

$db\_pass = 'mysql';

$charset = 'utf8';

$options = [PDO::ATTR\_ERRMODE => PDO::ERRMODE\_EXCEPTION,

PDO::ATTR\_DEFAULT\_FETCH\_MODE => PDO::FETCH\_ASSOC];

try{

$pdo = new PDO(

"$driver:host=$host;dbname=$db\_name;charset=$charset", $db\_user, $db\_pass, $options

);

}catch (PDOException $i){

die("Ошибка подключения к базе данных");

}

**Контроллер взаимодействия с данными пользователя «users.php»**

<?php include "e:/serverAmpps/Ampps/www/OnlineStore/app/database/db.php";

$errMsg = [];

function userAuth($user){

$\_SESSION['id'] = $user['id'];

$\_SESSION['login'] = $user['username'];

$\_SESSION['userroot'] = $user['userroot'];

if($\_SESSION['userroot']){

header('location: ' . "http://localhost/OnlineStore/" . "admin/users/index.php");

}else{

header('location: ' . "http://localhost/OnlineStore/");

}

}

$users = selectAll('users');

// Код для формы регистрации

if($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['button-reg'])){

$userroot = 0;

$login = trim($\_POST['login']);

$email = trim($\_POST['mail']);

$passF = trim($\_POST['pass-first']);

$passS = trim($\_POST['pass-second']);

$adres = trim($\_POST['adres']);

if($login === '' || $email === '' || $passF === '' || $adres === '' ){

array\_push($errMsg, "Не все поля заполнены!");

}elseif (mb\_strlen($login, 'UTF8') < 2){

array\_push($errMsg, "Логин должен быть более 2-х символов");

}elseif ($passF !== $passS) {

array\_push($errMsg, "Пароли в обеих полях должны соответствовать!");

}else{

$existence = selectOne('users', ['email' => $email]);

if($existence['email'] === $email){

array\_push($errMsg, "Пользователь с такой почтой уже зарегистрирован!");

}else{

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

$post = [

'userroot' => $userroot,

'username' => $login,

'email' => $email,

'password' => $pass,

'adres'=> $adres

];

$id = insert('users', $post);

$user = selectOne('users', ['id' => $id] );

userAuth($user);

}

}

}else{

$login = '';

$email = '';

}

// Код для формы авторизации

if($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['button-log'])){

$email = trim($\_POST['mail']);

$pass = trim($\_POST['password']);

if($email === '' || $pass === '') {

array\_push($errMsg, "Не все поля заполнены!");

}else{

$existence = selectOne('users', ['email' => $email]);

if($existence && password\_verify($pass, $existence['password'])){

userAuth($existence);

}else{

array\_push($errMsg, "Почта либо пароль введены неверно!");

}

}

}else{

$email = '';

}

// Код добавления пользователя в админке

if($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['create-user'])){

$userroot = 0;

$login = trim($\_POST['login']);

$email = trim($\_POST['mail']);

$passF = trim($\_POST['pass-first']);

$passS = trim($\_POST['pass-second']);

$adres = trim($\_POST['adres']);

if($login === '' || $email === '' || $passF === '' || $adres === '' ){

array\_push($errMsg, "Не все поля заполнены!");

}elseif (mb\_strlen($login, 'UTF8') < 2){

array\_push($errMsg, "Логин должен быть более 2-х символов");

}elseif ($passF !== $passS) {

array\_push($errMsg, "Пароли в обеих полях должны соответствовать!");

}else{

$existence = selectOne('users', ['email' => $email]);

if($existence['email'] === $email){

array\_push($errMsg, "Пользователь с такой почтой уже зарегистрирован!");

}else{

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

if (isset($\_POST['userroot'])) $userroot = 1;

$user = [

'userroot' => $userroot,

'username' => $login,

'email' => $email,

'password' => $pass,

'adres'=> $adres

];

$id = insert('users', $user);

$user = selectOne('users', ['id' => $id] );

header('location: ' . "http://localhost/OnlineStore/" . 'admin/users/index.php');

}

}

}else{

$login = '';

$email = '';

}

// Код удаления пользователя в админке

if($\_SERVER['REQUEST\_METHOD'] === 'GET' && isset($\_GET['delete\_id'])){

$id = $\_GET['delete\_id'];

delete('users', $id);

header('location: ' . "http://localhost/OnlineStore/" . 'admin/users/index.php');

}

// РЕДАКТИРОВАНИЕ ПОЛЬЗОВАТЕЛЯ ЧЕРЕЗ АДМИНКУ

if($\_SERVER['REQUEST\_METHOD'] === 'GET' && isset($\_GET['edit\_id'])){

$user = selectOne('users', ['id' => $\_GET['edit\_id']]);

#tt($user);

$id = $user['id'];

$userroot = $user['userroot'];

$username = $user['username'];

$email = $user['email'];

$adres = $user['adres'];

}

if($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['update-user'])){

$id = $\_POST['id'];

$mail = trim($\_POST['mail']);

$login = trim($\_POST['login']);

$passF = trim($\_POST['pass-first']);

$passS = trim($\_POST['pass-second']);

$adres = trim($\_POST['adres']);

$userroot = isset($\_POST['userroot']) ? 1 : 0;

if($login === ''){

array\_push($errMsg, "Не все поля заполнены!");

}elseif (mb\_strlen($login, 'UTF8') < 2){

array\_push($errMsg, "Логин должен быть более 2-х символов");

}elseif ($passF !== $passS) {

array\_push($errMsg, "Пароли в обеих полях должны соответствовать!");

}else{

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

if (isset($\_POST['userroot'])) $userroot = 1;

$user = [

'userroot' => $userroot,

'username' => $login,

#'email' => $mail,

'password' => $pass,

'adres'=> $adres

];

$user = update('users', $id, $user);

header('location: ' . "http://localhost/OnlineStore/" . 'admin/users/index.php');

}

}

**Пример написания функции составления запроса в базу данных:**

// Проверка выполнения запроса к БД

function dbCheckError($query){

$errInfo = $query->errorInfo();

if ($errInfo[0] !== PDO::ERR\_NONE){

echo $errInfo[2];

exit();

}

return true;

}

// Запрос на получение данных с одной таблицы

function selectAll($table, $params = []){

global $pdo;

$sql = "SELECT \* FROM $table";

if(!empty($params)){

$i = 0;

foreach ($params as $key => $value){

if (!is\_numeric($value)){

$value = "'".$value."'";

}

if ($i === 0){

$sql = $sql . " WHERE $key=$value";

}else{

$sql = $sql . " AND $key=$value";

}

$i++;

}

}

$query = $pdo->prepare($sql);

$query->execute();

dbCheckError($query);

return $query->fetchAll();

}

# Вывод

В заключении, благодаря проектированию, реализации и тестированию данного веб-приложения была создана надежная и функциональная интернет-платформа, обеспечивающая удобство использования, безопасность и высокую производительность. Процесс проектирования включал в себя архитектурное планирование, разработку пользовательского интерфейса и определение структуры базы данных. Реализация приложения включала в себя разработку клиентской и серверной частей, обеспечение безопасности и проведение тестирования. Тестирование, в свою очередь, позволило выявить и устранить потенциальные проблемы и гарантировать качество работы сайта.