Частное учреждение

профессиональная образовательная организация

ТЕХНИКУМ «БИЗНЕС И ПРАВО»

ЗАЩИЩЕНО С ОЦЕНКОЙ

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

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

(подпись и ФИО руководителя)

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_г.

**КУРСОВАЯ РАБОТА**

|  |  |
| --- | --- |
| Тема | РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ ДЛЯ ВЕБ-ПРИЛОЖЕИНЯ «БЛОГ ДЛЯ КОМПАНИИ» |
| По МДК 09.01 | Проектирование и разработка веб-приложений |
| Выполнил | Тубальцев Н.С. |
| Курс, группа | 4 курс, группа ИС-4 |
| Специальность | 09.02.07 Информационные системы и программирование |
| Научный руководитель | Мартыненко В.А. |

г. Белореченск  
2024 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 3

1 АНАЛИТИЧЕСКАЯ ЧАСТЬ 5

1.1 Анализ предметной области 5

1.2 Цели и задачи выполняемой работы 5

1.3 Метод реализации процесса проектирования 6

1.4 Обоснование проектных решений 6

2 ПРОЕКТНАЯ ЧАСТЬ 9

2.1 Техническое задание 9

2.2 Установка и настройка среды разработки 12

2.3 Руководство пользователя 12

2.4 Руководство администратора 20

ЗАКЛЮЧЕНИЕ 24

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 25

ПРИЛОЖЕНИЕ

ВВЕДЕНИЕ

Актуальность создания блога о компании обусловлена необходимостью эффективного удовлетворения потребностей пользователей в мире в условиях современного цифрового мира. В эпоху стремительного технологического прогресса и увеличения числа людей, работающих и обучающихся удаленно, наличие качественных устройств становится критически важным. Оптимизация доступа к разной информации через блог о компаниии помогает решить проблему нехватки времени у потребителей и предоставляет им широкий выбор продукции под любые нужды – будь то игровые компьютеры, устройства для работы с графикой или просто надежные офисные машины.

Оценка эффективности блога должна включать анализ его функциональности, удобства использования и доступности для широкой аудитории. Грамотно спроектированный и реализованный блог о компании способен значительно повысить доступность компьютерной техники, обеспечивая пользователям возможность легко находить и приобретать необходимые устройства. Это, в свою очередь, положительно сказывается на производительности труда и качестве жизни потребителей.

Объект исследования:

* объектом исследования является разработка блога «блог для компаниии»

Предмет исследования:

- предметом исследования являются процессы взаимодействия пользователей с блогом, а также механизмы, обеспечивающие высокий уровень сервиса и удовлетворенности клиентов.

Цель исследования:

- целью исследования является создание эффективного блога компании, способствующий улучшению качества обслуживания клиентов и увеличению продаж компьютерной техники.

Задачи исследования:

- изучить рынок компьютерных технологий и выявить актуальные потребности целевой аудитории;

- рассмотреть существующие блоги, определить их сильные и слабые стороны;

- выбрать ключевые функциональные требования к блогу, исходя из потребностей пользователей.

- спроектировать архитектуру и интерфейс блога, учитывающий удобство навигации и интуитивный дизайн;

- реализовать систему управления пользователями и избранными товарами, отображение продукции магазина;

- протестировать блог для компании с целью выявления и устранения возможных проблем.

Методы исследования:

- анализ существующих блогов на рынке электронной коммерции;

- тестирование визуальных и функциональных возможностей блога с участием реальных пользователей.

1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1 Анализ предметной области

Блог – информационная система, предназначенная для предоставления контента пользователям через Интернет. Его основная цель заключается в предоставлении пользователям возможности удобно и безопасно получать информацию о компьютерной технике, новостях и обзорах продуктов.

Для этого блог должен обеспечивать следующие функции:

* каталог статей с подробным описанием каждой статьи, фотографиями, в котором пользователь может пополнить блог, а администратор способен удалять, добавлять новые статьи;
* форма для поиска статей по ключевым словам.
* Система управления комментариями, которые будут отображаться под статьями.
* Личный кабинет, где пользователь может просматривать свои комментарии и подписки на статьи.

1.2 Цели и задачи выполняемой работы

Основными целями создания являются:

* обеспечение эффективного и организованного управления контентом и пользователями
* улучшение пользовательского опыта и увеличение объема продаж;
* помощь в предоставлении удобного интерфейса для ознакомления с блогом.

Для достижения поставленных целей система должна решать следующие задачи:

* предоставление возможности пользователям просматривать страницы магазина, а администраторами удалять товары пользователей;
* сортировка статей по категориям и тегам;
* предоставление возможности пользователям удобно просматривать данные, используя внедренные в систему функции.

1.3 Метод реализации процесса проектирования

При проектировании информационной системы был выбран метод типового проектирования ИС. Типовое проектирование ИС предполагает создание системы из готовых типовых элементов. Основополагающим требованием для применения методов типового проектирования является возможность декомпозиции, проектируемой ИС на множество составляющих компонентов (подсистем, комплексов задач, программных модулей и т.д.). Для достижения поставленных целей выбираются имеющиеся на рынке типовые проектные решения, которые настраиваются на особенности конкретного предприятия.

В процессе проектирования, исходя из выбранного метода, было выявлено некоторое множество составляющих компонентов, необходимых для целостности проекта:

* пользователи - регистрация и авторизация, хранение данных;
* задачи - создание, просмотр, удаление, сортировка, хранение данных;

1.4 Обоснование проектных решений

На данном этапе необходимо описать примерную архитектуру разрабатываемой системы.

Для того, чтобы создать информационную систему планирования рабочего дня, мы будем использовать архитектуру веб-приложения. Чтобы убедиться, что она нам точно подходит рассмотрим ее подробнее.

Архитектура веб-приложения – это структура, которая определяет, как веб-приложение должно быть построено. Она включает в себя различные компоненты, такие как пользовательский интерфейс, серверное приложение, базы данных и другие компоненты.

Пользовательский интерфейс: это средства, позволяющие пользователю эффективно взаимодействовать с устройствами компьютера достаточно удобным для себя образом. Визуально, пользовательский интерфейс представляет собой набор пиктограмм (иконок) для различных программ, кнопки для различных действий и окна, в которых отображаются данные при работе с той или иной программой. Современный интерфейс у многих программ визуально схож. Он может быть написан на HTML, CSS, JavaScript.

Серверное приложение: это то, что работает на сервере и обрабатывает запросы от пользовательского интерфейса. Это приложение, которое ожидает запросов от других приложений и отвечает на них, таким образом предоставляя "услугу" по их запросу. Сервер приложений обычно выделяется как среднее звено в трехуровневой клиент-серверной архитектуре (3-tier) Оно может быть написано на Python, Java, Node.js, Ruby, PHP и т.д.

База данных: это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляется системой управления базами данных (СУБД). Данные вместе с СУБД, а также приложения, которые с ними связаны, называются системой баз данных, или, для краткости, просто базой данных.

Архитектура: это способ организации компонентов веб-приложения для достижения целостности всей информационной системы. Это может быть монолитный, микро сервисный, серверный рендеринг, клиентский рендеринг, и т.д.

Преимущества архитектуры веб-приложения:

* масштабируемость - веб-приложения могут быть легко масштабированы, добавляя или удаляя сервера в зависимости от нагрузки;
* отказоустойчивость: если один сервер выйдет из строя, другие серверы могут продолжить обслуживать запросы;
* остойчивость - если один компонент веб-приложения выйдет из строя, другие компоненты могут продолжить работу;
* простота обновления - веб-приложения могут быть легко обновлены, добавляя новые функции или исправляя ошибки;
* изоляция - каждый компонент веб-приложения может быть разработан, тестирован и развернут независимо друг от друга;
* возможность использования различных технологий - веб-приложения могут использовать различные технологии для каждого компонента.

2 ПРОЕКТНАЯ ЧАСТЬ

2.1 Техническое задание

Полное наименование системы: информационная система блога компании.

Краткое наименование системы: ИС Блог о компании.

Назначение системы.

Инструменты, способные упростить и автоматизировать процессы поиска, выбора и просмотра информации. Такие информационные системы блогов представляют собой непревзойденный инструмент для оптимизации обучения в сети.

Характеристика объектов автоматизации.

Произведённым автоматическим процессом будет являться компьютерной техники. Платформа сохранит индивидуальные данные о пользователях, а также информацию о товарах и заказах.

Требования к системе.

При анализе предметной области разрабатываемой системы были выделены следующие сущности, представляющие своей совокупностью целостность всей информационной системы, на основе которых будет разрабатываться база данных системы:

* пользователи;
* статьи;
* комментарии;

База данных информационной системы должна состоять из следующих связанных таблиц:

* users – таблица, содержащая информацию о пользователях;
* articles– таблица, содержащая информацию о статьях;
* comments – таблица, содержащая информацию о комментариях;

Данные о пользователях должны размещаться в таблице users:

Таблица 2.1 – Таблица users в базе данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| id | integer | 11 | Идентификатор пользователя | Первичный ключ |
| email | varchar | 100 | Электронная почта пользователя | NOT NULL |
| username | varchar | 100 | Логин (имя пользователя) | NOT NULL |
| password | timestamp |  | Пароль пользователя | NOT NULL |
| role | еnum (‘admin’,’user’) |  | Роль пользователя в системе | NOT NULL |
|  |  |  |  |  |
| created\_at | integer | 255 | Дата создания учётной записи пользователя | NOT NULL |

Данные о товарах должны размещаться в таблице products:

Таблица 2.2 – Таблица products в базе данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| id | integer | 11 | Идентификатор задачи | AUTO INCREMENT |
| category | varchar | 255 | Категория компьютера (игровой, офисный, рабочая станция) | NOT NULL |
| name | varchar | 100 | Название компьютера | NOT NULL |
| image\_path | varchar | 255 | Путь до изображения компьютера | NOT NULL |
| cpu\_id | varchar | 50 | Идентификатор процессора | NOT NULL |
| cooling\_id | int | 11 | Идентификатор системы охлаждения | NOT NULL |
| gpu\_id | timestamp |  | Идентификатор видеокарты | NOT NULL |
| ram\_id | timestamp |  | Идентификатор процессора | NOT NULL |
| motherboard\_id | integer | 11 | Идентификатор материнской платы | NOT NULL |
| storage\_id | integer | 11 | Идентификатор системы хранения компьютера | NOT NULL |
| power\_supply\_id | integer | 11 | Идентификатор блока питания |  |
| price | decimal | 10,2 | Цена компьютера | NOT NULL |

Данные о избранном должны размещаться в таблице favorites:

Таблица 2.3 – Таблица favorites в базе данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| user\_id | integer | 11 | Идентификатор пользователя | NOT NULL |
| product\_id | integer | 11 | Идентификатор продукта | NOT NULL |

Данные о комплектующих должны размещаться в таблице components:

Таблица 2.4 – Таблица components в базе данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| id | integer | 11 | Идентификатор комплектующего | AUTO\_INCREMENT,  NOT NULL |
| type | varchar | 50 | Описание типа комплектующего | NOT NULL |
| brand | varchar | 50 | Описание бренда комплектующего | NOT NULL |
| model | varchar | 100 | Описание модели комплектующего |  |
| capacity | int | 11 | Описание ёмкости системы хранения |  |
| power | int | 11 | Описание мощности блока питания |  |
| interface | varchar | 50 | Описание интерфейса подключения |  |

Модель базы данных представлена на рисунке 2.1:

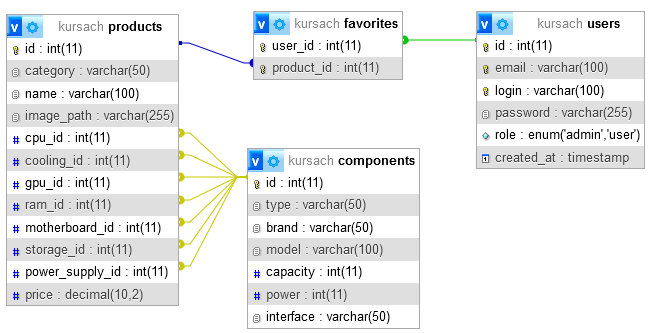


Рисунок 2.1 – Модель базы данных

2.2 Установка и настройка среды разработки

Выбор среды разработки: PhpStorm, Visual Studio Code.

Установка среды разработки: после выбора среды разработки необходимо ее установить на компьютер. Для этого можно использовать официальный сайт разработчика или пакетные менеджеры, такие как apt или Pacman.

Установка Apache и MySQL: Веб-приложение на PHP требует сервера Apache и базы данных MySQL для работы. Для установки Apache и MySQL можно использовать пакетные менеджеры, такие как apt-get или yum, или скачать их с официальных сайтов разработчиков.

Необходимо также установить PHP и настроить его для работы с Apache и MySQL.

2.3 Руководство пользователя

Требования к квалификации пользователя:

* знание основных навыков работы с компьютером - навигация в операционной системе, использование мыши и клавиатуры, открытие и закрытие программ и файлов;
* знание основных навыков работы в браузере - навигация по веб-сайту, использование ссылок, открытие и закрытие вкладок, использование поисковой строки;
* знание основных навыков работы с сетью - настройка сетевого подключения, использование Wi-Fi, настройка безопасности и конфиденциальности;
* знание основных навыков работы с веб-приложениями - регистрация и авторизация, настройка профиля, использование форм и полей ввода, взаимодействие с другими пользователями;
* знание основных навыков работы с контентом - просмотр видео и аудио, просмотр изображений и документов, скачивание файлов.

Требования к компьютеру пользователя:

* процессор с тактовой частотой 1 ГГц или выше;
* оперативная память 4096 Мб или более;
* объем свободного пространства на жестком диске не менее 1024 Мб;
* интернет-браузер Mozilla Firefox или Google Chrome последней версии.

Операции работы с веб-приложением:

Имя: Первый вход пользователя на сайт.

Описание: Неавторизованный пользователь впервые заходит на сайт.

Действующий субъект: Неавторизованный пользователь.

Сценарий:

* система выводит лэндинг-страницу;
* пользователь знакомится с услугами, доступными для использования.

На рисунке 2.2 изображена диаграмма последовательности для сценария первого входа на сайт:

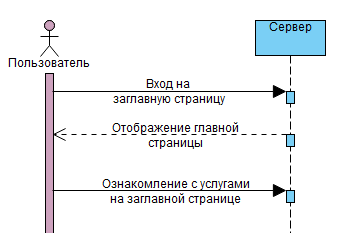


Рисунок 2.2 – Диаграмма последовательности сценария “Первый вход пользователя на сайт”

Имя: Авторизация пользователя.

Описание: для того, чтобы начать работать с системой, пользователю необходимо ввести свои данные для входа.

Действующий субъект: Неавторизованный пользователь.

Предусловие: Пользователь перешел на страницу для авторизации.

Сценарий:

* система отображает форму для входа в систему;
* пользователь вводит необходимые данные для входа в систему;
* пользователь нажимает на кнопку «Войти»;
* система сверяет данные, введенные пользователем, с данными в БД;
* система открывает доступ к функционалу сайта.

Расширение: если пользователь не зарегистрирован, он может нажать кнопку в модальном окне авторизации для регистрации в отдельном модальном окне.

На рисунке 2.3 изображена диаграмма последовательности для авторизации пользователя:

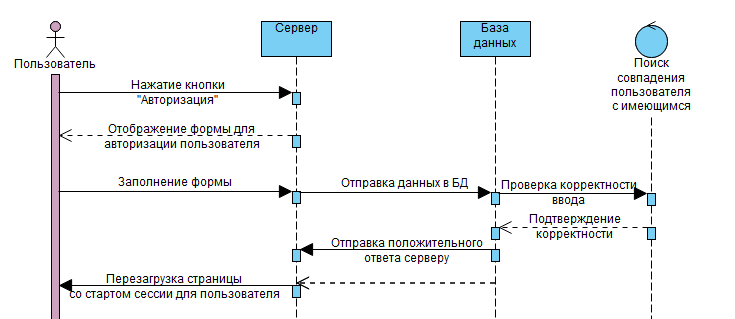


Рисунок 2.3 – Диаграмма последовательности для сценария “Авторизация пользователя”.

Имя: Регистрация пользователя.

Описание: для того, чтобы авторизоваться в системе, пользователю необходимо зарегистрироваться.

Действующий субъект: Неавторизованный пользователь.

Предусловие: Пользователь перешел на страницу для регистрации.

Сценарий:

* система выводит форму для регистрации;
* пользователь заполняет соответствующие поля;
* пользователь нажимает на кнопку «Зарегистрироваться».
* система сохраняет информацию о пользователе в базе данных;
* система переадресовывает пользователя на главную страницу.

Расширение: в случае некорректного формата введенных данных, при нажатии на кнопку «Зарегистрироваться», выведется сообщение с описанием ошибки.

На рисунке 2.4 изображена диаграмма последовательности для сценария регистрации пользователя:

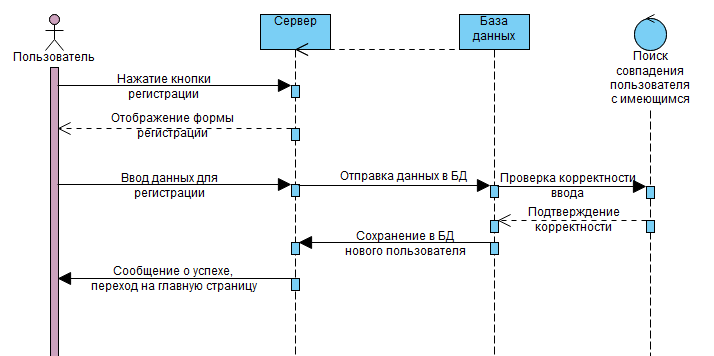


Рисунок 2.4 – диаграмма последовательности сценария “Регистрация пользователя”.

Имя: Добавление товара в каталог.

Описание: Пользователь теперь может добавлять товар.

Действующий субъект: Администратор.

Предусловие: Администратор перешел на страницу каталога и нажал кнопку “Добавить товар”.

Сценарий:

* система выводит форму для добавления;
* администратор заполняет соответствующие поля;
* администратор нажимает на кнопку «Готово».
* система сохраняет новую товар для каталога;
* система переадресовывает администратора на страницу каталога.

Расширение: в случае, если администратор передумал добавлять задачу, он нажимает на кнопку «Отмена» и система переадресовывает его на страницу каталога.

На рисунке 2.5 изображена диаграмма последовательности для сценария добавления товара в каталог:

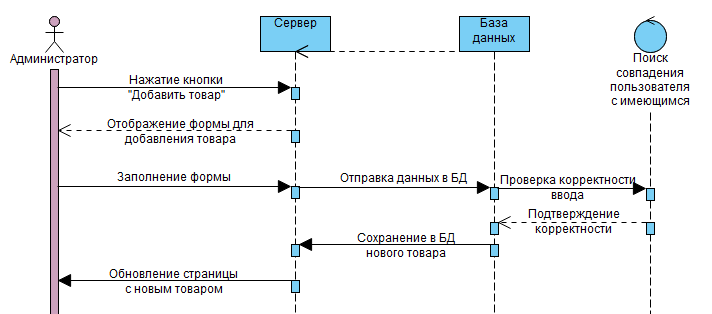


Рисунок 2.5 – Диаграмма последовательности для сценария “Добавление нового товара в каталог”

Имя: Редактирование товара.

Описание: Администратор может редактировать задачи.

Действующий субъект: Администратор.

Предусловие: Администратор нажал кнопку “редактировать” у карточки товара

Сценарий:

* система выводит форму для редактирования, в которой отображается старая информация о товаре;
* администратор редактирует соответствующие поля, если необходимо;
* пользователь нажимает на кнопку “Сохранить изменения”.
* система изменяет старые данные товара на новые;
* система переадресовывает администратора на страницу со всем и доступными товарами.

На рисунке 2.6 изображена диаграмма последовательности для сценария

“ Редактирование товара.”:

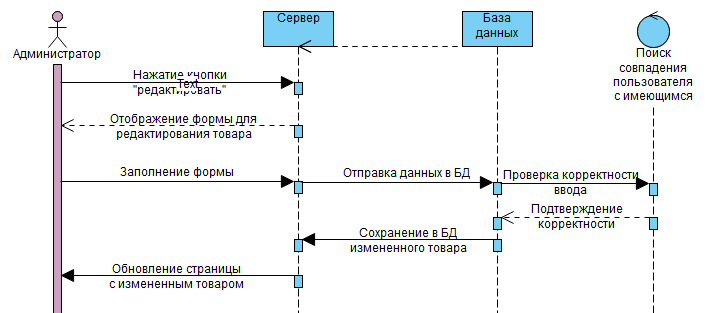


Рисунок 2.6 – диаграмма последовательности для сценария “Редактирование товара”

Имя: Удаление товара.

Описание: Администратор может удалять товары.

Действующий субъект: Администратор.

Предусловие: Администратор перешел на страницу с каталогом товаров.

Сценарий:

* система, при входе администратора на страницу каталога, выводит в карточке товара дополнительную кнопку для удаления товара;
* администратор нажимает на кнопку «Удалить»;
* система отправляет запрос к базе данных для удаления;
* система удаляет товар и обновляет страницу каталога товаров.

На рисунке 2.7 изображена диаграмма последовательности для сценария удаления товара:

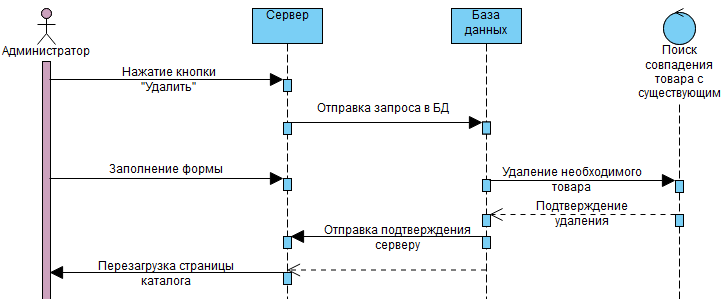


Рисунок 2.7 – Диаграмма последовательности для сценария “Удаление товара”

Имя: Редактирование пользователя.

Описание: Администратор может редактировать пользователей.

Действующий субъект: Администратор.

Предусловие: Администратор нажал кнопку “Редактировать” у карточки товара

Сценарий:

* система, при входе администратора на страницу “Личный кабинет”, выводит в таблице с пользователями дополнительную кнопку “Редактировать”, вызывающую модальное окно, где администратор редактирует соответствующие поля, если необходимо;
* пользователь нажимает на кнопку “Сохранить изменения”.
* система изменяет старые данные пользователя на новые;
* система отправляет запрос на базу данных для изменения пользователя;
* система переадресовывает администратора на страницу со всем и доступными товарами.

На рисунке 2.6 изображена диаграмма последовательности для сценария

“ Редактирование пользователя.”:

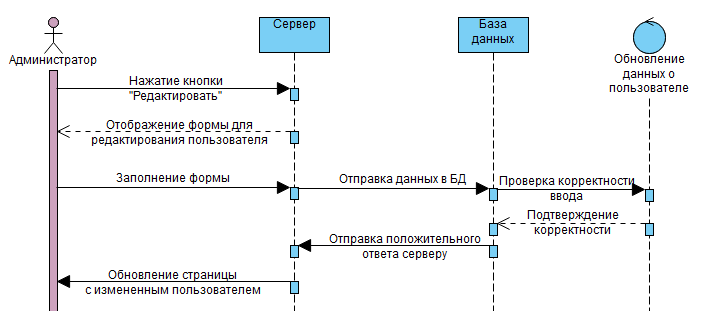


Рисунок 2.6 – диаграмма последовательности для сценария “Редактирование товара”

Имя: Удаление пользователя.

Описание: Администратор может удалять пользователей.

Действующий субъект: Администратор.

Предусловие: Администратор нажал кнопку “Удалить” в таблице с пользователями.

Сценарий:

* система, при входе администратора на страницу “Личный кабинет”, выводит в таблице с пользователями дополнительную кнопку для удаления пользователя;
* администратор нажимает на кнопку «Удалить»;
* система отправляет запрос на удаление пользователя из базы данных;
* база данных удаляет пользователя;
* система удаляет пользователя;
* база данных удаляет пользователя и отправляет системе ответ;
* система принимает ответ;
* система обновляет страницу таблицу с пользователями.

На рисунке 2.7 изображена диаграмма последовательности для сценария удаления пользователя:

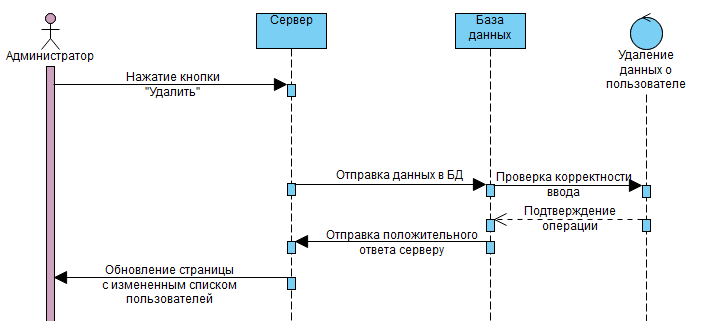


Рисунок 2.7 – Диаграмма последовательности для сценария “Удаление пользователя”

2.4 Руководство администратора

Предназначение:

Система интернет-магазина продажи компьютеров предназначена для организации и корректного отображения товаров, а также взаимодействия с пользователями. Она позволяет удобно создавать, редактировать, изменять и удалять товары, а также изменять и удалять пользователей.

Система интернет-магазина продажи компьютеров должна быть использована в информационной деятельности компании в сети Интернет. Она помогает повысить эффективность работы магазина, привлекать новых клиентов, знакомя их с брендом и предложениями компании.

Программное обеспечение:

Для администрирования веб-приложения администратору необходимо иметь следующее программное обеспечение:

* веб-браузер: Google Chrome(x64)/ Opera(x64), Mozilla Firefox (x64);
* текстовый редактор или IDE: Visual Studio Code, PhpStorm или Sublime Text;
* сервер приложений: Apache;
* база данных: MySQL.

В зависимости от конкретных потребностей и функций веб-приложения, администратору могут потребоваться дополнительные инструменты и технологии.

Инструкция по использованию API:

Главная страница:

* базовый URL: http://localhost/путь\_к\_папке/index.php;
* формат ответа: HTML-страница.

Авторизация:

Для доступа к API необходимо авторизоваться. Для этого нужно отправить POST-запрос на URL / путь\_к\_папке /login.php с параметрами:

* password – пароль.
* login или email – логин или адрес электронной почты

Регистрация:

Если у пользователя нет учетной записи, необходимо зарегистрироваться. Для этого нужно отправить POST-запрос на URL / путь\_к\_папке /register.php с параметрами:

* username – имя пользователя;
* email – имя пользователя;
* password – пароль.

Панель администратора:

* базовый URL: http://localhost/путь\_к\_папке/account.php с начатой сессией администратора;
* формат ответа: HTML-страница.

Добавление задачи:

Для добавления товара нужно отправить POST-запрос на URL products.php с параметрами:

* category – категория компьютера;
* name – наименование модели компьютера;
* image\_path – путь до изображения компьютера;
* cpu\_id – ID процессора;
* cooling\_id – ID системы охлаждения компьютера;
* gpu\_id – ID видеокарты;
* ram\_id – ID оперативной памяти;
* motherboard\_id – ID материнской платы;
* storage\_id – ID системы хранения компьютера;
* power\_supply\_id – ID блока питания компьютера;
* price – цена компьютера

Редактирование товара:

Для редактирования задачи нужно отправить POST-запрос на URL products.php с параметрами:

* category – категория компьютера;
* name – наименование модели компьютера;
* image\_path – путь до изображения компьютера;
* cpu\_id – ID процессора;
* cooling\_id – ID системы охлаждения компьютера;
* gpu\_id – ID видеокарты;
* ram\_id – ID оперативной памяти;
* motherboard\_id – ID материнской платы;
* storage\_id – ID системы хранения компьютера;
* power\_supply\_id – ID блока питания компьютера;
* price – цена компьютера.

Удаление задачи:

Для удаления задачи нужно отправить POST-запрос на URL products.php с параметрами:

* id – id модели товара.

Редактирование пользователя:

Для редактирования пользователя нужно отправить POST-запрос на URL account.php с параметрами:

* username – имя пользователя;
* email – имя пользователя;
* password – пароль.

Удаление пользователя:

Для удаления пользователя нужно отправить POST-запрос на URL account.php с параметрами:

* id – id пользователя.

ЗАКЛЮЧЕНИЕ

В процессе разработки интернет-магазина продажи компьютеров были реализованы ключевые принципы удобства использования и интуитивно понятного интерфейса, что способствует повышению удобства пользователей. Все цели, поставленные в рамках курсовой работы, были успешно достигнуты.

Создание многоуровневой архитектуры интернет-магазина обеспечило эффективное взаимодействие с базой данных и оптимизацию процессов планирования. Разработанный инструмент представляет собой ценный инструмент для пользователей, предоставляя базовый доступ к управлению товарами, пользователями, позволяя также отображать полезную для компании информацию.

Задачи были решены на уровне реализации многоуровневой архитектуры, обеспечения быстрого доступа к информации, а также реализации функций добавления, редактирования и удаления задач. Эти решения способствуют удобству использования сервера и повышают его функциональность.

Применение современных технологий, таких как язык программирования PHP, веб-сервер Apache, реляционная система управления базами данных MySQL и интерфейс для администрирования PHPMyAdmin, значительно улучшило процесс разработки, сократив время на ручные операции. Все данные, необходимые для функционирования веб-сервера, хранятся в базе данных, что обеспечивает надежность и стабильность работы приложения.

Созданное веб-приложение является неотъемлемой частью процесса интернет-торговли, предоставляя пользователям эффективный инструмент для управления товарами и пользователями, что в конечном итоге способствует повышению качества обслуживания и восприятия интернет-магазина.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Бройдо В.Л. Вычислительные системы, сети и телекоммуникации. 2-е изд.: учебник для вузов / В.Л. Бройдо -СПб: Питер, 2021, 703 с.
2. Васильев В.В., Сороколетова Н.В., Хливненко Л.В. Практикум по WЕВ-технологиям. – М.: ФОРУМ, 2022. – 416 с.
3. Веллинг Л. Разработка веб-приложений с помощью PHP и MySQL [Текст] / Л. Веллинг, Л. Томпсон. — 4-е издание. — М: Вильямс, 2019. — 846с.
4. Гвоздева Т.В., Баллад Б.А. Проектирование информационных систем: учебное пособие. - Ростов/н/Д.: Феникс, 2019. - 508 с. (гриф).
5. Гуда А.Н. Информатика и программирование: компьютерный практикум. – Ростов/н/Д.: Наука-Спектр; М.: Изд-во "Дашков и К", 2019. - 240 с. (гриф)
6. Зандстра М. PHP. Объекты, шаблоны и методики программирования. / М. Зандстра– М: Вильямс, 2018. 560с.
7. Измайлов В.Г., Лисьев Г.А., Озерова М.Ю., Трейбач А.Л. Проектирование web-приложений и программных систем в Open Source: учебное пособие. М.: ФЛИНТА, 2021. - 98 с.
8. Колисниченко Д. PHP и MySQL. Разработка веб-приложений. [Текст] / Д. Колисниченко. — СПб.: БХВ-Петербург: 2021. — 543с.
9. Колисниченко Д.Н. PHP 5/6 и MySQL 6. Разработка web-приложений / Д.Н. Колисниченко. – СПб.:БХВ-Петербург, 2022. 560 с.
10. Котеров Д.В. PHP 5. / Д.В. Котеров, А.Ф. Костарев. – СПб: БХВ Петербург, 2021, - 1120с.
11. Кузнецов М. Объектно-ориентированное программирование на PHP /М. Кузнецов, И. Симдянов. – СПб.: БХВ-Петербург, 2019. 608с.
12. Ленгсторф Д. PHP и jQuery для профессионалов [Текст] / Д. Ленгсторф. — М: Вильямс Ленгсторф: 2021. — 401с.
13. Ловэйн П. PHP 5. Объектно-ориентированное программирование. / П. Ловэйн. – М.: НТ Пресс, 2018. 224с.

ПРИЛОЖЕНИЕ

Листинг кода.

index.php – заглавная страница

<?php include 'header.php'; ?>

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Магазин ПК "Технодом"</title>

<link rel="stylesheet" href="assets/css/style.css">

</head>

<body>

<div class="container">

<?php require 'advantages.php' ?>

<section class="card-grid">

<div class="cards">

<?php

$cards = [

['img' => '/assets/images/about-us-pics/bestHD.png', 'title' => 'Лучшие компоненты', 'text' => 'Мы собираем только из оригинальных и новых компонентов, чтобы ваш новый компьютер при получении не потерял актуальности.'],

['img' => '/assets/images/about-us-pics/bestAssembly.jpg', 'title' => 'Высокое качество сборки', 'text' => 'Каждый компьютер тщательно собирается нашими специалистами.'],

['img' => '/assets/images/about-us-pics/testfm.webp', 'title' => 'Тестирование и настройка', 'text' => 'Каждый компьютер проходит этап установки ОС для того, чтобы пользователь сразу мог убедиться в работоспособности системы. А многочасовое прохождение тестов AIDA64, OCCT и прочих бенчмарков позволит исключить риски доставки нерабочего ПК.'],

['img' => '/assets/images/about-us-pics/sale.png', 'title' => 'С нами выгоднее', 'text' => 'Мы всегда стараемся предложить вам самые вкусные цены, которые только есть на рынке. Работаем как с физическими, так и юридическими лицами.'],

['img' => '/assets/images/about-us-pics/warranty.webp', 'title' => 'Гарантия', 'text' => 'Наша гарантия - 5 лет с полным доступом к вашему компьютеру (скажем нет пломбам!). Также наш сервисный центр для ремонта и обслуживания никогда не подведет, даже в самую трудную минуту.'],

['img' => '/assets/images/about-us-pics/shipping.png', 'title' => 'Быстрая доставка', 'text' => 'Бесплатная доставка по Белореченску. Без предоплат и прочих схем. Работаем с широкой номенклатурой транспортных компаний - доставим даже в глушь.']

];

foreach ($cards as $card): ?>

<div class="card">

<img src="<?= htmlspecialchars($card['img']) ?>" alt="<?= htmlspecialchars($card['title']) ?>">

<h3><?= htmlspecialchars($card['title']) ?></h3>

<p><?= nl2br(htmlspecialchars($card['text'])) ?></p>

</div>

<?php endforeach; ?>

</div>

</section>

<div class="about-us">

<div class="about-content">

<div class="about-image">

<img src="/assets/images/workspace.png" alt="О нас">

</div>

<div class="about-text">

<h2>О нас</h2>

<p>Наш интернет-магазин компьютеров был основан в 2022 году с целью предоставления качественных и недорогих компьютеров для всех потребителей. Мы стремимся предлагать только оригинальные комплектующие и тщательно собирать каждый компьютер под заказ.</p>

<p>Наша миссия — сделать процесс покупки компьютера простым и приятным для каждого клиента. Мы уверены, что выбор нас означает выбор надежности и высокого качества.</p>

</div>

</div>

</div>

</div>

<script src="/assets/js/script.js"></script>

<?php include 'footer.php' ?>

</body>

</html>

about.php – страница “о нас”

<?php include 'header.php'; ?>

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Магазин ПК "Технодом"</title>

<link rel="stylesheet" href="assets/css/style.css">

</head>

<body>

<div class="container">

<?php require 'advantages.php' ?>

<section class="card-grid">

<div class="cards">

<?php

$cards = [

['img' => '/assets/images/about-us-pics/bestHD.png', 'title' => 'Лучшие компоненты', 'text' => 'Мы собираем только из оригинальных и новых компонентов, чтобы ваш новый компьютер при получении не потерял актуальности.'],

['img' => '/assets/images/about-us-pics/bestAssembly.jpg', 'title' => 'Высокое качество сборки', 'text' => 'Каждый компьютер тщательно собирается нашими специалистами.'],

['img' => '/assets/images/about-us-pics/testfm.webp', 'title' => 'Тестирование и настройка', 'text' => 'Каждый компьютер проходит этап установки ОС для того, чтобы пользователь сразу мог убедиться в работоспособности системы. А многочасовое прохождение тестов AIDA64, OCCT и прочих бенчмарков позволит исключить риски доставки нерабочего ПК.'],

['img' => '/assets/images/about-us-pics/sale.png', 'title' => 'С нами выгоднее', 'text' => 'Мы всегда стараемся предложить вам самые вкусные цены, которые только есть на рынке. Работаем как с физическими, так и юридическими лицами.'],

['img' => '/assets/images/about-us-pics/warranty.webp', 'title' => 'Гарантия', 'text' => 'Наша гарантия - 5 лет с полным доступом к вашему компьютеру (скажем нет пломбам!). Также наш сервисный центр для ремонта и обслуживания никогда не подведет, даже в самую трудную минуту.'],

['img' => '/assets/images/about-us-pics/shipping.png', 'title' => 'Быстрая доставка', 'text' =>'Бесплатная доставка по Белореченску. Без предоплат и прочих схем. Работаем с широкой номенклатурой транспортных компаний - доставим даже в глушь.']

];

foreach ($cards as $card): ?>

<div class="card">

<img src="<?= htmlspecialchars($card['img']) ?>" alt="<?= htmlspecialchars($card['title']) ?>">

<h3><?= htmlspecialchars($card['title']) ?></h3>

<p><?= nl2br(htmlspecialchars($card['text'])) ?></p>

</div>

<?php endforeach; ?>

</div>

</section>

<div class="about-us">

<div class="about-content">

<div class="about-image">

<img src="/assets/images/workspace.png" alt="О нас">

</div>

<div class="about-text">

<h2>О нас</h2>

<p>Наш интернет-магазин компьютеров был основан в 2022 году с целью предоставления качественных и недорогих компьютеров для всех потребителей. Мы стремимся предлагать только оригинальные комплектующие и тщательно собирать каждый компьютер под заказ.</p>

<p>Наша миссия — сделать процесс покупки компьютера простым и приятным для каждого клиента. Мы уверены, что выбор нас означает выбор надежности и высокого качества.</p>

</div>

</div>

</div>

</div>

<script src="/assets/js/script.js"></script>

<?php include 'footer.php' ?>

</body>

</html>

account.php – личный кабинет

<?php

session\_start();

require 'db.php';

function showMessage($type, $message) {

return "<div class=\"$type\">$message</div>";

}

// Извлечение данных пользователей из базы данных

try {

$stmt = $conn->prepare("SELECT \* FROM users WHERE id = :user\_id");

$stmt->bindParam(':user\_id', $\_SESSION['user\_id']);

$stmt->execute();

$user = $stmt->fetch(PDO::FETCH\_ASSOC);

} catch (PDOException $e) {

$\_SESSION['error'] = "Ошибка базы данных: " . $e->getMessage();

header('Location: index.php');

exit;

}

$response = ['success' => false, 'message' => ''];

// Обработка формы добавления нового пользователя

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST['add\_user'])) {

$new\_login = htmlspecialchars(trim($\_POST['new\_login']));

$new\_password = password\_hash(htmlspecialchars(trim($\_POST['new\_password'])), PASSWORD\_DEFAULT);

$new\_email = htmlspecialchars(trim($\_POST['new\_email']));

$new\_role = htmlspecialchars(trim($\_POST['new\_role']));

if (empty($new\_login) || empty($new\_password) || empty($new\_email) || empty($new\_role)) {

$response['message'] = "Все поля обязательны.";

} else {

try {

$stmt = $conn->prepare("SELECT \* FROM users WHERE login = :login OR email = :email");

$stmt->bindParam(':login', $new\_login);

$stmt->bindParam(':email', $new\_email);

$stmt->execute();

$existingUser = $stmt->fetch(PDO::FETCH\_ASSOC);

if ($existingUser) {

$response['message'] = "Пользователь с таким логином или email уже существует.";

} else {

$stmt = $conn->prepare("INSERT INTO users (login, password, email, role) VALUES (:login, :password, :email, :role)");

$stmt->bindParam(':login', $new\_login);

$stmt->bindParam(':password', $new\_password);

$stmt->bindParam(':email', $new\_email);

$stmt->bindParam(':role', $new\_role);

if ($stmt->execute()) {

$response['success'] = true;

$response['message'] = "Пользователь успешно добавлен.";

} else {

$response['message'] = "Ошибка при добавлении пользователя.";

}

}

} catch (PDOException $e) {

$response['message'] = "Ошибка базы данных: " . $e->getMessage();

}

}

}

// Обработка формы редактирования пользователя

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST['edit\_user'])) {

$user\_id = htmlspecialchars(trim($\_POST['user\_id']));

$new\_login = htmlspecialchars(trim($\_POST['new\_login']));

$new\_password = htmlspecialchars(trim($\_POST['new\_password']));

$new\_email = htmlspecialchars(trim($\_POST['new\_email']));

$new\_role = htmlspecialchars(trim($\_POST['new\_role']));

if (empty($new\_login) || empty($new\_email) || empty($new\_role)) {

$response['message'] = "Логин, Email и Роль обязательны.";

} else {

try {

$stmt = $conn->prepare("SELECT \* FROM users WHERE (login = :login OR email = :email) AND id != :user\_id");

$stmt->bindParam(':login', $new\_login);

$stmt->bindParam(':email', $new\_email);

$stmt->bindParam(':user\_id', $user\_id);

$stmt->execute();

$existingUser = $stmt->fetch(PDO::FETCH\_ASSOC);

if ($existingUser) {

$response['message'] = "Пользователь с таким логином или email уже существует.";

} else {

$update\_query = "UPDATE users SET login = :login, email = :email, role = :role";

$params = [':login' => $new\_login, ':email' => $new\_email, ':role' => $new\_role];

if (!empty($new\_password)) {

$update\_query .= ", password = :password";

$params[':password'] = password\_hash($new\_password, PASSWORD\_DEFAULT);

}

$update\_query .= " WHERE id = :user\_id";

$params[':user\_id'] = $user\_id;

$stmt = $conn->prepare($update\_query);

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

$stmt->bindParam($key, $value);

}

if ($stmt->execute()) {

$response['success'] = true;

$response['message'] = "Данные пользователя успешно обновлены.";

} else {

$response['message'] = "Ошибка при обновлении данных пользователя.";

}

}

} catch (PDOException $e) {

$response['message'] = "Ошибка базы данных: " . $e->getMessage();

}

}

}

// Обработка формы удаления пользователя

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST['delete\_user'])) {

$user\_id = htmlspecialchars(trim($\_POST['user\_id']));

if (empty($user\_id)) {

$response['message'] = "ID пользователя не указан.";

} else {

try {

$stmt = $conn->prepare("DELETE FROM users WHERE id = :user\_id");

$stmt->bindParam(':user\_id', $user\_id);

if ($stmt->execute()) {

$response['success'] = true;

$response['message'] = "Пользователь успешно удален.";

} else {

$response['message'] = "Ошибка при удалении пользователя.";

}

} catch (PDOException $e) {

$response['message'] = "Ошибка базы данных: " . $e->getMessage();

}

}

}

// Получение списка всех пользователей (только для админов)

$users = [];

if ($user['role'] === 'admin') {

try {

$stmt = $conn->prepare("SELECT \* FROM users");

$stmt->execute();

$users = $stmt->fetchAll(PDO::FETCH\_ASSOC);

} catch (PDOException $e) {

$response['message'] = "Ошибка базы данных: " . $e->getMessage();

}

}

// Получение избранных товаров текущего пользователя

$favorites = [];

try {

$stmt = $conn->prepare("

SELECT

p.id,

p.category,

p.name,

p.image\_path,

p.price,

c1.model AS cpu\_model,

c2.model AS cooling\_model,

c3.model AS gpu\_model,

c4.model AS ram\_model,

c5.model AS motherboard\_model,

c6.model AS storage\_model,

c7.model AS power\_supply\_model

FROM products p

JOIN favorites f ON p.id = f.product\_id

LEFT JOIN components c1 ON p.cpu\_id = c1.id

LEFT JOIN components c2 ON p.cooling\_id = c2.id

LEFT JOIN components c3 ON p.gpu\_id = c3.id

LEFT JOIN components c4 ON p.ram\_id = c4.id

LEFT JOIN components c5 ON p.motherboard\_id = c5.id

LEFT JOIN components c6 ON p.storage\_id = c6.id

LEFT JOIN components c7 ON p.power\_supply\_id = c7.id

WHERE f.user\_id = :user\_id

");

$stmt->bindParam(':user\_id', $\_SESSION['user\_id']);

$stmt->execute();

$favorites = $stmt->fetchAll(PDO::FETCH\_ASSOC);

} catch (PDOException $e) {

$response['message'] = "Ошибка базы данных: " . $e->getMessage();

}

?>

<?php include "header.php"; ?>

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Магазин ПК "Технодом" | Профиль</title>

<link rel="stylesheet" href="assets/css/style.css">

<link rel="stylesheet" href="assets/css/product-card.css">

</head>

<body>

<main role="main">

<div class="profile-container">

<div class="profile-card">

<div class="avatar">

<img src="assets/images/standart-avatar.jpg" alt="Аватар пользователя" class="avatar-image">

</div>

<div class="profile-info">

<h2><?php echo htmlspecialchars($user['login']); ?></h2>

<p>Email: <?php echo htmlspecialchars($user['email']); ?></p>

<p>Дата регистрации: <?php echo htmlspecialchars(date('d.m.Y H:i', strtotime($user['created\_at']))); ?></p>

<p>Роль: <?php echo htmlspecialchars($user['role']); ?></p>

</div>

</div>

<div class="profile-content">

<?php if ($user['role'] === 'admin'): ?>

<h2>Админский раздел</h2>

<p>Добро пожаловать, администратор!</p>

<h3>Управление пользователями</h3>

<table>

<thead>

<tr>

<th>ID</th>

<th>Логин</th>

<th>Email</th>

<th>Роль</th>

<th>Дата регистрации</th>

<th>Действия</th>

</tr>

</thead>

<tbody>

<?php foreach ($users as $userItem): ?>

<tr>

<td><?php echo htmlspecialchars($userItem['id']); ?></td>

<td><?php echo htmlspecialchars($userItem['login']); ?></td>

<td><?php echo htmlspecialchars($userItem['email']); ?></td>

<td><?php echo htmlspecialchars($userItem['role']); ?></td>

<td><?php echo htmlspecialchars(date('d.m.Y H:i', strtotime($userItem['created\_at']))); ?></td>

<td>

<button onclick="openModal('editUser', '<?php echo htmlspecialchars($userItem['id']); ?>', '<?php echo htmlspecialchars($userItem['login']); ?>', '<?php echo htmlspecialchars($userItem['email']); ?>', '<?php echo htmlspecialchars($userItem['role']); ?>')">Редактировать</button>

<form action="" method="POST" style="display:inline;">

<input type="hidden" name="user\_id" value="<?php echo htmlspecialchars($userItem['id']); ?>">

<button type="submit" name="delete\_user">Удалить</button>

</form>

</td>

</tr>

<?php endforeach; ?>

</tbody>

</table>

<button onclick="openModal('addUser')">Добавить пользователя</button>

<h3>Управление товарами</h3>

<a href="products.php" class="admin-link">Перейти к управлению товарами</a>

<?php else: ?>

<h2>Мой профиль</h2>

<p>Добро пожаловать, <?php echo htmlspecialchars($user['login']); ?>!</p>

<h3>Мои избранные товары</h3>

<?php if (empty($favorites)): ?>

<p>Вы еще не добавили товары в избранное.</p>

<?php else: ?>

<div class="products-container">

<?php foreach ($favorites as $product): ?>

<div class="product-card">

<div class="image-container">

<img src="<?php echo htmlspecialchars($product['image\_path'] ?? ''); ?>" alt="Изображение товара">

</div>

<div class="details">

<div class="spec-item">

<div class="spec-title">Процессор</div>

<div class="spec-value"><?php echo htmlspecialchars($product['cpu\_model'] ?? 'Не указан'); ?></div>

</div>

<div class="spec-item">

<div class="spec-title">Охлаждение</div>

<div class="spec-value"><?php echo htmlspecialchars($product['cooling\_model'] ?? 'Не указан'); ?></div>

</div>

<div class="spec-item">

<div class="spec-title">Видеокарта</div>

<div class="spec-value"><?php echo htmlspecialchars($product['gpu\_model'] ?? 'Не указан'); ?></div>

</div>

<div class="spec-item">

<div class="spec-title">Оперативная память</div>

<div class="spec-value"><?php echo htmlspecialchars($product['ram\_model'] ?? 'Не указан'); ?></div>

</div>

<div class="spec-item">

<div class="spec-title">Материнская плата</div>

<div class="spec-value"><?php echo htmlspecialchars($product['motherboard\_model'] ?? 'Не указан'); ?></div>

</div>

<div class="spec-item">

<div class="spec-title">Накопитель</div>

<div class="spec-value"><?php echo htmlspecialchars($product['storage\_model'] ?? 'Не указан'); ?></div>

</div>

<div class="spec-item">

<div class="spec-title">Блок питания</div>

<div class="spec-value"><?php echo htmlspecialchars($product['power\_supply\_model'] ?? 'Не указан'); ?></div>

</div>

</div>

<div class="footer">

<div class="price"><?php echo number\_format($product['price'] ?? 0, 0, '', ' '); ?>₽</div>

<div class="action-buttons">

<form action="favorite\_action.php" method="post" style="display:inline;">

<input type="hidden" name="product\_id" value="<?php echo htmlspecialchars($product['id'] ?? 0); ?>">

<button type="submit" class="favorite-button favorited">Удалить из избранного</button>

</form>

</div>

</div>

</div>

<?php endforeach; ?>

</div>

<?php endif; ?>

<?php endif; ?>

<?php if ($response['success']): ?>

<?php echo showMessage('success-message', $response['message']); ?>

<?php elseif ($response['message']): ?>

<?php echo showMessage('error-message', $response['message']); ?>

<?php endif; ?>

</div>

</div>

</main>

<?php include 'modal.php'; ?>

<script src="assets/js/script.js"></script>

<?php include "footer.php"; ?>

</body>

</html>

advantages.php – преимущества

<?php

$advantages = [

[

'icon' => '/assets/images/advantages-icons/fast.png',

'alt' => 'Быстрая доставка',

'title' => 'Быстрая доставка',

'description' => 'Доставка в течение 2-3 рабочих дней. Мы используем современные методы доставки, чтобы обеспечить быструю и надежную поставку товаров.'

],

[

'icon' => '/assets/images/advantages-icons/reliability.png',

'alt' => 'Гарантия качества',

'title' => 'Гарантия качества',

'description' => 'Гарантия на все товары 1 год. Мы уверены в качестве нашей продукции и предоставляем надежную гарантию.'

],

[

'icon' => '/assets/images/advantages-icons/support.png',

'alt' => 'Круглосуточная поддержка',

'title' => 'Круглосуточная поддержка',

'description' => 'Поддержка клиентов 24/7. Наши специалисты всегда готовы ответить на ваши вопросы и помочь решить любые проблемы.'

],

[

'icon' => '/assets/images/advantages-icons/coupon.png',

'alt' => 'Специальные предложения',

'title' => 'Специальные предложения',

'description' => 'Регулярные скидки и акции. Будьте в курсе наших лучших предложений и экономьте на покупках.'

],

[

'icon' => '/assets/images/advantages-icons/best.png',

'alt' => 'Аутентичность продукции',

'title' => 'Аутентичность продукции',

'description' => 'Только оригинальные комплектующие. Мы гарантируем, что все наши товары аутентичны и соответствуют описанию.'

]

];

?>

<h1 class="section-title">Почему именно мы? Мы предлагаем клиентам следующее:</h1>

<div class="advantages-section">

<?php foreach ($advantages as $advantage): ?>

<div class="advantage-card">

<div class="advantage-icon-container">

<img src="<?php echo htmlspecialchars($advantage['icon']); ?>" alt="<?php echo htmlspecialchars($advantage['alt']); ?>" class="advantage-icon">

</div>

<div class="advantage-content">

<h3><?php echo htmlspecialchars($advantage['title']); ?></h3>

<p><?php echo htmlspecialchars($advantage['description']); ?></p>

</div>

</div>

<?php endforeach; ?>

</div>

auth\_service.php – сервис для обработки авторизации

<?php

session\_start();

function isAdmin() {

if (isset($\_SESSION['user\_id'])) {

$user\_id = intval($\_SESSION['user\_id']);

try {

require 'db.php';

$stmt = $conn->prepare("SELECT role FROM users WHERE id = ?");

$stmt->execute([$user\_id]);

$user = $stmt->fetch(PDO::FETCH\_ASSOC);

return $user && $user['role'] === 'admin';

} catch (PDOException $e) {

die("Ошибка: " . $e->getMessage());

}

}

return false;

}

?>

Contact.php – страница “контакты”

<?php include 'header.php'; ?>

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Магазин ПК "Технодом" | Контакты </title>

<link rel="stylesheet" href="assets/css/style.css">

</head>

<body>

<main>

<section class="contact-section">

<div class="container">

<h2>Контакты</h2>

<div class="contact-info">

<div class="contact-item">

<h3>Телефонная поддержка</h3>

<ul>

<li><a href="tel:+71234567890">+7 (123) 456-78-90</a></li>

<li><a href="tel:+71234567891">+7 (123) 456-78-91</a></li>

</ul>

</div>

<div class="contact-item">

<h3>Электронная почта</h3>

<ul>

<li><a href="mailto:individuals@example.com">individuals@example.com</a> (индивидуальные

клиенты)

</li>

<li><a href="mailto:business@example.com">business@example.com</a> (юридические лица)</li>

</ul>

</div>

<div class="contact-item">

<h3>Адрес</h3>

<p>Белореченск, ул. Чапаева, 48</p>

</div>

</div>

</div>

</section>

<section class="map-section">

<div class="container">

<h2>Местоположение магазина</h2>

<div id="map" style="width: 100%; height: 400px;"></div>

</div>

</section>

</main>

<script src="https://api-maps.yandex.ru/2.1/?apikey=YOUR\_YANDEX\_MAPS\_API\_KEY&lang=ru\_RU"></script>

<script>

ymaps.ready(function () {

var myMap = new ymaps.Map('map', {

center: [44.758376, 39.866514],

zoom: 15

});

var myPlacemark = new ymaps.Placemark([44.758376, 39.866514], {

hintContent: 'Интернет-магазин готовых компьютеров',

balloonContent: 'Белореченск, ул. Чапаева, 48'

});

myMap.geoObjects.add(myPlacemark);

});

</script>

</body>

</html>

db.php – конфигурация для БД

<?php

$servername = "localhost";

$username = "tester";

$password = "123456";

$dbname = "kursach";

try {

$conn = new PDO("mysql:host=$servername;dbname=$dbname;charset=utf8", $username, $password);

$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

error\_log("Успешное подключение");

} catch (PDOException $e) {

error\_log("Подключение провалилось: " . $e->getMessage());

die("Ошибка подключения: " . htmlspecialchars($e->getMessage()));

}

?>

favorite\_action.php – сервис для взаимодействия с избранным

<?php

session\_start();

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

$product\_id = isset($\_POST['product\_id']) ? intval($\_POST['product\_id']) : 0;

$user\_id = isset($\_SESSION['user\_id']) ? intval($\_SESSION['user\_id']) : 0;

if ($user\_id > 0 && $product\_id > 0) {

require 'db.php';

try {

// Проверка, существует ли уже такая запись в таблице favorites

$stmt = $conn->prepare("SELECT \* FROM favorites WHERE user\_id = ? AND product\_id = ?");

$stmt->execute([$user\_id, $product\_id]);

$favorite = $stmt->fetch();

if (!$favorite) {

// Если записи нет, создается новая

$stmt = $conn->prepare("INSERT INTO favorites (user\_id, product\_id) VALUES (?, ?)");

$stmt->execute([$user\_id, $product\_id]);

echo "Продукт добавлен в избранное!";

} else {

// Если запись существует, она удаляется

$stmt = $conn->prepare("DELETE FROM favorites WHERE user\_id = ? AND product\_id = ?");

$stmt->execute([$user\_id, $product\_id]);

echo "Продукт удален из избранного!";

}

} catch (PDOException $e) {

echo "Ошибка: " . $e->getMessage();

}

} else {

echo "Пользователь не авторизован или неверные данные.";

}

} else {

echo "Неверный метод запроса.";

}

header("Location: account.php");

exit();

?>

footer.php – подвал сайта

<body>

<div class="footer">

<div class="footer\_\_inner container">

<div class="footer\_\_info">

<p>© 2023 Интернет-магазин "Технодом"</p>

<p>Политика конфиденциальности | Условия использования</p>

</div>

<div class="footer\_\_links">

<h3>Полезные ссылки</h3>

<ul>

<li><a href="index.php">Главная</a></li>

<li><a href="about.php">О нас</a></li>

<li><a href="products.php">Продукты</a></li>

<li><a href="contact.php">Контакты</a></li>

</ul>

</div>

<div class="footer\_\_sitemap">

<h3>Карта сайта</h3>

<ul>

<li><a href="index.php">Главная</a></li>

<li><a href="products.php">Каталог</a></li>

<li><a href="news.php">Новости</a></li>

<li><a href="assets">Блог</a></li>

</ul>

</div>

</div>

</div>

header.php – заголовок сайта

<header class="header">

<div class="header\_\_inner container">

<a href="index.php" class="logo-link">

<img src="assets/images/logo.png" alt="Логотип компании" class="logo">

<h1 class="sr-only">Магазин компьютерной техники</h1>

</a>

<nav class="nav">

<ul class="nav\_\_list">

<li><a href="about.php" class="nav-link">О нас</a></li>

<li><a href="products.php" class="nav-link">Каталог</a></li>

<li><a href="contact.php" class="nav-link">Контакты</a></li>

</ul>

</nav>

<div class="buttons-group">

<button id="accountBtn" class="auth-button" title="Личный кабинет">

<img src="assets/icons/user.png" alt="Личный кабинет" class="icon">

</button>

<button id="logoutBtn" class="auth-button">

<img src="assets/icons/logout.png" alt="Выход" class="icon">

</button>

<button id="openModalBtn" class="auth-button">

<img src="assets/icons/login.png" alt="Авторизация" class="icon">

</button>

</div>

</div>

</header>

<?php include 'modal.php'; ?>

<script src="assets/js/script.js"></script>

login.php – логика авторизации

<?php

session\_start();

require 'db.php';

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

if (!isset($\_POST['loginOrEmail']) || !isset($\_POST['password'])) {

$\_SESSION['error'] = "Не все поля заполнены.";

} else {

$loginOrEmail = htmlspecialchars(trim($\_POST['loginOrEmail']));

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

try {

$stmt = $conn->prepare("SELECT \* FROM users WHERE email = :loginOrEmail OR login = :loginOrEmail");

$stmt->bindParam(':loginOrEmail', $loginOrEmail);

$stmt->execute();

$user = $stmt->fetch(PDO::FETCH\_ASSOC);

if ($user) {

if ($password === $user['password']) {

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

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

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

$\_SESSION['redirect\_url'] = $\_SERVER['HTTP\_REFERER'];

header('Location: redirect.php');

exit;

} else {

$\_SESSION['error'] = "Неверный пароль.";

}

} else {

$\_SESSION['error'] = "Пользователь не найден.";

}

} catch (PDOException $e) {

$\_SESSION['error'] = "Ошибка базы данных: " . $e->getMessage();

}

}

header('Location: ' . $\_SERVER['HTTP\_REFERER']);

exit;

}

?>

logout.php – сброс пользователя

<?php

session\_start();

session\_unset();

session\_destroy();

?>

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Выход</title>

<script>

window.onload = function() {

window.location.href = document.referrer;

};

</script>

</head>

<body>

</body>

</html>

modal.php – модальное окно

<div id="modalWindow" class="modal">

<div class="modal-content">

<span class="close">&times;</span>

<h2 id="modalTitle"></h2>

<div id="modalContent"></div>

</div>

</div>

news.php – новости сайта

<div class="news-container">

<div class="news-card">

<img src="/assets/images/news/news1.png" alt="Новость 1">

<div class="news-content">

<h3>Новые сборки игровых ПК с RTX 4090</h3>

<p>Представляем вашему вниманию наши новейшие сборки игровых компьютеров, оснащенные мощными видеокартами NVIDIA GeForce RTX 4090.

Эти машины готовы к любым игровым вызовам и гарантируют максимальную производительность.</p>

<span class="date">Дата публикации: 11.11.2024</span>

</div>

</div>

<div class="news-card">

<img src="/assets/images/news/news2.jpg" alt="Новость 2">

<div class="news-content">

<h3>Компьютеры для работы и учебы со скидками до 15%</h3>

<p>Специальное предложение для студентов и офисных работников! До конца недели действует скидка до 15% на модели компьютеров,

оптимизированные для работы и учебы. Надежность, скорость и доступная цена – все в одном устройстве.</p>

<span class="date">Дата публикации: 10.11.2024</span>

</div>

</div>

<div class="news-card">

<img src="/assets/images/news/news3.png" alt="Новость 3">

<div class="news-content">

<h3>Бесплатная доставка при покупке компьютера дороже 50 000 рублей.</h3>

<p>Теперь вы можете получить бесплатную доставку вашего нового компьютера, если его стоимость превышает 50 000 рублей.

Сделайте заказ сегодня и наслаждайтесь быстрой доставкой без дополнительных затрат!</p>

<span class="date">Дата публикации: 09.11.2024</span>

</div>

</div>

</div>

pc-ads.php – контейнеры для заглавной страницы

<h1 class="section-title">Модельный ряд. Что подойдет именно вам?</h1>

<div class="all-categories">

<div class="big-container">

<h1>Игровые компьютеры</h1>

<p>Компьютеры для победителей</p>

<img src="assets/images/gaming-pc.png" alt="Игровой компьютер" class="category-image">

<a href="index.php?page=products" class="button">Подробнее</a>

</div>

<div class="small-containers">

<div class="small-container">

<h2>Офисные ПК</h2>

<p>Максимальная производительность для Excel</p>

<img src="assets/images/office-pc.png" alt="Офисный компьютер" class="category-image">

<a href="index.php?page=products" class="button">Подробнее</a>

</div>

<div class="small-container">

<h2>Рабочие станции</h2>

<p>Всё для разработчиков и художников</p>

<img src="assets/images/workspace.png" alt="Рабочая станция" class="category-image">

<a href="index.php?page=products" class="button">Подробнее</a>

</div>

</div>

</div>

product\_service.php – логика работы с товарами

<?php

require 'db.php';

function getProducts($conn, $params = []) {

$whereClauses = [];

$bindParams = [];

if (!empty($params['category'])) {

$whereClauses[] = "p.category = ?";

$bindParams[] = $params['category'];

}

if (!empty($params['cpu'])) {

$whereClauses[] = "p.cpu\_id = ?";

$bindParams[] = $params['cpu'];

}

if (!empty($params['cooling'])) {

$whereClauses[] = "p.cooling\_id = ?";

$bindParams[] = $params['cooling'];

}

if (!empty($params['gpu'])) {

$whereClauses[] = "p.gpu\_id = ?";

$bindParams[] = $params['gpu'];

}

if (!empty($params['ram'])) {

$whereClauses[] = "p.ram\_id = ?";

$bindParams[] = $params['ram'];

}

if (!empty($params['motherboard'])) {

$whereClauses[] = "p.motherboard\_id = ?";

$bindParams[] = $params['motherboard'];

}

if (!empty($params['storage'])) {

$whereClauses[] = "p.storage\_id = ?";

$bindParams[] = $params['storage'];

}

if (!empty($params['power\_supply'])) {

$whereClauses[] = "p.power\_supply\_id = ?";

$bindParams[] = $params['power\_supply'];

}

$whereClause = !empty($whereClauses) ? "WHERE " . implode(" AND ", $whereClauses) : "";

$sql = "

SELECT p.\*,

c\_cpu.model AS cpu\_model,

c\_cooling.model AS cooling\_model,

c\_gpu.model AS gpu\_model,

c\_ram.model AS ram\_model,

c\_motherboard.model AS motherboard\_model,

c\_storage.model AS storage\_model,

c\_power\_supply.model AS power\_supply\_model

FROM products p

LEFT JOIN components c\_cpu ON p.cpu\_id = c\_cpu.id

LEFT JOIN components c\_cooling ON p.cooling\_id = c\_cooling.id

LEFT JOIN components c\_gpu ON p.gpu\_id = c\_gpu.id

LEFT JOIN components c\_ram ON p.ram\_id = c\_ram.id

LEFT JOIN components c\_motherboard ON p.motherboard\_id = c\_motherboard.id

LEFT JOIN components c\_storage ON p.storage\_id = c\_storage.id

LEFT JOIN components c\_power\_supply ON p.power\_supply\_id = c\_power\_supply.id

$whereClause

ORDER BY p.price ASC

";

try {

$stmt = $conn->prepare($sql);

$stmt->execute($bindParams);

return $stmt->fetchAll(PDO::FETCH\_ASSOC);

} catch (PDOException $e) {

die("Ошибка: " . $e->getMessage());

}

}

function getComponents($conn) {

try {

$stmt = $conn->prepare("SELECT id, model FROM components");

$stmt->execute();

return $stmt->fetchAll(PDO::FETCH\_ASSOC);

} catch (PDOException $e) {

die("Ошибка: " . $e->getMessage());

}

}

function addProduct($conn, $data) {

try {

$stmt = $conn->prepare("

INSERT INTO products (category, cpu\_id, cooling\_id, gpu\_id, ram\_id, motherboard\_id, storage\_id, power\_supply\_id, price, image\_path)

VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)

");

$stmt->execute([

$data['new\_category'],

$data['new\_cpu\_id'],

$data['new\_cooling\_id'],

$data['new\_gpu\_id'],

$data['new\_ram\_id'],

$data['new\_motherboard\_id'],

$data['new\_storage\_id'],

$data['new\_power\_supply\_id'],

$data['new\_price'],

$data['new\_image\_path']

]);

return $stmt->rowCount() > 0;

} catch (PDOException $e) {

return false;

}

}

function editProduct($conn, $data) {

try {

$stmt = $conn->prepare("

UPDATE products SET

category = ?,

cpu\_id = ?,

cooling\_id = ?,

gpu\_id = ?,

ram\_id = ?,

motherboard\_id = ?,

storage\_id = ?,

power\_supply\_id = ?,

price = ?,

image\_path = ?

WHERE id = ?

");

$stmt->execute([

$data['new\_category'],

$data['new\_cpu\_id'],

$data['new\_cooling\_id'],

$data['new\_gpu\_id'],

$data['new\_ram\_id'],

$data['new\_motherboard\_id'],

$data['new\_storage\_id'],

$data['new\_power\_supply\_id'],

$data['new\_price'],

$data['new\_image\_path'],

$data['product\_id']

]);

return $stmt->rowCount() > 0;

} catch (PDOException $e) {

return false;

}

}

function deleteProduct($conn, $product\_id) {

try {

$stmt = $conn->prepare("DELETE FROM products WHERE id = ?");

$stmt->execute([$product\_id]);

return $stmt->rowCount() > 0;

} catch (PDOException $e) {

return false;

}

}

function getFavorites($conn, $user\_id) {

try {

$stmt = $conn->prepare("SELECT product\_id FROM favorites WHERE user\_id = ?");

$stmt->execute([$user\_id]);

return $stmt->fetchAll(PDO::FETCH\_COLUMN);

} catch (PDOException $e) {

return [];

}

}

?>

Products.php – страница каталога, логика подбора товаров

<?php

require 'db.php';

require 'auth\_service.php';

require 'product\_service.php';

$isAdmin = isAdmin();

// параметры формы

$category = isset($\_GET['category']) ? $\_GET['category'] : '';

$cpu = isset($\_GET['cpu']) ? $\_GET['cpu'] : '';

$cooling = isset($\_GET['cooling']) ? $\_GET['cooling'] : '';

$gpu = isset($\_GET['gpu']) ? $\_GET['gpu'] : '';

$ram = isset($\_GET['ram']) ? $\_GET['ram'] : '';

$motherboard = isset($\_GET['motherboard']) ? $\_GET['motherboard'] : '';

$storage = isset($\_GET['storage']) ? $\_GET['storage'] : '';

$power\_supply = isset($\_GET['power\_supply']) ? $\_GET['power\_supply'] : '';

$params = [

'category' => $category,

'cpu' => $cpu,

'cooling' => $cooling,

'gpu' => $gpu,

'ram' => $ram,

'motherboard' => $motherboard,

'storage' => $storage,

'power\_supply' => $power\_supply

];

$products = getProducts($conn, $params);

$components = getComponents($conn);

$user\_id = isset($\_SESSION['user\_id']) ? intval($\_SESSION['user\_id']) : 0;

$favorites = getFavorites($conn, $user\_id);

$response = ['success' => false, 'message' => ''];

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

if (isset($\_POST['add\_product'])) {

$new\_category = htmlspecialchars(trim($\_POST['new\_category']));

$new\_cpu\_id = htmlspecialchars(trim($\_POST['new\_cpu\_id']));

$new\_cooling\_id = htmlspecialchars(trim($\_POST['new\_cooling\_id']));

$new\_gpu\_id = htmlspecialchars(trim($\_POST['new\_gpu\_id']));

$new\_ram\_id = htmlspecialchars(trim($\_POST['new\_ram\_id']));

$new\_motherboard\_id = htmlspecialchars(trim($\_POST['new\_motherboard\_id']));

$new\_storage\_id = htmlspecialchars(trim($\_POST['new\_storage\_id']));

$new\_power\_supply\_id = htmlspecialchars(trim($\_POST['new\_power\_supply\_id']));

$new\_price = htmlspecialchars(trim($\_POST['new\_price']));

$new\_image\_path = htmlspecialchars(trim($\_POST['new\_image\_path']));

if (empty($new\_category) || empty($new\_cpu\_id) || empty($new\_cooling\_id) || empty($new\_gpu\_id) || empty($new\_ram\_id) || empty($new\_motherboard\_id) || empty($new\_storage\_id) || empty($new\_power\_supply\_id) || empty($new\_price) || empty($new\_image\_path)) {

$response['message'] = "Все поля обязательны.";

} else {

if (addProduct($conn, [

'new\_category' => $new\_category,

'new\_cpu\_id' => $new\_cpu\_id,

'new\_cooling\_id' => $new\_cooling\_id,

'new\_gpu\_id' => $new\_gpu\_id,

'new\_ram\_id' => $new\_ram\_id,

'new\_motherboard\_id' => $new\_motherboard\_id,

'new\_storage\_id' => $new\_storage\_id,

'new\_power\_supply\_id' => $new\_power\_supply\_id,

'new\_price' => $new\_price,

'new\_image\_path' => $new\_image\_path

])) {

$response['success'] = true;

$response['message'] = "Товар успешно добавлен.";

header("Location: index.php");

exit();

} else {

$response['message'] = "Ошибка при добавлении товара.";

}

}

} elseif (isset($\_POST['edit\_product'])) {

$product\_id = htmlspecialchars(trim($\_POST['product\_id']));

$new\_category = htmlspecialchars(trim($\_POST['new\_category']));

$new\_cpu\_id = htmlspecialchars(trim($\_POST['new\_cpu\_id']));

$new\_cooling\_id = htmlspecialchars(trim($\_POST['new\_cooling\_id']));

$new\_gpu\_id = htmlspecialchars(trim($\_POST['new\_gpu\_id']));

$new\_ram\_id = htmlspecialchars(trim($\_POST['new\_ram\_id']));

$new\_motherboard\_id = htmlspecialchars(trim($\_POST['new\_motherboard\_id']));

$new\_storage\_id = htmlspecialchars(trim($\_POST['new\_storage\_id']));

$new\_power\_supply\_id = htmlspecialchars(trim($\_POST['new\_power\_supply\_id']));

$new\_price = htmlspecialchars(trim($\_POST['new\_price']));

$new\_image\_path = htmlspecialchars(trim($\_POST['new\_image\_path']));

if (empty($new\_category) || empty($new\_cpu\_id) || empty($new\_cooling\_id) || empty($new\_gpu\_id) || empty($new\_ram\_id) || empty($new\_motherboard\_id) || empty($new\_storage\_id) || empty($new\_power\_supply\_id) || empty($new\_price) || empty($new\_image\_path)) {

$response['message'] = "Все поля обязательны.";

} else {

if (editProduct($conn, [

'product\_id' => $product\_id,

'new\_category' => $new\_category,

'new\_cpu\_id' => $new\_cpu\_id,

'new\_cooling\_id' => $new\_cooling\_id,

'new\_gpu\_id' => $new\_gpu\_id,

'new\_ram\_id' => $new\_ram\_id,

'new\_motherboard\_id' => $new\_motherboard\_id,

'new\_storage\_id' => $new\_storage\_id,

'new\_power\_supply\_id' => $new\_power\_supply\_id,

'new\_price' => $new\_price,

'new\_image\_path' => $new\_image\_path

])) {

$response['success'] = true;

$response['message'] = "Данные товара успешно обновлены.";

header("Location: index.php");

exit();

} else {

$response['message'] = "Ошибка при обновлении данных товара.";

}

}

} elseif (isset($\_POST['delete\_product'])) {

$product\_id = htmlspecialchars(trim($\_POST['product\_id']));

if (empty($product\_id)) {

$response['message'] = "ID товара не указан.";

} else {

if (deleteProduct($conn, $product\_id)) {

$response['success'] = true;

$response['message'] = "Товар успешно удален.";

header("Location: index.php");

exit();

} else {

$response['message'] = "Ошибка при удалении товара.";

}

}

}

}

?>

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Магазин ПК "Технодом"</title>

<link rel="stylesheet" href="assets/css/style.css">

<link rel="stylesheet" href="assets/css/product-card.css">

</head>

<?php include 'header.php'; ?>

<main role="main">

<h2 class="section-title">Каталог</h2>

<div class="search-and-products">

<div class="search-container">

<?php include 'search-form.php'; ?>

</div>

<div class="products-container">

<?php if ($isAdmin): ?>

<div class="admin-panel">

<button onclick="openModal('addProduct')">Добавить товар</button>

</div>

<?php endif; ?>

<?php

$categories = array\_unique(array\_column($products, 'category'));

foreach ($categories as $category): ?>

<section class="category">

<h2><?php echo htmlspecialchars($category); ?></h2>

<?php

$categoryProducts = array\_filter($products, function ($product) use ($category) {

return $product['category'] === $category;

});

if (empty($categoryProducts)) {

echo "<p>Нет товаров в этой категории.</p>";

} else {

foreach ($categoryProducts as $product):

$isFavorited = in\_array($product['id'], $favorites);

?>

<div class="product-card">

<div class="image-container">

<img src="<?php echo htmlspecialchars($product['image\_path']); ?>" alt="Изображение товара">

</div>

<div class="details">

<div class="spec-item">

<div class="spec-title">Процессор</div>

<div class="spec-value"><?php echo htmlspecialchars($product['cpu\_model']); ?></div>

</div>

<div class="spec-item">

<div class="spec-title">Охлаждение</div>

<div class="spec-value"><?php echo htmlspecialchars($product['cooling\_model']); ?></div>

</div>

<div class="spec-item">

<div class="spec-title">Видеокарта</div>

<div class="spec-value"><?php echo htmlspecialchars($product['gpu\_model']); ?></div>

</div>

<div class="spec-item">

<div class="spec-title">Оперативная память</div>

<div class="spec-value"><?php echo htmlspecialchars($product['ram\_model']); ?></div>

</div>

<div class="spec-item">

<div class="spec-title">Материнская плата</div>

<div class="spec-value"><?php echo htmlspecialchars($product['motherboard\_model']); ?></div>

</div>

<div class="spec-item">

<div class="spec-title">Накопитель</div>

<div class="spec-value"><?php echo htmlspecialchars($product['storage\_model']); ?></div>

</div>

<div class="spec-item">

<div class="spec-title">Блок питания</div>

<div class="spec-value"><?php echo htmlspecialchars($product['power\_supply\_model']); ?></div>

</div>

</div>

<div class="footer">

<div class="price"><?php echo number\_format($product['price'], 0, '', ' '); ?>₽</div>

<div class="action-buttons">

<form action="favorite\_action.php" method="post" style="display:inline;">

<input type="hidden" name="product\_id" value="<?php echo htmlspecialchars($product['id']); ?>">

<button type="submit" class="favorite-button <?php echo $isFavorited ? 'favorited' : ''; ?>">

<?php echo $isFavorited ? 'Удалить из избранного' : 'В избранное'; ?>

</button>

</form>

<?php if ($isAdmin): ?>

<button onclick="openModal('editProduct', '<?php echo htmlspecialchars($product['id']); ?>', '<?php echo htmlspecialchars($product['category']); ?>', '<?php echo htmlspecialchars($product['cpu\_id']); ?>', '<?php echo htmlspecialchars($product['cooling\_id']); ?>', '<?php echo htmlspecialchars($product['gpu\_id']); ?>', '<?php echo htmlspecialchars($product['ram\_id']); ?>', '<?php echo htmlspecialchars($product['motherboard\_id']); ?>', '<?php echo htmlspecialchars($product['storage\_id']); ?>', '<?php echo htmlspecialchars($product['power\_supply\_id']); ?>', '<?php echo htmlspecialchars($product['price']); ?>', '<?php echo htmlspecialchars($product['image\_path']); ?>')" class="edit-button">

Редактировать

</button>

<form action="" method="POST" style="display:inline;">

<input type="hidden" name="product\_id" value="<?php echo htmlspecialchars($product['id']); ?>">

<button type="submit" name="delete\_product" class="delete-button">

Удалить

</button>

</form>

<?php endif; ?>

</div>

</div>

</div>

<?php endforeach;

} ?>

</section>

<?php endforeach; ?>

</div>

</div>

</main>

<?php include 'modal.php'; ?>

<script src="assets/js/script.js"></script>

<script type="application/json" id="componentsData">

[

{"id": 1, "type": "Процессор", "brand": "AMD", "model": "AMD Ryzen 9 5900X", "capacity": null, "power": null, "interface": null},

{"id": 2, "type": "Процессор", "brand": "AMD", "model": "AMD Ryzen 9 7950X", "capacity": null, "power": null, "interface": null},

{"id": 3, "type": "Процессор", "brand": "AMD", "model": "AMD Ryzen 7 5800X3D", "capacity": null, "power": null, "interface": null},

{"id": 4, "type": "Процессор", "brand": "AMD", "model": "AMD Ryzen 7 7700X", "capacity": null, "power": null, "interface": null},

{"id": 5, "type": "Процессор", "brand": "AMD", "model": "AMD Ryzen 5 5600X", "capacity": null, "power": null, "interface": null},

{"id": 6, "type": "Процессор", "brand": "AMD", "model": "AMD Ryzen 5 5600G", "capacity": null, "power": null, "interface": null},

{"id": 7, "type": "Процессор", "brand": "AMD", "model": "AMD Ryzen 5 7600X", "capacity": null, "power": null, "interface": null},

{"id": 8, "type": "Процессор", "brand": "AMD", "model": "AMD Ryzen 5 3500X", "capacity": null, "power": null, "interface": null},

{"id": 9, "type": "Процессор", "brand": "Intel", "model": "Intel Core i9-13900K", "capacity": null, "power": null, "interface": null},

{"id": 10, "type": "Процессор", "brand": "Intel", "model": "Intel Core i9-12900KS", "capacity": null, "power": null, "interface": null},

{"id": 11, "type": "Процессор", "brand": "Intel", "model": "Intel Core i9-11900K", "capacity": null, "power": null, "interface": null},

{"id": 12, "type": "Процессор", "brand": "Intel", "model": "Intel Core i9-10980XE", "capacity": null, "power": null, "interface": null},

{"id": 13, "type": "Процессор", "brand": "Intel", "model": "Intel Core i7-13700K", "capacity": null, "power": null, "interface": null},

{"id": 14, "type": "Процессор", "brand": "Intel", "model": "Intel Core i7-12700K", "capacity": null, "power": null, "interface": null},

{"id": 15, "type": "Процессор", "brand": "Intel", "model": "Intel Core i7-11700K", "capacity": null, "power": null, "interface": null},

{"id": 16, "type": "Процессор", "brand": "Intel", "model": "Intel Core i7-10700K", "capacity": null, "power": null, "interface": null},

{"id": 17, "type": "Процессор", "brand": "Intel", "model": "Intel Core i7-9700K", "capacity": null, "power": null, "interface": null},

{"id": 18, "type": "Процессор", "brand": "Intel", "model": "Intel Core i5-13600K", "capacity": null, "power": null, "interface": null},

{"id": 19, "type": "Процессор", "brand": "Intel", "model": "Intel Core i5-12600K", "capacity": null, "power": null, "interface": null},

{"id": 20, "type": "Процессор", "brand": "Intel", "model": "Intel Core i5-12400F", "capacity": null, "power": null, "interface": null},

{"id": 21, "type": "Процессор", "brand": "Intel", "model": "Intel Core i5-10400F", "capacity": null, "power": null, "interface": null},

{"id": 22, "type": "Процессор", "brand": "Intel", "model": "Intel Core i3-12100F", "capacity": null, "power": null, "interface": null},

{"id": 23, "type": "Процессор", "brand": "Intel", "model": "Intel Xeon W-3275", "capacity": null, "power": null, "interface": null},

{"id": 24, "type": "Процессор", "brand": "Intel", "model": "Intel Xeon W-3375", "capacity": null, "power": null, "interface": null},

{"id": 25, "type": "Процессор", "brand": "Intel", "model": "Intel Xeon E5-2690 v4", "capacity": null, "power": null, "interface": null},

{"id": 26, "type": "Видеокарта", "brand": "Nvidia", "model": "Nvidia GeForce RTX 3060 Ti", "capacity": null, "power": null, "interface": null},

{"id": 27, "type": "Видеокарта", "brand": "Nvidia", "model": "Nvidia GeForce RTX 3070 Ti", "capacity": null, "power": null, "interface": null},

{"id": 28, "type": "Видеокарта", "brand": "Nvidia", "model": "Nvidia GeForce RTX 3080 Ti", "capacity": null, "power": null, "interface": null},

{"id": 29, "type": "Видеокарта", "brand": "Nvidia", "model": "Nvidia GeForce RTX 4090", "capacity": null, "power": null, "interface": null},

{"id": 30, "type": "Видеокарта", "brand": "Nvidia", "model": "Nvidia GeForce RTX 4070 Ti", "capacity": null, "power": null, "interface": null},

{"id": 31, "type": "Видеокарта", "brand": "AMD", "model": "AMD Radeon RX 6800 XT", "capacity": null, "power": null, "interface": null},

{"id": 32, "type": "Видеокарта", "brand": "AMD", "model": "AMD Radeon RX 6900 XT", "capacity": null, "power": null, "interface": null},

{"id": 33, "type": "Видеокарта", "brand": "AMD", "model": "AMD Radeon RX 6950 XT", "capacity": null, "power": null, "interface": null},

{"id": 34, "type": "Видеокарта", "brand": "AMD", "model": "AMD Radeon RX 7800 XT", "capacity": null, "power": null, "interface": null},

{"id": 35, "type": "Охлаждение", "brand": "Воздушное", "model": "Noctua NH-D15", "capacity": null, "power": null, "interface": null},

{"id": 36, "type": "Охлаждение", "brand": "Воздушное", "model": "Cooler Master Hyper 212 EVO", "capacity": null, "power": null, "interface": null},

{"id": 37, "type": "Охлаждение", "brand": "Воздушное", "model": "Deepcool AS500 PLUS", "capacity": null, "power": null, "interface": null},

{"id": 38, "type": "Охлаждение", "brand": "Жидкостное", "model": "ASUS ROG RYUO III", "capacity": null, "power": null, "interface": null},

{"id": 39, "type": "Охлаждение", "brand": "Жидкостное", "model": "NZXT Kraken Z73", "capacity": null, "power": null, "interface": null},

{"id": 40, "type": "Охлаждение", "brand": "Жидкостное", "model": "Arctic Liquid Freezer II 360", "capacity": null, "power": null, "interface": null},

{"id": 41, "type": "ОЗУ", "brand": "DDR4", "model": "Corsair Vengeance LPX 16GB 3200MHz", "capacity": 16, "power": null, "interface": "DDR4"},

{"id": 42, "type": "ОЗУ", "brand": "DDR4", "model": "Corsair Vengeance LPX 32GB 3200MHz", "capacity": 32, "power": null, "interface": "DDR4"},

{"id": 43, "type": "ОЗУ", "brand": "DDR4", "model": "Corsair Vengeance LPX 64GB 6000MHz", "capacity": 64, "power": null, "interface": "DDR4"},

{"id": 44, "type": "ОЗУ", "brand": "DDR4", "model": "Crucial Ballistix MAX 16GB 4000MHz", "capacity": 16, "power": null, "interface": "DDR4"},

{"id": 45, "type": "ОЗУ", "brand": "DDR4", "model": "Kingston Fury Beast 8GB 2666MHz", "capacity": 8, "power": null, "interface": "DDR4"},

{"id": 46, "type": "ОЗУ", "brand": "DDR5", "model": "Corsair Dominator Platinum RGB DDR5 32GB 5200MHz", "capacity": 32, "power": null, "interface": "DDR5"},

{"id": 47, "type": "ОЗУ", "brand": "DDR5", "model": "G.Skill Trident Z5 RGB DDR5 64GB 6000MHz", "capacity": 64, "power": null, "interface": "DDR5"},

{"id": 48, "type": "Материнская плата", "brand": "AMD", "model": "ASUS ROG Strix X670E-E Gaming", "capacity": null, "power": null, "interface": null},

{"id": 49, "type": "Материнская плата", "brand": "AMD", "model": "MSI MEG X570 ACE", "capacity": null, "power": null, "interface": null},

{"id": 50, "type": "Материнская плата", "brand": "AMD", "model": "Gigabyte B550 Aorus Elite", "capacity": null, "power": null, "interface": null},

{"id": 51, "type": "Материнская плата", "brand": "Intel", "model": "ASRock B660M-ITX/ac", "capacity": null, "power": null, "interface": null},

{"id": 52, "type": "Материнская плата", "brand": "Intel", "model": "Gigabyte Z790 AORUS ELITE AX", "capacity": null, "power": null, "interface": null},

{"id": 53, "type": "Материнская плата", "brand": "Intel", "model": "Asus Prime Z490-A", "capacity": null, "power": null, "interface": null},

{"id": 54, "type": "Накопитель", "brand": "SSD", "model": "Samsung 990 PRO 2TB NVMe", "capacity": 2000, "power": null, "interface": "NVMe"},

{"id": 55, "type": "Накопитель", "brand": "SSD", "model": "WD Black SN850X 2TB NVMe", "capacity": 2000, "power": null, "interface": "NVMe"},

{"id": 56, "type": "Накопитель", "brand": "SSD", "model": "Crucial P5 Plus 1TB NVMe", "capacity": 1000, "power": null, "interface": "NVMe"},

{"id": 57, "type": "Накопитель", "brand": "HDD", "model": "Seagate BarraCuda 1TB SATA", "capacity": 1000, "power": null, "interface": "SATA"},

{"id": 58, "type": "Накопитель", "brand": "HDD", "model": "Western Digital Red Pro 6TB SATA", "capacity": 6000, "power": null, "interface": "SATA"},

{"id": 59, "type": "Накопитель", "brand": "HDD", "model": "Toshiba N300 4TB SATA", "capacity": 4000, "power": null, "interface": "SATA"},

{"id": 60, "type": "Блок питания", "brand": "Corsair", "model": "Be Quiet! BN345 1000W", "capacity": null, "power": 1000, "interface": null},

{"id": 61, "type": "Блок питания", "brand": "Corsair", "model": "RMx Series RM750x 750W", "capacity": null, "power": 750, "interface": null},

{"id": 62, "type": "Блок питания", "brand": "Corsair", "model": "AXi Series AX1600i 1600W", "capacity": null, "power": 1600, "interface": null},

{"id": 63, "type": "Блок питания", "brand": "Thermaltake", "model": "Toughpower GF1 ARGB 1050W", "capacity": null, "power": 1050, "interface": null},

{"id": 64, "type": "Блок питания", "brand": "Thermaltake", "model": "Toughpower PF1 ARGB 850W", "capacity": null, "power": 850, "interface": null},

{"id": 65, "type": "Блок питания", "brand": "EVGA", "model": "SuperNOVA 650 G5 650W", "capacity": null, "power": 650, "interface": null},

{"id": 66, "type": "Блок питания", "brand": "EVGA", "model": "SuperNOVA 500 G5 500W", "capacity": null, "power": 500, "interface": null},

{"id": 67, "type": "Блок питания", "brand": "Seasonic", "model": "Focus Plus Gold 1200 G4 1200W", "capacity": null, "power": 1200, "interface": null},

{"id": 68, "type": "Блок питания", "brand": "Seasonic", "model": "Prime TX-1600 Titanium 1600W", "capacity": null, "power": 1600, "interface": null},

{"id": 69, "type": "Блок питания", "brand": "Seasonic", "model": "Prime Ultra Titanium 1500W", "capacity": null, "power": 1500, "interface": null}

]

</script>

<?php include 'footer.php'; ?>

redirect.php – логика перенаправления

<?php

session\_start();

if (isset($\_SESSION['redirect\_url'])) {

$redirect\_url = $\_SESSION['redirect\_url'];

unset($\_SESSION['redirect\_url']);

header('Location: ' . $redirect\_url);

exit;

} else {

header('Location: D:\xampp\htdocs\kursach\index.php');

exit;

}

?>

register.php – логика регистрации пользователя

<?php

session\_start();

require 'db.php';

$response = array('success' => false, 'message' => '');

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

if (!isset($\_POST['email']) || !isset($\_POST['login']) || !isset($\_POST['password'])) {

$\_SESSION['error'] = "Не все поля заполнены.";

} else {

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

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

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

if (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {

$\_SESSION['error'] = "Неверный формат электронной почты.";

} elseif (strlen($password) < 6) {

$\_SESSION['error'] = "Пароль должен быть не менее 6 символов.";

} else {

try {

$stmt = $conn->prepare("SELECT \* FROM users WHERE email = :email OR login = :login");

$stmt->bindParam(':email', $email);

$stmt->bindParam(':login', $login);

$stmt->execute();

$user = $stmt->fetch(PDO::FETCH\_ASSOC);

if ($user) {

$\_SESSION['error'] = "Пользователь с таким email или логином уже существует.";

} else {

$stmt = $conn->prepare("INSERT INTO users (email, login, password) VALUES (:email, :login, :password)");

$stmt->bindParam(':email', $email);

$stmt->bindParam(':login', $login);

$stmt->bindParam(':password', $password);

if ($stmt->execute()) {

$\_SESSION['success'] = "Регистрация успешна. Вы можете войти в систему.";

$\_SESSION['redirect\_url'] = $\_SERVER['HTTP\_REFERER'];

header('Location: redirect.php');

exit;

} else {

$\_SESSION['error'] = "Ошибка при регистрации.";

}

}

} catch (PDOException $e) {

$\_SESSION['error'] = "Ошибка базы данных: " . $e->getMessage();

}

}

}

header('Location: ' . $\_SERVER['HTTP\_REFERER']);

exit;

}

?>

search-form.php – логика поиска, форма поиска товара по комплектующим

<?php

$dsn = 'mysql:host=localhost;dbname=kursach;charset=utf8mb4';

$username = 'tester';

$password = '123456';

try {

$conn = new PDO($dsn, $username, $password);

$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

// Получение списка категорий

$categoriesStmt = $conn->query("SELECT DISTINCT category FROM products");

$categories = $categoriesStmt->fetchAll(PDO::FETCH\_COLUMN);

// Получение списка комплектующих

$cpuStmt = $conn->query("SELECT id, model FROM components WHERE type = 'Процессор'");

$cpus = $cpuStmt->fetchAll(PDO::FETCH\_ASSOC);

$coolingStmt = $conn->query("SELECT id, model FROM components WHERE type = 'Охлаждение'");

$coolings = $coolingStmt->fetchAll(PDO::FETCH\_ASSOC);

$gpuStmt = $conn->query("SELECT id, model FROM components WHERE type = 'Видеокарта'");

$gpus = $gpuStmt->fetchAll(PDO::FETCH\_ASSOC);

$ramStmt = $conn->query("SELECT id, model FROM components WHERE type = 'ОЗУ'");

$rams = $ramStmt->fetchAll(PDO::FETCH\_ASSOC);

$motherboardStmt = $conn->query("SELECT id, model FROM components WHERE type = 'Материнская плата'");

$motherboards = $motherboardStmt->fetchAll(PDO::FETCH\_ASSOC);

$storageStmt = $conn->query("SELECT id, model FROM components WHERE type = 'Накопитель'");

$storages = $storageStmt->fetchAll(PDO::FETCH\_ASSOC);

$powerSupplyStmt = $conn->query("SELECT id, model FROM components WHERE type = 'Блок питания'");

$powerSupplies = $powerSupplyStmt->fetchAll(PDO::FETCH\_ASSOC);

} catch (PDOException $e) {

die("Ошибка: " . $e->getMessage());

}

?>

<form class="search-form" method="GET" action="products.php">

<input type="hidden" name="page" value="products">

<label for="category">Категория:</label>

<select name="category" id="category">

<option value="">Любая</option>

<?php foreach ($categories as $cat): ?>

<option value="<?php echo htmlspecialchars($cat); ?>" <?php if (isset($\_GET['category']) && $\_GET['category'] == $cat) echo 'selected'; ?>>

<?php echo htmlspecialchars($cat); ?>

</option>

<?php endforeach; ?>

</select>

<label for="cpu">Процессор:</label>

<select name="cpu" id="cpu">

<option value="">Любой</option>

<?php foreach ($cpus as $cpuItem): ?>

<option value="<?php echo htmlspecialchars($cpuItem['id']); ?>" <?php if (isset($\_GET['cpu']) && $\_GET['cpu'] == $cpuItem['id']) echo 'selected'; ?>>

<?php echo htmlspecialchars($cpuItem['model']); ?>

</option>

<?php endforeach; ?>

</select>

<label for="cooling">Охлаждение:</label>

<select name="cooling" id="cooling">

<option value="">Любое</option>

<?php foreach ($coolings as $coolingItem): ?>

<option value="<?php echo htmlspecialchars($coolingItem['id']); ?>" <?php if (isset($\_GET['cooling']) && $\_GET['cooling'] == $coolingItem['id']) echo 'selected'; ?>>

<?php echo htmlspecialchars($coolingItem['model']); ?>

</option>

<?php endforeach; ?>

</select>

<label for="gpu">Видеокарта:</label>

<select name="gpu" id="gpu">

<option value="">Любая</option>

<?php foreach ($gpus as $gpuItem): ?>

<option value="<?php echo htmlspecialchars($gpuItem['id']); ?>" <?php if (isset($\_GET['gpu']) && $\_GET['gpu'] == $gpuItem['id']) echo 'selected'; ?>>

<?php echo htmlspecialchars($gpuItem['model']); ?>

</option>

<?php endforeach; ?>

</select>

<label for="ram">Оперативная память:</label>

<select name="ram" id="ram">

<option value="">Любая</option>

<?php foreach ($rams as $ramItem): ?>

<option value="<?php echo htmlspecialchars($ramItem['id']); ?>" <?php if (isset($\_GET['ram']) && $\_GET['ram'] == $ramItem['id']) echo 'selected'; ?>>

<?php echo htmlspecialchars($ramItem['model']); ?>

</option>

<?php endforeach; ?>

</select>

<label for="motherboard">Материнская плата:</label>

<select name="motherboard" id="motherboard">

<option value="">Любая</option>

<?php foreach ($motherboards as $motherboardItem): ?>

<option value="<?php echo htmlspecialchars($motherboardItem['id']); ?>" <?php if (isset($\_GET['motherboard']) && $\_GET['motherboard'] == $motherboardItem['id']) echo 'selected'; ?>>

<?php echo htmlspecialchars($motherboardItem['model']); ?>

</option>

<?php endforeach; ?>

</select>

<label for="storage">Накопитель:</label>

<select name="storage" id="storage">

<option value="">Любой</option>

<?php foreach ($storages as $storageItem): ?>

<option value="<?php echo htmlspecialchars($storageItem['id']); ?>" <?php if (isset($\_GET['storage']) && $\_GET['storage'] == $storageItem['id']) echo 'selected'; ?>>

<?php echo htmlspecialchars($storageItem['model']); ?>

</option>

<?php endforeach; ?>

</select>

<label for="power\_supply">Блок питания:</label>

<select name="power\_supply" id="power\_supply">

<option value="">Любой</option>

<?php foreach ($powerSupplies as $powerSupplyItem): ?>

<option value="<?php echo htmlspecialchars($powerSupplyItem['id']); ?>" <?php if (isset($\_GET['power\_supply']) && $\_GET['power\_supply'] == $powerSupplyItem['id']) echo 'selected'; ?>>

<?php echo htmlspecialchars($powerSupplyItem['model']); ?>

</option>

<?php endforeach; ?>

</select>

<input type="submit" value="Найти">

</form>

setup.php – конфигурационный файл для создания таблиц БД и их заполнения данными

<?php

require 'db.php';

function executeQuery($conn, $sql, $message) {

try {

$conn->exec($sql);

echo htmlspecialchars($message) . "<br>";

} catch (PDOException $e) {

error\_log("Ошибка при выполнении запроса: " . $e->getMessage());

die("Ошибка при выполнении запроса: " . htmlspecialchars($e->getMessage()));

}

}

// Создание таблицы пользователей

$sql\_users = "CREATE TABLE IF NOT EXISTS users (

id INT AUTO\_INCREMENT PRIMARY KEY,

email VARCHAR(100) UNIQUE NOT NULL,

login VARCHAR(100) UNIQUE NOT NULL,

password VARCHAR(255) NOT NULL,

role ENUM('admin', 'user') NOT NULL DEFAULT 'user',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

)";

executeQuery($conn, $sql\_users, "Таблица пользователей создана успешно!");

// Создание таблицы для комплектующих

$sql\_components = "CREATE TABLE IF NOT EXISTS components (

id INT AUTO\_INCREMENT PRIMARY KEY,

type VARCHAR(50) NOT NULL,

brand VARCHAR(50) NOT NULL,

model VARCHAR(100) DEFAULT NULL,

capacity INT DEFAULT NULL,

power INT DEFAULT NULL,

interface VARCHAR(50) DEFAULT NULL

)";

executeQuery($conn, $sql\_components, "Таблица комплектующих создана успешно!");

// Создание таблицы для продуктов

$sql\_products = "CREATE TABLE IF NOT EXISTS products (

id INT AUTO\_INCREMENT PRIMARY KEY,

category VARCHAR(50) NOT NULL,

name VARCHAR(100) NOT NULL,

image\_path VARCHAR(255) NOT NULL,

cpu\_id INT,

cooling\_id INT,

gpu\_id INT,

ram\_id INT,

motherboard\_id INT,

storage\_id INT,

power\_supply\_id INT,

price DECIMAL(10, 2) NOT NULL,

FOREIGN KEY (cpu\_id) REFERENCES components(id),

FOREIGN KEY (cooling\_id) REFERENCES components(id),

FOREIGN KEY (gpu\_id) REFERENCES components(id),

FOREIGN KEY (ram\_id) REFERENCES components(id),

FOREIGN KEY (motherboard\_id) REFERENCES components(id),

FOREIGN KEY (storage\_id) REFERENCES components(id),

FOREIGN KEY (power\_supply\_id) REFERENCES components(id)

)";

executeQuery($conn, $sql\_products, "Таблица продуктов создана успешно!");

// Создание таблицы для избранного

$sql\_favorites = "CREATE TABLE IF NOT EXISTS favorites (

user\_id INT NOT NULL,

product\_id INT NOT NULL,

PRIMARY KEY (user\_id, product\_id),

FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE,

FOREIGN KEY (product\_id) REFERENCES products(id) ON DELETE CASCADE

)";

executeQuery($conn, $sql\_favorites, "Таблица избранного создана успешно!");

// Заполнение таблицы комплектующих данными

$sql\_insert\_components = "

INSERT INTO components (type, brand, model, capacity, power, interface) VALUES

('Процессор', 'AMD', 'AMD Ryzen 9 5900X', NULL, NULL, NULL),

('Процессор', 'AMD', 'AMD Ryzen 9 7950X', NULL, NULL, NULL),

('Процессор', 'AMD', 'AMD Ryzen 7 5800X3D', NULL, NULL, NULL),

('Процессор', 'AMD', 'AMD Ryzen 7 7700X', NULL, NULL, NULL),

('Процессор', 'AMD', 'AMD Ryzen 5 5600X', NULL, NULL, NULL),

('Процессор', 'AMD', 'AMD Ryzen 5 5600G', NULL, NULL, NULL),

('Процессор', 'AMD', 'AMD Ryzen 5 7600X', NULL, NULL, NULL),

('Процессор', 'AMD', 'AMD Ryzen 5 3500X', NULL, NULL, NULL),

('Процессор', 'Intel', 'Intel Core i9-13900K', NULL, NULL, NULL),

('Процессор', 'Intel', 'Intel Core i9-12900KS', NULL, NULL, NULL),

('Процессор', 'Intel', 'Intel Core i9-11900K', NULL, NULL, NULL),

('Процессор', 'Intel', 'Intel Core i9-10980XE', NULL, NULL, NULL),

('Процессор', 'Intel', 'Intel Core i7-13700K', NULL, NULL, NULL),

('Процессор', 'Intel', 'Intel Core i7-12700K', NULL, NULL, NULL),

('Процессор', 'Intel', 'Intel Core i7-11700K', NULL, NULL, NULL),

('Процессор', 'Intel', 'Intel Core i7-10700K', NULL, NULL, NULL),

('Процессор', 'Intel', 'Intel Core i7-9700K', NULL, NULL, NULL),

('Процессор', 'Intel', 'Intel Core i5-13600K', NULL, NULL, NULL),

('Процессор', 'Intel', 'Intel Core i5-12600K', NULL, NULL, NULL),

('Процессор', 'Intel', 'Intel Core i5-12400F', NULL, NULL, NULL),

('Процессор', 'Intel', 'Intel Core i5-10400F', NULL, NULL, NULL),

('Процессор', 'Intel', 'Intel Core i3-12100F', NULL, NULL, NULL),

('Процессор', 'Intel', 'Intel Xeon W-3275', NULL, NULL, NULL),

('Процессор', 'Intel', 'Intel Xeon W-3375', NULL, NULL, NULL),

('Процессор', 'Intel', 'Intel Xeon E5-2690 v4', NULL, NULL, NULL),

('Видеокарта', 'Nvidia', 'Nvidia GeForce RTX 3060 Ti', NULL, NULL, NULL),

('Видеокарта', 'Nvidia', 'Nvidia GeForce RTX 3070 Ti', NULL, NULL, NULL),

('Видеокарта', 'Nvidia', 'Nvidia GeForce RTX 3080 Ti', NULL, NULL, NULL),

('Видеокарта', 'Nvidia', 'Nvidia GeForce RTX 4090', NULL, NULL, NULL),

('Видеокарта', 'Nvidia', 'Nvidia GeForce RTX 4070 Ti', NULL, NULL, NULL),

('Видеокарта', 'AMD', 'AMD Radeon RX 6800 XT', NULL, NULL, NULL),

('Видеокарта', 'AMD', 'AMD Radeon RX 6900 XT', NULL, NULL, NULL),

('Видеокарта', 'AMD', 'AMD Radeon RX 6950 XT', NULL, NULL, NULL),

('Видеокарта', 'AMD', 'AMD Radeon RX 7800 XT', NULL, NULL, NULL),

('Охлаждение', 'Воздушное', 'Noctua NH-D15', NULL, NULL, NULL),

('Охлаждение', 'Воздушное', 'Cooler Master Hyper 212 EVO', NULL, NULL, NULL),

('Охлаждение', 'Воздушное', 'Deepcool AS500 PLUS', NULL, NULL, NULL),

('Охлаждение', 'Жидкостное', 'ASUS ROG RYUO III', NULL, NULL, NULL),

('Охлаждение', 'Жидкостное', 'NZXT Kraken Z73', NULL, NULL, NULL),

('Охлаждение', 'Жидкостное', 'Arctic Liquid Freezer II 360', NULL, NULL, NULL),

('ОЗУ', 'DDR4', 'Corsair Vengeance LPX 16GB 3200MHz', 16, NULL, 'DDR4'),

('ОЗУ', 'DDR4', 'Corsair Vengeance LPX 32GB 3200MHz', 32, NULL, 'DDR4'),

('ОЗУ', 'DDR4', 'Corsair Vengeance LPX 64GB 6000MHz', 64, NULL, 'DDR4'),

('ОЗУ', 'DDR4', 'Crucial Ballistix MAX 16GB 4000MHz', 16, NULL, 'DDR4'),

('ОЗУ', 'DDR4', 'Kingston Fury Beast 8GB 2666MHz', 8, NULL, 'DDR4'),

('ОЗУ', 'DDR5', 'Corsair Dominator Platinum RGB DDR5 32GB 5200MHz', 32, NULL, 'DDR5'),

('ОЗУ', 'DDR5', 'G.Skill Trident Z5 RGB DDR5 64GB 6000MHz', 64, NULL, 'DDR5'),

('Материнская плата', 'AMD', 'ASUS ROG Strix X670E-E Gaming', NULL, NULL, NULL),

('Материнская плата', 'AMD', 'MSI MEG X570 ACE', NULL, NULL, NULL),

('Материнская плата', 'AMD', 'Gigabyte B550 Aorus Elite', NULL, NULL, NULL),

('Материнская плата', 'Intel', 'ASRock B660M-ITX/ac', NULL, NULL, NULL),

('Материнская плата', 'Intel', 'Gigabyte Z790 AORUS ELITE AX', NULL, NULL, NULL),

('Материнская плата', 'Intel', 'Asus Prime Z490-A', NULL, NULL, NULL),

('Накопитель', 'SSD', 'Samsung 990 PRO 2TB NVMe', 2000, NULL, 'NVMe'),

('Накопитель', 'SSD', 'WD Black SN850X 2TB NVMe', 2000, NULL, 'NVMe'),

('Накопитель', 'SSD', 'Crucial P5 Plus 1TB NVMe', 1000, NULL, 'NVMe'),

('Накопитель', 'HDD', 'Seagate BarraCuda 1TB SATA', 1000, NULL, 'SATA'),

('Накопитель', 'HDD', 'Western Digital Red Pro 6TB SATA', 6000, NULL, 'SATA'),

('Накопитель', 'HDD', 'Toshiba N300 4TB SATA', 4000, NULL, 'SATA'),

('Блок питания', 'Corsair', 'Be Quiet! BN345 1000W', NULL, 1000, NULL),

('Блок питания', 'Corsair', 'RMx Series RM750x 750W', NULL, 750, NULL),

('Блок питания', 'Corsair', 'AXi Series AX1600i 1600W', NULL, 1600, NULL),

('Блок питания', 'Thermaltake', 'Toughpower GF1 ARGB 1050W', NULL, 1050, NULL),

('Блок питания', 'Thermaltake', 'Toughpower PF1 ARGB 850W', NULL, 850, NULL),

('Блок питания', 'EVGA', 'SuperNOVA 650 G5 650W', NULL, 650, NULL),

('Блок питания', 'EVGA', 'SuperNOVA 500 G5 500W', NULL, 500, NULL),

('Блок питания', 'Seasonic', 'Focus Plus Gold 1200 G4 1200W', NULL, 1200, NULL),

('Блок питания', 'Seasonic', 'Prime TX-1600 Titanium 1600W', NULL, 1600, NULL),

('Блок питания', 'Seasonic', 'Prime Ultra Titanium 1500W', NULL, 1500, NULL)

";

executeQuery($conn, $sql\_insert\_components, "Таблица комплектующих заполнена данными успешно!");

// Заполнение таблицы продуктов данными

$sql\_insert\_products = "

INSERT INTO products (category, name, image\_path, cpu\_id, cooling\_id, gpu\_id, ram\_id, motherboard\_id, storage\_id, power\_supply\_id, price) VALUES

('Игровые ПК', 'Игровой ПК 1', 'assets/images/product.png', 4, 6, 1, 7, 10, 12, 15, 900000.00),

('Игровые ПК', 'Игровой ПК 2', 'assets/images/product.png', 5, 6, 2, 8, 11, 13, 16, 450000.00),

('Игровые ПК', 'Игровой ПК 3', 'assets/images/product.png', 18, 6, 2, 8, 11, 14, 16, 300000.00),

('Офисные ПК', 'Офисный ПК 1', 'assets/images/product.png', 20, 6, 1, 8, 11, 13, 16, 280000.00),

('Офисные ПК', 'Офисный ПК 2', 'assets/images/product.png', 21, 6, 2, 8, 11, 13, 16, 260000.00),

('Офисные ПК', 'Офисный ПК 3', 'assets/images/product.png', 22, 6, 1, 8, 11, 14, 16, 230000.00),

('Рабочие станции', 'Рабочая станция 1', 'assets/images/product.png', 23, 6, 10, 7, 11, 12, 17, 1300000.00),

('Рабочие станции', 'Рабочая станция 2', 'assets/images/product.png', 5, 6, 2, 8, 11, 13, 16, 750000.00),

('Рабочие станции', 'Рабочая станция 3', 'assets/images/product.png', 18, 6, 2, 8, 11, 14, 16, 530000.00),

ON DUPLICATE KEY UPDATE category=VALUES(category), name=VALUES(name), image\_path=VALUES(image\_path), cpu\_id=VALUES(cpu\_id), cooling\_id=VALUES(cooling\_id), gpu\_id=VALUES(gpu\_id), ram\_id=VALUES(ram\_id), motherboard\_id=VALUES(motherboard\_id), storage\_id=VALUES(storage\_id), power\_supply\_id=VALUES(power\_supply\_id), price=VALUES(price)

";

executeQuery($conn, $sql\_insert\_products, "Таблица продуктов заполнена данными успешно!");

// Создание суперпользователя

$sql\_admin\_create = "INSERT INTO users (email, login, password, role, created\_at)

VALUES ('admin@index.ru', 'adminka', 'password', 'admin', NOW())

ON DUPLICATE KEY UPDATE email=VALUES(email), login=VALUES(login), password=VALUES(password), role=VALUES(role), created\_at=VALUES(created\_at)";

executeQuery($conn, $sql\_admin\_create, "Суперпользователь создан успешно!");

?>

slider.php – форма для слайдера

<body>

<div class="slider">

<div class="slides">

<div class="slide">

<img src="/assets/slides/slide1.jpg" alt="Слайд 1">

<div class="slide-content">

<h2>Мощные игровые ПК по специальным ценам!</h2>

<p>Успейте приобрести мощные игровые компьютеры по сниженным ценам!

Наши специалисты собрали самые производительные конфигурации специально для вас.</p>

</div>

</div>

<div class="slide">

<img src="/assets/slides/slide2.jpg" alt="Слайд 2">

<div class="slide-content">

<h2>Компьютеры для бизнеса и офиса</h2>

<p>Эффективность и надежность – ключевые качества наших бизнес-компьютеров. Оптимальные решения для офисов

любого размера.</p>

</div>

</div>

<div class="slide">

<img src="/assets/slides/slide3.jpg" alt="Слайд 3">

<div class="slide-content">

<h2>Рабочие станции для дизайнеров и видеомонтажеров</h2>

<p>Создавайте шедевры на профессиональном уровне! Наши рабочие станции оснащены мощными процессорами и

видеокартами, которые идеально подходят для работы с графикой и видео.</p>

</div>

</div>

</div>

<div class="prev-btn-container">

<button class="prev-btn">&#10094;</button>

</div>

<div class="next-btn-container">

<button class="next-btn">&#10095;</button>

</div>

<div class="dots-container">

<span class="dot active"></span>

<span class="dot"></span>

<span class="dot"></span>

</div>

</div>

<script src="/assets/js/script.js"></script>

</body>

script.js – javascript-код для всего проекта

// Модальное окно

var modal = document.getElementById("modalWindow");

var span = document.getElementsByClassName("close")[0];

var modalTitle = document.getElementById("modalTitle");

var modalContent = document.getElementById("modalContent");

if (span) {

span.onclick = function () {

modal.style.display = "none";

}

}

window.onclick = function (event) {

if (event.target === modal) {

modal.style.display = "none";

}

}

function openModal(action, productId, category, cpuId, coolingId, gpuId, ramId, motherboardId, storageId, powerSupplyId, price, imagePath, userId, login, email, role) {

modal.style.display = "block";

if (action === 'addProduct') {

modalTitle.textContent = 'Добавление нового товара';

modalContent.innerHTML = `

<form action="products.php" method="POST">

<label for="new\_category">Категория:</label>

<input type="text" id="new\_category" name="new\_category" required>

<label for="new\_cpu\_id">Процессор:</label>

<select id="new\_cpu\_id" name="new\_cpu\_id" required>

${generateComponentOptions(cpuId, 'Процессор')}

</select>

<label for="new\_cooling\_id">Охлаждение:</label>

<select id="new\_cooling\_id" name="new\_cooling\_id" required>

${generateComponentOptions(coolingId, 'Охлаждение')}

</select>

<label for="new\_gpu\_id">Видеокарта:</label>

<select id="new\_gpu\_id" name="new\_gpu\_id" required>

${generateComponentOptions(gpuId, 'Видеокарта')}

</select>

<label for="new\_ram\_id">Оперативная память:</label>

<select id="new\_ram\_id" name="new\_ram\_id" required>

${generateComponentOptions(ramId, 'ОЗУ')}

</select>

<label for="new\_motherboard\_id">Материнская плата:</label>

<select id="new\_motherboard\_id" name="new\_motherboard\_id" required>

${generateComponentOptions(motherboardId, 'Материнская плата')}

</select>

<label for="new\_storage\_id">Накопитель:</label>

<select id="new\_storage\_id" name="new\_storage\_id" required>

${generateComponentOptions(storageId, 'Накопитель')}

</select>

<label for="new\_power\_supply\_id">Блок питания:</label>

<select id="new\_power\_supply\_id" name="new\_power\_supply\_id" required>

${generateComponentOptions(powerSupplyId, 'Блок питания')}

</select>

<label for="new\_price">Цена:</label>

<input type="number" id="new\_price" name="new\_price" step="0.01" required>

<label for="new\_image\_path">Путь к изображению:</label>

<input type="text" id="new\_image\_path" name="new\_image\_path" required>

<button type="submit" name="add\_product">Добавить товар</button>

</form>

`;

} else if (action === 'editProduct') {

modalTitle.textContent = 'Редактирование товара';

modalContent.innerHTML = `

<form action="products.php" method="POST">

<input type="hidden" id="edit\_product\_id" name="product\_id" value="${productId}">

<label for="edit\_category">Категория:</label>

<input type="text" id="edit\_category" name="new\_category" value="${category}" required>

<label for="edit\_cpu\_id">Процессор:</label>

<select id="edit\_cpu\_id" name="new\_cpu\_id" required>

${generateComponentOptions(cpuId, 'Процессор')}

</select>

<label for="edit\_cooling\_id">Охлаждение:</label>

<select id="edit\_cooling\_id" name="new\_cooling\_id" required>

${generateComponentOptions(coolingId, 'Охлаждение')}

</select>

<label for="edit\_gpu\_id">Видеокарта:</label>

<select id="edit\_gpu\_id" name="new\_gpu\_id" required>

${generateComponentOptions(gpuId, 'Видеокарта')}

</select>

<label for="edit\_ram\_id">Оперативная память:</label>

<select id="edit\_ram\_id" name="new\_ram\_id" required>

${generateComponentOptions(ramId, 'ОЗУ')}

</select>

<label for="edit\_motherboard\_id">Материнская плата:</label>

<select id="edit\_motherboard\_id" name="new\_motherboard\_id" required>

${generateComponentOptions(motherboardId, 'Материнская плата')}

</select>

<label for="edit\_storage\_id">Накопитель:</label>

<select id="edit\_storage\_id" name="new\_storage\_id" required>

${generateComponentOptions(storageId, 'Накопитель')}

</select>

<label for="edit\_power\_supply\_id">Блок питания:</label>

<select id="edit\_power\_supply\_id" name="new\_power\_supply\_id" required>

${generateComponentOptions(powerSupplyId, 'Блок питания')}

</select>

<label for="edit\_price">Цена:</label>

<input type="number" id="edit\_price" name="new\_price" value="${price}" step="0.01" required>

<label for="edit\_image\_path">Путь к изображению:</label>

<input type="text" id="edit\_image\_path" name="new\_image\_path" value="${imagePath}" required>

<button type="submit" name="edit\_product">Сохранить изменения</button>

</form>

`;

} else if (action === 'addUser') {

modalTitle.textContent = 'Добавление нового пользователя';

modalContent.innerHTML = `

<form action="account.php" method="POST">

<label for="new\_login">Логин:</label>

<input type="text" id="new\_login" name="new\_login" required>

<label for="new\_email">Email:</label>

<input type="email" id="new\_email" name="new\_email" required>

<label for="new\_password">Пароль:</label>

<input type="password" id="new\_password" name="new\_password" required>

<label for="new\_role">Роль:</label>

<select id="new\_role" name="new\_role" required>

<option value="user">Пользователь</option>

<option value="admin">Администратор</option>

</select>

<button type="submit" name="add\_user">Добавить пользователя</button>

</form>

`;

} else if (action === 'editUser') {

modalTitle.textContent = 'Редактирование пользователя';

modalContent.innerHTML = `

<form action="account.php" method="POST">

<input type="hidden" id="edit\_user\_id" name="user\_id" value="${userId}">

<label for="edit\_login">Логин:</label>

<input type="text" id="edit\_login" name="new\_login" value="${login}" required>

<label for="edit\_email">Email:</label>

<input type="email" id="edit\_email" name="new\_email" value="${email}" required>

<label for="edit\_password">Новый пароль (оставьте пустым, если не меняете):</label>

<input type="password" id="edit\_password" name="new\_password">

<label for="edit\_role">Роль:</label>

<select id="edit\_role" name="new\_role" required>

<option value="user" ${role === 'user' ? 'selected' : ''}>Пользователь</option>

<option value="admin" ${role === 'admin' ? 'selected' : ''}>Администратор</option>

</select>

<button type="submit" name="edit\_user">Сохранить изменения</button>

</form>

`;

} else if (action === 'login') {

modalTitle.textContent = 'Авторизация';

modalContent.innerHTML = `

<form id="authForm" action="login.php" method="POST">

<label for="loginOrEmail">Логин или Электронная почта:</label>

<input type="text" id="loginOrEmail" name="loginOrEmail" required>

<label for="password">Пароль:</label>

<input type="password" id="password" name="password" required>

<button type="submit">Войти</button>

</form>

<p id="toggleLink">Нет аккаунта? <a href="#" id="registerLink">Зарегистрироваться</a></p>

`;

document.getElementById('registerLink').addEventListener('click', function (event) {

event.preventDefault();

openModal('register');

});

} else if (action === 'register') {

modalTitle.textContent = 'Регистрация';

modalContent.innerHTML = `

<form id="registerForm" action="register.php" method="POST">

<label for="regEmail">Электронная почта:</label>

<input type="email" id="regEmail" name="email" required>

<label for="regLogin">Логин:</label>

<input type="text" id="regLogin" name="login" required>

<label for="regPassword">Пароль:</label>

<input type="password" id="regPassword" name="password" required>

<button type="submit">Зарегистрироваться</button>

</form>

<p id="toggleLink">Уже есть аккаунт? <a href="#" id="loginLink">Войти</a></p>

`;

document.getElementById('loginLink').addEventListener('click', function (event) {

event.preventDefault();

openModal('login');

});

}

}

function closeModal() {

modal.style.display = "none";

}

function generateComponentOptions(selectedId = null, type = null) {

const componentsData = JSON.parse(document.getElementById('componentsData').textContent);

let filteredComponents = componentsData;

if (type) {

filteredComponents = componentsData.filter(component => component.type === type);

}

return filteredComponents.map(component =>

`<option value="${component.id}" ${selectedId === component.id.toString() ? 'selected' : ''}>

${component.brand} - ${component.model}

</option>`

).join('');

}

// Обработчики событий для кнопок

document.addEventListener('DOMContentLoaded', function () {

var accountBtn = document.getElementById('accountBtn');

var logoutBtn = document.getElementById('logoutBtn');

var openModalBtn = document.getElementById('openModalBtn');

if (accountBtn) {

accountBtn.addEventListener('click', function () {

window.location.href = 'account.php';

});

}

if (logoutBtn) {

logoutBtn.addEventListener('click', function () {

window.location.href = 'logout.php';

});

}

if (openModalBtn) {

openModalBtn.addEventListener('click', function () {

openModal('login');

});

}

// Обработчик для кнопки "Добавить товар" (только для админов)

var addProductBtn = document.querySelector('.admin-panel button');

if (addProductBtn) {

addProductBtn.addEventListener('click', function () {

openModal('addProduct');

});

}

// Удаление из избранного

var favoriteButtons = document.querySelectorAll('.favorite-button.favorited');

favoriteButtons.forEach(button => {

button.addEventListener('click', function (event) {

event.preventDefault();

var form = button.closest('form');

if (form) {

form.submit();

}

});

});

});

document.addEventListener('DOMContentLoaded', function () {

var accountBtn = document.getElementById('accountBtn');

var logoutBtn = document.getElementById('logoutBtn');

var openModalBtn = document.getElementById('openModalBtn');

if (accountBtn) {

accountBtn.addEventListener('click', function () {

window.location.href = 'account.php';

});

}

if (logoutBtn) {

logoutBtn.addEventListener('click', function () {

window.location.href = 'logout.php';

});

}

if (openModalBtn) {

openModalBtn.addEventListener('click', function () {

openModal('login');

});

}

var addProductBtn = document.querySelector('.admin-panel button');

if (addProductBtn) {

addProductBtn.addEventListener('click', function () {

openModal('addProduct');

});

}

});

// Слайдер

document.addEventListener('DOMContentLoaded', function () {

const slides = document.querySelectorAll('.slide');

const prevBtn = document.querySelector('.prev-btn');

const nextBtn = document.querySelector('.next-btn');

const dotsContainer = document.querySelector('.dots-container');

const dots = document.querySelectorAll('.dot');

let currentSlide = 0;

function showSlide(index) {

if (index < 0) {

currentSlide = slides.length - 1;

} else if (index >= slides.length) {

currentSlide = 0;

} else {

currentSlide = index;

}

const offset = -currentSlide \* 100;

document.querySelector('.slides').style.transform = `translateX(${offset}%)`;

dots.forEach(dot => dot.classList.remove('active'));

dots[currentSlide].classList.add('active');

}

prevBtn.addEventListener('click', () => {

showSlide(currentSlide - 1);

});

nextBtn.addEventListener('click', () => {

showSlide(currentSlide + 1);

});

dotsContainer.addEventListener('click', (event) => {

if (event.target.classList.contains('dot')) {

const dotIndex = Array.from(dots).indexOf(event.target);

showSlide(dotIndex);

}

});

setInterval(() => {

showSlide(currentSlide + 1);

}, 5000);

showSlide(currentSlide);

});

style.css – стили для всего проекта

/\* Импорт css-стилей, для избежания перегрузки кодом одного файла \*/

@import "header.css";

@import "footer.css";

@import "news.css";

@import "slider.css";

@import "advantages.css";

@import "pc-ads.css";

@import "about.css";

@import "product-card.css";

@import "search-form.css";

@import "modal.css";

@import "registration.css";

@import "profile.css";

@import "contact.css";

@import "catalog.css";

@import "admin-table.css";

@import "register-form.css";

/\* Общие стили \*/

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: 'IBM Plex Sans', sans-serif;

background-color: #202020;

color: white;

}

.container {

margin: 0 auto;

max-width: 80vw;

padding: 0 20px;

}

main {

width: 100%;

}

.sr-only {

position: absolute;

width: 1px;

height: 1px;

padding: 0;

overflow: hidden;

clip: rect(0, 0, 0, 0);

white-space: nowrap;

clip-path: inset(50%);

border: 0;

}

/\* Стиль для названия страницы \*/

.section-title {

margin-top: 40px;

text-align: left;

font-size: 32px;

margin-bottom: 40px;

}

.message {

font-weight: bold;

margin-top: 10px;

}

.error-message {

color: darkred;

}

.success-message {

color: green;

}

/\* Стили для форм \*/

form {

display: flex;

flex-direction: column;

}

label {

margin-bottom: 5px;

font-weight: bold;

}

input[type="text"],

input[type="email"],

input[type="password"],

input[type="number"],

textarea,

select {

padding: 10px;

margin-bottom: 15px;

border: 1px solid #ccc;

border-radius: 5px;

font-size: 16px;

width: 100%;

}

button {

padding: 10px;

background-color: rebeccapurple;

color: white;

border: none;

border-radius: 5px;

font-size: 16px;

cursor: pointer;

transition: background-color 0.3s ease;

width: 100%;

}

button:hover {

background-color: purple;

}

p {

margin-top: 15px;

text-align: center;

}

a,

.admin-link,

.user-link {

color: rebeccapurple;

text-decoration: none;

font-weight: bold;

margin-bottom: 10px;

display: block;

}

a:hover,

.admin-link:hover,

.user-link:hover {

color: purple;

}