Частное учреждение

профессиональная образовательная организация

ТЕХНИКУМ «БИЗНЕС И ПРАВО»

ЗАЩИЩЕНО С ОЦЕНКОЙ

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

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

(подпись и ФИО руководителя)

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

**ОТЧЕТ О ВЫПОЛНЕНИИ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

|  |  |
| --- | --- |
|  | Разработка веб-приложения «Электронный магазин» |
| По ПМ 09 | Проектирование и разработка веб-приложений |
| Выполнил | Грибенюков К.Н |
| 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](#_Toc161041174)

[1. АНАЛИТИЧЕСКАЯ ЧАСТЬ 4](#_Toc161041175)

[1.1 Анализ предметной области 4](#_Toc161041176)

[1.1.1 Определение концепций и принципов логистики 4](#_Toc161041177)

[1.1.2 Определение требований к системе 4](#_Toc161041178)

[1.2 Анализ существующих решений 4](#_Toc161041179)

[1.3 Проектирование базы данных 5](#_Toc161041180)

[1.4 Цели и задачи выполняемой системы 5](#_Toc161041181)

[1.5 Метод реализации процесса проектирования 5](#_Toc161041182)

[1.6 Обоснование проектных решений 6](#_Toc161041183)

[2. ПРОЕКТНАЯ ЧАСТЬ 15](#_Toc161041184)

[2.1 Техническое задание 15](#_Toc161041185)

[2.2 Программное обеспечение информационной системы 16](#_Toc161041186)

[2.3 Руководство пользователя 17](#_Toc161041187)

[2.4 Руководство администратора 18](#_Toc161041188)

[ЗАКЛЮЧЕНИЕ 19](#_Toc161041189)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 20](#_Toc161041190)

[ПРИЛОЖЕНИЕ 21](#_Toc161041191)

# ВВЕДЕНИЕ

С развитием интернет-технологий и повсеместным использованием мобильных устройств, электронная коммерция стала неотъемлемой частью современного бизнеса. Одной из наиболее популярных категорий товаров в сфере онлайн-торговли являются мобильные телефоны. Интернет-магазины, специализирующиеся на продаже мобильных устройств, предлагают широкий ассортимент товаров, различные условия покупки и доставки, а также возможность получения подробной информации о товарах.

Целью данной курсовой работы является разработка концепции и создание интернет-магазина по продаже мобильных телефонов. В работе рассматриваются основные этапы создания интернет-магазина, начиная от анализа рынка и потребительских предпочтений, выбора платформы для интернет-магазина и его функциональных возможностей, до разработки дизайна и рекламной стратегии.

Актуальность выбранной темы обусловлена не только повышением интереса потребителей к покупке товаров через интернет, но и возрастающей конкуренцией среди интернет-магазинов. Важным аспектом при создании интернет-магазина является учет современных тенденций в сфере электронной коммерции и удовлетворение требований потребителей к удобству покупки и качеству обслуживания.

В работе будут рассмотрены основные этапы создания интернет-магазина, включая анализ рынка, определение целевой аудитории, выбор платформы и технических решений, разработку дизайна и функционала сайта, а также вопросы продвижения и рекламы.

# 1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

## Анализ предметной области

В данном разделе будет проведен анализ предметной области, которая включает в себя изучение концепций, принципов и терминов, связанных с продажей. Это поможет понять, какие проблемы могут быть решены приложением, и какие функции оно должно иметь.

### 1.1.1 Определение концепций и принципов логистики

Первым шагом в анализе предметной области будет изучение основных концепций и принципов продажи. Это поможет понять, какие проблемы могут возникнуть в процессе продажи, и как их можно решить с помощью веб-приложения.

### 1.1.2 Определение требований к системе

На основе изученных концепций и принципов продаж, а также анализа существующих решений, будут определены требования к системе, которые должно удовлетворять веб-приложение.

## Анализ существующих решений

В данном разделе приведен анализ существующих решений с продажей. Это поможет понять, какие проблемы решаются в настоящее время, и какие функции могут быть полезны для разработки нового приложения.

## 1.3 Проектирование базы данных

В данном разделе будет приведено проектирование базы данных для положения. Это включает в себя определение структуры базы данных, выбор подходящего типа данных и структуры данных, а также проектирование отношений между таблицами

## 1.4 Цели и задачи выполняемой системы

Основные цели системы:

1. Разработать удобное веб-приложение для оформления заказов и их покупки.
2. Реализовать функциональность регистрации и авторизации пользователей.

Задачами являются:

1. Достичь поставленных целей;
2. Реализовать функционал для просмотра историй покупок со всеми подробностями.
3. Создать отчет о выполненной работе и результатах обучения.

## 1.5 Метод реализации процесса проектирования

При проектировании веб-приложения, которое использует сторонние API для вывода данных, можно использовать следующий подход:

1) Анализ требований:

1. Определение какие данные необходимо получить из API.
2. Определить как эти данные должны быть представлены в пользовательском интерфейсе.

2) Интеграция с API:

1. Использование PHP для создания запросов к API и получения данных.
2. Обработка ответов API, чтобы извлечь нужные данные.

3) Отображение данных:

1. Используется HTML, CSS и Bootstrap для создания использования пользовательского интерфейса.
2. Выделение полученных данных на страницу с использованием PHP.

4) Обработка ошибок:

1. Реализация обработки и отлавливания ошибок при загрузке API.
2. Ошибки связанных с API будут не доступны для обычных пользователей, вся информация об ошибках будет отображаться в логах системы.

## 1.6 Обоснование проектных решений

Использование определенного стека технологий – можно обосновать использование PHP, Bootstrap, CSS и HTML. Для разработки веб-приложения, это обеспечивает кросс-браузерную совместимость, легкость в изучении и использовании, а так же широкую допустимость библиотек и фреймворков.

Интеграция со сторонними API – обоснование интеграции сторонними API для получения данных о странах. Это позволит упросить процесс получения информации для заполнения заказа а так же его более точного отслеживания.

Использование определённой архитектуры – MVC(Model-View-Controller) для организации кода и его поддерживаемой и масштабируемости.

Система MVC архитектуры, представленных на рисунке 1.1:

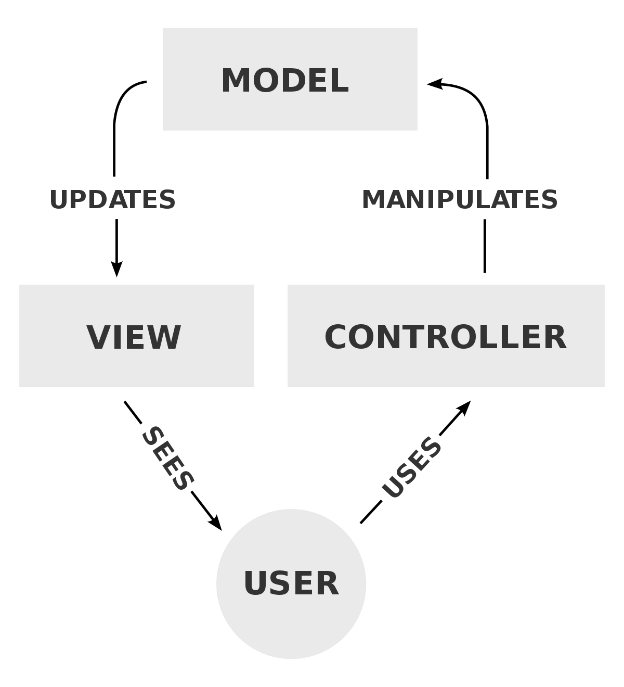


Рисунок 1.1 – MVC архитектура

Выбор определенного способа хранения данных – MySQL, для хранена информации о пользователях, заказах, подсчет посещений, подсчет заказов, хранений другой информации о пользователях или о заказах.

Требования к системе разработаны в соответствии со стандартом качества программного обеспечения ISO 9126:2001, описывающим многоуровневую модель характеристик качества и соответствующий им набор атрибутов.

Требования к системе (международные грузоперевозки) представлены в таблице 1.1.

Таблица 1.1 – Требования к системе учебного курса

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибуты | Метки | | Требования |
| 1 | 2 | | 3 |
| Удобство | | | |
| Простота использования | Среднее время, необходимое  пользователю, чтобы найти  необходимый элемент в программе | | Поиск отдельного элемента интерфейса не должен занимать более 1 секунды |
| Привлекательность | Соответствие интерфейса  требованиям | | 99% элементов должны быть работоспособны для  пользователя |
| Обучаемость | Показатель, затрачиваемый  пользователями на обучение | | Пользователь должен изучить работу ПО за первые 3-5 минут |
|  |  | |  |
| Производительность | | | |
| Временная эффективность | Время выполнения компонента программы | - Приложение должно  обслуживать любую функцию не дольше 1 сек  (не включая задержки в сети); | |
| Эффективность использования ресурсов | Объемы ресурсов требуемых для выполнения задач | - Постоянная память,  используемая программой – не более 10МБ;  - Приложение должно потреблять не более 3МБ  памяти на каждый неактивный сеанс с  пользователем; | |
| 1 | 2 | 3 | |
| Переносимость | | | |
| Удобство установки | Легкость установки | Необходима установка:  xampp, mysql, admin php | |
| Способность к сосуществованию | Способность ПО сосуществовать с другими  программами в общем окружении | Программа не должна  занимать большие объемы памяти устройства (максимум 3МБ) | |
| Удобство замены другого ПО данным | Совместимость структур данных | - Должна поддерживать  внедрение новых версий;  - Система должна быть применима вместо других  программных систем для решения тех же задач в определенном окружении. | |

Требования к численности и квалификации персонала системы и режиму его работы:

В системе существуют 2 группы пользователей: Пользователь и Администратор. Пользователи выполняют функцию: просмотра, заказа, редактирования.

Администратор имеет возможность удаление информации пользователей, добавление, редактирование товаров.

Пользователи системы должны иметь опыт работы с персональным компьютером на базе операционных систем Microsoft Windows на уровне пользователя.

В таблице 1.2 представлены требования к надежности.

Таблица 1.2 – Требования к надёжности

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибуты | Метрики | | Требования | |
| 1 | | 2 | | 3 |
| Надёжность | | | | |
| Устойчивость к отказу | Коэффициент аварийных | | - Не менее 90% ошибок | |
|  | отказов | | в программе должны  обрабатываться без  экстренного завершения;  - Обеспечение бесперебойного питание  активного сетевого  оборудования; | |
| Способность к восстановлению | Отношение количества числа успешного восстановление работы системы к ее аварийному завершению за определённый промежуток времени | | - Восстановление  работоспособности  системы не должно превышать 5 минут; | |
| Уровень зрелости | Среднее время работы без сбоев | | - Система должна  работать 24 часа в сутки;  - Система должна оповещать  - Система должна оповещать пользователя при некорректных действиях в системе;  - Данные, которые вводит пользователь, должны сохраняться. | |

Средняя доступность приложения должна составлять не менее 99%

Среднее время между сбоями — это среднее время, за которое компонент или модуль может выполнять свои функции без перерыва.

Измеряется от начала работы до момента следующего сбоя.

Среднее временя работы без сбоев должно составлять не менее 1500 часов.

В таблице 1.3 представлен перечень и критерии отказов для каждой функции, по которой задаются требования по надежности.

Таблица 1.3 – Перечень и критерии отказов для каждой функции

|  |  |
| --- | --- |
| Функции | Критерии отказа |
| 1 | 2 |
| Отображение данных | Отсутствие соединения с базой данных |
| Добавление | Отсутствие соединения с базой данных |
| Редактирование | Отсутствие соединения с базой данных |
| Удаление | Отсутствие соединения с базой данных |

Требования к функциям web-приложения, выполняемым программным обеспечением

Имя: Вход пользователя

Описание: Пользователь заходит на главную страницу

Действующий субъект: Пользователь

Предусловие: пользователь зашел на сайт для оформления товара Сценарий:

1. Пользователь зарегистрировался на сайте
2. Авторизировался под своими учетными данными
3. Зашел на главную страницу
4. Выбрал товар
5. В корзине заполнил данные

Расширения: в случае не верно введенных данных при авторизации будет возникать ошибка «Неверный логин или пароль», в случае не верно заполненных полей заказа, будет выводиться ошибка «Введены не корректные данные, попробуйте позже».

Для функционирования системы требуется наличие сервера и клиентских компьютеров, на которых будет производиться работа с системой. Перед вводом системы в действие необходимо:

1. интернет-соединение;
2. обеспечить каждого специалиста доступом к web-хостингу;
3. иметь данные для доступа к админ-панели;
4. база данных MySQL.

Разрабатываемая система требует в своей архитектуре наличие БД-сервера со свободной реляционной системой управления – MySQL.

База данных информационной системы должна состоять из следующих связанных таблиц:

1. products – таблица содержащая данные о посещении пользователей.
2. users – таблица содержащая информацию о пользователях.
3. cart – таблица содержащая информацию о заказах

Данные о пользователях представлены в таблице users 1.4.

Таблица 1.4 – Данные о пользователях

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| id | int | 11 | Идентификатор пользователя | Первичный ключ |
| username | Varchar | 25 | Название категории | NOT NULL |
| Password\_hash | Long | 255 | Пароль пользователя | NOT NULL |
| email | Varchar | 30 | Email пользователя | NOT NULL |
| First\_name | Straing | 30 | Имя | NOT NULL |
| Last\_name | Straing | 30 | Отчество | NOT NULL |
| status\_id | int | 11 | status | NOT NULL |
| Created\_at | date | 50 | Дата регистрации | DATE |
| countries | varchar | 30 | cont | NOT NULL |

Данные о коризне должны размещаться в таблице cart .5.

Таблица 1.5 – Данные о корзины

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| id | Int | 11 | Идентификатор посещения | Первичный ключ |
| product\_id | name | 50 | ID Продукта | NOT NULL |
| user\_id | int | 100 | ID пользователя | NOT NULL |
| price | int | 255 | Цена | NOT NULL |

Данные о товаров должны размещаться в таблице cart 1.6.

Таблица 1.6. – Данные о товарах.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничения |
| Id | Int | 255 | Индентификатор заказа | Первичный ключ |
| User\_id | Int | 255 | Индентификатор пользователя | User id |
| image | Varchar | 255 | изображение | NOT NULL |
| description | text | 255 | описание | NOT NULL |
| price | Decimal | 255 | цена | NOT NULL |

Данные о контенте странице «Соглашения» размещается в таблице products 1.7.

Рисунок 1.2. – окно оформления товара.

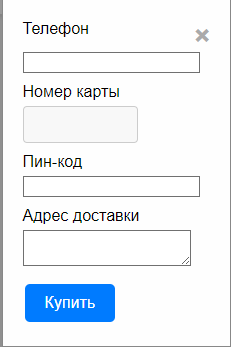


Рисунок 1.3. - профиль пользователя.

На рисунке 1.3 представлен профиль пользователя.

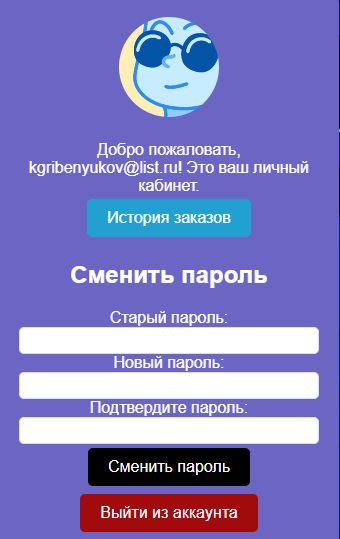
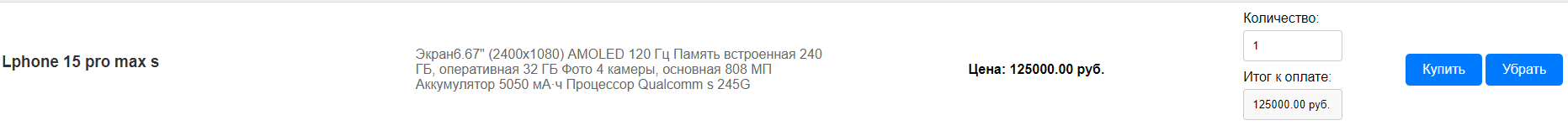


Рисунок 1.4. - страница просмотра заказа.

На рисунке 1.4 – представлена страница просмотра конкретного заказа.



На данном этапе разберем, то как будем вести себя система при выполнение выделенных модулей. Для этого мы будем использовать диаграммы состояний в нотации UML.

На рисунке 2.1 представлена общая диаграмма состояний. При переходе пользователем в программу пользователь будет находиться в том, состоянии, когда ему необходимо авторизироваться в системе, чтобы продолжить работу с системой и осуществлять деятельность согласно выделенным вариантам использования.

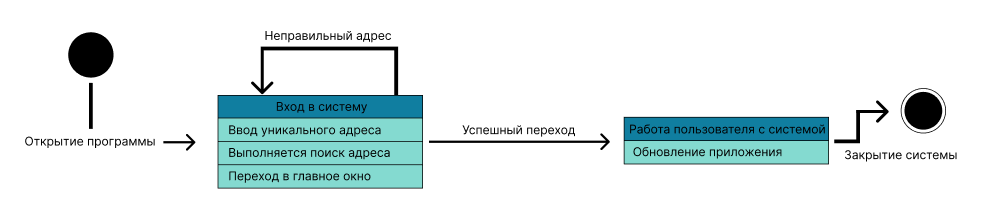


Рисунок 2.1 – Диаграмма состояний добавление данных

На рисунке 2.2 изображена декомпозиция состояния «Редактирования».

Сотрудник должен нажать на кнопку «Редактировать». После этого приложение обновится и обновит данные в базе данных. Далее сотрудник может просмотреть все изменённые данные.

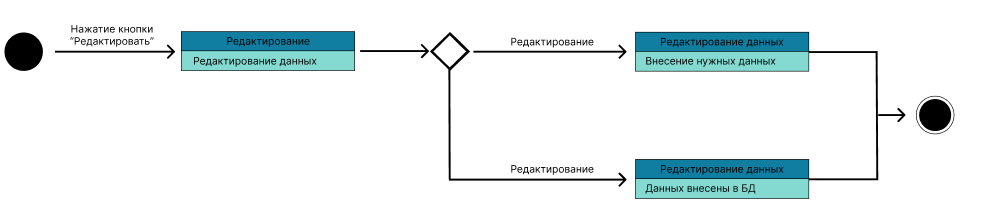


Рисунок 2.2 – Диаграмма состояний редактирования данных

# 2. ПРОЕКТНАЯ ЧАСТЬ

## 2.1 Техническое задание

Система управления контентом (административная часть сайта) должна предоставлять возможность добавления, редактирования и удаления содержимого статических и динамических страниц. Также должна быть предусмотрена возможность добавления информации без отображения на сайте.

Наполнение различных сайтов, функционирование которых поддерживается одной и той же инсталляцией системы, должно храниться под управлением единой СУБД.

Система управления контентом должна иметь стандартный для Windows интерфейс, отвечающий следующим требованиям:

1. реализация в графическом оконном режиме;
2. единый стиль оформления;
3. интуитивно понятное назначение элементов интерфейса;
4. отображение на экране только тех возможностей, которые доступны конкретному пользователю;
5. отображение на экране только необходимой для решения текущей прикладной задачи информации;
6. отображение на экране хода длительных процессов обработки;
7. диалог с пользователем должен быть оптимизирован для выполнения типовых и часто используемых операций;
8. для операций по массовому вводу информации должна быть предусмотрена минимизация количества нажатий на клавиатуру для выполнения стандартных действий.

HTML (от английского HyperText Markup Language) — это язык гипертекстовой разметки текста. Он нужен, чтобы размещать на веб-странице элементы: текст, картинки, таблицы и видео.

HTML состоит из тегов — команд, которые указывают браузеру, как отображать помещённый в них текст. Это и есть элементы веб-страницы. У каждого тега есть имя, которое заключается в угловые скобки < и >.

Для реализации интерактивных элементов сервероной части должны использоваться языки php, javascript.

Для реализации динамических страниц должен использоваться язык PHP.

Объем одной стандартной загружаемой страницы сайта в среднем не должен превышать 170 kb.

Объем flash-заставки не должен превышать 300 Kb.

CSS (Cascading Style Sheets, каскадные таблицы стилей) — язык описания внешнего вида HTML-документа. Это одна из базовых технологий в современном интернете. Практически ни один сайт не обходится без CSS, поэтому HTML и CSS действуют в единой связке.

Каскадные таблицы стилей работают с HTML, но это совершенно другой язык. HTML структурирует документ и упорядочивает информацию, а CSS взаимодействует с браузером, чтобы придать документу оформление.

## 2.2 Программное обеспечение информационной системы

SQL Server является одной из наиболее популярных систем управления базами данных (СУБД) в мире. Данная СУБД подходит для самых различных проектов: от небольших приложений до больших высоконагруженных проектов.

SQL Server был создан компанией Microsoft. Первая версия вышла в 1987 году. А текущей версией является версия 2022, которая вышла в ноябре 2022 году и которая будет использоваться в текущем руководстве.

SQL Server долгое время был исключительно системой управления базами данных для Windows, однако начиная с версии 16 эта система доступна и на Linux.

Необходимое программное обеспечение для персонального компьютера:

1) Операционная система: Windows 10/Linux/macOS

Сайт должен быть доступен для полнофункционального просмотра с помощью следующих браузеров:

1. Opera 6.0 и выше;
2. Google;
3. Yandex.

JavaScript - предназначен для написания сценариев для активных HTML-страниц. Язык JavaScript не имеет никакого отношения к языку Java. Java разработан фирмой SUN. JavaScript - фирмой Netscape Communication Corporation. Первоначальное название - LiveScript. После завоевания языком Java всемирной известности LiveScript из коммерческих соображений переименовали в JavaScript.

2.3 Руководство пользователя

Авторизация и Регистрация: для доступа к профилю пользователя и оформлению заказа необходимо зарегистрироваться и авторизироваться, после этого на главной странице появится форма для оформления заказа.

Если у вас нет аккаунта, вы можете создать его, нажав «Создать аккаунт» и заполнив необходимую информацию, включая имя, логин, пароль.

После успешной авторизации или регистрации вы попадете на главную страницу.

Профиль и Настройки:

В личном кабинете есть возможность смены пароля, при нажатии на “сменить пароль” должен заполнить поля и нажать на кнопку сменить.

Выход из Аккаунта:

В левой части экрана будет кнопка “Выход” при нажатии на нее происходит ди авторизация из аккаунта.

2.4 Руководство администратора

Авторизация:

Как администратор, вы авторизуетесь так же, как и обычный пользователь, нажимая на «Авторизация» и вводя свой логин и пароль.

Роль администратора определена в базе данных как «admin».

Доступ к Админ-Панели:

После успешной авторизации под Администратором у вас появляется доступ к “Админ-панели”(в персональном кабинете), там выводится вся информация о сайте, так же есть возможность редактирования, добавления и удаления.

Нажмите на этот пункт, чтобы перейти в административную панель.

Обзор Пользователей:

В админ-панели отображается таблица со всеми пользователями, включая их данные.

Каждая строка содержит кнопки «Удалить» для управления данными пользователя.

Функции Администрирования:

Удалить: Кнопка «Удалить» удаляет профиль пользователя. Подтвердите действие, чтобы завершить удаление.

Безопасность:

Выход из Админ-Панели:

Для завершения сеанса в админ-панели выберите «Выйти» из вашего личного кабинета.

# ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы была разработана концепция интернет-магазина по продаже мобильных телефонов. Были рассмотрены основные этапы создания и запуска интернет-магазина, начиная от анализа рынка и потребительских предпочтений, выбора платформы для интернет-магазина и его функциональных возможностей, до разработки дизайна и рекламной стратегии.

В ходе работы было выявлено, что создание успешного интернет-магазина требует не только хорошего знания целевой аудитории и конкурентной среды, но и профессионального подхода к разработке и продвижению сайта. Особое внимание было уделено выбору платформы для интернет-магазина, его дизайна и функциональности, а также эффективным методам продвижения и рекламы.

Интернет-магазин по продаже мобильных телефонов представляет собой перспективное и конкурентоспособное предприятие в условиях современного рынка. Важно строить работу магазина на принципах качественного обслуживания, удобства для покупателей и постоянного совершенствования ассортимента и сервиса.

Это программное обеспечение включает в себя:

1. язык программирования PHP, HTML, Bootstrap, CSS, JS;

2. реляционная система управления БД MySQL;

Всё информационное наполнение сайта находится в БД.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Вандюк, Джон К. CMS Drupal. Руководство по разработке системы управления сайтом / Вандюк, Джон К., Мэтт Вестгейт,. - М.: Вильямс, 2019. - 400 c.
2. Гаевский, А.Ю. 100% самоучитель. Создание Web-страниц и Web-сайтов. HTML и JavaScript / А.Ю. Гаевский, В.А. Романовский. - М.: Триумф, 2020. - 464 c.
3. Дакетт, Джон HTML и CSS. Разработка и дизайн веб-сайтов / Джон Дакетт. - Москва: Наука, 2019. - 480 c.
4. Дронов, В. JavaScript в Web-дизайне / В. Дронов. - М.: СПб: БХВ, 2019. - 880 c.
5. Дронов, В. PHP, MySQL и Dreamweaver. Разработка интерактивных Web-сайтов / В. Дронов. - М.: БХВ-Петербург, 2019. - 480 c.
6. Дронов, В.А. PHP 5/6, MySQL 5/6 и Dreamweaver CS4. Разработка интерактивных Web-сайтов / В.А. Дронов. - М.: БХВ-Петербург, 2019. - 820 c.
7. Дронов, Владимир JavaScript и AJAX в Web-дизайне / Владимир Дронов. - Москва: Высшая школа, 2019. - 736 c.
8. Дронов, Владимир PHP, MySQL и Dreamweaver MX 2004. Разработка интерактивных Web-сайтов / Владимир Дронов. - М.: "БХВ-Петербург", 2019. - 448 c.
9. Колисниченко, Д. PHP 5/6 и MySQL 6. Разработка Web-приложений / Д. Колисниченко. - М.: БХВ-Петербург, 2021. - 560 c.
10. Колисниченко, Д.Н. PHP 5/6 и MySQL 6. Разработка Web-приложений / Д.Н. Колисниченко. - М.: БХВ-Петербург, 2021. - 364 c.

# ПРИЛОЖЕНИЕ

Account.php

<?php

session\_start();

// Проверяем, есть ли сессия

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

// Если сессия есть, перенаправляем пользователя на страницу личного кабинета

header('Location: personal\_account.php');

exit;

}

// Генерация случайного числа для капчи

$captcha = rand(1000, 9999);

// Сохранение капчи в сессии

$\_SESSION['captcha'] = $captcha;

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Bapst - Личный кабинет</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

overflow: hidden;

background-color: #f0f0f0;

}

header {

background-color: #6B65C3;

color: white;

padding: 10px 0;

text-align: center;

}

nav ul {

list-style-type: none;

padding: 0;

}

nav ul li {

display: inline;

margin: 0 10px;

}

nav ul li a {

color: white;

text-decoration: none;

}

footer {

background-color: #333;

color: white;

text-align: center;

padding: 10px 0;

position: fixed;

bottom: 0;

width: 100%;

}

.logo {

width: 100px;

height: auto;

}

.training-item {

border: 1px solid #ddd;

padding: 10px;

margin-bottom: 10px;

background-color: #fff;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

.training-item h3 {

margin-top: 0;

color: #333;

}

.training-item p {

margin: 5px 0;

color: #666;

}

.training-item strong {

color: #000;

}

.training-item button {

background-color: #007bff;

color: white;

border: none;

padding: 10px 20px;

text-align: center;

text-decoration: none;

display: inline-block;

font-size: 16px;

margin: 4px 2px;

cursor: pointer;

border-radius: 5px;

}

.training-item button:hover {

background-color: #0056b3;

}

.training-item img {

max-width: 20%;

height: auto;

}

.site-title {

font-size: 2em;

font-weight: bold;

text-transform: uppercase;

letter-spacing: 2px;

margin: 0;

padding: 10px 0;

}

header {

display: flex;

align-items: center;

}

.logo {

margin-right: 10px; /\* Добавляем отступ между логотипом и названием сайта \*/

}

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f0f0f0;

}

header {

background-color: #6B65C3;

color: white;

padding: 10px 20px;

display: flex;

align-items: center;

}

.logo {

height: 50px;

margin-right: 20px; /\* Отступ от логотипа до названия \*/

max-width:60px

}

.site-title {

font-size: 2em;

font-weight: bold;

text-transform: uppercase;

letter-spacing: 2px;

margin: 0;

}

nav {

margin-left: auto; /\* Отступ от навигации до правого края \*/

}

nav ul {

list-style-type: none;

padding: 0;

display: flex;

}

nav ul li {

margin-left: 10px;

}

nav ul li a {

color: white;

text-decoration: none;

}

main {

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;

padding: 20px;

}

.account-section {

background-color: #fff;

padding: 20px;

margin-bottom: 200px;

border-radius: 5px;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

text-align: center;

max-width: 300px;

}

.account-section h2 {

color: #333;

}

.account-section form {

display: flex;

flex-direction: column;

align-items: center;

}

.account-section label {

margin-bottom: 5px;

color: #666;

}

.account-section input[type="text"],

.account-section input[type="email"],

.account-section input[type="password"] {

padding: 10px;

margin-bottom: 15px;

border: 1px solid #ddd;

border-radius: 5px;

width: 100%;

}

.account-section button {

padding: 10px;

background-color: #050505;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

width: 100%;

}

.account-section button {

/\* ... (предыдущие стили для кнопки) ... \*/

transition: transform 0.3s ease-in-out; /\* Добавляем плавную анимацию \*/

}

.account-section button:active {

transform: scale(0.95) translateY(-5px); /\* Уменьшаем размер и смещаем вверх \*/

}

.account-section button:hover {

background-color: ;

}

.hidden {

display: none;

}

.switch-form {

margin-top: 10px;

color: #666;

}

.switch-form button {

background: none;

border: none;

color: #050505;

cursor: pointer;

padding: 0;

}

.switch-form button:hover {

color: #0056b3;

}

.switch-form button:disabled {

color: #ccc;

cursor: not-allowed;

}

.site-title-link {

color: white; /\* Change the color to white \*/

text-decoration: none; /\* Remove the underline \*/

}

.site-title-link:hover {

text-decoration: underline; /\* Add underline on hover \*/

}

</style>

</head>

<body>

<header>

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

<h1 class="site-title">

<a class="site-title-link" href="main.php">Bapst</a>

</h1>

<nav>

<ul>

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

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

<li><a href="product.php">О компании</a></li>

<li><a href="partners.php">Партнеры, агенты</a></li>

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

<li><a href="cart.php">Корзина</a></li>

<li><a href="account.php">Личный кабинет</a></li>

</ul>

</nav>

</header>

<script>

</script>

<main>

<!-- Форма авторизации -->

<div class="account-section" id="login-section">

<?php

if (isset($\_SESSION['message'])) {

echo "<p>{$\_SESSION['message']}</p>";

unset($\_SESSION['message']);

}

?>

<h2>Авторизация</h2>

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

<label for="auth-email">Email:</label>

<input type="email" id="auth-email" name="email" required>

<label for="auth-password">Пароль:</label>

<input type="password" id="auth-password" name="password" required>

<div class="form-group">

<label for="captcha">Капча:</label>

<input type="text" class="form-control" id="captcha" name="captcha" required>

<p>Введите число: <?php echo $captcha; ?></p> <!-- Отображение капчи -->

</div>

<button type="submit">Войти</button>

</form>

<div class="switch-form">

<button id="register-btn">Нет аккаунта? Зарегистрироваться</button>

</div>

</div>

<script>

function validateForm() {

var firstName = document.forms["registrationForm"]["username"].value;

var lastName = document.forms["registrationForm"]["lastname"].value;

var password = document.forms["registrationForm"]["password"].value;

// Проверка пароля на минимальную длину

if (password.length < 8) {

alert("Пароль должен содержать не менее 8 символов.");

return false;

}

// Проверка имени и фамилии на заглавные буквы и отсутствие цифр

var namePattern = /^[A-ZА-Я][a-zа-я]\*$/;

if (!namePattern.test(firstName) || !namePattern.test(lastName)) {

alert("Имя и фамилия должны начинаться с заглавной буквы и не должны содержать цифр.");

return false;

}

return true;

}

</script>

</head>

<body>

<div class="account-section hidden" id="register-section">

<h2>Регистрация</h2>

<form action="register.php" name="registrationForm" onsubmit="return validateForm()" method="post">

<label for="reg-email">Email:</label>

<input type="email" id="reg-email" name="email" required>

<label for="reg-username">Имя:</label>

<input type="text" id="reg-username" name="username" required>

<label for="reg-lastname">Фамилия:</label>

<input type="text" id="reg-lastname" name="lastname" required>

<label for="reg-password">Пароль:</label>

<input type="password" id="reg-password" name="password" required><br>

<button type="submit" name="go">Зарегистрироваться</button>

</form>

<div class="switch-form">

<button id="login-btn">Уже есть аккаунт? Войти</button>

</div>

</div>

</main>

</form>

<script>

// Скрипт для переключения между формами

document.getElementById('register-btn').addEventListener('click', function() {

document.getElementById('login-section').classList.add('hidden');

document.getElementById('register-section').classList.remove('hidden');

});

document.getElementById('login-btn').addEventListener('click', function() {

document.getElementById('register-section').classList.add('hidden');

document.getElementById('login-section').classList.remove('hidden');

});

</script>

<footer>

<p>&copy; 2024 Мобильные устройства. Все права защищены.</p>

</footer>

</body>

</html>

Add\_main

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Add Product</title>

    <!-- Подключение Bootstrap CSS -->

    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

</head>

<style>

       .add-button {

            display: inline-block;

            margin-top: 20px;

            margin-bottom: 20px;

        }

        .add-button a {

            background-color: #007BFF;

            color: white;

            padding: 10px 20px;

            text-decoration: none;

            border-radius: 5px;

            transition: background-color 0.3s ease;

        }

        .add-button a:hover {

            background-color: #0056b3;

        }

    </style>

<body>

    <div class="container mt-5">

        <h2>Добавление товара</h2>

        <form action="add\_product.php" method="post">

            <div class="form-group">

                <label for="image">Изображение:</label>

                <input type="text" class="form-control" id="image" name="image" required>

            </div>

            <div class="form-group">

                <label for="name">Название:</label>

                <input type="text" class="form-control" id="name" name="name" required>

            </div>

            <div class="form-group">

                <label for="description">Описание:</label>

                <textarea class="form-control" id="description" name="description" required></textarea>

            </div>

            <div class="form-group">

                <label for="price">Цена:</label>

                <input type="number" class="form-control" id="price" name="price" step="0.01" required>

            </div>

            <button type="submit" name="add\_product" class="btn btn-primary">Добавить товар</button>

            <a href="admin\_panel.php">Вернуться</a>

        </form>

    </div>

    <!-- Подключение Bootstrap JS и jQuery -->

    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</body>

</html>

Add\_product.php

<?php

// Start the session to access the user's ID

session\_start();

// Database connection details

$dbHost = 'localhost';

$dbUsername = 'root';

$dbPassword = '';

$dbName = 'shop';

// Create a database connection

$conn = new mysqli($dbHost, $dbUsername, $dbPassword, $dbName);

// Check connection

if ($conn->connect\_error) {

    die("Connection failed: " . $conn->connect\_error);

}

// Check if the form is submitted

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

    // Retrieve the form data

    $image = $\_POST['image'];

    $name = $\_POST['name'];

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

    $price = $\_POST['price'];

    // Проверка наличия сообщения в сессии

if (isset($\_SESSION['message'])) {

    // Вывод сообщения с помощью JavaScript

    echo '<script>alert("' . htmlspecialchars($\_SESSION['message']) . ' 😊");</script>';

    // Удаление сообщения из сессии после его вывода

    unset($\_SESSION['message']);

}

    // Prepare the SQL statement to insert the product with NULL users\_id

    $stmt = $conn->prepare("INSERT INTO products (image, name, description, price) VALUES (?, ?, ?, ?)");

    $stmt->bind\_param("sssd", $image, $name, $description, $price);

    // Execute the product insert statement

    if ($stmt->execute()) {

        // Redirect to a success page or back to the form

        header("Location: admin\_panel.php");

        exit;

    } else {

        // Handle the error

        echo "Error inserting product: " . $stmt->error;

    }

    // Close the statement

    $stmt->close();

}

// Close the database connection

$conn->close();

?>

Add\_to\_cart.php

<?php

require\_once 'db.php';

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

    // Проверяем, существует ли параметр product\_id

    if (isset($\_POST['product\_id'])) {

        // Получаем ID товара из POST-запроса

        $product\_id = $\_POST['product\_id'];

        session\_start();

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

            echo '<script>alert("Для добавления товара в корзину необходимо войти в аккаунт.😉"); window.location.href = "account.php";</script>';

            exit;

        }

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

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

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

        $stmt = $pdo->prepare('INSERT INTO cart (product\_id, user\_id, quantity, price) VALUES (:product\_id, :user\_id, :quantity, :price)');

        $stmt->execute(['product\_id' => $product\_id, 'user\_id' => $user\_id, 'quantity' => $quantity, 'price' => $price]);

        header('Location: cart.php');

        exit;

    }

}

?>

Admin\_panel.php

<?php

// personal\_account.php

session\_start();

// Подключение к базе данных

$dsn = 'mysql:host=localhost;dbname=shop';

$username = 'root';

$password = '';

try {

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

} catch (PDOException $e) {

    echo 'Подключение не удалось: ' . $e->getMessage();

}

// Получение списка пользователей из базы данных

$stmt = $db->query("SELECT id, email FROM users");

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

// Предположим, что у вас есть подключение к базе данных и переменная $db

$stmt = $db->query("SELECT id, name, description, price FROM products");

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

// Получение списка пользователей из базы данных

$stmt = $db->query("SELECT id, email, created\_at FROM users");

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

// Обработка удаления пользователя

if (isset($\_POST['delete\_user'])) {

    $userId = $\_POST['user\_id'];

    // Удаление пользователя из базы данных

    $stmt = $db->prepare("DELETE FROM users WHERE id = :id");

    $stmt->execute(['id' => $userId]);

    // Установка сообщения об успешном удалении

    $\_SESSION['message'] = "Пользователь с ID $userId успешно удален.";

    // Перенаправление на страницу админ-панели

    header('Location: admin\_panel.php');

    exit;

}

// Проверка, вошел ли пользователь

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

    // Если пользователь не вошел, перенаправляем на страницу входа

    header('Location: login.php');

    exit;

}

// Отображение личного кабинета пользователя

$welcomeMessage = "Добро пожаловать, " . $\_SESSION['username'] . "! Это ваш личный кабинет.";

if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['change\_password'])) {

    // Получение данных из формы

    $oldPassword = $\_POST['old\_password'];

    $newPassword = $\_POST['new\_password'];

    $confirmPassword = $\_POST['confirm\_password'];

    // Проверка, совпадает ли старый пароль с текущим паролем пользователя

    if (password\_verify($oldPassword, $\_SESSION['password'])) {

        // Проверка, совпадают ли новый пароль и подтверждение пароля

        if ($newPassword === $confirmPassword) {

            // Хеширование нового пароля

            $hashedPassword = password\_hash($newPassword, PASSWORD\_DEFAULT);

            // Обновление пароля в базе данных

            $stmt = $db->prepare("UPDATE users SET password = :password WHERE id = :id");

            $stmt->execute(['password' => $hashedPassword, 'id' => $\_SESSION['user\_id']]);

            // Обновление даты изменения пароля в базе данных

            $stmt = $db->prepare("UPDATE users SET password\_change\_date = NOW() WHERE id = :id");

            $stmt->execute(['id' => $\_SESSION['user\_id']]);

            // Обновление пароля в сессии

            $\_SESSION['password'] = $hashedPassword;

            // Установка сообщения в сессию

            $\_SESSION['message'] = "Пароль успешно изменен!";

        } else {

            // Установка сообщения в сессию

            $\_SESSION['message'] = "Новый пароль и подтверждение пароля не совпадают.";

        }

    } else {

        // Установка сообщения в сессию

        $\_SESSION['message'] = "Старый пароль введен неверно.";

    }// Проверка, совпадает ли старый пароль с текущим паролем пользователя

    if (password\_verify($oldPassword, $\_SESSION['password'])) {

        // Проверка, совпадают ли новый пароль и подтверждение пароля

        if ($newPassword === $confirmPassword) {

            // Хеширование нового пароля

            $hashedPassword = password\_hash($newPassword, PASSWORD\_DEFAULT);

            // Обновление пароля в базе данных

            $stmt = $db->prepare("UPDATE users SET password = :password WHERE id = :id");

            $stmt->execute(['password' => $hashedPassword, 'id' => $\_SESSION['user\_id']]);

            // Обновление даты изменения пароля в базе данных

            $stmt = $db->prepare("UPDATE users SET password\_change\_date = NOW() WHERE id = :id");

            $stmt->execute(['id' => $\_SESSION['user\_id']]);

            // Обновление пароля в сессии

            $\_SESSION['password'] = $hashedPassword;

            // Установка сообщения в сессию

            $\_SESSION['message'] = "Пароль успешно изменен!";

        } else {

            // Установка сообщения в сессию

            $\_SESSION['message'] = "Новый пароль и подтверждение пароля не совпадают.";

        }

    } else {

        // Установка сообщения в сессию

        $\_SESSION['message'] = "Старый пароль введен неверно.";

    }

    // Перенаправление на страницу личного кабинета

    header('Location: account.php');

    exit;

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Bapst - Админ панель</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            margin: 0;

            padding: 0;

            background-color: #f0f0f0;

        }

        header {

            background-color: #333;

            color: white;

            padding: 10px 0;

            text-align: center;

        }

        nav ul {

            list-style-type: none;

            padding: 0;

        }

        nav ul li {

            display: inline;

            margin: 0 10px;

        }

        nav ul li a {

            color: white;

            text-decoration: none;

        }

        main {

            padding: 20px;

        }

        footer {

            background-color: #333;

            color: white;

            text-align: center;

            padding: 10px 0;

            bottom: 0;

            width: 100%;

            position:fixed;

        }

        .logo {

            width: 100px;

            height: auto;

        }

        .site-title {

            font-size: 2em;

            font-weight: bold;

            text-transform: uppercase;

            letter-spacing: 2px;

            margin: 0;

            padding: 10px 0;

        }

        header {

            display: flex;

            align-items: center;

        }

        .logo {

            margin-right: 10px; /\* Добавляем отступ между логотипом и названием сайта \*/

        }

        body {

            font-family: Arial, sans-serif;

            margin: 0;

            padding: 0;

            background-color: #ffffff;

        }

        header {

            background-color: #6B65C3;

            color: white;

            padding: 10px 20px;

            display: flex;

            align-items: center;

        }

        .logo {

            height: 50px;

            margin-right: 20px; /\* Отступ от логотипа до названия \*/

            max-width:60px

        }

        .site-title {

            font-size: 2em;

            font-weight: bold;

            text-transform: uppercase;

            letter-spacing: 2px;

            margin: 0;

        }

        nav {

            margin-left: auto; /\* Отступ от навигации до правого края \*/

        }

        nav ul {

            list-style-type: none;

            padding: 0;

            display: flex;

        }

        nav ul li {

            margin-left: 10px;

        }

        nav ul li a {

            color: white;

            text-decoration: none;

        }

        main {

            padding: 2  0px;

        }

        .event-container {

            display: flex;

            flex-wrap: wrap;

            justify-content: center;

            gap: 20px;

        }

        .event-card {

            background-color: #fff;

            border-radius: 10px;

            box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

            padding: 20px;

            text-align: center;

            transition: transform 0.3s ease;

        }

        .event-card:hover {

            transform: translateY(-5px);

        }

        .event-card img {

            max-width: 100%;

            height: auto;

            border-radius: 5px;

            cursor: pointer;

            width: 200px; /\* Фиксированная ширина \*/

            height: auto; /\* Высота автоматически подстраивается \*/

        }

        .event-card h2 {

            margin-top: 10px;

        }

        .event-card p {

            margin-bottom: 0;

        }

        h1 {

            text-align: center;

        }

        .event-container {

            display: flex;

            flex-wrap: wrap;

            justify-content: center;

            gap: 20px;

        }

        .event-card {

            background-color: #fff;

            border-radius: 10px;

            box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

            padding: 20px;

            text-align: center;

            transition: transform 0.3s ease;

            width: 250px;

        }

        .event-card:hover {

            transform: translateY(-5px);

        }

        .event-card img {

            max-width: 100%;

            height: auto;

            border-radius: 5px;

            cursor: pointer;

            width: 200px;

            height: auto;

        }

        .event-card h4 {

            margin-top: 10px;

            font-size: 1.2em;

            color: #333;

        }

        .event-card p {

            margin-bottom: 0;

            font-size: 0.9em;

            color: #666;

        }

        .event-card a {

            text-decoration: none;

            color: #007BFF;

        }

        .event-card a:hover {

            text-decoration: underline;

        }

        .content {

            padding: 20px;

            background: #ffffff;

            margin-top: 20px;

            border-radius: 5px;

            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

        }

        .content p {

            line-height: 1.6;

            color: #333;

        }

        .content table {

            width: 100%;

            border-collapse: collapse;

            margin-top: 20px;

        }

        .content table th,

        .content table td {

            padding: 10px;

            text-align: left;

            border-bottom: 1px solid #ddd;

        }

        .content table th {

            background-color: #f4f4f4;

        }

        .logout-button {

    background-color: #a10b0b;

    color: white;

    padding: 10px 10px;

    text-align: center;

    text-decoration: none;

    display: inline-block;

    font-size: 16px;

    margin: 4px 2px;

    cursor: pointer;

    border: none;

    border-radius: 5px;

    transition: background-color 0.3s ease;

}

.welcome-block {

            background-color: #ebebeb;

            padding: 10px;

            margin-bottom: 10px;

        }

        .change-password-block {

            background-color: #ffffff;

            padding: 10px;

            max-width:250px;

            margin:0 auto;

        }

        .change-button{

            background-color: #000;

            color: white; /\* Белый текст \*/

            padding: 10px 20px; /\* Внутренние отступы \*/

            text-align: center; /\* Выравнивание текста по центру \*/

            text-decoration: none; /\* Удаление подчеркивания \*/

            display: inline-block; /\* Отображение кнопки в строке с другими элементами \*/

            font-size: 16px; /\* Размер шрифта \*/

            margin: 4px 2px; /\* Внешние отступы \*/

            cursor: pointer; /\* Изменение курсора при наведении \*/

            border: none; /\* Удаление границ \*/

            border-radius: 5px; /\* Скругление углов \*/

            transition: background-color 0.3s ease; /\* Плавное изменение цвета фона \*/

        }

        body {

            font-family: Arial, sans-serif;

        }

        .welcome-block {

            background-color: #d1cfe7;

            padding: 10px;

            margin-bottom: 10px;

        }

        .logout-block {

            background-color: #e0e0e0;

            padding: 10px;

            margin-bottom: 10px;

        }

        .delete-user-block {

            background-color: #d0d0d0;

            padding: 10px;

        }

        .user-list {

            margin-top: 20px;

        }

        .add-button {

            display: inline-block;

            margin-top: 20px;

            margin-bottom: 20px;

        }

      </style>

    </head>

<body>

    <header>

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

        <h1 class="site-title">

        <h1 class="site-title">

    <a class="site-title-link" href="main.php">Bapst</a>

</h1>

        <nav>

            <ul>

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

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

                <li><a href="product.php">О компании</a></li>

                <li><a href="partners.php">Партнеры, агенты</a></li>

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

                <li><a href="cart.php">Корзина</a></li>

                <li><a href="account.php">Личный кабинет</a></li>

            </ul>

        </nav>

    </header>

    <main>

    <div class="welcome-block">

        <h2>Добро пожаловать, <?php echo $\_SESSION['username']; ?>!</h2>

        <p>Вы вошли как администратор.</p>

        <div class="welcome-block">

    <h2 >Сменить пароль</h2>

        <form action="change\_password.php" method="post">

            <label for="old\_password">Старый пароль:</label><br>

            <input type="password" id="old\_password" name="old\_password" required><br>

            <label for="new\_password">Новый пароль:</label><br>

            <input type="password" id="new\_password" name="new\_password" required><br>

            <label for="confirm\_password">Подтвердите пароль:</label><br>

            <input type="password" id="confirm\_password" name="confirm\_password" required><br>

            <button type="submit" name="change\_password" class="change-button">Сменить пароль</button>

             </form>

            <form action="logout.php" method="post"><br>

            <button type="submit" name="logout" class="logout-button">Выйти из аккаунта</button>

               </form> </div></div>

        <div class="add-button">

    <a href="delete\_product1.php">Перейти к удолению продукта</a>

    </div>

    <div class="add-button">

    <a href="delete\_user.php">Перейти к удолению пользователю</a>

    </div><br>

        <div class="add-button">

    <a href="add\_main.php">Перейти к добавлению</a>

    </div>

    <div class="add-button">

    <a href="change\_prod.php">Перейти к редактированию</a>

    </div>

       </main>

    <footer>

        <p>&copy; 2024 Мобильные устройства. Все права защищены.</p>

    </footer>

</body>

</html>

Main.php

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Bapst - Мобильные устройства</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            margin: 0;

            padding: 0;

            background-color: #f0f0f0;

        }

             nav ul li {

            display: inline;

            margin: 0 10px;

        }

        nav ul li a {

            color: white;

            text-decoration: none;

        }

        main {

            padding: 20px;

        }

        footer {

            background-color: #333;

            color: white;

            text-align: center;

            padding: 10px 0;

            bottom: 0;

            width: 100%;

        }

       .site-title-link {

    color: white; /\* Change the color to white \*/

    text-decoration: none; /\* Remove the underline \*/

}

.site-title-link:hover {

    text-decoration: underline; /\* Add underline on hover \*/

}

    </style>

</head>

<body>

    <header>

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

        <h1 class="site-title">

    <a class="site-title-link" href="main.php">Bapst</a>

</h1>

        <nav>

            <ul>

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

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

                <li><a href="product.php">О компании</a></li>

                <li><a href="partners.php">Партнеры, агенты</a></li>

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

                <li><a href="cart.php">Корзина</a></li>

                <li><a href="account.php">Личный кабинет</a></li>

            </ul>

        </nav>

    </header>

      <main class="container">

        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

        <div id="carouselExampleIndicators" class="carousel slide" data-bs-ride="carousel">

            <div class="carousel-indicators">

                <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="0" class="active" aria-current="true" aria-label="Slide 1"></button>

                <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="1" aria-label="Slide 2"></button>

                <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="2" aria-label="Slide 3"></button>

            </div>

            <div class="carousel-inner">

                <div class="carousel-item active">

                    <img src="png/slider.png" class="d-block w-100" alt="Телефон 1">

                </div>

                <div class="carousel-item">

                    <img src="png/phone1.png" class="d-block w-100" alt="Телефон 2">

                </div>

                <div class="carousel-item">

                    <img src="png/phone2.png" class="d-block w-100" alt="Телефон 3">

                </div>

            </div>

            <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide="prev">

                <span class="carousel-control-prev-icon" aria-hidden="true"></span>

                <span class="visually-hidden">Previous</span>

            </button>

            <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide="next">

                <span class="carousel-control-next-icon" aria-hidden="true"></span>

                <span class="visually-hidden">Next</span>

            </button>

        </div>

              <main class="container">

                    <h1 class="text-center my-5">Мобильная техника</h1>

        <?php

// Подключаем файл db.php

require\_once 'db.php';

// Выполняем запрос к базе данных для получения данных о товарах

$stmt = $pdo->query('SELECT \* FROM products');

// Отображаем товары на странице

$count = 0;

while ($row = $stmt->fetch(PDO::FETCH\_ASSOC)) {

    // Если счетчик равен 0, начинаем новый ряд

    if ($count == 0) {

        echo '<div class="row">';

    }

    echo '<div class="col-md-3 mb-4">';

    echo '<div class="product-item">';

    echo '<img src="' . $row['image'] . '" alt="' . $row['name'] . '">';

    echo '<h3>' . $row['name'] . '</h3>';

    echo '<p>' . $row['description'] . '</p>';

    echo '<strong>Цена: ' . $row['price'] . ' руб.</strong>';

    echo '<form action="add\_to\_cart.php" method="post">';

    echo '<input type="hidden" name="product\_id" value="' . $row['id'] . '">';

    echo '<button type="submit" class="btn btn-primary" name="add\_to\_cart">В корзину</button>';

    echo '<a href="peper.php?id=' . $row['id'] . '" class="btn btn-primary">О товаре</a>';

    echo '</form>';

    echo '</div>';

    echo '</div>';

        //счетчик

    $count++;

    //Если счетчик равен 4, закрываем текущий ряд и сбрасываем счетчик

    if ($count == 4) {

        echo '</div>';

        $count = 0;

    }

}

//Если счетчик не равен 0, закрываем последний ряд

if ($count != 0) {

    echo '</div>';

}

?>

      <div class="container-map">

             <br>   <h3 align="center">Мы находимся здесь</h3>

             <div class="iframe-container">

        <iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d2286.795154658112!2d36.38877557729758!3d55.029282272825014!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x46cad57996c3522f%3A0x6d3aaaef8db6fa8!2z0JLQviDQn9GD0YjQutC40L3QsCDQtNC-0Lwg0LrQvtC70L7RgtGD0YjQutC40L3QsA!5e0!3m2!1sru!2sru!4v1708070851128!5m2!1sru!2sru" width="11000" height="450" style="border:0;" allowfullscreen="" loading="lazy" referrerpolicy="no-referrer-when-downgrade"></iframe>

    </div>

        </div>

    </main>

         <footer>

        <p>&copy; 2024 Мобильные устройства. Все права защищены.</p>

    </footer>

        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

</body>

</html>

Peper.php

<?php

// Подключение к базе данных

$db = new mysqli('localhost', 'root', '', 'shop');

// Проверка соединения

if ($db->connect\_error) {

    die("Connection failed: " . $db->connect\_error);

}

// Получение ID товара из GET-параметра

$productId = $\_GET['id'] ?? null;

// Получение данных товара

$query = "SELECT \* FROM products WHERE id = ?";

$stmt = $db->prepare($query);

$stmt->bind\_param("i", $productId);

$stmt->execute();

$result = $stmt->get\_result();

$product = $result->fetch\_assoc();

$db->close();

?>

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <title>Страница товара</title>

    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css">

</head>

<body>

    <div class="container mt-5">

        <div class="row">

            <div class="col-md-6">

                <img src="<?php echo $product['image']; ?>" alt="<?php echo $product['name']; ?>" class="img-fluid">

            </div>

            <div class="col-md-6">

                <h1><?php echo $product['name']; ?></h1>

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

                <p>Цена: <?php echo $product['price']; ?> руб.</p>

                <div class="form-group">

                    <label for="colorSelect">Выберите цвет:</label>

                    <select class="form-control" id="colorSelect">

                        <option>Белый</option>

                        <option>Черный</option>

                        <option>Золотой</option>

                        <!-- Добавьте другие цвета по необходимости -->

                    </select>

                </div>

                <a href="main.php" class="btn btn-primary"><i class="fas fa-arrow-left"></i> Вернуться</a>

            </div>

        </div>

        <div class="row mt-4">

    <div class="col">

        <a href="404.html" class="btn btn-secondary btn-block"><i class="fas fa-info-circle"></i> Подробнее о товаре</a>

    </div>

    <div class="col">

        <a href="404.html" class="btn btn-secondary btn-block"><i class="fas fa-truck"></i> Доставка и оплата</a>

    </div>

    <div class="col">

        <a href="404.html" class="btn btn-secondary btn-block"><i class="fas fa-shield-alt"></i> Гарантия</a>

    </div>

    <div class="col">

        <a href="404.html" class="btn btn-secondary btn-block"><i class="fas fa-sync-alt"></i> Обмен и возврат</a>

    </div>

</div></div> </div>

    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.1/dist/umd/popper.min.js"></script>

    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</body>

</html>

Personal\_account.php

<?php

// personal\_account.php

session\_start();

// Подключение к базе данных

$dsn = 'mysql:host=localhost;dbname=shop';

$username = 'root';

$password = '';

try {

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

} catch (PDOException $e) {

    echo 'Подключение не удалось: ' . $e->getMessage();

}

// Проверка, вошел ли пользователь

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

    // Если пользователь не вошел, перенаправляем на страницу входа

    header('Location: login.php');

    exit;

}

// Получаем пользователя из базы данных

$stmt = $db->prepare('SELECT \* FROM users WHERE id = :id');

$stmt->execute(['id' => $\_SESSION['user\_id']]);

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

// Проверка роли пользователя и перенаправление

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

    // Если пользователь является администратором, перенаправляем на admin\_panel.php

    header('Location: admin\_panel.php');

    exit;

}

// Отображение личного кабинета пользователя

$welcomeMessage = "Добро пожаловать, " . $\_SESSION['username'] . "! Это ваш личный кабинет.";

if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['change\_password'])) {

    // Получение данных из формы

    $oldPassword = $\_POST['old\_password'];

    $newPassword = $\_POST['new\_password'];

    $confirmPassword = $\_POST['confirm\_password'];

    // Проверка, совпадает ли старый пароль с текущим паролем пользователя

    if (password\_verify($oldPassword, $\_SESSION['password'])) {

        // Проверка, совпадают ли новый пароль и подтверждение пароля

        if ($newPassword === $confirmPassword) {

            // Хеширование нового пароля

            $hashedPassword = password\_hash($newPassword, PASSWORD\_DEFAULT);

            // Обновление пароля в базе данных

            $stmt = $db->prepare("UPDATE users SET password = :password WHERE id = :id");

            $stmt->execute(['password' => $hashedPassword, 'id' => $\_SESSION['user\_id']]);

            // Обновление даты изменения пароля в базе данных

            $stmt = $db->prepare("UPDATE users SET password\_change\_date = NOW() WHERE id = :id");

            $stmt->execute(['id' => $\_SESSION['user\_id']]);

            // Обновление пароля в сессии

            $\_SESSION['password'] = $hashedPassword;

            // Установка сообщения в сессию

            $\_SESSION['message'] = "Пароль успешно изменен!";

        } else {

            // Установка сообщения в сессию

            $\_SESSION['message'] = "Новый пароль и подтверждение пароля не совпадают.";

        }

    } else {

        // Установка сообщения в сессию

        $\_SESSION['message'] = "Старый пароль введен неверно.";

    }

    // Перенаправление на страницу личного кабинета

    header('Location: account.php');

    exit;

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Bapst - Личный кабинет</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            margin: 0;

            padding: 0;

            background-color: #f0f0f0;

        }

        header {

            background-color: #6B65C3;

            color: white;

            padding: 10px 0;

            text-align: center;

        }

        nav ul {

            list-style-type: none;

            padding: 0;

        }

        nav ul li {

            display: inline;

            margin: 0 10px;

        }

        nav ul li a {

            color: white;

            text-decoration: none;

        }

        main {

            padding: 20px;

        }

        footer {

            background-color: #333;

            color: white;

            text-align: center;

            padding: 10px 0;

            position: fixed;

            bottom: 0;

            width: 100%;

        }

        .logo {

            width: 100px;

            height: auto;

        }

        .training-item {

            border: 1px solid #ddd;

            padding: 10px;

            margin-bottom: 10px;

            background-color: #fff;

            box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

        }

        .training-item h3 {

            margin-top: 0;

            color: #333;

        }

        .training-item p {

            margin: 5px 0;

            color: #666;

        }

        .training-item strong {

            color: #000;

        }

        .training-item button {

            background-color: #007bff;

            color: white;

            border: none;

            padding: 10px 20px;

            text-align: center;

            text-decoration: none;

            display: inline-block;

            font-size: 16px;

            margin: 4px 2px;

            cursor: pointer;

            border-radius: 5px;

        }

        .training-item button:hover {

            background-color: #0056b3;

        }

        .training-item img {

            max-width: 20%;

            height: auto;

        }

        .site-title {

            font-size: 2em;

            font-weight: bold;

            text-transform: uppercase;

            letter-spacing: 2px;

            margin: 0;

            padding: 10px 0;

        }

        header {

            display: flex;

            align-items: center;

        }

        .logo {

            margin-right: 10px; /\* Добавляем отступ между логотипом и названием сайта \*/

        }

        body {

            font-family: Arial, sans-serif;

            margin: 0;

            padding: 0;

            background-color: #f0f0f0;

            overflow: hidden; /\* Убираем скроллбар \*/

 }

        header {

            background-color: #6B65C3;

            color: white;

            padding: 10px 20px;

            display: flex;

            align-items: center;

        }

        .logo {

            height: 50px;

            margin-right: 20px; /\* Отступ от логотипа до названия \*/

            max-width:60px

        }

        .site-title {

            font-size: 2em;

            font-weight: bold;

            text-transform: uppercase;

            letter-spacing: 2px;

            margin: 0;

        }

        nav {

            margin-left: auto; /\* Отступ от навигации до правого края \*/

        }

        nav ul {

            list-style-type: none;

            padding: 0;

            display: flex;

        }

        nav ul li {

            margin-left: 10px;

        }

        nav ul li a {

            color: white;

            text-decoration: none;

        }

        main {

            padding: 2  0px;

        }

        .event-container {

            display: flex;

            flex-wrap: wrap;

            justify-content: center;

            gap: 20px;

        }

        .event-card {

            background-color: #fff;

            border-radius: 10px;

            box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

            padding: 20px;

            text-align: center;

            transition: transform 0.3s ease;

        }

        .event-card:hover {

            transform: translateY(-5px);

        }

        .event-card img {

            max-width: 100%;

            height: auto;

            border-radius: 5px;

            cursor: pointer;

            width: 200px; /\* Фиксированная ширина \*/

            height: auto; /\* Высота автоматически подстраивается \*/

        }

        .event-card h2 {

            margin-top: 10px;

        }

        .event-card p {

            margin-bottom: 0;

        }

        h1 {

            text-align: center;

        }

        .event-container {

            display: flex;

            flex-wrap: wrap;

            justify-content: center;

            gap: 20px;

        }

        .event-card {

            background-color: #fff;

            border-radius: 10px;

            box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

            padding: 20px;

            text-align: center;

            transition: transform 0.3s ease;

            width: 250px;

        }

        .event-card:hover {

            transform: translateY(-5px);

        }

        .event-card img {

            max-width: 100%;

            height: auto;

            border-radius: 5px;

            cursor: pointer;

            width: 200px;

            height: auto;

        }

        .event-card h4 {

            margin-top: 10px;

            font-size: 1.2em;

            color: #333;

        }

        .event-card p {

            margin-bottom: 0;

            font-size: 0.9em;

            color: #666;

        }

        .event-card a {

            text-decoration: none;

            color: #007BFF;

        }

        .event-card a:hover {

            text-decoration: underline;

        }

        .content {

            padding: 20px;

            background: #ffffff;

            margin-top: 20px;

            border-radius: 5px;

            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

        }

        .content p {

            line-height: 1.6;

            color: #333;

        }

        .content table {

            width: 100%;

            border-collapse: collapse;

            margin-top: 20px;

        }

        .content table th,

        .content table td {

            padding: 10px;

            text-align: left;

            border-bottom: 1px solid #ddd;

        }

        .content table th {

            background-color: #f4f4f4;

        }

        .content img {

            max-width: 40%;

            height: auto;

            display: block;

            margin: 50px auto;

            border-radius: 5px;

            box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);

        }

        .change-button{

            background-color: #000;

            color: white; /\* Белый текст \*/

            padding: 10px 20px; /\* Внутренние отступы \*/

            text-align: center; /\* Выравнивание текста по центру \*/

            text-decoration: none; /\* Удаление подчеркивания \*/

            display: inline-block; /\* Отображение кнопки в строке с другими элементами \*/

            font-size: 16px; /\* Размер шрифта \*/

            margin: 4px 2px; /\* Внешние отступы \*/

            cursor: pointer; /\* Изменение курсора при наведении \*/

            border: none; /\* Удаление границ \*/

            border-radius: 5px; /\* Скругление углов \*/

            transition: background-color 0.3s ease; /\* Плавное изменение цвета фона \*/

        }

           .logout-button {

            background-color: #a10b0b; /\* Красный цвет фона \*/

            color: white; /\* Белый текст \*/

            padding: 10px 20px; /\* Внутренние отступы \*/

            text-align: center; /\* Выравнивание текста по центру \*/

            text-decoration: none; /\* Удаление подчеркивания \*/

            display: inline-block; /\* Отображение кнопки в строке с другими элементами \*/

            font-size: 16px; /\* Размер шрифта \*/

            margin: 4px 2px; /\* Внешние отступы \*/

            cursor: pointer; /\* Изменение курсора при наведении \*/

            border: none; /\* Удаление границ \*/

            border-radius: 5px; /\* Скругление углов \*/

            transition: background-color 0.3s ease; /\* Плавное изменение цвета фона \*/

        }

        .schange-button{

            background-color: #21a0d1;

            color: white; /\* Белый текст \*/

            padding: 10px 20px; /\* Внутренние отступы \*/

            text-align: center; /\* Выравнивание текста по центру \*/

            text-decoration: none; /\* Удаление подчеркивания \*/

            display: inline-block; /\* Отображение кнопки в строке с другими элементами \*/

            font-size: 16px; /\* Размер шрифта \*/

            margin: 4px 2px; /\* Внешние отступы \*/

            cursor: pointer; /\* Изменение курсора при наведении \*/

            border: none; /\* Удаление границ \*/

            border-radius: 5px; /\* Скругление углов \*/

            transition: background-color 0.3s ease; /\* Плавное изменение цвета фона \*/

        }

        .logout-button:hover {

            background-color: #da190b; /\* Темно-красный цвет фона при наведении \*/

        }

        .account-section {

            background-color: #6b65c3;

            padding: 20px;

            margin-bottom: 200px;

            border-radius: 5px;

            box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

            text-align: center;

            max-width: 300px;

            height: 700px;

        }

        .avatar-image {

            width: 100px;

            height: 100px;

            border-radius: 50%;

            margin-bottom: 20px;

        }

        .background-image {

            position: fixed;

            top: 0;

            left: 0;

            width: 100%;

            height: 100%;

            object-fit: cover;

            z-index: -1;

        }

        main {

            display: ;

            flex-direction: column;

            align-items: center;

            justify-content: center;

            min-height: 100vh;

            text-align: center;

        }

        .welcome-message {

            color: white; /\* Изменение цвета текста \*/

        }

        form {

    grid-template-columns: auto auto;

  gap: 10px; /\* Отступ между элементами \*/

}

label {

  text-align: right; /\* Выравнивание текста по правому краю \*/

}

input {

  width: 100%; /\* Ширина поля ввода \*/

  padding: 5px; /\* Внутренние отступы \*/

  border-radius: 5px; /\* Скругление углов \*/

  border: 1px solid #ddd; /\* Граница \*/

  box-sizing: border-box; /\* Учитывать границу в ширине \*/

}   .site-title-link {

    color: white; /\* Change the color to white \*/

    text-decoration: none; /\* Remove the underline \*/

}

.site-title-link:hover {

    text-decoration: underline; /\* Add underline on hover \*/

}

    </style>

    </head>

<body>

        <header>

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

        <h1 class="site-title">

    <a class="site-title-link" href="main.php">Bapst</a>

</h1>

 <nav>

            <ul>

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

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

                <li><a href="product.php">О компании</a></li>

                <li><a href="partners.php">Партнеры, агенты</a></li>

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

                <li><a href="cart.php">Корзина</a></li>

                <li><a href="account.php">Личный кабинет</a></li>

            </ul>

        </nav>

    </header>

    <main>

               <div class="account-section" id="login-section">

            <!-- Добавление аватара -->

        <img src="png/avatar.png" alt="Аватар пользователя" class="avatar-image">

                <!-- Вывод сообщения внутри тела HTML -->

        <div class="welcome-message">

            <?php echo $welcomeMessage; ?>

        <!-- Форма для смены пароля -->

        </form>

            <form id="orderHistoryForm" action="order\_history.php" method="post">

            <button type="submit" class="schange-button">История заказов</button>

</form>

        <h2 >Сменить пароль</h2>

        <form action="change\_password.php" method="post">

            <label for="old\_password">Старый пароль:</label>

            <input type="password" id="old\_password" name="old\_password" required><br>

            <label for="new\_password">Новый пароль:</label>

            <input type="password" id="new\_password" name="new\_password" required><br>

            <label for="confirm\_password">Подтвердите пароль:</label>

            <input type="password" id="confirm\_password" name="confirm\_password" required><br>

            <button type="submit" name="change\_password" class="change-button">Сменить пароль</button>

      </form>

 </div>

  <!-- Форма для выхода из аккаунта -->

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

            <button type="submit" name="logout" class="logout-button">Выйти из аккаунта</button>

        </form>

            <div class="switch-form">

            </div>

        </div>

    </main>

    <!-- Добавление фона -->

    <img src="png/5.png" alt="Фон" class="background-image">

    <footer>

        <p>&copy; 2024 Мобильные устройства. Все права защищены.</p>

    </footer>

</body>

</html>