Частное учреждение

профессиональная образовательная организация

ТЕХНИКУМ «БИЗНЕС И ПРАВО»

ЗАЩИЩЕНО С ОЦЕНКОЙ

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

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

(подпись и ФИО руководителя)

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

**КУРСОВАЯ РАБОТА**

|  |  |
| --- | --- |
| Тема | Разработка серверной части веб-приложения «Доставка готовой продукции» |
| По МДК 09.01 | Проектирование и разработка веб-приложений |
| Выполнил | Ханахян А.А. |
| Курс, группа | 4 курс, группа ИС-4 |
| Специальность | 09.02.07 Информационные системы и программирование |
| Научный руководитель | Мартыненко В.А. |

г. Белореченск

2023 г.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc153439075)

[1. АНАЛИТИЧЕСКАЯ ЧАСТЬ 4](#_Toc153439076)

[1.1 Анализ предметной области 4](#_Toc153439077)

[1.2 Цели и задачи выполняемой системы 4](#_Toc153439078)

[1.3 Метод реализации процесса проектирования 4](#_Toc153439079)

[1.4 Обоснование проектных решений 5](#_Toc153439080)

[2. ПРОЕКТНАЯ ЧАСТЬ 18](#_Toc153439081)

[2.1 Техническое задание 18](#_Toc153439082)

[2.2 Установка и настройка среды разработки 19](#_Toc153439083)

[2.3 Руководство пользователя 21](#_Toc153439084)

[2.4 Руководство администратора 22](#_Toc153439085)

[ЗАКЛЮЧЕНИЕ 23](#_Toc153439086)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 24](#_Toc153439087)

ПРИЛОЖЕНИЕ

# ВВЕДЕНИЕ

Веб-приложения для доставки готовой продукции становятся все более популярными в современном мире, обеспечивая удобство и эффективность как для потребителей, так и для поставщиков. Разработка серверной части такого приложения играет ключевую роль в обеспечении его надежной и масштабируемой работы. В данном отчете рассматривается процесс разработки серверной части веб-приложения "Доставки готовой продукции"

Объект исследования – Веб-приложение для доставки готовой продукции.

Предмет исследования – предметом исследования являются процессы размещения информации.

Цель исследование – рассмотреть и установить повышение эффективности процессов поиска, добавления, редактирования, удаления, информации на сайте за счет разработки и внедрения информационной системы доставки продукции.

Задачи исследования:

* изучить предметную область;
* рассмотреть ряд существующих разработок для решения поставленной цели;
* выбрать метод реализации работы доставки готовых продукций;
* спроектировать работу разрабатываемой системы доставок готовых продукций;
* разработать информационную систему;
* протестировать разработанную информационную доставки готовой продукции.

# 1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

* 1. Анализ предметной области

Разрабатываемая система предназначена для более удобного и лёгкого заказа продукции.

Система будет очень удобной, легкой в своем использовании. Она позволит пользователям создавать свои учетные записи, редактировать и удалять их, а также они смогут добавлять товары в корзину. Доступный и понятный интерфейс сайта сделает время провождение на нем более простым и эффективным для пользователя

## 1.2 Цели и задачи выполняемой системы

Основными целями системы:

* удобное авторизация на сайте;
* удобное представление информации посетителям сайта;  
   изменение их параметров;
* удобное интерфейс;
* удаление учетной записи;
* обеспечить систему автоматическим обновлением данных

Задачами являются:

* достичь поставленных целей;
* автоматическое обновление данных;
* представить посетителям сайта информацию о продукции
* оформить отчёт;
* освоить навыки, полученные в ходе разработки данного сайта;
* сделать структурированную программу.

## 1.3 Метод реализации процесса проектирования

При разработке веб-приложения «Доставка готовой продукции» внедряются инновационные методы, направленные на обеспечение эффективного управления контентом. Процесс проектирования системы включает в себя управление пользователями, обеспечивая легкий доступ к товарам.

При внедрении системы будет достигнуто:

* добавление пользователя;
* редактирование пользователя;
* удаление пользователя;
* просмотр товаров;
* обеспечение удобной и правильной работы системы.

Объекты автоматизации включают в себя:

* обеспечивать максимальную работоспособность;
* быть надежной и технологичной;
* обслуживание должно проводиться минимальным количеством персонала;
* Регулярное обновление и совершенствование товаров сайта.

## 1.4 Обоснование проектных решений

Разрабатываемый веб-приложение «Доставка готовой продукции» будет реализован в формате трехзвенной клиент-серверной архитектуры. Вот обоснование выбора такой архитектуры:

Разделение на три уровня:

Эта архитектура разделяет приложение на три основных уровня: представление данных, бизнес-логику и уровень данных. Это позволит более гибко управлять и развивать каждую часть приложения независимо.

Единое веб-приложение:

Трехзвенная архитектура позволяет объединить серверную и клиентскую части в рамках одного веб-приложения. Это способствует удобству управления и поддержке всего контента сайта.

Использование технологий:

Применение PHP, HTML, JavaScript и CSS позволит создать динамический интерфейс, реализовать бизнес-логику и форматировать контент веб-приложения.

Совместная архитектура:

Веб-приложения будет использовать единую архитектуру для взаимодействия серверной и клиентской частей, обеспечивая единообразие и согласованность при работе с данными.

Отсутствие необходимости в отдельных клиентах и серверах:

Не потребуется развертывание и обслуживание отдельных серверов для обеспечения работы учебника, так как все ресурсы будут интегрированы в рамках веб-приложения.

Преимущества трехзвенной архитектуры:

* Гибкость и масштабируемость системы.
* Повышенный уровень безопасности.
* Эффективная производительность за счет распределения задач между сервером и клиентом.

Работа с базой данных:

Использование MySQL для хранения информации и управления данными, обеспечивающее структурированное хранение товара.

На рисунке 1 представлена структура трехзвенной клиент-серверной архитектуры.

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Автоматически созданное описание

Рисунок 1 - 3-х звенная клиент серверная архитектура

Это обеспечит создание сайта с доставкой готовой продукции с удобным интерфейсом, гибкостью, безопасностью и высокой производительностью, упрощая взаимодействие пользователей с материалами обучения.

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 (товары на сайте) представлены в таблице 1.1.

Таблица 1.1 – Требования к системе

|  |  |  |  |
| --- | --- | --- | --- |
| 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  работе | | Пользователь должен изучить работу ПО за первые 3-5 минут  использования |
| Производительность | | | |
| Временная эффективность | Время выполнения компонента программы | - Приложение должно  обслуживать любую функцию не дольше 1 сек  (не включая задержки в сети); | |

Продолжение таблицы 1.1 - Требования к системе

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 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; |
| 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 | - Система должна быть  разработана по общим принципам и не требовать навыков от пользователя;  - 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 1 группы пользователей: пользователи. Пользователи выполняет функцию:

* просмотр данных,
* внесение данных,
* редактирование данных,
* удаления данных.

В таблице 1.2 представлены требования к надежности.

Таблица 1.2 – Требования к надёжности

|  |  |  |
| --- | --- | --- |
| Атрибуты | Метрики | Требования |
| Надёжность | | |
| 1 | 2 | 3 |
| Устойчивость к отказу | Коэффициент аварийных  отказов | - Не менее 90% ошибок  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 пользователи должны сохраняться. |
| Средняя доступность |  | Средняя доступность изображение должна составлять не менее 99% |
| Среднее время между сбоями |  | Среднее время между сбоями должно быть не менее 2 часа |

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 3 представлен перечень и критерии отказов для каждой функции, по которой задаются требования по надежности.

Таблица 3 – перечень и критерии отказов для каждой функции

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

Имя: вход пользователя

Описание: Пользователь может войти на сайт, используя свои учетные данные.

Действующий субъект: Пользователь

Предусловие: Пользователь имеет учетную запись в системе

Сценарий:

* Пользователь открывает окно авторизации.
* Вводит свой логин и пароль.
* Система проверяет введенные данные.
* В случае успешной аутентификации, пользователь входит в систему и получает доступ к персонализированному контенту.
* В случае некорректных учетных данных, система выводит сообщение об ошибке.

Расширения: при отсутствии соединения с сервером при нажатии на

кнопку «Войти» выведется сообщение об ошибке.

На рисунке 1 представлена диаграмма последовательности сценария «Вход пользователя».

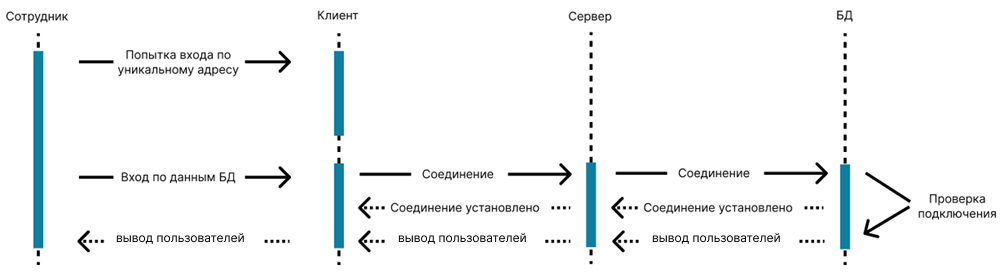


Рисунок 1 - диаграмма последовательности сценария «Вход пользователя».

Имя: регистрация пользователя

Описание: Новый пользователь может зарегистрироваться в системе для получения доступа к товарам.

Действующий субъект: Новый пользователь

Предусловие: Пользователь отсутствует в системе и желает создать учетную запись.

Сценарий:

* Пользователь открывает страницу регистрации.
* Пользователь вводит необходимую информацию: логин, пароль, подтверждение пароля, электронная почта и аватарку.
* Пользователь проходит каптчу.
* Система проверяет корректность введенных данных.
* В случае успешной проверки, новый аккаунт пользователя создан

Расширения: при отсутствии соединения с сервером при нажатии на

кнопку «Зарегистрироваться» выведется сообщение об ошибке.

На рисунке 2 представлена диаграмма последовательности сценария «Регистрация пользователя».

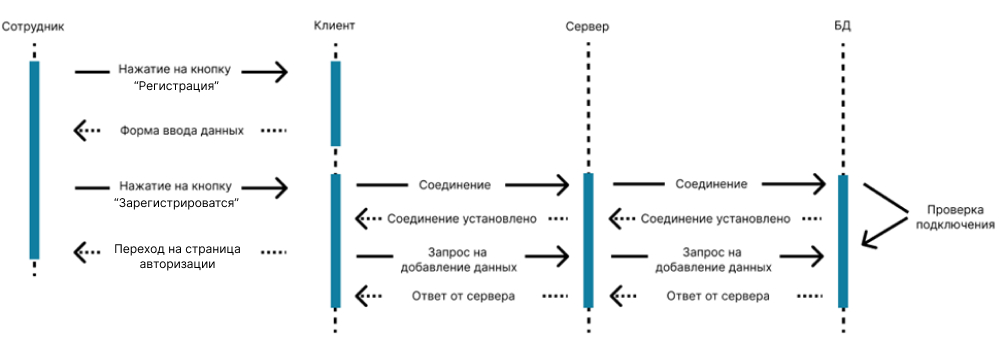


Рисунок 2 - Диаграмма последовательности сценария «Регистрация пользователя»

Имя: редактирование данных

Описание: сотрудник имеет возможность редактировать данные своей учетной записи

Действующий субъект: Сотрудник

Предусловие: Пользователь имеет учетную запись в системе и хочет её редактировать

Сценарий:

* Пользователь открывает главную страницу
* Пользователь нажимает на кнопку «Редактировать»
* Пользователь вносит необходимые изменения в данных своей учетной записи
* Система проверяет корректность введенных изменений.
* При успешном редактирование, пользователь должен входит с изменёнными данными.

Расширения: при отсутствии соединения с сервером при нажатии на

кнопку «Редактировать» выведется сообщение об ошибке.

На рисунке 3 представлена диаграмма последовательности сценария

"Редактирования".

Изображение выглядит как текст, линия, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3 – Диаграмма последовательности сценария "Редактирования

Имя: удаления профиля пользователя

Описание: Пользователь может удалить свою учетную запись.

Действующий субъект: Пользователь

Предусловие: Пользователь авторизован в системе и имеет доступ к управлению своими данными.

Сценарий:

* Пользователь открывает главную страницу
* Пользователь нажимает на кнопку «Удалить»
* Пользователь подтверждает удаление.
* Система проверяет возможность удаления и производит соответствующие операции.

Расширения: при отсутствии соединения с сервером при нажатии на

кнопку «Удалить» выведется сообщение об ошибке.

На рисунке 4 представлена диаграмма последовательности сценария «Удаление профиля пользователя».

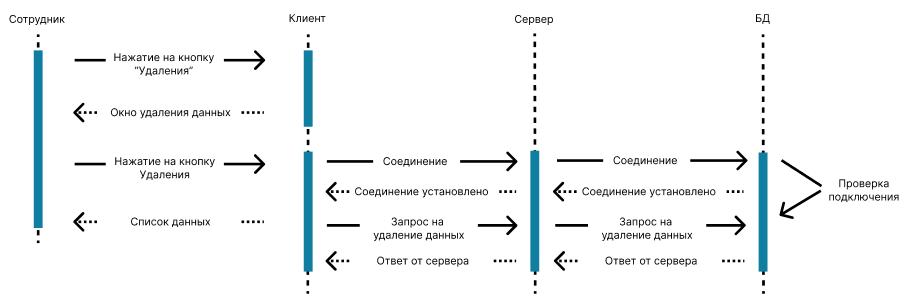


Рисунок 4 - диаграмма последовательности сценария «Удаление профиля пользователя»

Для эффективного функционирования системы «Доставка готовой продукции» необходимо обеспечить наличие сервера и персональных компьютеров для клиентов, на которых будет производиться работа с системой. Перед внедрением системы необходимо выполнить следующие шаги:

* установить подключение к интернету;
* предоставить каждому специалисту персональный компьютер;
* установить программное обеспечение на каждый компьютер;
* обеспечить наличие программы xampp;
* установить и настроить базу данных MySQL.

Разрабатываемая система требует наличия в своей архитектуре сервера базы данных с реляционной системой управления, такой как MySQL.

Структура базы данных информационной системы включает в себя следующую таблицу:

* users – таблица содержащая информацию о пользователях.
* sets – таблица содержащая информацию о товара.
* pizza – таблица содержащая информацию о товара.
* salads – таблица содержащая информацию о товара.

Данные о пользователях должны размещаться в таблице users:

Таблица 2.1 – Таблица users в базе данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| id | int | 100 | Идентификатор категории | Первичный ключ |
| name | varchar | 356 | Название категории | NOT NULL |
| login | varchar | 255 | login | NOT NULL |
| email | varchar | 255 | email | NOT NULL |
| password | varchar | 500 | password | NOT NULL |
| avatar | varchar | 600 | avatar | NOT NULL |
| role | varchar | 255 | role | NOT NULL |

Данные о товаре должны размещаться в таблице sets :

Таблица 2.2 – Таблица sets в базе данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| id | int | 100 | Идентификатор категории | Первичный ключ |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| sushi\_name | varchar | 255 | Название товара | NOT NULL |
| description | varchar | 255 | Информация о товаре | NOT NULL |
| price | varchar | 255 | Цена товара | NOT NULL |
| img | varchar | 255 | Изображение товара | NOT NULL |

Данные о товаре должны размещаться в таблице pizza:

Таблица 2.3 – Таблица pizza в базе данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| id | int | 100 | Идентификатор категории | Первичный ключ |
| pizza\_name | varchar | 355 | Название товара | NOT NULL |
| description | varchar | 255 | Информация о товаре | NOT NULL |
| price | varchar | 255 | Цена товара | NOT NULL |
| img | varchar | 255 | Изображение товара | NOT NULL |

Данные о товаре должны размещаться в таблице salads :

Таблица 2.3 – Таблица salads в базе данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| id | int | 100 | Идентификатор категории | Первичный ключ |
| Salads\_name | varchar | 355 | Название товара | NOT NULL |
| description | varchar | 255 | Информация о товаре | NOT NULL |
| price | varchar | 255 | Цена товара | NOT NULL |
| img | varchar | 255 | Изображение товара | NOT NULL |

На рисунке 5 представлено окно добавления(регистрации) новых пользователей в базу данных.

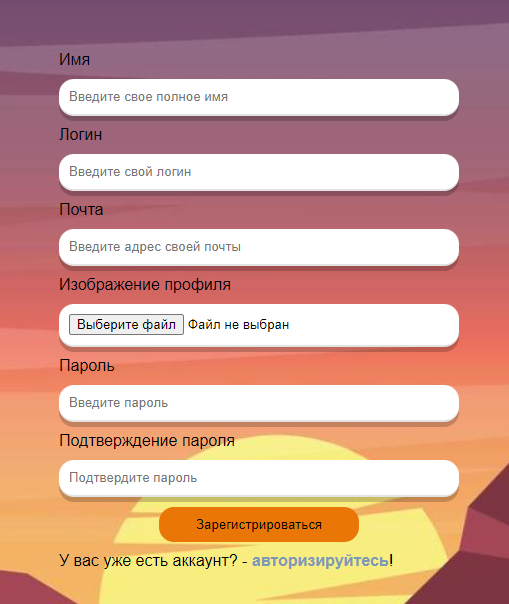


Рисунок 5 – Окно добавления новых пользователей

На рисунке 6 представлено окно авторизации пользователя.

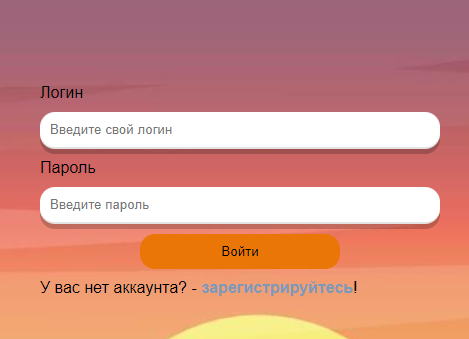


Рисунок 6 – Окно авторизации пользователя

На рисунке 7 представлено окно просмотра продукции.

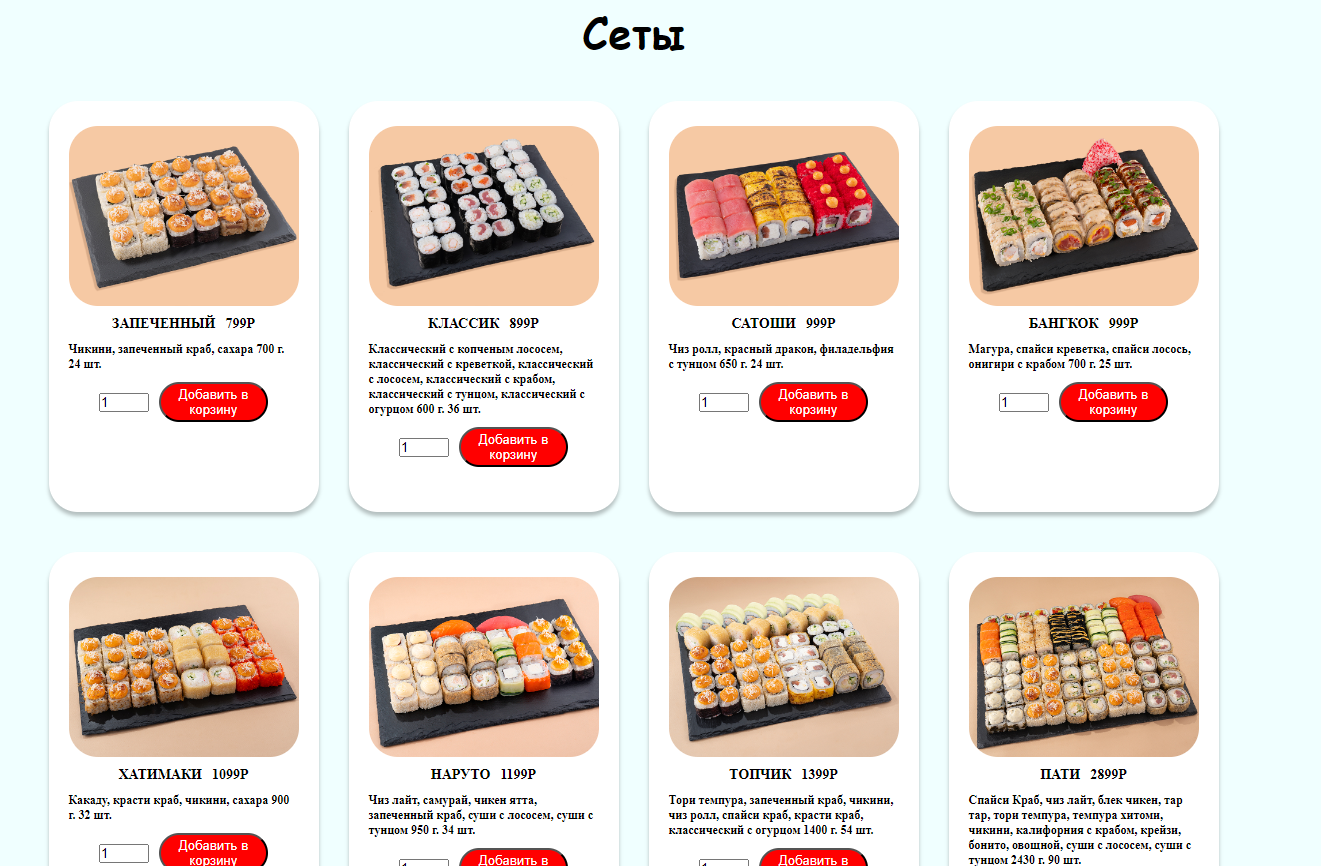


Рисунок 7 – Окно просмотра продукции

На рисунке 8 – представлено окно редактирование пользователя.

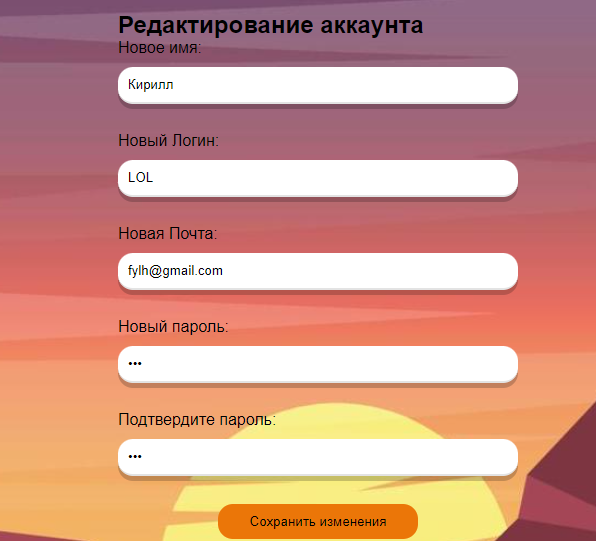


Рисунок 8 – Окно показа выбранного товара

На рисунке 9 – представлено окно удаления пользователя.

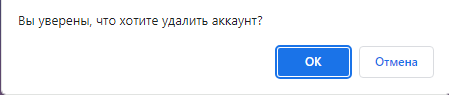


Рисунок 9 – Окно удаления товара

# 2. ПРОЕКТНАЯ ЧАСТЬ

## 2.1 Техническое задание

Разрабатываемый веб-приложения «Доставка готовой продукции» представляет собой платформу для просмотра товара. Проект будет реализован с использованием технологий PHP, HTML, CSS и JavaScript.

Система управления пользователями включает функционал регистрации и управления профилями, обеспечивая пользователям возможность создавать учетные записи и входить в свои профили.

Основные цели системы включают:

* Разработку удобного интерфейса для веб-приложения.
* Реализацию функциональности регистрации и авторизации пользователей.
* Обеспечение системы автоматического обновления обучающего контента.

Задачи включают:

* Достижение поставленных целей.
* Автоматическое обновление данных.
* Освоение навыков, полученных на учении.
* Создание отчета о выполненной работе и результатах обучения.

Метод реализации процесса проектирования включает инновационные подходы, направленные на обеспечение эффективной время провождения на сайте.

Цель внедрения системы:

Объекты автоматизации включают:

* Максимальную интерактивность и учебную эффективность системы.
* Технологичную и надежную платформу для просмотра товара.
* Минимизацию человеческого вмешательства в процессы обновления и поддержки.
* Регулярное обновление и совершенствование материалов веб-приложения.

Результатом проекта ожидается создание инновационного веб-приложения для доставки готовой продукции. Дополнительно предусмотрена возможность управления пользовательской базой данных для эффективного хранения и обработки информации.

## 2.2 Установка и настройка среды разработки

Для обеспечения эффективного функционирования веб-приложения «Доставка готовой продукции» планируется организация хранения данных внутри проекта, за исключением медиа-файлов, таких как фотографии, которые будут сохранены в файловой системе. В базе данных будут храниться информация о товарах и данные пользователей.

Принципы информационного обеспечения системы включают использование трехзвенной клиент-серверной архитектуры, где серверная часть управляет базой данных, а клиентская часть обеспечивает взаимодействие с пользователями. Для управления данными выбрана реляционная система управления базами данных MySQL, обеспечивающая структурированное хранение результатов тестов и данных пользователей.

Система управления базами данных (СУБД) MySQL выполняет функции управления данными на дисках и в оперативной памяти с использованием дискового кэша, а также поддерживает языки БД для определения данных и манипулирования данными.

Для создания статических страниц и шаблонов веб-приложения будут использованы языки HTML 5.0 и CSS 3.0. HTML 5.0 включает в себя современные элементы и атрибуты для гибкой и современной разработки, в то время как CSS 3.0 предоставляет расширенные возможности для стилизации и анимации элементов.

Интерактивные элементы серверной части веб-приложения будут реализованы с использованием языков PHP и JavaScript. PHP, язык программирования общего назначения, будет применяться для обработки данных на сервере и взаимодействия с базой данных MySQL. JavaScript, скриптовый язык программирования, обеспечит интерактивность веб-приложения, взаимодействие с пользователем и динамическое обновление содержимого страниц.

Для обеспечения полноценного функционирования веб-приложения «Доставка готовой продукции» необходимо наличие следующего программного обеспечения:

– Операционная система: Windows 10/Linux/macOS;

– Сервер: MySQL Server 8.0;

– XAMPP.

MySQL Server и MySQL являются надежными и широко используемыми системами управления базами данных (СУБД), предоставляющими эффективное хранение данных. MySQL, выпущенная в 1995 году, является открытым программным обеспечением и поддерживает разнообразные проекты, включая веб-приложения.

Необходимое программное обеспечение для персонального компьютера:

Операционная система: Windows 10

Сайт «Доставка готовой продукции» будет поддерживать полнофункциональный просмотр через следующие браузеры:

* Google Chrome
* Yandex

JavaScript предназначен для создания сценариев и обеспечения интерактивности на HTML-страницах. Важно отметить, что JavaScript не имеет никакого отношения к языку программирования Java. JavaScript был разработан компанией Netscape Communication Corporation, а его первоначальное название было LiveScript. В последствии, в целях коммерческой стратегии, его переименовали в JavaScript.

Для функционирования сайта необходимо следующее техническое обеспечение со следующими минимальными характеристиками:

– процессор – intel core i3;

– оперативная память – 512 Mb RAM;

## 2.3 Руководство пользователя

Авторизация и Регистрация:

Для доступа ко всем товарам необходимо авторизоваться. Нажмите на кнопку «Авторизация».

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

После успешной авторизации или регистрации вы попадете на страницу товаров.

Структура товаров:

Товары представлены в виде карточек, внутри которых находятся вся информация.

Редактирование профиля:

Наведясь на аватарку пользователя всплывает меню с кнопкой “редактировать” чтобы изменить информацию в вашем профиле.

Также доступна опция «Удалить аккаунт» для удаления профиля.

Выход из Аккаунта:

В личном кабинете вы можете выбрать «Выйти» для безопасного завершения сеанса.

## 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 появится новый пункт в верхнем правом углу – «Админ\_Панель».

Нажмите на этот пункт, чтобы перейти в административную панель.

Обзор товаров:

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

Каждая строка содержит кнопки «Редактировать» и «Удалить» для управления данными товара.

Функции Администрирования:

Редактировать: Нажмите на кнопку «Редактировать» рядом с товаром для внесения в него изменений.

Удалить: Кнопка «Удалить» удаляет товар.

Создать: На странице админ-панели есть кнопка «Создать», которая позволяет администратору добавлять новые товары.

# ЗАКЛЮЧЕНИЕ

При разработке приложения было сделано простота использования и удобство интерфейса, а также защищенность от постороннего вмешательства.

Также были достигнуты цели курсовой работы. Удалось создать трехуровневую архитектуру приложения. Цель работы достигнута, а именно создано приложение для работы с базой данных.

Куда проще вносить данные о товаре и хранить их там.

Задачи решены, а именно создать трехуровневую архитектуру приложения, быстрый доступ к базе данных, добавление пользователя, редактировать и удалять информацию о своем аккаунте пользователя, а также просмотр товаров.

В результате выполнения курсовой, был создан динамический сайт для внесения и хранения там данных о товарах. Для его разработки применялось актуальное программное обеспечение, без помощи которого не было бы возможности добиться поставленной цели. Это программное обеспечение включает в себя:

* язык программирования PHP;
* веб-сервер XAMPP;
* реляционная система управления БД MySQL;
* приложение PHP MyAdmin.

С помощью этих инструментов значительно сокращена ручная работа и повторяющийся код. Чтобы добиться ЧПУ без применения bootstrap и дополнительных библиотек, необходимо вносить много изменений в файлы.

Таким образом, затрачено меньше времени на разработку сайта. Всё информационное наполнение сайта находится в БД.

# СПИСОК ИСПОЛЬЗУЕЫХ ИСТОЧНИКОВ

1. Ванда, Джон К. CMS Drupal. Руководство по разработке системы управления сайтом / Ванда, Джон К., Мэтт Вестгот,. - 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. Даёт, Джон HTML и CSS. Разработка и дизайн веб-сайтов / Джон Дает. - 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.
11. Кузнецов PHP. Практика создания Web-сайтов / Кузнецов, М.В. и. - М.: БХВ-Петербург, 2020. - 895 c.
12. Кузнецов, М. PHP 5. Практика создания Web-сайтов / М. Кузнецов, И. Седанов, С. Голышев. - М.: БХВ-Петербург, 2020. - 960 c.
13. Леонтьев, Борис PHP 5.0 для начинающих, или как создать динамический WEB-сайт / Борис Леонтьев. - М.: Новый издательский дом, 2019. - 176 c.
14. Пташкин, Алексей HTML в Web-дизайне / Алексей Пташкин. - Москва: Машиностроение, 20. - 400 c.

Краль, А. PHP - это просто. Программируем для Web-сайта / А. Краль. - М.: БХВ-Петербург, 2021. - 368 c.

ПРИЛОЖЕНИЕ

Приложение A: Main.php

<?php

session\_start();

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

    header('Location: /');

}

$db = new PDO("mysql:host=localhost;dbname=kyrs", "root", "");

$info = [];

// Получение данных о сетах

if ($query = $db->query("SELECT \* FROM `sets`")) {

    $info['sets'] = $query->fetchAll(PDO::FETCH\_ASSOC);

} else {

    print\_r($db->errorInfo());

}

// Получение данных о пиццах

if ($query = $db->query("SELECT \* FROM `pizza`")) {

    $info['pizza'] = $query->fetchAll(PDO::FETCH\_ASSOC);

} else {

    print\_r($db->errorInfo());

}

// Получение данных о салатах

if ($query = $db->query("SELECT \* FROM `salads`")) {

    $info['salads'] = $query->fetchAll(PDO::FETCH\_ASSOC);

} else {

    print\_r($db->errorInfo());

}

// Обработка добавления в корзину

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

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

    $productName = $\_POST['product\_name'];

    $productPrice = $\_POST['product\_price'];

    $quantity = $\_POST['quantity'];

    $cartItem = [

        'id' => $productId,

        'name' => $productName,

        'price' => $productPrice,

        'quantity' => $quantity,

    ];

    // Создаем корзину в сессии, если её еще нет

    if (!isset($\_SESSION['cart'])) {

        $\_SESSION['cart'] = [];

    }

    // Добавляем товар в корзину

    $\_SESSION['cart'][] = $cartItem;

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

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

    <title>Главная страница</title>

</head>

<body>

<ul class="topnav">

    <li><a class="btn-13" href="/Курсовая/Main.php">Меню</a></li>

    <li><a class="btn-13" href="/Курсовая/about Us.php">О нас</a></li>

    <li><a class="btn-13" href="/Курсовая/contacts.php">Контакты</a></li>

    <li><a class="btn-13" href="/Курсовая/cart.php">Корзина</a></li>

    <h2 class="user\_name"><?= $\_SESSION['user']['name'] ?></h2>

    <a href="#" class="email"><?= $\_SESSION['user']['email'] ?></a>

    <a href="vendor/logout.php" class="logout">Выход</a>

    <div class="avatar-container">

    <img class="avatar" src="<?= $\_SESSION['user']['avatar'] ?>" onmouseover="showMenu()" onmouseout="hideMenu()">

    <div class="menu" id="menu" onmouseover="showMenu()" onmouseout="hideMenu()">

    <a href="/Курсовая/account\_editing.php">Редактировать аккаунт</a>

  </div>

  </div>

  <script>

function showMenu() {

  document.getElementById("menu").style.display = "block";

}

function hideMenu() {

  document.getElementById("menu").style.display = "none";

}

  </script>

</ul>

<hr>

<!-- Слайдер -->

<main>

    <div class="slider-container">

        <div class="slider">

            <img src="12.jpg" alt="Сет суши">

            <img src="13.jpg" alt="Пицца">

            <img src="14.jpg" alt="Салат">

        </div>

        <button class="prev-button" aria-label="Посмотреть предыдущий слайд">&lt;</button>

        <button class="next-button" aria-label="Посмотреть следующий слайд">&gt</button>

    </div>

</main>

<script>

    const slider = document.querySelector('.slider');

    const prevButton = document.querySelector('.prev-button');

    const nextButton = document.querySelector('.next-button');

    const slides = Array.from(slider.querySelectorAll('img'));

    const slideCount = slides.length;

    let slideIndex = 0;

    prevButton.addEventListener('click', showPreviousSlide);

    nextButton.addEventListener('click', showNextSlide);

    function showPreviousSlide() {

        slideIndex = (slideIndex - 1 + slideCount) % slideCount;

        updateSlider();

    }

    function showNextSlide() {

        slideIndex = (slideIndex + 1) % slideCount;

        updateSlider();

    }

    function updateSlider() {

        slides.forEach((slide, index) => {

            if (index === slideIndex) {

                slide.style.display = 'block';

            } else {

                slide.style.display = 'none';

            }

        });

    }

    updateSlider();

</script>

<!-- Слайдер -->

<!-- ... Ваш HTML код ... -->

<h3 class="sets">Сеты</h3>

<div class="container">

    <?php foreach ($info['sets'] as $data): ?>

        <div class="cards">

            <form method="post" action="">

                <input type="hidden" name="product\_id" value="<?= $data['id'] ?>">

                <input type="hidden" name="product\_name" value="<?= $data['sushi\_name'] ?>">

                <input type="hidden" name="product\_price" value="<?= $data['price'] ?>">

                <ul class="product-list">

                    <li class="product-item">

                         <img class="imgs" src="<?= $data['img']; ?>">

                    </li>

                    <li class="product-item">

                         <h3 class="sushi\_name"><?= $data['sushi\_name']; ?></h3>

                <h3 class="price"><?= $data['price']; ?></h3>

                    </li>

                    <li class="product-item">

                        <h3 class="description"><?= $data['description']; ?></h3>

                    </li>

                    <li class="product-item">

                         <input class="input" type="number" name="quantity" value="1" min="1">

                <button class="butt" type="submit" name="add\_to\_cart">Добавить в корзину</button>

                    </li>

                </ul>

            </form>

        </div>

    <?php endforeach; ?>

     <?php

    if ($query = $db->query("SELECT \* FROM `pizza`")) {

        $pizzaInfo = $query->fetchAll(PDO::FETCH\_ASSOC);

    } else {

        print\_r($db->errorInfo());

    }

    ?>

    <h3 class="sets">Пиццы</h3>

    <div class="container">

        <?php foreach ($pizzaInfo as $data): ?>

            <div class="cards">

                <form method="post" action="">

                    <input type="hidden" name="product\_id" value="<?= $data['id'] ?>">

                    <input type="hidden" name="product\_name" value="<?= $data['pizza\_name'] ?>">

                    <input type="hidden" name="product\_price" value="<?= $data['price'] ?>">

                <ul class="product-list">

                    <li class="product-item">

                      <img class="imgs" src="<?= $data['img']; ?>">

                    </li>

                    <li class="product-item">

                    <h3 class="sushi\_name"><?= $data['pizza\_name']; ?></h3>

                    <h3 class="price"><?= $data['price']; ?></h3>

                    </li>

                    <li class="product-item">

                    <h3 class="description"><?= $data['description']; ?></h3>

                    </li>

                    <li class="product-item">

                    <input class="input" type="number" name="quantity" value="1" min="1">

                    <button class="butt" type="submit" name="add\_to\_cart">Добавить в корзину</button>

                    </li>

                </ul>

                </form>

            </div>

        <?php endforeach; ?>

    </div>

    <?php

    if ($query = $db->query("SELECT \* FROM `salads`")) {

        $saladInfo = $query->fetchAll(PDO::FETCH\_ASSOC);

    } else {

        print\_r($db->errorInfo());

    }

    ?>

<h3 class="sets">Салаты</h3>

<div class="container">

    <?php foreach ($saladInfo as $data): ?>

        <div class="cards">

            <form method="post" action="">

                <input type="hidden" name="product\_id" value="<?= $data['id'] ?>">

                <input type="hidden" name="product\_name" value="<?= $data['Salads\_name'] ?>">

                <input type="hidden" name="product\_price" value="<?= $data['price'] ?>">

                <ul class="product-list">

                    <li class="product-item">

                <img class="imgs" src="<?= $data['img']; ?>">

                </li>

                <li class="product-item">

                <h3 class="sushi\_name"><?= $data['Salads\_name']; ?></h3>

                <h3 class="price"><?= $data['price']; ?></h3>

                </li>

                <li class="product-item">

                <h3 class="description"><?= $data['description']; ?></h3>

                </li>

                <li class="product-item">

                <input class="input" type="number" name="quantity" value="1" min="1">

                <button class="butt" type="submit" name="add\_to\_cart">Добавить в корзину</button>

                </li>

                </ul>

            </form>

        </div>

    <?php endforeach; ?>

</div>

<!-- Падвал -->

<div class="pod">

    <a href="tel:89189544121" class="pod1"> 8(918)954-41-21,</a>

    <a href="tel:89189544121" class="pod2"> 8(918)954-41-21</a>

</div>

<!-- Падвал -->

</body>

</html>

Приложение B: Cart.php

<?php

session\_start();

if (isset($\_POST['clear\_cart'])) {

    // Clear the cart items in the session

    $\_SESSION['cart'] = [];

    // session\_unset();

    $\_POST = [];

}

$cartItems = isset($\_SESSION['cart']) ? $\_SESSION['cart'] : [];

$totalPrice = 0;

foreach ($cartItems as $item) {

    $totalPrice += (float)$item['price'] \* (int)$item['quantity'];

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

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

    <title>Корзина</title>

</head>

<body>

<!-- ... Навигационная понел ... -->

<ul class="topnav">

    <li><a class="btn-13" href="/Курсовая/Main.php">Меню</a></li>

    <li><a class="btn-13" href="/Курсовая/about Us.php">О нас</a></li>

    <li><a class="btn-13" href="/Курсовая/contacts.php">Контакты</a></li>

    <li><a class="btn-13" href="/Курсовая/cart.php">Корзина</a></li>

</ul>

<hr>

<!-- Навигационная понел -->

<main>

    <div class="slider-container">

        <div class="slider">

            <img src="1.jpeg" alt="Сет суши">

            <img src="2.jpeg" alt="Пицца">

            <img src="3.jpeg" alt="Салат">

        </div>

        <button class="prev-button" aria-label="Посмотреть предыдущий слайд">&lt;</button>

        <button class="next-button" aria-label="Посмотреть следующий слайд">&gt</button>

    </div>

</main>

<script>

    const slider = document.querySelector('.slider');

    const prevButton = document.querySelector('.prev-button');

    const nextButton = document.querySelector('.next-button');

    const slides = Array.from(slider.querySelectorAll('img'));

    const slideCount = slides.length;

    let slideIndex = 0;

    prevButton.addEventListener('click', showPreviousSlide);

    nextButton.addEventListener('click', showNextSlide);

    function showPreviousSlide() {

        slideIndex = (slideIndex - 1 + slideCount) % slideCount;

        updateSlider();

    }

    function showNextSlide() {

        slideIndex = (slideIndex + 1) % slideCount;

        updateSlider();

    }

    function updateSlider() {

        slides.forEach((slide, index) => {

            if (index === slideIndex) {

                slide.style.display = 'block';

            } else {

                slide.style.display = 'none';

            }

        });

    }

    updateSlider();

</script>

<!-- Слайдер -->

<div class="container">

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

    <p>Ваша корзина пуста.</p>

<?php else: ?>

    <table>

        <tr class="cary">

            <th>Название</th>

            <th>Цена за шт.</th>

            <th>Количество</th>

            <th>Итого</th>

        </tr>

        <?php foreach ($cartItems as $item): ?>

            <tr>

                <td><?= $item['name'] ?></td>

                <td><?= $item['price'] ?></td>

                <td><?= $item['quantity'] ?></td>

                <td><?= (float)$item['price'] \* (int)$item['quantity'] ?>₽</td>

            </tr>

        <?php endforeach; ?>

        <tr>

            <td colspan="3">Общая стоимость:</td>

            <td><?= $totalPrice ?>₽</td>

        </tr>

    </table>

    <form method="post" action="">

        <button class="butto" type="submit" name="clear\_cart">Очистить корзину</button>

    </form>

<?php endif; ?>

</div>

</body>

</html>

Приложение C: about Us.php

<?php

$db = new PDO ("mysql:host=localhost;dbname=kyrs", "root", "");

$info = [];

if($query = $db->query ("SELECT \* FROM `sets`")) {

  $info = $query->fetchAll( PDO::FETCH\_ASSOC);

} else {

    print\_r($db->errorInfo());

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

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

    <title>Document</title>

</head>

<body>

<!-- Навигационная понель -->

<ul class="topnav">

  <li><a class="btn-13" href="/Курсовая/Main.php">Меню</a></li>

  <li><a class="btn-13" href="/Курсовая/about Us.php">О нас</a></li>

  <li><a class="btn-13" href="/Курсовая/contacts.php">Контакты</a></li>

  <li><a class="btn-13" href="/Курсовая/cart.php">Корзина</a></li>

</ul>

<hr>

<!-- Навигационная понел -->

<!-- Слайдер -->

<main>

    <div class="slider-container">

    <div class="slider">

      <img src="1.jpeg" alt="Сет суши">

      <img src="2.jpeg" alt="Пицца">

      <img src="3.jpeg" alt="Салат">

    </div>

    <button class="prev-button" aria-label="Посмотреть предыдущий слайд">&lt;</button>

    <button class="next-button" aria-label="Посмотреть следующий слайд">&gt</button>

  </div>

  </main>

  <script>

    const slider = document.querySelector('.slider');

const prevButton = document.querySelector('.prev-button');

const nextButton = document.querySelector('.next-button');

const slides = Array.from(slider.querySelectorAll('img'));

const slideCount = slides.length;

let slideIndex = 0;

prevButton.addEventListener('click', showPreviousSlide);

nextButton.addEventListener('click', showNextSlide);

function showPreviousSlide() {

  slideIndex = (slideIndex - 1 + slideCount) % slideCount;

  updateSlider();

}

function showNextSlide() {

  slideIndex = (slideIndex + 1) % slideCount;

  updateSlider();

}

function updateSlider() {

  slides.forEach((slide, index) => {

    if (index === slideIndex) {

      slide.style.display = 'block';

    } else {

      slide.style.display = 'none';

    }

  });

}

updateSlider();

  </script>

<!-- Слайдер -->

<!-- Коталог-сеты -->

<h3 class="about1">О НАС</h3>

<div class="about">

<h3 class="about2">RED DRAGON - сеть, работающая в формате take away & delivery</h3>

<h3 class="about3">RED DRAGON предлагает Вам насладиться всеми ключевыми сочетаниями

   вкусов японской, паназиатской, итальянской и европейской кухонь.<br> Мы используем только

    самые лучшие, свежие ингриденты, ведь нашим главным приоритетом является высокое качество

     наших блюд.</h3>

</div>

<!-- Коталог-сеты -->

<!-- Падвал -->

<div class="pod">

    <a href="tel:89189544121" class="pod1"> 8(918)954-41-21,</a>

    <a href="tel:89189544121" class="pod2"> 8(918)954-41-21</a>

</div>

<!-- Падвал -->

</body>

</html>

Приложение D: contacts.php

<?php

$db = new PDO ("mysql:host=localhost;dbname=kyrs", "root", "");

$info = [];

if($query = $db->query ("SELECT \* FROM `sets`")) {

  $info = $query->fetchAll( PDO::FETCH\_ASSOC);

} else {

    print\_r($db->errorInfo());

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

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

    <title>Document</title>

</head>

<body>

<!-- Навигационная понель -->

<ul class="topnav">

  <li><a class="btn-13" href="/Курсовая/Main.php">Меню</a></li>

  <li><a class="btn-13" href="/Курсовая/about Us.php">О нас</a></li>

  <li><a class="btn-13" href="/Курсовая/contacts.php">Контакты</a></li>

  <li><a class="btn-13" href="/Курсовая/cart.php">Корзина</a></li>

</ul>

<hr>

<!-- Навигационная понел -->

<!-- Слайдер -->

<main>

    <div class="slider-container">

    <div class="slider">

      <img src="1.jpeg" alt="Сет суши">

      <img src="2.jpeg" alt="Пицца">

      <img src="3.jpeg" alt="Салат">

    </div>

    <button class="prev-button" aria-label="Посмотреть предыдущий слайд">&lt;</button>

    <button class="next-button" aria-label="Посмотреть следующий слайд">&gt</button>

  </div>

  </main>

  <script>

    const slider = document.querySelector('.slider');

const prevButton = document.querySelector('.prev-button');

const nextButton = document.querySelector('.next-button');

const slides = Array.from(slider.querySelectorAll('img'));

const slideCount = slides.length;

let slideIndex = 0;

prevButton.addEventListener('click', showPreviousSlide);

nextButton.addEventListener('click', showNextSlide);

function showPreviousSlide() {

  slideIndex = (slideIndex - 1 + slideCount) % slideCount;

  updateSlider();

}

function showNextSlide() {

  slideIndex = (slideIndex + 1) % slideCount;

  updateSlider();

}

function updateSlider() {

  slides.forEach((slide, index) => {

    if (index === slideIndex) {

      slide.style.display = 'block';

    } else {

      slide.style.display = 'none';

    }

  });

}

updateSlider();

  </script>

<!-- Слайдер -->

<!-- Информация -->

<h3 class="cont1">Контакты</h3>

<h3 class="cont2">Телефон: 8(918)954-41-21</h3>

<h3 class="cont3">Адрес: г. Белореченск, ул.Ленина 159а</h3>

 <iframe class="cont4" src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d2832.5480740448465!2d39.861853237751106!3d44.76963067963744!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x40f0c5dba597b24f%3A0x837bdc152b3c3841!2z0YPQuy4g0JvQtdC90LjQvdCwLCAxNTnQsCwg0JHQtdC70L7RgNC10YfQtdC90YHQuiwg0JrRgNCw0YHQvdC-0LTQsNGA0YHQutC40Lkg0LrRgNCw0LksIDM1MjYzMA!5e0!3m2!1sru!2sru!4v1700155331135!5m2!1sru!2sru"

  width="600" height="450" style="border:0;" allowfullscreen="" loading="lazy"

  referrerpolicy="no-referrer-when-downgrade"></iframe>

<!-- Информация -->

<!-- Падвал -->

<div class="pod">

    <a href="tel:89189544121" class="pod1"> 8(918)954-41-21,</a>

    <a href="tel:89189544121" class="pod2"> 8(918)954-41-21</a>

</div>

<!-- Падвал -->

</body>

</html>

Приложение E: index.php

<?php

session\_start();

// if ($\_SESSION['user']) {

//      header('Location: Main.php');

// }

?>

<!doctype html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Авторизация и регистрация</title>

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

</head>

<body>

<!-- Форма авторизации -->

    <form action="vendor/signin.php" method="post">

        <label>Логин</label>

        <input type="text" name="login" placeholder="Введите свой логин"required>

        <label>Пароль</label>

        <input type="password" name="password" placeholder="Введите пароль"required>

        <button type="submit">Войти</button>

        <p>

            У вас нет аккаунта? - <a href="/Курсовая/register.php">зарегистрируйтесь</a>!

        </p>

        <?php

        if(!empty($\_SESSION['message'])){

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

                echo '<p class="msg"> ' . $\_SESSION['message'] . ' </p>';

            }

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

        }

        ?>

    </form>

</body>

</html>

Приложение F: register.php

<?php

    session\_start();

    // if ($\_SESSION['user']) {

    //     header('Location: Main.php');

    // }

?>

<!doctype html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Авторизация и регистрация</title>

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

</head>

<body>

    <!-- Форма регистрации -->

    <form action="vendor/signup.php" method="post" enctype="multipart/form-data">

        <label>Имя</label>

        <input type="text" name="name" placeholder="Введите свое полное имя"required>

        <label>Логин</label>

        <input type="text" name="login" placeholder="Введите свой логин"required>

        <label>Почта</label>

        <input type="email" name="email" placeholder="Введите адрес своей почты"required>

        <label>Изображение профиля</label>

        <input type="file" name="avatar">

        <label>Пароль</label>

        <input type="password" name="password" placeholder="Введите пароль"required>

        <label>Подтверждение пароля</label>

        <input type="password" name="password\_confirm" placeholder="Подтвердите пароль"required>

<!-- Капча кнопка -->

        <img src="captcha.php"alt="Captcha" class="captcha"><br>

        <input type="text" id="captcha" name="captcha\_code" class="input\_password" placeholder="Код с картинки"required><br>

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

        <p>

            У вас уже есть аккаунт? - <a href="/Курсовая/index.php">авторизируйтесь</a>!

        </p>

        <?php

        if(!empty($\_SESSION['message'])){

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

                echo '<p class="msg"> ' . $\_SESSION['message'] . ' </p>';

            }

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

        }

        ?>

    </form>

</body>

</html>

Приложение G: glavnoe.css

\* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

body {

  background-image: url('blob:https://web.telegram.org/93729da3-75a2-42ca-be53-db5fbc01bd29');

  background-repeat: no-repeat;

  background-size: cover;

  background-color: #f0ffff;

}

/\* Понель навигации  \*/

.topnav {

    background-color: #f0ffff;

    overflow: hidden;

    display: flex;

  flex-direction: row;

  justify-content: center;

}

ul {

    list-style-type: none;

    margin: 0;

    padding: 0;

    overflow: hidden;

  }

  li {

    display: inline;

  }

.btn-13 {

    text-align: center;

    display: inline-block;

    padding: 15px 40px;

    cursor: pointer;

    letter-spacing: 2px;

    position:relative;

    overflow:hidden;

    margin: 0 20px;

    color: rgb(0, 0, 0);

    font-size: 32px;

  text-align: center;

  padding: 14px 16px;

  text-decoration: none;

  }

  .btn-13:before {

    content: "";

    position: absolute;

    width: 0;

    background : #000000;

    left: 45%;

    height: 2px;

    bottom: 0;

    transition: all .3s;

    opacity: 0.7;

  }

  .btn-13:hover:before {

    width: 100%;

    left:0;

  }

  .avatar {

    margin-top: 5px;

    width: 60px;

    height: 60px;

    border-radius: 100px;

    border: 3px;

  }

  .user\_name {

    text-align: right;

    margin-top: 5px;

  }

  .email {

margin-top: 5px;

  }

  .logout {

    margin-top: 25px;

    margin: 10px;

  }

  .menu {

    display: none;

    position: absolute;

    background-color: #ffffff;

    min-width: 100px;

    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);

    padding: 12px;

    z-index: 1;

  }

  .menu a {

    display: block;

    color: black;

    padding: 8px 16px;

    text-decoration: none;

  }

  .menu a:hover {

    background-color: #ffffff;

  }

/\* Понель навигации  \*/

  hr {

    border: none;

    background-color: rgb(0, 0, 0);

    height: 2px;

   }

/\* Слайдер \*/

  .slider-container {

    position: relative;

    width: 600px;

    height: 400px;

    margin: 0 auto;

    overflow: hidden;

    margin-top: 20px;

  }

  .slider {

    display: flex;

    transition: transform 0.5s ease-in-out;

  }

  .slider img {

    width: 100%;

    height: 100%;

    object-fit: cover;

  }

  .prev-button,

  .next-button {

    position: absolute;

    top: 50%;

    transform: translateY(-50%);

    width: 50px;

    height: 50px;

    background-color: transparent;

    border: none;

    font-size: 24px;

    color: white;

  }

  .prev-button {

    left: 10px;

  }

  .next-button {

    right: 10px;

  }

  /\* Слайдер \*/

  /\* Коталог \*/

  .sets {

    font-size: 42px;

    font-weight: bold;

    margin-top: 30px;

    text-align: center;

    font-family: cursive;

  }

.container {

    width: 100%;

    max-width: 1200px;

    margin: 0 auto;

    padding: 0 15px;

    display: flex;

    flex-direction: row;

    flex-wrap: wrap;

    justify-content: space-between;

}

.cards {

    width: 270px;

    height: auto;

    border-radius: 29px;

    box-shadow: 0px 4px 4px 0px rgba(0, 0, 0, 0.25);

    background-color: rgb(255, 255, 255);

    margin-top: 40px;

    padding-bottom: 10px;

    padding-left: 5px;

    padding-right: 5px;

  }

.imgs {

    width: 230px;

    height: 180px;

    margin-left: 0px;

    margin-top: 10px;

    border-radius: 29px;

}

.sushi\_name {

    font-size: 14px;

    font-weight: bold;

    margin-right: 10px;

}

.quantity{

  margin-right: 10px;

}

.product-item{

  align-items: center;

  margin-bottom: 10px;

}

.price {

    font-size: 14px;

    font-weight: bold;

}

.description {

    font-size: 12px;

}

.input {

width: 50px;

display: block;

float: left;

margin-bottom: 10px;

}

.butt {

  width: 109px;

  height: 40px;

  border-radius: 20px;

background-color: red;

color: #ffffff;

margin-left: 10px;

margin-bottom: 10px;

}

.product-list{

  display: flex;

  flex-direction: column;

  align-items: center;

  padding: 15px;

}

.product-item{

  display: flex;

  justify-content: space-around;

}

   /\* Коталог \*/

/\* Подвал \*/

.pod {

  margin-bottom: 20px;

  margin-top: 50px;

  margin-left: 430px;

  display: flex;

}

.pod1 {

  font-size: 17px;

  font-weight: bold;

  font-family: cursive;

  float: left;

  margin-right: 30px;

  color: #000000;

  text-decoration: none;

}

.pod2 {

  font-size: 17px;

  font-weight: bold;

  font-family: cursive;

  float: right;

  color: #000000;

  text-decoration: none;

}

/\* Подвал \*/

Приложение H: main.css

\* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

body {

    height: 100vh;

    display: flex;

    align-items: center;

    justify-content: center;

    font-family: Montserrat, sans-serif;

    background-image: url('https://static.vecteezy.com/system/resources/previews/008/890/811/non\_2x/sunset-at-beach-illustration-nature-summer-wallpaper-vector.jpg');

    background-repeat: no-repeat;

    background-size: cover;

}

a {

    color: #eb7608dc;

    font-weight: bold;

    text-decoration: none;

}

p {

    margin: 10px 0;

}

form {

    display: flex;

    flex-direction: column;

    width: 400px;

}

input {

    background-color: #ffffff;

    margin: 10px 0;

    padding: 10px;

    border: unset;

    border-bottom: 2px solid #e3e3e3;

    outline: none;

    border-radius: 15px;

    box-shadow: 0 5px 0 0 rgba(0,0,0,0.2);

}

button {

    padding: 10px;

    background: #eb7608;

    border: unset;

    cursor: pointer;

    align-items: center;

    margin-top: 0%;

    margin-left: 25%;

    border-radius: 15px;

    width: 50%;

    margin-bottom: 30px;

}

button:hover{

    box-shadow: 0 5px 0 0 rgba(0, 0, 0, 0.479);

    transition: 0.5s;

}

.msg {

    border: 2px solid #ffa908;

    border-radius: 3px;

    padding: 10px;

    text-align: center;

    font-weight: bold;

}

.logout {

    color: #b7625c;

}

.edite {

    margin-bottom: 50px;

    text-align: center;

}

.delet {

    padding: 10px;

    background: #eb7608;

    border: unset;

    cursor: pointer;

    border-radius: 15px;

    width: 50%;

    color: #000;

    margin-left: 90px;

    font-size: 15px;

}

.otmena {

    padding: 10px;

    background: #eb7608;

    border: unset;

    cursor: pointer;

    border-radius: 15px;

    width: 50%;

    color: #000;

    font-size: 15px;

}