|  |
| --- |
| **ФЕДЕРАЛЬНОЕ АГЕНТСТВО ВОЗДУШНОГО ТРАНСПОРТА**  **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ**  **БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**  **ВЫСШЕГО ОБРАЗОВАНИЯ**  **«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**  **УНИВЕРСИТЕТ ГРАЖДАНСКОЙ АВИАЦИИ» (МГТУ ГА)** |
| Кафедра ВМКСС  Курсовая работа  защищена с оценкой  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись преподавателя, дата)  КУРСОВАЯ РАБОТА  по дисциплине «Базы данных»  Вариант № 10  Тема: Создание базы данных «Доставка пиццы»  Выполнил студент группы ЭВМб-3-1:  Попов Тимофей Андреевич  (Ф.И.О)    Руководитель: Профессор, д.т.н.  Егорова Алла Альбертовна (звание, степень, Ф.И.О.)  МОСКВА – 2024 |

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ВОЗДУШНОГО ТРАНСПОРТА**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ**

**БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ГРАЖДАНСКОЙ АВИАЦИИ» (МГТУ ГА)**

**Кафедра ВМКСС**

ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ (РАБОТУ)

По дисциплине ***Базы данных***

Студента: **Попова Тимофея Андреевича**

(фамилия, имя, отчество)

Группы: **ЭВМб-3-1**

1. Наименование темы ***Создание базы данных «Доставка пиццы»***
2. Задание и основные характеристики: Создать базу данных, содержащую сведения о заданном объекте («Доставка пиццы») и разработать интерактивную систему для обработки данных базы.
3. Объем: пояснительная записка на \_\_\_\_\_\_\_\_\_\_\_\_\_\_ листах, графическая часть на \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ листах формата.

Руководитель: ***д.т.н.Егорова А.А.***

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

Студент: ***Попов Т.А.***

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

СОДЕРЖАНИЕ

**ЗАДАНИЕ...…………………...……………………………………………………...…………………...4**

**ВВЕДЕНИЕ…………………...……………………………………………………...…….…………...4.1**

[1. ЗАДАНИЕ НА ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ 5](#_Toc168320983)

[2. КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ ДАННЫХ 6](#_Toc168320984)

[3. ЛОГИЧЕСКАЯ МОДЕЛЬ ДАННЫХ 8](#_Toc168320985)

[3.1 Логическая структура и отношения реляционной базы данных 8](#_Toc168320986)

[3.3 Запросы 14](#_Toc168320987)

[4. РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ 17](#_Toc168320988)

[4.1 Реализация и наполнение базы данных 17](#_Toc168320989)

[4.2 Триггеры 21](#_Toc168320990)

[4.3 Пользователи базы данных 23](#_Toc168320991)

[5. РЕАЛИЗАЦИЯ ВЕБ-ИНТЕРФЕЙСА БАЗЫ ДАННЫХ 24](#_Toc168320992)

[5.1 UML-диаграмма деятельности 24](#_Toc168320993)

[5.2 Интеллект-карта системы 25](#_Toc168320994)

[6. ЗАКЛЮЧЕНИЕ 26](#_Toc168320992)

[7. ЛИТЕРАТУРА 27](#_Toc168320992)

[ПРИЛОЖЕНИЕ A. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 28](#_Toc168320996)

[Интерфейс администратора 28](#_Toc168320997)

[Интерфейс пользователя 34](#_Toc168320998)

[ПРИЛОЖЕНИЕ В. ИСХОДНЫЙ КОД 38](#_Toc168320999)

**ЗАДАНИЕ**

Создать базу данных, содержащую сведения о заданном объекте (в соответствии с вариантом) и разработать интерактивную систему для обработки данных базы. Система должна обеспечивать:

* ведение БД: просмотр, корректировку, удаление, дополнение;
* контроль ограничения целостности (ограничение значений атрибутов и структурные ограничения);
* использование нескольких пользователей в СУБД (минимум – администратор и пользователь) и соответственно программный интерфейс с разными уровнями доступа (например, администратор имеет право на все - создание, удаление, изменение, извлечение, а пользователь - только на извлечение);
* формирование выходных форм в соответствии с заданиями,
* извлечение информации по запросам пользователей в соответствии с бизнес-правилами и функционалом системы.
* API-доступ к стороннему ресурсу (курс доллара/евро, прогноз погоды, достопримечательности города и т.п.).

В качестве программного обеспечения можно использовать локальный сервер Denver. При желании студента допустимо выбирать и другие аналогичные технологии, заменяющие отдельные функции, реализованные через компоненты Denver вплоть до полного его исключения. (Например, использование технологии ASP.Net с моделью MVC на базе SQL–Server). Обоснованное использование других программных средств, языков программирования, протоколов, форматов и пр., используемых при проектировании и сопровождении баз данных, для представления данных, аутентификации, обработки и т.п. не является в работе недостатком и поощряется.

**ВВЕДЕНИЕ**

В современном информационном обществе создание эффективных и функциональных веб-сайтов с базами данных становится все более актуальным и востребованным заданием. В рамках данной курсовой работы был поставлен проект - разработка веб-сайта для доставки пиццы с использованием базы данных, обеспечивающей функциональность и удобство как для администратора, так и для пользователей.

Основной целью данного проекта является создание информационной системы, позволяющей администратору пиццерии управлять контентом веб-сайта. Пользователям, в свою очередь, предоставляется возможность просмотра меню, реализации заказа просмотра погоды. Также представлены выходные документы для курьера в виде записки с заказом клиента.

Для достижения поставленной цели, в курсовой работе используются современные программные средства и технологии, такие как: Beget (хостинг-провайдер), языки html и php, CSS-шаблоны, фреймворк «Bootstrap», СУБД phpMyAdmin 4.9.7.

Результатом курсовой работы будет полнофункциональный веб-сайт доставки пиццы с базой данных, предоставляющий удобный и интуитивно понятный интерфейс как для администратора, так и для пользователей. Созданный сайт будет способствовать улучшению управления доставкой пиццы, повышению доступности информации для пользователей и улучшению пользовательского опыта взаимодействия с доставкой.

**1. ЗАДАНИЕ НА ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ**

В результате анализа предметной области по открытым источникам в сети Интернет были выполнены следующие действия:

* уточнено задание на выполнение курсового проекта;
* разработаны бизнес-правила курсового проекта.

Предметрная область

-Доставка пиццы

База данных должна содержать сведения о следующих объектах:

Курьеры – фио, табельный номер, паспортные данные.

Пицца- название,вес,цена,состав,размер (2 размера)(5 видов пиццы )  
Клиенты- (лояльная карта создается и в ней же указывается ) ФИО,номер телефона ,email,индивидуальная скидка , купленные ранее товары (в виде чека)  
Выходные документы

-Заказ

-Путевой лист

-Бизнес-правила

- Карта лояльности создается на каждого клиента  
-За каждым автомобилем закреплено не больше 2 курьеров  
-Номер телефона является и номером карты лояльности  
- У каждой карты сохраняются все чеки своего покупателя  
- Одна карта лояльности на один телефон  
- Если мед книжке больше года , то нужно переделывать

**2. КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ ДАННЫХ**

Для разработки базы данных «Доставка пиццы» необходима реализация следующих сущностей:

1. Машины. Информация о машинах курьеров.
2. Доставка. Информация о доставке.
3. Работник. Информация о работниках.
4. Клиент. Информация о клиентах.
5. Медицинская\_информация. Информация для работников на кухне
6. Заказ. Информация о заказе.
7. Чек. Информация об оплате заказа
8. Пицца. Информация о пиццах в ассортименте.
9. Пицца\_чек. Информация о пицце в заказе.
10. Пост. Пост работника.
11. Программа\_лояльности. Информация о программе лояльности клиентов.
12. Размер\_пиццы. Информация о размерах пицц.
13. Статус. Информация о статусе заказа.
14. Способ оплаты. Информация о способе оплаты.

ER-диаграмма проектируемой БД представлена далее на рис. 1.

Изображение выглядит как текст, диаграмма, линия, снимок экрана

Автоматически созданное описание

Рис. 1 – ER-диаграмма проектируемой БД

# 3. ЛОГИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

## 3.1 Логическая структура и отношения реляционной базы данных

На этапе разработки логической модели данных была составлена логическая структура базы данных, для атрибутов указаны соответствующие типы данных, выполнена проверка на соответствие логической структуры третьей нормальной форме (рис. 2).

Проведём анализ каждой таблицы и проверим их соответствие третьей нормальной форме.

1. Таблица «Машина»

* Табельный номер, бренд, модель, цвет, регистрационный номер.
* Здесь каждый атрибут явно зависит от первичного ключа (ID\_машина). В данном случае нет зависимостей между не ключевыми атрибутами, поэтому таблица "Машина" соответствует третьей нормальной форме.

1. Таблица “Доставка”.

* Заказ\_ID, дата, адрес, сумма\_доставки, работник\_ID.
* Каждый атрибут зависит от первичного ключа (ID\_доставка), следовательно, она соответствует третьей нормальной форме.

1. Таблица “Работник”

* ФИО, номер\_телефона, email, пост\_ID, медицинская\_информация, машина\_ID.
* Каждый атрибут зависит от первичного ключа (ID\_работника), следовательно, она соответствует третьей нормальной форме.

1. Таблица «Клиент»

* Фамилия, имя, адрес, email, номер\_телефона, программа\_лояльности\_ID.
* Каждый атрибут зависит от первичного ключа (ID\_клиент), следовательно, она соответствует третьей нормальной форме.

1. Таблица «Медицинская информация»

* Да\_или\_нет, дата.
* Каждый атрибут зависит от первичного ключа (ID\_медицинская\_информация), следовательно, она соответствует третьей нормальной форме.

1. Таблица «Заказ»

* Фамилия, имя, номер\_телефона, адрес, дата, сумма\_заказа, статус\_ID, работник\_ID.
* Каждый атрибут зависит от первичного ключа (ID\_заказ), следовательно, она соответствует третьей нормальной форме.

1. Таблица «Чек»

* Заказ\_ID, сумма\_к\_оплате, дата, способ\_оплаты\_ID.
* Каждый атрибут зависит от первичного ключа (ID\_чек), следовательно, она соответствует третьей нормальной форме.

1. Таблица «Пицца»

* Название\_пиццы ,цена пиццы , размер\_ID , вес.
* Каждый атрибут зависит от первичного ключа (ID\_пиццы), следовательно, она соответствует третьей нормальной форме.

1. Таблица «Пицца\_чек»

* Заказ\_iD , пиццы\_ID , количество.
* В данной таблице имеется только один атрибут, который зависит от первичного ключа (ID\_чек), поэтому она также соответствует третьей нормальной форме.

1. Таблица «Пост»

* Пост.
* В данной таблице имеется только один атрибут, который зависит от первичного ключа (ID\_пост), поэтому она также соответствует третьей нормальной форме.

1. Таблица «Программа\_лояльности»

* Номер\_телефона , скидка.
* В данной таблице имеется только один атрибут, который зависит от первичного ключа (ID\_програма\_лояльности), поэтому она также соответствует третьей нормальной форме.

1. Таблица «Размер\_пиццы»

* Размер
* В данной таблице имеется только один атрибут, который зависит от первичного ключа (ID\_размер), поэтому она также соответствует третьей нормальной форме.

1. Таблица «Статус»

* Статус
* В данной таблице имеется только один атрибут, который зависит от первичного ключа (ID\_статус), поэтому она также соответствует третьей нормальной форме.

1. Таблица «Способ\_оплаты»

* Тип\_оплаты
* В данной таблице имеется только один атрибут, который зависит от первичного ключа (ID\_способ\_оплаты), поэтому она также соответствует третьей нормальной форме.

Таким образом, на основе предоставленных данных, все таблицы, которые мы перечислили, соответствуют третьей нормальной форме.

Отношения, представленные в логической структуре, описаны в таб. 1-15.

Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт

Автоматически созданное описание

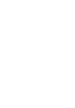


Рис. 2 – Логическая структура базы данных «Сеть кинотеатров».

*Таблица 1. Отношение «car»*

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип и ограничение** | **Комментарий** |
| ID\_car | int(11) | Ключевой атрибут |
| tabel\_number | int(11) |  |
| Brand | varchar(30) |  |
| Model | varchar(30) |  |
| Color | varchar(30) |  |
| Register\_number | varchar(30) |  |

*Таблица 2. Отношение «delivery»*

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип и ограничение** | **Комментарий** |
| ID\_Delivery | int(11) | Ключевой атрибут |
| Order\_ID | int(11) |  |
| Date\_and\_time | Datetime |  |
| Address | varchar(30) |  |
| Sum\_delivery | int(11) |  |
| Employees\_ID | int(11) |  |

*Таблица 3. Отношение «employees»*

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип и ограничение** | **Комментарий** |
| ID\_Employees | int(11) | Ключевой атрибут |
| name\_employees | varchar(30) |  |
| number\_Employees | varchar(30) |  |
| email\_Employees | varchar(30) |  |
| Post\_ID | int(11) |  |
| Medical\_ID | int(11) |  |
| Car\_ID | int(11) |  |

*Таблица 4. Отношение «klient»*

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип и ограничение** | **Комментарий** |
| ID\_klient | int(11) | Ключевой атрибут |
| Surname | varchar(30) |  |
| Name | varchar(30) |  |
| Address | varchar(30) |  |
| email | varchar(30) |  |
| Number | varchar(30) |  |
| program\_loyalty\_ID | varchar(30) |  |

*Таблица 5. Отношение «medical»*

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип и ограничение** | **Комментарий** |
| ID\_Medical | int(11) | Ключевой атрибут |
| Yes\_or\_no | text |  |
| Date\_and\_time | date |  |

*Таблица 6. Отношение «order\_online»*

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип и ограничение** | **Комментарий** |
| ID\_order\_online | int(11) | Ключевой атрибут |
| Surname | varchar(30) |  |
| Name | varchar(30) |  |
| Number | varchar(30) |  |
| Address | varchar(30) |  |
| Date\_and\_time | Datetime |  |
| Sum\_order | int(11) |  |
| status-ID | int(11) |  |
| employees\_id | int(11) |  |

*Таблица 7. Отношение «payment»*

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип и ограничение** | **Комментарий** |
| ID\_Payment | int(11) | Ключевой атрибут |
| Order\_ID | int(11) |  |
| Sum\_pay | int(11) |  |
| Date\_and\_time | Datetime |  |
| Way\_ID | int(11) |  |

*Таблица 8. Отношение «pizza»*

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип и ограничение** | **Комментарий** |
| ID\_Pizza | int(11) | Ключевой атрибут |
| Name\_pizza | varchar(30) |  |
| Price\_pizza | int(11) |  |
| Size\_ID | int(11) |  |
| Weight | int(11) |  |

*Таблица 9. Отношение «pizza\_check»*

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип и ограничение** | **Комментарий** |
| ID\_check | int(11) | Ключевой атрибут |
| order\_online\_ID | int(11) |  |
| Pizza\_ID | int(11) |  |
| Quantity | int(11) |  |

*Таблица 10. Отношение «post»*

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип и ограничение** | **Комментарий** |
| ID\_post | int(11) | Ключевой атрибут |
| Post | varchar(30) |  |

*Таблица 11. Отношение «program\_loyalty»*

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип и ограничение** | **Комментарий** |
| ID\_program\_loyalty | int(11) | Ключевой атрибут |
| Number | varchar(30) |  |
| Discount | int(11) |  |

*Таблица 12. Отношение «size\_pizza»*

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип и ограничение** | **Комментарий** |
| ID\_size | int(11) | Ключевой атрибут |
| size | int(11) |  |

*Таблица 13. Отношение «status»*

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип и ограничение** | **Комментарий** |
| ID\_Status | int(11) | Ключевой атрибут |
| Status | varchar(30) |  |

*Таблица 14. Отношение «way»*

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип и ограничение** | **Комментарий** |
| ID\_way | int(11) | Ключевой атрибут |
| Type\_of\_payment | varchar(30) |  |

**3.3 Запросы**

Примеры запросов, используемых в веб-интерфейсе базы данных, представлены в таб. 16.

*Таблица 16. Примеры используемых запросов*

|  |  |
| --- | --- |
| Вывод всей таблицы «klient» | SELECT \* FROM klient |
| Вывести все пиццы и варианты размеров с их ценами | SELECT pizza.Name\_pizza AS "название пиццы", pizza.Price\_pizza AS "цена пиццы", size\_pizza.size AS "размер пиццы"  FROM pizza  JOIN size\_pizza ON size\_pizza.ID\_size = pizza.Size\_ID |
| Вывести клиентов с суммами заказов >2500 рублей | SELECT klient."Surname" AS "Фамилия", klient."Name" AS "Имя", payment."Sum\_pay" AS "Сумма заказа"  FROM klient  JOIN payment ON payment."ID\_Payment" = klient."ID\_klient"  WHERE payment."Sum\_pay" >= 2500; |
| Вывести адрес доставки и общую сумму заказа клиента с номером телефона 89672030056 | SELECT klient.Sumname AS "Фамилия клиента", klient.Name AS "Имя клиента", klient.Address AS "Адрес проживания", klient.Number AS "Номер телефона", klient.Sum\_pay AS 'Сумма оплаты'  FROM klient  JOIN payment ON payment.ID\_Payment = klient.ID\_klient  WHERE klient.Number LIKE '89672030056%' |
| Вывести адрес и сумму доставки доставщика с именем Михаил | SELECT employees.name AS "ФИО доставщика", delivery.Address AS "Адресс доставки", SUM(delivery.Sum\_delivery) AS "Цена доставки"  FROM employees  JOIN delivery ON employees.ID\_Employees = delivery.Employees\_ID  WHERE employees.name LIKE '%Михаил%'  GROUP BY employees.name, delivery.Address |
| Вывести какие есть пиццы с size=15 и их цену | SELECT pizza Name\_pizza AS 'Название пиццы', Price\_pizza AS  "Цена пиццы", Size 'Размер пиццы"  FROM pizza  JOIN size\_ pizza ON size\_ pizza ID\_size = pizza Size\_|D  WHERE Size =15 |
| Обновление данных о клиентах | UPDATE klient  SET Surname = '$pizzaSurname', Name = '$pizzaName', Address = $adress, email = $email, Number = $number, program\_loyalty\_ID= $loyaltyId  WHERE ID\_klient = $pizzaId |

Продемонстрируем работу некоторых запросов:

1. Вывести адрес и сумму доставки доставщика с именем.(рис. 3).

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 3. Вывести адрес и сумму доставки доставщика с именем.

1. Вывести заказы за 21 марта (рис. 4).

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 4. Вывести заказы за 21 марта

**4. РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ**

**4.1 Реализация и наполнение базы данных**

В ходе выполнения этапа реализации было произведено наполнение созданной схемы в соответствии с состоянием предметной области. Примеры наполненных отношений приведены на рис. (5-18).

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 5. «Отношение “car”».

Изображение выглядит как текст, Шрифт, число, снимок экрана

Автоматически созданное описание

Рисунок 6. « Отношение “delivery”».

Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание

Рисунок 7. « Отношение “employees”».

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 8. « Отношение “klient”».

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 9. « Отношение “medical”».

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 10. « Отношение “order\_online”».

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 11. « Отношение “payment”».

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 12. « Отношение “pizza”».

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 13. «Отношение “post”».

Изображение выглядит как текст, снимок экрана, число, Параллельный

Автоматически созданное описание

Рисунок 14. «Отношение “pizza\_check”».

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 15. «Отношение “program\_loyalty”».

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 16. «size\_pizza».

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 17. «Отношение “status”».

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 18. «Отношение “way”».

**4.2 Триггеры**

На данном этапе реализации базы данных для поддержания её целостности были разработаны триггеры (рис. 20-21).

1. Напишем триггер, который не будет разрешать привязывать к одной машине больше 2 курьеров

CREATE TRIGGER `car` BEFORE INSERT ON `employees`

FOR EACH ROW BEGIN

IF NEW.Car\_ID = '1' THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'За данной машиной привязано больше 2 курьеров ,выберете другой автомобиль';

END IF;

END

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 20. Результат работы 1-го триггера

2)Напишем триггер, который не даст сделать заказ не на текущую дату.

CREATE TRIGGER 'order\_data' BEFORE INSERT ON 'order'

FOR EACH ROW BEGIN

IF NEW. Date\_and\_time

1= CURDATE() THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Заказ можно сделать только на сегодняшний день';

END IF;

END

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 21. Результат работы 2-го триггера

**4.3 Пользователи базы данных**

Для использования интерактивной системы обработки данных были разработаны два уровня привилегий (два пользователя): администратор и пользователь (рис. 22). Администратор имеет все возможные в системе привилегии (SELECT, INSERT, UPDATE, DELETE) на уровне базы данных. Пользователь не имеет привилегий.

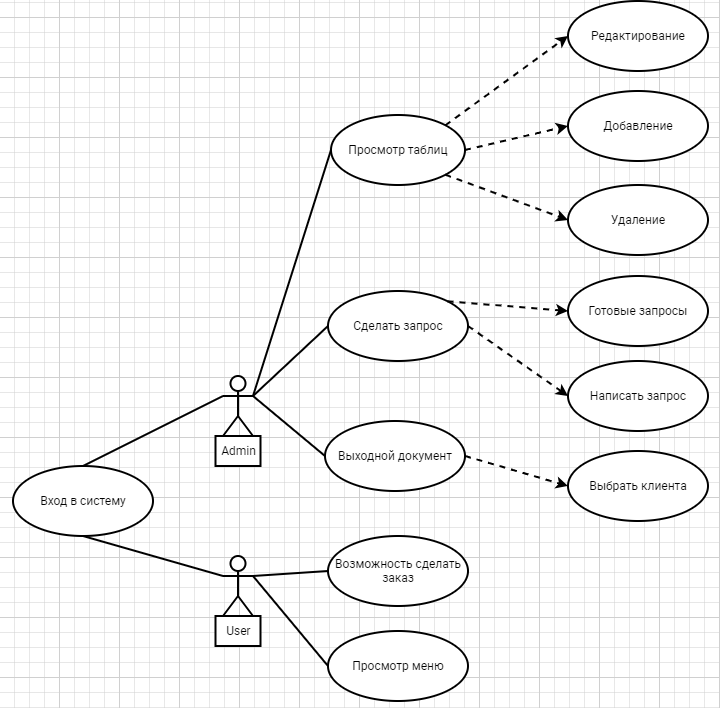


Рисунок 22. «UML-диаграмма прецедентов использования»

**5. РЕАЛИЗАЦИЯ ВЕБ-ИНТЕРФЕЙСА БАЗЫ ДАННЫХ**

**5.1 UML-диаграмма деятельности**

На данной диаграмме представлена логика авторизации пользователей сайта.

Изображение выглядит как диаграмма, текст, зарисовка, линия

Автоматически созданное описание

Рисунок 23. «Диаграмма деятельности “Авторизация”».

**5.2 Интеллект-карта системы**

**Изображение выглядит как рисунок, зарисовка, Штриховая графика, диаграмма

Автоматически созданное описание**

Рисунок 24. «Интеллект-карта веб-интерфейса БД».

**ЗАКЛЮЧЕНИЕ**

В заключении курсовой работы можно подвести итоги проведенных работ и охарактеризовать полученные результаты. В данной работе был проанализирован процесс создания веб-сайта с базой данных для доставки пиццы, на основе чего был разработан и реализован полнофункциональный продукт, отвечающий поставленным целям и требованиям.

В рамках данного проекта была создана информационная система, обеспечивающая администратору удобное управление контентом, а пользователям - удобный доступ к актуальной информации. Благодаря использованию современных технологий и программных средств, сайт предоставляет интуитивно понятный интерфейс, что способствует улучшению пользовательского опыта.

Таким образом, созданный веб-сайт с базой данных для пиццерии представляет собой важный инструмент для современного информационного общества, обеспечивая удобство, функциональность и доступность как для администраторов, так и для пользователей.

**ЛИТЕРАТУРА**

## 1. Джон Дакетт: «HTML и CSS. Разработка и дизайн веб-сайтов»

2. <https://gpt-chatbot.ru/>

3. <https://github.com/>

4. А.Н. Маслобоев **«**Базовые возможности языка HTML по разработке и оформлению Web-страниц**»**

**ПРИЛОЖЕНИЕ A. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

**Интерфейс администратора**

1.Для входа в интерактивную систему управления сайтом доставки пиццы, пожалуйста перейдите на сайт <http://localhost/denwer/index.php>. На открывшейся странице вы увидите приветственный текст и кнопку «Войти» (рис. 25), после нажатия которой появится модальное окно, в котором вы найдете форму для авторизации. Введите в соответствующие поля свои учетные данные – логин и пароль, выданные администратором. Затем нажмите кнопку «Войти», чтобы перейти к использованию всех возможностей системы.

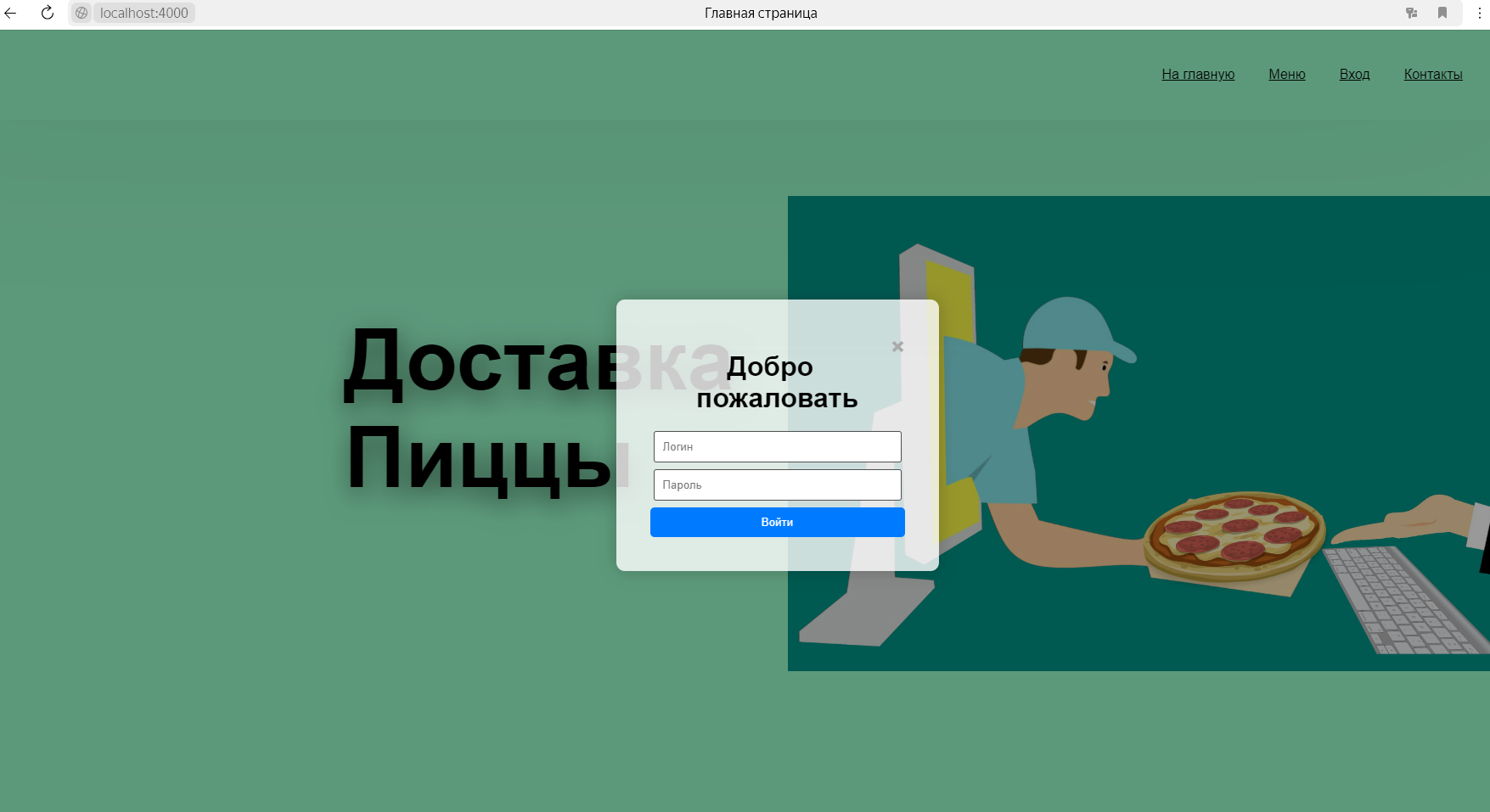


Рисунок 25. «Главная страница».

Далее у нас представлен интерфейс главной страницы администратора(рис.26).

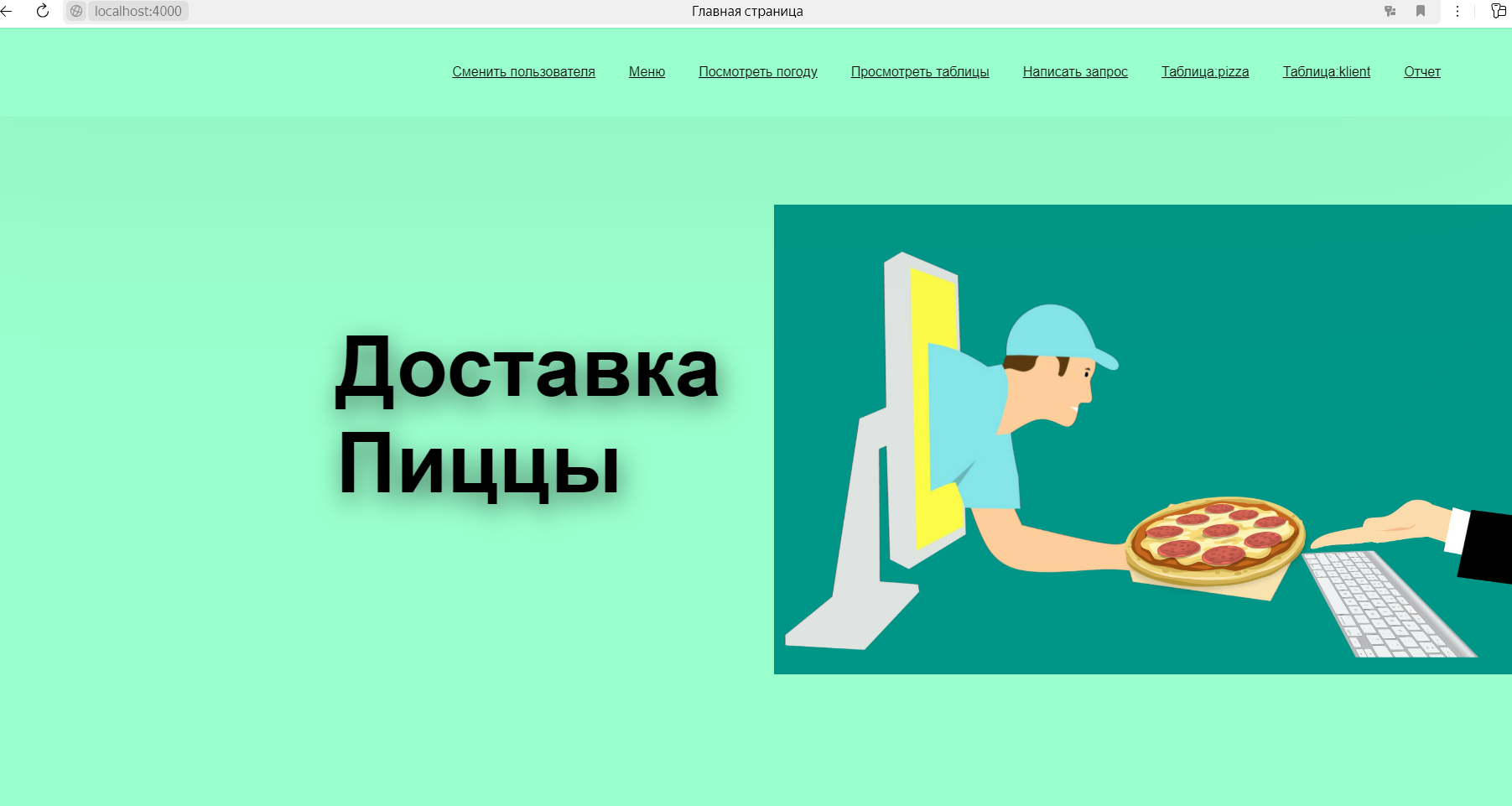


Рисунок 26. «Главная страница администратора».

2.На страничке администратора у нас представлены кнопки для работы с нашей базой данных. Перейдем на вкладку с таблицей “pizza”, в котоую мы можем добавлять, изменять и удалять данные.(рис 27).

Изображение выглядит как текст, снимок экрана, число, Параллельный

Автоматически созданное описание

Рисунок 27. «Вкладка с таблицей “pizza”»

3.При переходе на вкладку “Таблица ‘klient’ ” мы так же можем добавлять, изменять и редактировать данные.(рис. 28)

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Рисунок 28. «Вкладка с таблицей “klient ”»

4.При переходе на вкладку “Просмотр таблиц” мы можем вывести данные из любой таблицы(рис.29)

Изображение выглядит как текст, снимок экрана, число

Автоматически созданное описание

Рисунок 29. «Вкладка с просмотром таблиц»

5.При переходе на вкладку “Написать запрос” мы можем реализовать запрос к нашей базе данных или воспользоваться готовыми запросами(рис. 30-31)

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рисунок 30. «Вкладка с реализацией запросов»

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 31. «Вкладка с реализацией готовых запросов»

6.При переходе на вкладку “Отчет” мы можем получить выходной документ по каждому заказу наших клиентов.(рис.32-33)

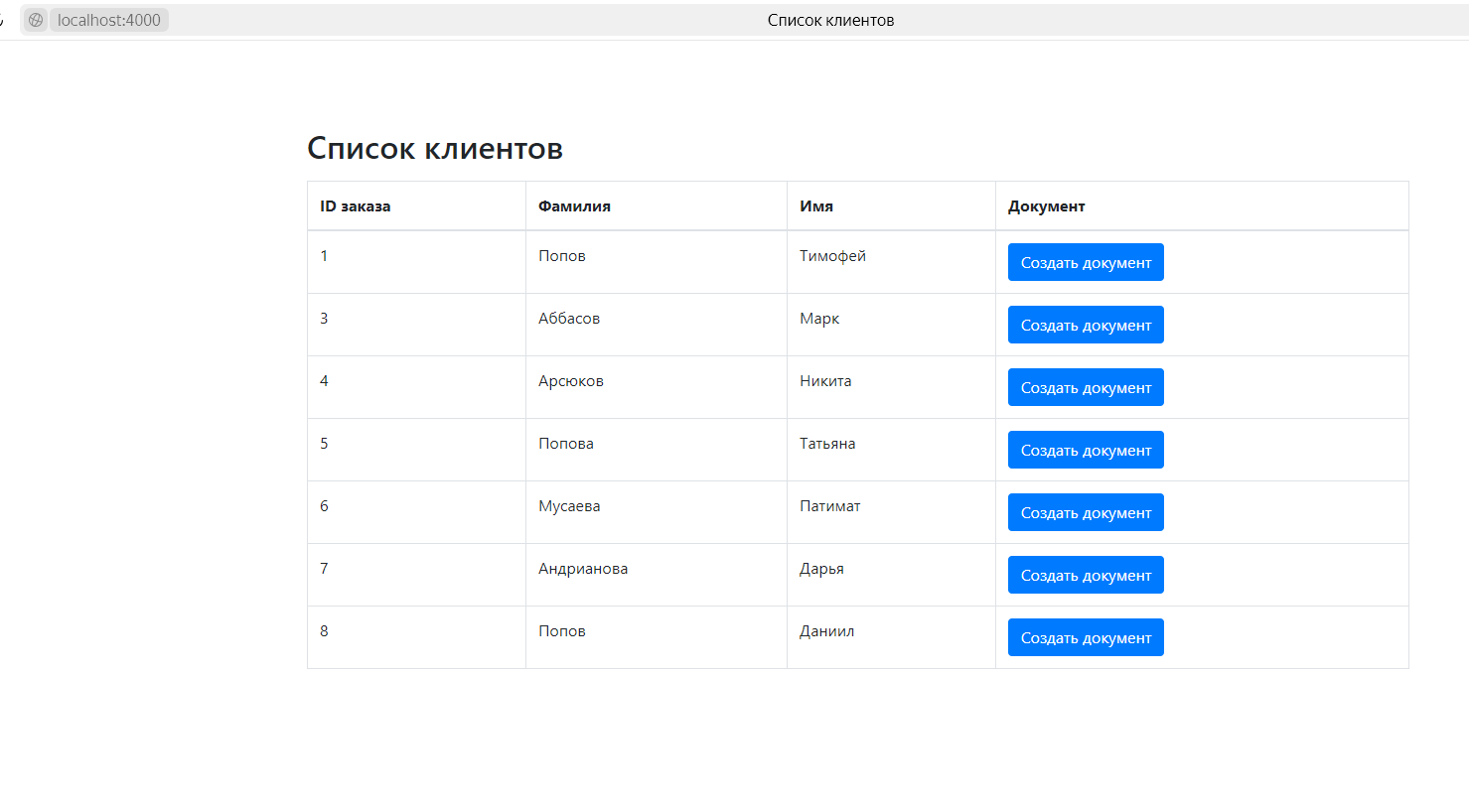


Рисунок 32. «Вкладка со списком клиентов для отчета»

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рисунок 33. «Текстовый документ о заказе»

7.Так же возможна реализации API-функции перейдя по вкладке “Прогноз погоды”. (рис.34)

Изображение выглядит как текст, снимок экрана, Шрифт, диаграмма

Автоматически созданное описание

Рисунок 34. «Вкладка прогноз погоды»

**Интерфейс пользователя**

1.Для входа в интерактивную систему управления сайтом доставки пиццы, пожалуйста перейдите на сайт <http://localhost/denwer/index.php>. На открывшейся странице вы увидите приветственный текст и кнопку «Войти» (рис. 35), после нажатия которой появится модальное окно, в котором вы найдете форму для авторизации. Введите в соответствующие поля свои учетные данные – логин и пароль, выданные администратором. Затем нажмите кнопку «Войти», чтобы перейти к использованию всех возможностей системы.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 35. «Авторизация »

2.Входим как зарегистрированный пользователь на главную страницу, где можно перейти на вкладку Сделать заказ, Меню, Просмотр погоды для получение нужной информации(рис. 36)

Изображение выглядит как текст, снимок экрана, веб-страница, Веб-сайт

Автоматически созданное описание

Рисунок 36. «Интерфейс пользователя»

3.Переходм на вкладку “Контакты” где мы можем ознакомиться с информацией о предприятии.(рис.37)

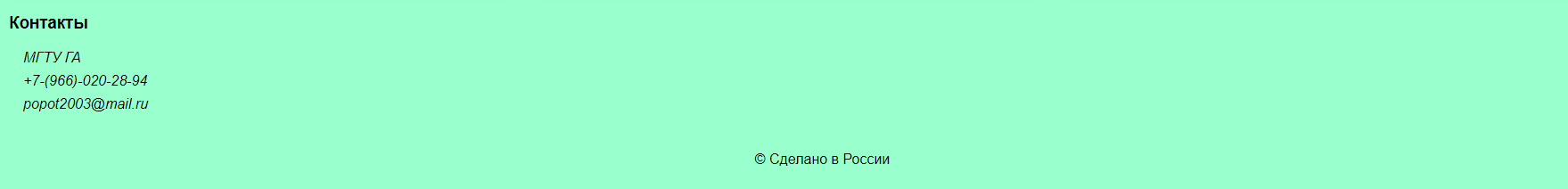


Рисунок 37. «Вкладка контакты»

4.Переходим на вкладку “Сделать заказ” для реализации заказа.(рис.38)

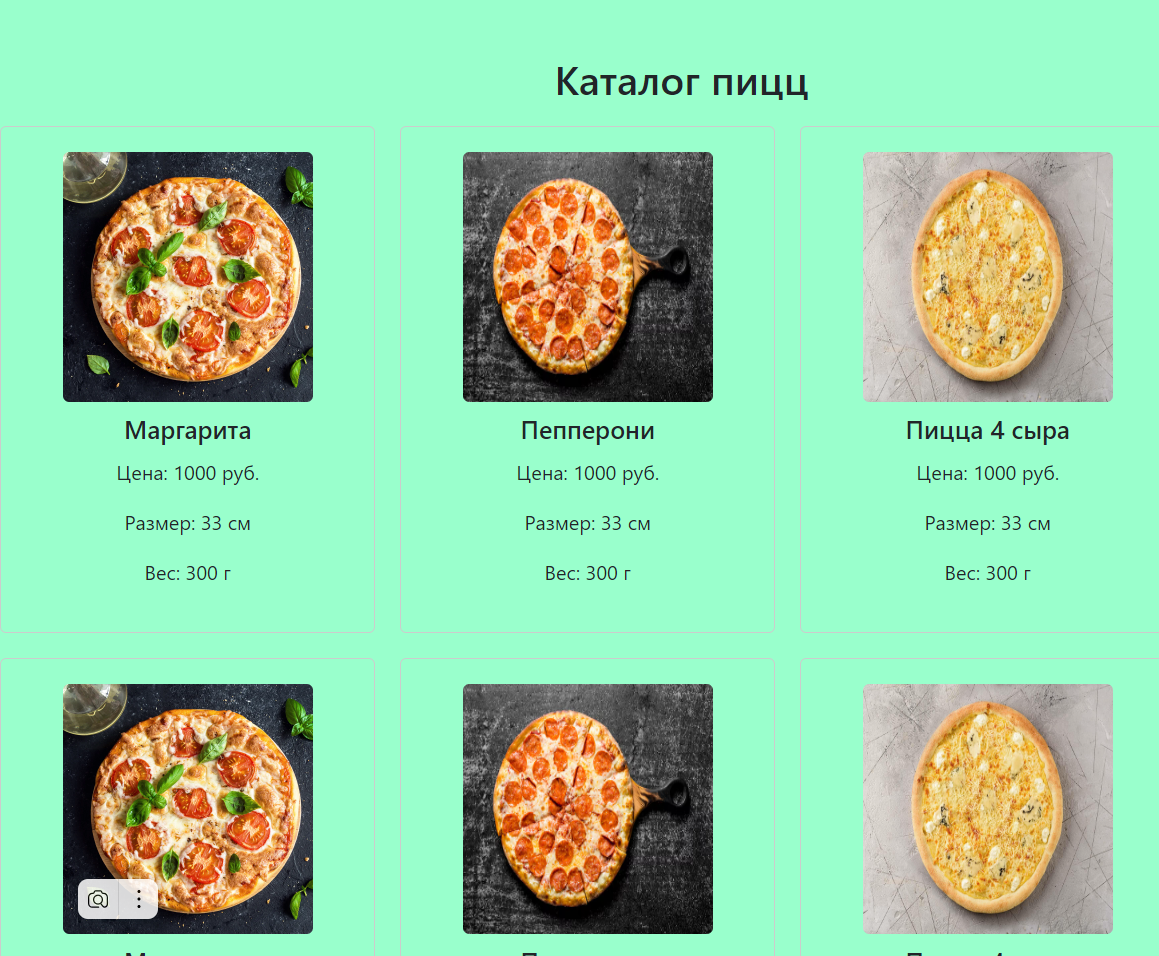


Рисунок 38. «Вкладка меню»

При нажатии на интересующую нас пиццы снизу страницы появляется кнопка “Перейти к корзине”.(рис.39)

Изображение выглядит как текст, Быстрое питание, меню, снимок экрана

Автоматически созданное описание

Рисунок 39. «Вкладка меню»

При нажатии на кнопку “Перейти к корзине” нас переносит на страницу где мы можем ввести свои данные и ознакомиться с суммой заказа.(рис. 40)

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 40. «Вкладка с оформлением заказа»

После введения пользователя своих данных и нажатия на кнопку “Оформить заказ” наш заказ сохраняется в базе данных .(рис.41)

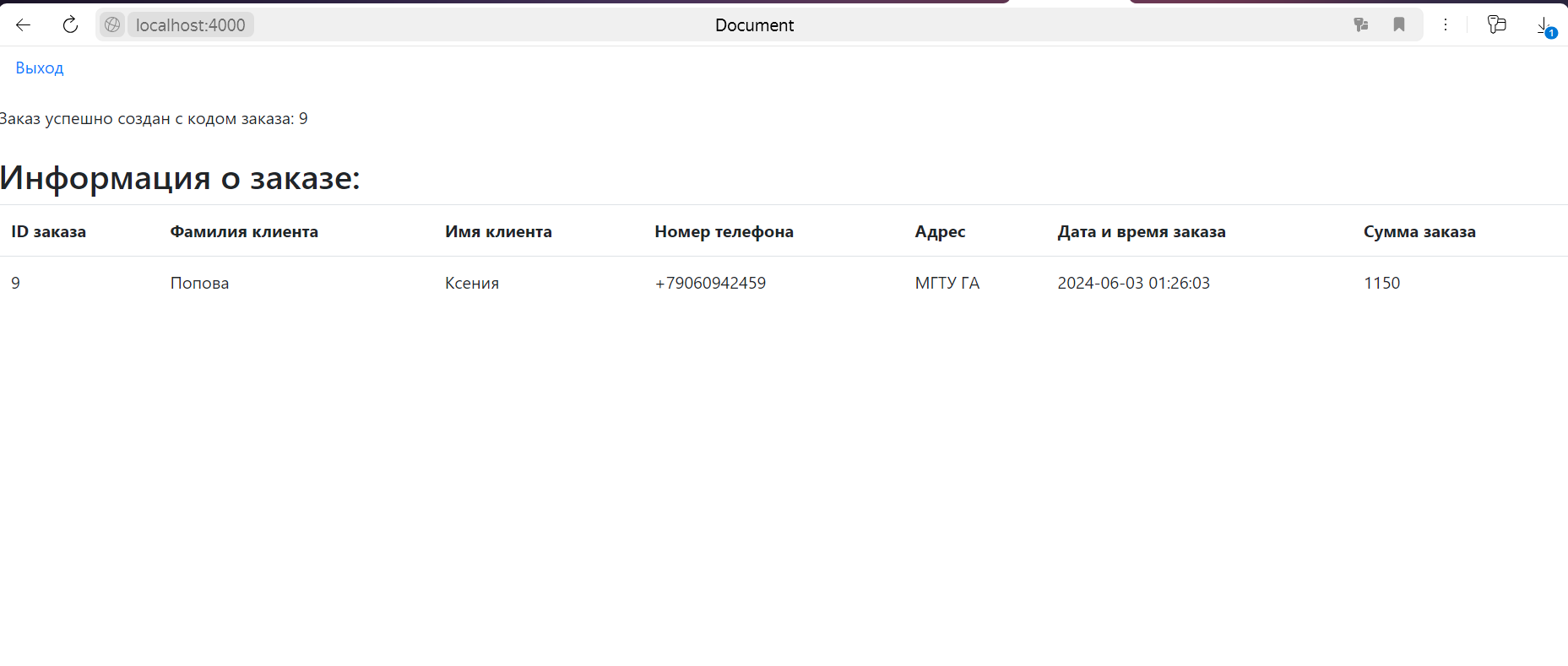


Рисунок 41. «Вкладка заказа»

**ПРИЛОЖЕНИЕ В. ИСХОДНЫЙ КОД**

Код основных файлов приведён ниже.

**Файл «index.php»**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Главная страница</title>

<style>

body{

background-color: #99FFCC;

}

</style>

<link rel="stylesheet" href="style\_main.css">

</head>

<body>

<div id="home" class="w3-content">

<header>

<a href="#" class="logo">Pizza Pie</a>

<div class="bx bx-menu" id="menu-icon"></div>

<ul class="navbar">

<a href="#home">На главную</a>

<a href="#menu">Меню</a>

<a id="loginBtn" href="#home">Вход</a>

<a href="#contact">Контакты</a>

</ul>

</header>

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

<div class="modal-content">

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

<h1>Добро пожаловать </h1>

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

<input type="text" name="username" placeholder="Логин">

<input type="password" name="password" placeholder="Пароль">

<input type="submit" value="Войти">

</form>

</div>

</div>

<div id="loginModal1" class="modal1">

<div class="modal-content1">

<span class="close1">&times;</span>

<a class="nuipogoda-iframe-informer" data-nuipogoda="informer2" href="https://nuipogoda.ru" style="width:284px;height:326px;display:block;box-shadow: 0 0 5px #999;">НУ И ПОГОДА</a><script>(function(a,f,g){var e=a.createElement(f);e.async=1;e.src=g; a=a.getElementsByTagName(f)[0];a.parentNode.insertBefore(e,a)})(document,'script','//nuipogoda.ru/informer/nuipogoda.js');</script>

</div>

</div>

<section class="home" id="home">

<div class="home-text">

<h1>Доставка Пиццы</h1>

</div>

<div class="home-img">

<img src="home.jpg" alt="">

</div>

</section>

<section class="menu" id="menu">

<div class="heading">

<span>МЕНЮ</span>

<h2>Лучшее для лучших</h2>

</div>

<div class="menu-container">

<!-- Box 1 -->

<div class="box">

<div class="box-img">

<img src="2.jpg" alt="">

</div>

<h2>Пицца 4 сыра</h2>

<h3>Горгондзола, моцарелла,<br> фонтина и пармезан</h3>

<span>От 600</span>

<i class='bx bx-cart-alt'></i>

</div>

<!-- Box 2 -->

<div class="box">

<div class="box-img">

<img src="3.avif" alt="">

</div>

<h2>Пицца пеперони</h2>

<h3>Cвинина,говядина,<br> паприка,острый перец</h3>

<span>От 500</span>

<i class='bx bx-cart-alt'></i>

</div>

<!-- Box 3 -->

<div class="box">

<div class="box-img">

<img src="5.jpg" alt="">

</div>

<h2>Пицца маргарита</h2>

<h3>Дрожжевое тесто, оливковое масло,<br> томаты, листья базилика, моцарелла</h3>

<span>От 550</span>

<i class='bx bx-cart-alt'></i>

</div>

</div>

</section>

<!-- Contact -->

<section class="contact" id="contact">

<div class="contact-box address">

<h3>Контакты</h3>

<i class='bx bxs-map' ><span>МГТУ ГА</span></i>

<i class='bx bxs-phone' ><span>+7-(966)-020-28-94</span></i>

<i class='bx bxs-envelope' ><span>popot2003@mail.ru</span></i>

</div>

</section>

<!-- Copyright -->

<div class="copyright">

<p>© Сделано в России</p>

</div>

<!-- Scroll Reveal -->

<script src="https://unpkg.com/scrollreveal"></script>

<!-- Link To JavaScript -->

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

<script>

var modal = document.getElementById("loginModal");

var btn = document.getElementById("loginBtn");

var span = document.getElementsByClassName("close")[0];

btn.onclick = function() {

modal.style.display = "flex";

}

span.onclick = function() {

modal.style.display = "none";

}

window.onclick = function(event) {

if (event.target == modal) {

modal.style.display = "none";

}

}

</script>

</body>

</html>

**Файл «full.php»**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Root</title>

<link rel="stylesheet" href="1.css">

</head>

<body>

<header>

<div class="bx bx-menu" id="menu-icon"></div>

<ul class="navbar">

<a href="index2.php">Выход</a>

</ul>

</header>

<?php

$serverName = "localhost";

$userName = "root";

$password = "";

$dbname = "deliverypizza";

$conn = new mysqli($serverName, $userName, $password, $dbname);

// Проверка подключения

if ($conn->connect\_error) {

die("Ошибка подключения: " . $conn->connect\_error);

}

// Запрос для получения списка таблиц в базе данных

$sql\_table\_name = "SHOW TABLES FROM $dbname";

$result\_table\_name = $conn->query($sql\_table\_name);

// Формирование меню для выбора таблицы

echo "<form class='.nav\_\_form form' method='post' action='".$\_SERVER['PHP\_SELF']."'>";

echo "<ul class='inside'>";

while($row\_table = $result\_table\_name->fetch\_row()) {

$table\_name = $row\_table[0];

echo "<div class='wrapper'>";

echo"<article class='flow'>";

echo "<div class='list-container'>";

echo "<li class='nav\_\_item'><label class='form\_\_label'><input class='nav\_\_radio' type='radio' name='table' value='$table\_name'><span class='nav\_\_text'>$table\_name</span></label></li>";

echo "</div>";

echo"</article>";

echo "</div>";

}

echo "</ul>";

echo "<input class='nav\_\_button button' type='submit' value='Показать таблицу'>";

echo "</form>";

?>

<?php

//Функция вывода таблицы

function getTable($result) {

if ($result->num\_rows > 0) {

echo "<table class='table'><tr class='table\_\_row'>";

while ($fieldinfo = $result->fetch\_field()) {

echo "<th class='table\_\_heading'>".$fieldinfo->name."</th>";

}

echo "</tr>";

while($row\_data = $result->fetch\_assoc()) {

echo "<tr class='table\_\_row'>";

foreach ($row\_data as $key => $value) {

echo "<td class='table\_\_data'>".$value."</td>";

}

echo "</tr>";

}

echo "</table>";

}

}

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

switch (isset($\_POST)) {

// Вывод выбранной таблицы

case $\_POST['table']:

// Получение данных из поля

$selected\_table = $\_POST['table'];

$sql\_data = "SELECT \* FROM $selected\_table";

// Выполнение запроса

$result\_data = $conn->query($sql\_data);

//Вывод результата

echo "<h2 class='table\_\_title'>Таблица: $selected\_table</h2>";

getTable($result\_data);

break;

// Вывод запроса

case $\_POST['sql\_query']:

//Получение запроса из поля

$sql\_query = $\_POST['sql\_query'];

// Проверка на то, что поле не пустое

if (!$sql\_query) {

$sql\_query = '#';

}

// Выполнение SQL запроса

$result = $conn->query($sql\_query);

// Вывод результата запроса

echo "<h2 class='table\_\_title'>Результат запроса:</h2>";

getTable($result);

break;

}

}

// Закрытие соединения с базой данных

$conn->close();

?>

</body>

</html>

**Файл «listener.php»**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Главная страница</title>

<style>

body{

background-color: #99FFCC;

}

</style>

<link rel="stylesheet" href="style\_main.css">

</head>

<body>

<div id="home" class="w3-content">

<header>

<a href="#" class="logo">Pizza Pie</a>

<div class="bx bx-menu" id="menu-icon"></div>

<ul class="navbar">

<a href="catalog.php">Сделать заказ</a>

<a href="#menu">Контакты</a>

<a id="loginBtn1" href="#home">Посмотреть погоду</a>

</ul>

</header>

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

<div class="modal-content">

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

<h1>Добро пожаловать </h1>

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

<input type="text" name="username" placeholder="Логин">

<input type="password" name="password" placeholder="Пароль">

<input type="submit" value="Войти">

</form>

</div>

</div>

<div id="loginModal2" class="modal2">

<div class="modal-content2">

<span class="close2">&times;</span>

<h1>Недостаточно прав, войдите под "root" пользователем</h1>

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

<input type="text" name="username" placeholder="Логин">

<input type="password" name="password" placeholder="Пароль">

<input type="submit" value="Войти">

</form>

</div>

</div>

<div id="loginModal1" class="modal1">

<div class="modal-content1">

<span class="close1">&times;</span>

<a class="nuipogoda-iframe-informer" data-nuipogoda="informer2" href="https://nuipogoda.ru" style="width:284px;height:326px;display:block;box-shadow: 0 0 5px #999;">НУ И ПОГОДА</a><script>(function(a,f,g){var e=a.createElement(f);e.async=1;e.src=g; a=a.getElementsByTagName(f)[0];a.parentNode.insertBefore(e,a)})(document,'script','//nuipogoda.ru/informer/nuipogoda.js');</script>

</div>

</div>

<section class="home" id="home">

<div class="home-text">

<h1>Доставка Пиццы</h1>

</div>

<div class="home-img">

<img src="home.jpg" alt="">

</div>

</section>

<section class="menu" id="menu">

<div class="heading">

<span>МЕНЮ</span>

<h2>Лучшее для лучших</h2>

</div>

<div class="menu-container">

<!-- Box 1 -->

<div class="box">

<div class="box-img">

<img src="2.jpg" alt="">

</div>

<h2>Пицца 4 сыра</h2>

<h3>Горгондзола, моцарелла,<br> фонтина и пармезан</h3>

<span>От 600</span>

<i class='bx bx-cart-alt'></i>

</div>

<!-- Box 2 -->

<div class="box">

<div class="box-img">

<img src="3.avif" alt="">

</div>

<h2>Пицца пеперони</h2>

<h3>Cвинина,говядина,<br> паприка,острый перец</h3>

<span>От 500</span>

<i class='bx bx-cart-alt'></i>

</div>

<!-- Box 3 -->

<div class="box">

<div class="box-img">

<img src="5.jpg" alt="">

</div>

<h2>Пицца маргарита</h2>

<h3>Дрожжевое тесто, оливковое масло,<br> томаты, листья базилика, моцарелла</h3>

<span>От 550</span>

<i class='bx bx-cart-alt'></i>

</div>

</section>

<!-- Contact -->

<section class="contact" id="contact">

<div class="contact-box address">

<h3>Контакты</h3>

<i class='bx bxs-map' ><span>МГТУ ГА</span></i>

<i class='bx bxs-phone' ><span>+7-(966)-020-28-94</span></i>

<i class='bx bxs-envelope' ><span>popot2003@mail.ru</span></i>

</div>

</section>

<!-- Copyright -->

<div class="copyright">

<p>© Сделано в России</p>

</div>

<!-- Scroll Reveal -->

<script src="https://unpkg.com/scrollreveal"></script>

<!-- Link To JavaScript -->

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

<script>

var modal1 = document.getElementById("loginModal1");

var btn1 = document.getElementById("loginBtn1");

var span1 = document.getElementsByClassName("close1")[0];

btn1.onclick = function() {

modal1.style.display = "flex";

}

span1.onclick = function() {

modal1.style.display = "none";

}

window1.onclick = function(event) {

if (event.target == modal1) {

modal1.style.display = "none";

}

}

</script>

<script>

var modal2 = document.getElementById("loginModal2");

var btn2 = document.getElementById("loginBtn2");

var span2 = document.getElementsByClassName("close2")[0];

btn2.onclick = function() {

modal2.style.display = "flex";

}

span2.onclick = function() {

modal2.style.display = "none";

}

window2.onclick = function(event) {

if (event.target == modal2) {

modal2.style.display = "none";

}

}

</script>

<script>

var modal = document.getElementById("loginModal");

var btn = document.getElementById("loginBtn");

var span = document.getElementsByClassName("close")[0];

btn.onclick = function() {

modal.style.display = "flex";

}

span.onclick = function() {

modal.style.display = "none";

}

window.onclick = function(event) {

if (event.target == modal) {

modal.style.display = "none";

}

}

</script>

</body>

</html>

**Файл «index2.php»**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Главная страница</title>

<style>

body{

background-color: #99FFCC;

}

</style>

<link rel="stylesheet" href="style\_main.css">

</head>

<body>

<div id="home" class="w3-content">

<header>

<a href="#" class="logo">Pizza Pie</a>

<div class="bx bx-menu" id="menu-icon"></div>

<ul class="navbar">

<a id="loginBtn" href="#home">Сменить пользователя</a>

<a href="#menu">Меню</a>

<a id="loginBtn1" href="#home">Посмотреть погоду</a>

<a href="full.php">Просмотреть таблицы</a>

<a href="sql.php">Написать запрос</a>

<a href="table1.php">Таблица:pizza</a>

<a href="table2.php">Таблица:klient</a>

<a href="t2.php">Отчет</a>

</ul>

</header>

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

<div class="modal-content">

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

<h1>Добро пожаловать </h1>

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

<input type="text" name="username" placeholder="Логин">

<input type="password" name="password" placeholder="Пароль">

<input type="submit" value="Войти">

</form>

</div>

</div>

<div id="loginModal1" class="modal1">

<div class="modal-content1">

<span class="close1">&times;</span>

<a class="nuipogoda-iframe-informer" data-nuipogoda="informer2" href="https://nuipogoda.ru" style="width:284px;height:326px;display:block;box-shadow: 0 0 5px #999;">НУ И ПОГОДА</a><script>(function(a,f,g){var e=a.createElement(f);e.async=1;e.src=g; a=a.getElementsByTagName(f)[0];a.parentNode.insertBefore(e,a)})(document,'script','//nuipogoda.ru/informer/nuipogoda.js');</script>

</div>

</div>

<section class="home" id="home">

<div class="home-text">

<h1>Доставка Пиццы</h1>

</div>

<div class="home-img">

<img src="home.jpg" alt="">

</div>

</section>

<section class="menu" id="menu">

<div class="heading">

<span>МЕНЮ</span>

<h2>Лучшее для лучших</h2>

</div>

<div class="menu-container">

<!-- Box 1 -->

<div class="box">

<div class="box-img">

<img src="2.jpg" alt="">

</div>

<h2>Пицца 4 сыра</h2>

<h3>Горгондзола, моцарелла,<br> фонтина и пармезан</h3>

<span>От 600</span>

<i class='bx bx-cart-alt'></i>

</div>

<!-- Box 2 -->

<div class="box">

<div class="box-img">

<img src="3.avif" alt="">

</div>

<h2>Пицца пеперони</h2>

<h3>Cвинина,говядина,<br> паприка,острый перец</h3>

<span>От 500</span>

<i class='bx bx-cart-alt'></i>

</div>

<!-- Box 3 -->

<div class="box">

<div class="box-img">

<img src="5.jpg" alt="">

</div>

<h2>Пицца маргарита</h2>

<h3>Дрожжевое тесто, оливковое масло,<br> томаты, листья базилика, моцарелла</h3>

<span>От 550</span>

<i class='bx bx-cart-alt'></i>

</div>

</section>

<!-- Contact -->

<section class="contact" id="contact">

<div class="contact-box address">

<h3>Контакты</h3>

<i class='bx bxs-map' ><span>МГТУ ГА</span></i>

<i class='bx bxs-phone' ><span>+7-(966)-020-28-94</span></i>

<i class='bx bxs-envelope' ><span>popot2003@mail.ru</span></i>

</div>

</section>

<!-- Copyright -->

<div class="copyright">

<p>© Сделано в России</p>

</div>

<!-- Scroll Reveal -->

<script src="https://unpkg.com/scrollreveal"></script>

<!-- Link To JavaScript -->

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

<script>

var modal1 = document.getElementById("loginModal1");

var btn1 = document.getElementById("loginBtn1");

var span1 = document.getElementsByClassName("close1")[0];

btn1.onclick = function() {

modal1.style.display = "flex";

}

span1.onclick = function() {

modal1.style.display = "none";

}

window1.onclick = function(event) {

if (event.target == modal1) {

modal1.style.display = "none";

}

}

</script>

<script>

var modal = document.getElementById("loginModal");

var btn = document.getElementById("loginBtn");

var span = document.getElementsByClassName("close")[0];

btn.onclick = function() {

modal.style.display = "flex";

}

span.onclick = function() {

modal.style.display = "none";

}

window.onclick = function(event) {

if (event.target == modal) {

modal.style.display = "none";

}

}

</script>

</body>

</html>

**Файл «login.php»**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

</head>

<body>

<?php

// Проверка наличия данных в форме

if(isset($\_POST['username']) && isset($\_POST['password'])) {

$userName = $\_POST['username'];

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

// Проведение проверки ролей

if($userName == 'root' && $password == '') {

// Перенаправление на страницу для пользователя "root"

header("Location: index2.php");

}

elseif($userName == 'timofey' && $password == '123') {

// Перенаправление на страницу для пользователя "listeners"

header("Location: listener.php");

} else {

echo "Неправильное имя пользователя или пароль";

}

}

?>

</body>

</html>

**Файл «cart.php»**

<?php

session\_start();

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

$\_SESSION['cart'] = $\_POST;

}

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

?>

<!DOCTYPE html>

<html>

<head>

<title>Корзина</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">

</head>

<body>

<style>

body {

background-color: #99FFCC;

}

.cart {

margin-top: 20px;

}

</style>

<div class="container">

<h2>Корзина</h2>

<div class="cart">

<form method="POST" action="order.php">

<div id="cart\_items">

<?php

$total = 0;

foreach ($cartItems['Name\_pizza'] as $index => $namePizza) {

$price = $cartItems['Price\_pizza'][$index];

$quantity = $cartItems['Quantity'][$index];

$sum = $price \* $quantity;

$total += $sum;

echo "

<div class='cart-item' data-id='{$index}'>

<p>{$namePizza} - {$price} руб. x {$quantity} шт. = {$sum} руб.</p>

<input type='hidden' name='Name\_pizza[]' value='{$namePizza}'>

<input type='hidden' name='Price\_pizza[]' value='{$price}'>

<input type='hidden' name='Quantity[]' value='{$quantity}'>

</div>";

}

?>

<h3>Итого: <?php echo $total; ?> руб.</h3>

</div>

<div class="form-group">

<input type="text" name="Surname" class="form-control" placeholder="Введите фамилию" required>

</div>

<div class="form-group">

<input type="text" name="Name" class="form-control" placeholder="Введите имя" required>

</div>

<div class="form-group">

<input type="text" name="Number" class="form-control" placeholder="Телефон" required>

</div>

<div class="form-group">

<input type="text" name="Address" class="form-control" placeholder="Адрес" required>

</div>

<button type="submit" class="btn btn-primary">Оформить заказ</button>

</form>

</div>

</div>

</body>

</html>

**Файл «catalog.php»**

<?php

include 'connect.php';

?>

<!DOCTYPE html>

<html>

<head>

<title>Каталог пицц</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">

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

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script>

</head>

<body>

<header>

<div class="bx bx-menu" id="menu-icon"></div>

<ul class="navbar">

<a href="listener.php">Выход</a>

</ul>

</header>

<style>

body {

background-color: #99FFCC;

}

.catalog {

display: flex;

flex-wrap: wrap;

}

.pizza {

width: 300px;

margin: 10px;

padding: 20px;

border: 1px solid #ccc;

border-radius: 5px;

text-align: center;

cursor: pointer;

}

.pizza img {

width: 200px;

height: 200px;

border-radius: 5px;

margin-bottom: 10px;

}

.pizza h3 {

font-size: 20px;

margin-bottom: 10px;

}

.pizza p {

font-size: 16px;

}

.cart {

margin-top: 20px;

}

</style>

<div class="container">

<h2><center>Каталог пицц</center></h2>

<div class="catalog">

<?php

// Запрос для получения данных о пиццах

$pizzaQuery = $mysqli->query("SELECT ID\_Pizza, Name\_pizza, Price\_pizza,size , Weight FROM pizza

Join size\_pizza on size\_pizza.ID\_size = pizza.Size\_ID");

while ($row = $pizzaQuery->fetch\_assoc()) {

echo "

<div class='pizza' data-id='{$row['ID\_Pizza']}' data-name='{$row['Name\_pizza']}' data-price='{$row['Price\_pizza']}'>

<img src='photo/{$row['ID\_Pizza']}.jpg' alt='{$row['Name\_pizza']}'>

<h3>{$row['Name\_pizza']}</h3>

<p>Цена: {$row['Price\_pizza']} руб.</p>

<p>Размер: {$row['size']} см</p>

<p>Вес: {$row['Weight']} г</p>

</div>";

}

?>

</div>

<div class="cart">

<form method="POST" action="cart.php">

<div id="cart\_items"></div>

<button type="submit" class="btn btn-primary">Перейти к корзине</button>

</form>

</div>

</div>

<script>

$(document).ready(function() {

$('.pizza').click(function() {

var id = $(this).data('id');

var name = $(this).data('name');

var price = $(this).data('price');

var cartItem = `

<div class="cart-item" data-id="${id}">

<p>${name} - ${price} руб.</p>

<input type="hidden" name="Name\_pizza[]" value="${name}">

<input type="hidden" name="Price\_pizza[]" value="${price}">

<input type="number" name="Quantity[]" value="1" min="1" class="form-control" required>

<button type="button" class="btn btn-danger" onclick="removeCartItem(this)">Удалить</button>

</div>`;

$('#cart\_items').append(cartItem);

});

});

function removeCartItem(button) {

$(button).parent('.cart-item').remove();

}

</script>

</body>

</html>

**Файл «connect.php»**

<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "deliverypizza";

$mysqli = new mysqli($servername, $username, $password, $dbname);

if ($mysqli->connect\_error) {

die("Ошибка подключения: " . $mysqli->connect\_error);

}

?>

**Файл «** **generate\_document.php»**

<?php

// Подключение к базе данных

$host = 'localhost';

$username = 'root';

$password = '';

$database = 'deliverypizza';

$conn = new mysqli($host, $username, $password, $database);

if ($conn->connect\_error) {

die("Ошибка подключения: " . $conn->connect\_error);

}

// Получение ID заказа из URL

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

// Запрос к базе данных для получения заказа клиента

$sql = "SELECT \* FROM order\_online WHERE ID\_order\_online = ?";

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

$stmt->bind\_param('i', $order\_id);

$stmt->execute();

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

if ($result->num\_rows > 0) {

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

// Создание текстового документа

$filename = 'заказ\_клиента\_' . $order\_id . '.txt';

$file = fopen($filename, 'w');

fwrite($file, "Заказ клиента:\n");

fwrite($file, "--------------\n");

fwrite($file, "Номер заказа: " . $row['ID\_order\_online'] . "\n");

fwrite($file, "Фамилия: " . $row['Surname'] . "\n");

fwrite($file, "Имя: " . $row['Name'] . "\n");

fwrite($file, "Адрес: " . $row['Address'] . "\n");

fwrite($file, "Номер телефона: " . $row['Number'] . "\n");

fwrite($file, "Сумма заказа: " . $row['Sum\_order'] . "\n");

// Запрос к базе данных для получения информации о пиццах в заказе

$pizzaSql = "

SELECT p.Name\_pizza, pc.Quantity, sz.size

FROM pizza\_check pc

JOIN pizza p ON pc.Pizza\_ID = p.ID\_Pizza

JOIN size\_pizza sz ON p.Size\_ID = sz.ID\_size

WHERE pc.order\_online\_ID = ?";

$pizzaStmt = $conn->prepare($pizzaSql);

$pizzaStmt->bind\_param('i', $order\_id);

$pizzaStmt->execute();

$pizzaResult = $pizzaStmt->get\_result();

if ($pizzaResult->num\_rows > 0) {

fwrite($file, "\nПиццы в заказе:\n");

fwrite($file, "----------------\n");

while ($pizzaRow = $pizzaResult->fetch\_assoc()) {

fwrite($file, "Название пиццы: " . $pizzaRow['Name\_pizza'] . "\n");

fwrite($file, "Размер: " . $pizzaRow['size'] . " см\n");

fwrite($file, "Количество: " . $pizzaRow['Quantity'] . "\n");

fwrite($file, "----------------\n");

}

} else {

fwrite($file, "\nПиццы в заказе не найдены.\n");

}

fclose($file);

// Вывод текстового документа

header('Content-Description: File Transfer');

header('Content-Type: application/octet-stream');

header('Content-Disposition: attachment; filename=' . basename($filename));

header('Content-Length: ' . filesize($filename));

readfile($filename);

// Удаление файла после скачивания

unlink($filename);

} else {

echo "Заказ не найден";

}

// Закрытие соединения с базой данных

$stmt->close();

$conn->close();

?>

**Файл «** **login.php»**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

</head>

<body>

<?php

// Проверка наличия данных в форме

if(isset($\_POST['username']) && isset($\_POST['password'])) {

$userName = $\_POST['username'];

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

// Проведение проверки ролей

if($userName == 'root' && $password == '') {

// Перенаправление на страницу для пользователя "root"

header("Location: index2.php");

}

elseif($userName == 'timofey' && $password == '123') {

// Перенаправление на страницу для пользователя "listeners"

header("Location: listener.php");

} else {

echo "Неправильное имя пользователя или пароль";

}

}

?>

</body>

</html>

**Файл «** **order.php»**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">

</head>

<body>

<header>

<div class="bx bx-menu" id="menu-icon"></div>

<ul class="navbar">

<a href="listener.php">Выход</a>

</ul>

</header>

</body>

</html>

<?php

include 'connect.php';

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

$klientSurname = $\_POST['Surname'];

$klientName = $\_POST['Name'];

$klientNumber = $\_POST['Number'];

$klientAddress = $\_POST['Address'];

$currentDateTime = date('Y-m-d H:i:s');

// Получаем максимальные значения ID\_check и ID\_order\_online

$resultCheck = $mysqli->query("SELECT MAX(ID\_check) AS maxID\_check FROM pizza\_check");

$rowCheck = $resultCheck->fetch\_assoc();

$nextID\_check = $rowCheck['maxID\_check'] + 1;

$resultOrder = $mysqli->query("SELECT MAX(ID\_order\_online) AS maxID\_order\_online FROM order\_online");

$rowOrder = $resultOrder->fetch\_assoc();

$nextID\_order\_online = $rowOrder['maxID\_order\_online'] + 1;

$namePizzas = $\_POST['Name\_pizza'];

$prices = $\_POST['Price\_pizza'];

$quantities = $\_POST['Quantity'];

$status = 1;

// Добавлено получение размеров пицц из формы

$total = 0;

session\_start();

if (isset($\_SESSION['number']) && $\_SESSION['number'] == 2) {

$\_SESSION['number'] = 2;

$empli = 2;

} else {

$\_SESSION['number'] = 4;

$empli = 4;

}

$mysqli->begin\_transaction();

try {

foreach ($namePizzas as $index => $namePizza) {

$price = $prices[$index];

$quantity = $quantities[$index];

// Размер пиццы

// Обновленный запрос для получения ID\_Pizza по названию пиццы и её размеру

$stmt = $mysqli->prepare("

SELECT p.ID\_Pizza

FROM pizza p

JOIN size\_pizza s ON p.Size\_ID = s.ID\_size

WHERE p.Name\_pizza = ? AND Price\_pizza = ?

");

$stmt->bind\_param("si", $namePizza, $price);

$stmt->execute();

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

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

$pizza = $product['ID\_Pizza'];

$orderSum = $price \* $quantity;

$total += $orderSum;

$stmt = $mysqli->prepare("INSERT INTO pizza\_check (ID\_check, order\_online\_ID, Pizza\_ID, Quantity) VALUES (?, ?, ?, ?)");

$stmt->bind\_param("iiii", $nextID\_check, $nextID\_order\_online, $pizza, $quantity);

$stmt->execute();

$nextID\_check++;

}

$stmt = $mysqli->prepare("INSERT INTO order\_online (ID\_order\_online, Surname, Name, Number, Address, Date\_and\_time, Sum\_order, status\_ID, employees\_id) VALUES (?, ?, ?, ?, ?, ?, ?,?,?)");

$stmt->bind\_param("isssssssi", $nextID\_order\_online, $klientSurname, $klientName, $klientNumber, $klientAddress, $currentDateTime, $total, $status, $empli);

$stmt->execute();

$mysqli->commit();

echo "Заказ успешно создан с кодом заказа: $nextID\_order\_online";

echo "<br><br><h2>Информация о заказе:</h2>";

echo "<table class='table'>";

echo "<tr><th>ID заказа</th><th>Фамилия клиента</th><th>Имя клиента</th><th>Номер телефона</th><th>Адрес</th><th>Дата и время заказа</th><th>Сумма заказа</th></tr>";

echo "<tr><td>$nextID\_order\_online</td><td>$klientSurname</td><td>$klientName</td><td>$klientNumber</td><td>$klientAddress</td><td>$currentDateTime</td><td>$total</td></tr>";

echo "</table>";

} catch (Exception $e) {

$mysqli->rollback();

echo "Ошибка создания заказа: " . $e->getMessage();

}

$stmt->close();

}

$mysqli->close();

?>

**Файл «sql.php»**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Root</title>

<link rel="stylesheet" href="1.css">

</head>

<body>

<header>

<div class="bx bx-menu" id="menu-icon"></div>

<ul class="navbar">

<a href="index2.php">Выход</a>

<a href="sql2.php">Готовые запросы</a>

</ul>

</header>

<?php

$serverName = "localhost";

$userName = "root";

$password = "";

$dbname = "deliverypizza";

$conn = new mysqli($serverName, $userName, $password, $dbname);

// Проверка подключения

if ($conn->connect\_error) {

die("Ошибка подключения: " . $conn->connect\_error);

}

?>

<div class="sql-container">

<form method="post" action="<?php echo $\_SERVER['SCRIPT\_NAME']; ?>">

<label class="sql-label" for="sql\_query"><span class="sql-span">Введите SQL запрос:</span></label><br>

<textarea name="sql\_query" rows="4" cols="37" placeholder='Введите запрос'><?= $\_POST['sql\_query']; ?></textarea><br>

<input class="button" type="submit" value="Выполнить запрос">

<?php

//Функция вывода таблицы

function getTable($result) {

if ($result->num\_rows > 0) {

echo "<table class='table'><tr class='table\_\_row'>";

while ($fieldinfo = $result->fetch\_field()) {

echo "<th class='table\_\_heading'>".$fieldinfo->name."</th>";

}

echo "</tr>";

while($row\_data = $result->fetch\_assoc()) {

echo "<tr class='table\_\_row'>";

foreach ($row\_data as $key => $value) {

echo "<td class='table\_\_data'>".$value."</td>";

}

echo "</tr>";

}

echo "</table>";

}

}

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

switch (isset($\_POST)) {

// Вывод запроса

case $\_POST['sql\_query']:

//Получение запроса из поля

$sql\_query = $\_POST['sql\_query'];

// Проверка на то, что поле не пустое

if (!$sql\_query) {

$sql\_query = '#';

}

// Выполнение SQL запроса

$result = $conn->query($sql\_query);

// Вывод результата запроса

echo "<h2 class='table\_\_title'>Результат запроса:</h2>";

getTable($result);

break;

}

}

// Закрытие соединения с базой данных

$conn->close();

?>

</form>

<section class="sql-table">

</secton>

</div>

</body>

</html>

**Файл «sql2.php»**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Root</title>

<link rel="stylesheet" href="2.css">

</head>

<body>

<header>

<div class="bx bx-menu" id="menu-icon"></div>

<ul class="navbar">

<a href="index2.php">Выход</a>

</ul>

</header>

<?php

$serverName = "localhost";

$userName = "root";

$password = "";

$dbname = "deliverypizza";

$conn = new mysqli($serverName, $userName, $password, $dbname);

// Проверка подключения

if ($conn->connect\_error) {

die("Ошибка подключения: " . $conn->connect\_error);

}

?>

<div class="sql-container">

<form method="post" action="<?php echo $\_SERVER['SCRIPT\_NAME']; ?>">

<label class="sql-label" for="sql\_query"><span class="sql-span">Введите SQL запрос:</span></label><br>

<form method="post">

<textarea id="myTextarea" name="sql\_query" rows="4" cols="37" placeholder='Введите запрос'onChange='updateText("myInput")'><?= $\_POST['sql\_query']; ?></textarea><br>

<div class="list">

<h2>Готовые запросы</h2>

<ul>

<li><span><button onclick="autoFill()">Вывести имена курьеров и их общее количество заказов</button></span></li>

<li><span><button onclick="autoFill1()">Вывести выручку за каждый месяц</button></span></li>

<li><span><button onclick="autoFill2()">Вывести среднее время доставки курьеров за первый месяц 2023 года</button></span></li>

</ul>

</div>

>

<script>

function autoFill() {

document.getElementById("myTextarea").value += "SELECT name\_employees AS 'Имя курьера ', COUNT( \* ) AS 'Количество заказов' FROM `order\_online` JOIN employees ON order\_online.employees\_id = employees.ID\_Employees GROUP BY employees\_id";

}

</script>

<script>

function autoFill1() {

document.getElementById("myTextarea").value += "SELECT MONTH( Date\_order ) AS 'месяца', SUM( Sum\_order ) AS 'Выручка за месяц' FROM `order\_online` GROUP BY MONTH( Date\_order )";

}

</script>

<script>

function autoFill2() {

document.getElementById("myTextarea").value += "SELECT AVG\_date AS 'Дата', AVG AS 'Среднее время доставки', name\_employees AS 'Имя курьера' FROM avg\_delivery JOIN employees ON employees.ID\_Employees = avg\_delivery.employees\_id WHERE EXTRACT( MONTH FROM AVG\_date ) = 1";

}

</script>

</form>

<?php

//Функция вывода таблицы

function getTable($result) {

if ($result->num\_rows > 0) {

echo "<table class='table'><tr class='table\_\_row'>";

while ($fieldinfo = $result->fetch\_field()) {

echo "<th class='table\_\_heading'>".$fieldinfo->name."</th>";

}

echo "</tr>";

while($row\_data = $result->fetch\_assoc()) {

echo "<tr class='table\_\_row'>";

foreach ($row\_data as $key => $value) {

echo "<td class='table\_\_data'>".$value."</td>";

}

echo "</tr>";

}

echo "</table>";

}

}

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

switch (isset($\_POST)) {

// Вывод запроса

case $\_POST['sql\_query']:

//Получение запроса из поля

$sql\_query = $\_POST['sql\_query'];

// Проверка на то, что поле не пустое

if (!$sql\_query) {

$sql\_query = '#';

}

// Выполнение SQL запроса

$result = $conn->query($sql\_query);

// Вывод результата запроса

echo "<h2 class='table\_\_title'>Результат запроса:</h2>";

getTable($result);

break;

}

}

// Закрытие соединения с базой данных

$conn->close();

?>

<section class="sql-table">

</secton>

<script>

function fillText() {

document.getElementById("myInput").value = "привет ";

}

</script>

</form>

</div>

</body>

</html>

**Файл «t2.php»**

<?php

// Подключение к базе данных

$host = 'localhost';

$username = 'root';

$password = '';

$database = 'deliverypizza';

$conn = new mysqli($host, $username, $password, $database);

if ($conn->connect\_error) {

die("Ошибка подключения: " . $conn->connect\_error);

}

?>

<!DOCTYPE html>

<html>

<head>

<title>Список клиентов</title>

<div class="bx bx-menu" id="menu-icon"></div>

<ul class="navbar">

<a href="index2.php">Выход</a>

</ul>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">

</head>

<body>

<div class="container">

<h2 class="mt-5">Список клиентов</h2>

<table class="table table-bordered mt-3">

<thead>

<tr>

<th>ID заказа</th>

<th>Фамилия</th>

<th>Имя</th>

<th>Документ</th>

</tr>

</thead>

<tbody>

<?php

// Запрос к базе данных для получения списка клиентов

$sql = "SELECT ID\_order\_online, Surname, Name FROM order\_online";

$result = mysqli\_query($conn, $sql);

if (mysqli\_num\_rows($result) > 0) {

while ($row = mysqli\_fetch\_assoc($result)) {

echo "<tr>

<td>{$row['ID\_order\_online']}</td>

<td>{$row['Surname']}</td>

<td>{$row['Name']}</td>

<td><a href='generate\_document.php?id={$row['ID\_order\_online']}' class='btn btn-primary'>Создать документ</a></td>

</tr>";

}

} else {

echo "<tr><td colspan='4'>Нет клиентов</td></tr>";

}

// Закрытие соединения с базой данных

mysqli\_close($conn);

?>

</tbody>

</table>

</div>

</body>

</html>

**Файл «table1.php»**

<!DOCTYPE html>

<html>

<head>

<title>Pizza</title>

</head>

<body>

<header>

<div class="bx bx-menu" id="menu-icon"></div>

<ul class="navbar">

<a href="index2.php">Выход</a>

</ul>

</header>

<?php

// Подключение к базе данных

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "deliverypizza";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect\_error) {

die("Ошибка подключения: " . $conn->connect\_error);

}

// Обработка запроса на удаление клиента

if (isset($\_GET['delete'])) {

$pizzaId = $\_GET['delete'];

$sql = "DELETE FROM pizza WHERE ID\_Pizza = $pizzaId";

$conn->query($sql);

}

// Обработка запроса на добавление клиента

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

$pizzaId = $\_POST['ID\_Pizza'];

$pizzaName = $\_POST['Name\_pizza'];

$pizzaPrice = $\_POST['Price\_pizza'];

$pizzaSize = $\_POST['Size\_ID'];

$pizzaWeight = $\_POST['Weight'];

if ($pizzaSize ==33){

$pizzaSize = 3;

}

elseif($pizzaSize == 25){

$pizzaSize = 2;

}

elseif($pizzaSize == 15){

$pizzaSize = 1;

};

$sql = "INSERT INTO pizza (ID\_Pizza, Name\_pizza,Price\_pizza, Size\_ID,Weight) VALUES ( '$pizzaId','$pizzaName', '$pizzaPrice', '$pizzaSize', '$pizzaWeight')";

$conn->query($sql);

}

// Обработка запроса на редактирование клиента

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

$pizzaId = $\_POST['ID\_Pizza'];

$pizzaName = $\_POST['Name\_pizza'];

$pizzaPrice = $\_POST['Price\_pizza'];

$pizzaSize = $\_POST['Size\_ID'];

$pizzaWeight = $\_POST['Weight'];

$sql = "UPDATE pizza SET Name\_pizza = '$pizzaName', Price\_pizza = '$pizzaPrice', Size\_ID = '$pizzaSize', Weight = '$pizzaWeight' WHERE ID\_Pizza = $pizzaId";

$conn->query($sql);

}

// Получение списка клиентов из базы данных

$sql = "SELECT \* FROM pizza join size\_pizza on pizza.Size\_ID = size\_pizza.ID\_size";

$result = $conn->query($sql);

$pizzas = $result->fetch\_all(MYSQLI\_ASSOC);

$conn->close();

?>

<html>

<head>

<!-- Подключение стилей Bootstrap -->

<link rel="stylesheet" href="2.css">

</head>

<body>

<div class="container">

<h2>Таблица: Пицца</h2>

<!-- Форма добавления/редактирования клиента -->

<form class="form-style-8" method="POST" action="">

<div class="form-row">

<div class="col-md-3 mb-3">

<input type="text" name="ID\_Pizza" class="form-control" placeholder="Введите id" required>

</div>

<div class="col-md-3 mb-3">

<input type="text" name="Name\_pizza" class="form-control" placeholder="Введите название пиццы" required>

</div>

<div class="col-md-3 mb-3">

<input type="text" name="Price\_pizza" class="form-control" placeholder="Введите цену пиццы" required>

</div>

<div class="col-md-3 mb-3">

<input type="text" name="Size\_ID" class="form-control" placeholder="Введите размер пиццы" required>

</div>

<div class="col-md-3 mb-3">

<input type="text" name="Weight" class="form-control" placeholder="Введите вес пиццы" required>

</div>

<div class="col-md-3">

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

<button type="submit" name="save" class="btn btn-primary">Сохранить</button>

</div>

</div>

</form>

<!-- Список клиентов -->

<table border='1'>

<tr>

<th>Название</th>

<th>Цена пиццы</th>

<th>Размер</th>

<th>Вес</th>

<th></th>

<th></th>

</tr>

<ul class="list-group">

<?php foreach ($pizzas as $client): ?>

<div class="row">

<div class="col-md-3">

<td> <strong><?php echo $client['Name\_pizza']; ?></strong> </td>

</div>

<div class="col-md-3">

<td> <?php echo $client['Price\_pizza']; ?></td>

</div>

<div class="col-md-3">

<td> <?php echo $client['size']; ?></td>

</div>

<div class="col-md-3">

<td> <?php echo $client['Weight']; ?></td>

</div>

<div class="col-md-3">

<td> <a href="?delete=<?php echo $client['ID\_Pizza']; ?>" class="btn btn-danger">Удалить</a></td>

<td> <a href="#" onclick="editClient('<?php echo $client['ID\_Pizza']; ?>', '<?php echo $client['Name\_pizza']; ?>', '<?php echo $client['Price\_pizza']; ?>', '<?php echo $client['size']; ?>', '<?php echo $client['Weight']; ?>')" class="btn btn-primary">Редактировать</a></td>

</div>

</tr>

</div>

<?php endforeach; ?>

</ul>

</div>

<script>

function editClient(id, Name\_pizza, Price\_pizza, Size\_ID, Weight) {

document.querySelector('input[name="ID\_Pizza"]').value = id;

document.querySelector('input[name="Name\_pizza"]').value = Name\_pizza;

document.querySelector('input[name="Price\_pizza"]').value = Price\_pizza;

document.querySelector('input[name="Size\_ID"]').value = Size\_ID;

document.querySelector('input[name="Weight"]').value = Weight;

}

</script>

<!-- Подключение скриптов Bootstrap -->

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script>

</body>

</html>

**Файл «table2.php»**

<!DOCTYPE html>

<html>

<head>

<title>Клиент</title>

</head>

<body>

<header>

<div class="bx bx-menu" id="menu-icon"></div>

<ul class="navbar">

<a href="index2.php">Выход</a>

</ul>

</header>

<?php

// Подключение к базе данных

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "deliverypizza";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect\_error) {

die("Ошибка подключения: " . $conn->connect\_error);

}

if (isset($\_GET['delete'])) {

$klientId = $\_GET['delete'];

$sql = "DELETE FROM klient WHERE ID\_klient = $klientId";

$conn->query($sql);

}

// Обработка запроса на добавление клиента

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

$klientId = $\_POST['ID\_klient'];

$klientSurame = $\_POST['Surname'];

$klientName = $\_POST['Name'];

$klientAddress = $\_POST['Address'];

$klientEmail = $\_POST['email'];

$klientNumber = $\_POST['Number'];

$klientProgram = $\_POST['program\_loyalty\_ID'];

$sql = "INSERT INTO klient (ID\_klient, Surname, Name, Address, email, Number, program\_loyalty\_ID) VALUES ( '$klientId','$klientSurame', '$klientName', '$klientAddress', '$klientEmail','$klientNumber','$klientProgram')";

$conn->query($sql);

}

// Обработка запроса на редактирование клиента

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

$klientId = $\_POST['ID\_klient'];

$klientSurame = $\_POST['Surname'];

$klientName = $\_POST['Name'];

$klientAddress = $\_POST['Address'];

$klientEmail = $\_POST['email'];

$klientNumber = $\_POST['Number'];

$klientProgram = $\_POST['program\_loyalty\_ID'];

$sql = "UPDATE klient SET Surname = '$klientSurame', Name = '$klientName', Address = '$klientAddress', email = '$klientEmail', Number = '$klientNumber', program\_loyalty\_ID = '$klientProgram' WHERE ID\_klient = $klientId ";

$conn->query($sql);

}

// Получение списка клиентов из базы данных

$sql = "SELECT \* FROM klient";

$result = $conn->query($sql);

$pizzas = $result->fetch\_all(MYSQLI\_ASSOC);

$conn->close();

?>

<html>

<head>

<!-- Подключение стилей Bootstrap -->

<link rel="stylesheet" href="2.css">

</head>

<body>

<div class="container">

<h2>Таблица:Клиенты</h2>

<!-- Форма добавления/редактирования клиента -->

<form class="form-style-8" method="POST" action="">

<div class="form-row">

<div class="col-md-3 mb-3">

<input type="text" name="ID\_klient" class="form-control" placeholder="Введите id" required>

</div>

<div class="col-md-3 mb-3">

<input type="text" name="Surname" class="form-control" placeholder="Введите Фамилию" required>

</div>

<div class="col-md-3 mb-3">

<input type="text" name="Name" class="form-control" placeholder="Введите Имя" required>

</div>

<div class="col-md-3 mb-3">

<input type="text" name="Address" class="form-control" placeholder="Введите адрес" required>

</div>

<div class="col-md-3 mb-3">

<input type="text" name="email" class="form-control" placeholder="Введите email" required>

</div>

<div class="col-md-3 mb-3">

<input type="text" name="Number" class="form-control" placeholder="Введите номер телефона" required>

</div>

<div class="col-md-3 mb-3">

<input type="text" name="program\_loyalty\_ID" class="form-control" placeholder="Введите программу лояльности " required>

</div>

<div class="col-md-3">

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

<button type="submit" name="save" class="btn btn-primary">Сохранить</button>

</div>

</div>

</form>

<!-- Список клиентов -->

<table border='1'>

<tr>

<th>Фамилия</th>

<th>Имя</th>

<th>Адрес</th>

<th>email</th>

<th>Номер телефона</th>

<th>Программа лояльности</th>

<th></th>

<th></th>

</tr>

<ul class="list-group">

<?php foreach ($pizzas as $client): ?>

<div class="row">

<div class="col-md-3">

<td> <strong><?php echo $client['Surname']; ?></strong> </td>

</div>

<div class="col-md-3">

<td> <?php echo $client['Name']; ?></td>

</div>

<div class="col-md-3">

<td> <?php echo $client['Address']; ?></td>

</div>

<div class="col-md-3">

<td> <?php echo $client['email']; ?></td>

</div>

<div class="col-md-3">

<td> <?php echo $client['Number']; ?></td>

</div>

<div class="col-md-3">

<td> <?php echo $client['program\_loyalty\_ID']; ?></td>

</div>

<div class="col-md-3">

<td> <a href="?delete=<?php echo $client['ID\_klient']; ?>" class="btn btn-danger">Удалить</a></td>

<td> <a href="#" onclick="editClient('<?php echo $client['ID\_klient']; ?>', '<?php echo $client['Surname']; ?>', '<?php echo $client['Name']; ?>', '<?php echo $client['Address']; ?>', '<?php echo $client['email']; ?>', '<?php echo $client['Number']; ?>', '<?php echo $client['program\_loyalty\_ID']; ?>')" class="btn btn-primary">Редактировать</a></td>

</div>

</tr>

</div>

<?php endforeach; ?>

</ul>

</div>

<script>

function editClient(id, Surname, Name, Address, email, Number, program\_loyalty\_ID) {

document.querySelector('input[name="ID\_klient"]').value = id;

document.querySelector('input[name="Surname"]').value = Surname;

document.querySelector('input[name="Name"]').value = Name;

document.querySelector('input[name="Address"]').value = Address;

document.querySelector('input[name="email"]').value = email;

document.querySelector('input[name="Number"]').value = Number;

document.querySelector('input[name="program\_loyalty\_ID"]').value = program\_loyalty\_ID;

}

</script>

<!-- Подключение скриптов Bootstrap -->

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script>

</body>

</html>

**Файл «1.css»**

@import url(https://fonts.googleapis.com/css?family=Roboto:400,500,700,300,100);

body {

background-color: #99FFCC;

font-family: "Roboto", helvetica, arial, sans-serif;

font-size: 16px;

font-weight: 400;

text-rendering: optimizeLegibility;

}

div.table-title {

display: block;

margin: auto;

max-width: 600px;

padding:5px;

width: 100%;

}

.table-title h3 {

color: #fafafa;

font-size: 30px;

font-weight: 400;

font-style:normal;

font-family: "Roboto", helvetica, arial, sans-serif;

text-shadow: -1px -1px 1px rgba(0, 0, 0, 0.1);

text-transform:uppercase;

}

/\*\*\* Table Styles \*\*/

.table-fill {

background: white;

border-radius:3px;

border-collapse: collapse;

height: 320px;

margin: auto;

max-width: 600px;

padding:5px;

width: 100%;

box-shadow: 0 5px 10px rgba(0, 0, 0, 0.1);

animation: float 5s infinite;

}

th {

color:#D5DDE5;;

background:#1b1e24;

border-bottom:4px solid #9ea7af;

border-right: 1px solid #343a45;

font-size:23px;

font-weight: 100;

padding:24px;

text-align:left;

text-shadow: 0 1px 1px rgba(0, 0, 0, 0.1);

vertical-align:middle;

}

th:first-child {

border-top-left-radius:3px;

}

th:last-child {

border-top-right-radius:3px;

border-right:none;

}

tr {

border-top: 1px solid #C1C3D1;

border-bottom-: 1px solid #C1C3D1;

color:#666B85;

font-size:16px;

font-weight:normal;

text-shadow: 0 1px 1px rgba(256, 256, 256, 0.1);

}

tr:hover td {

background:#4E5066;

color:#FFFFFF;

border-top: 1px solid #22262e;

}

tr:first-child {

border-top:none;

}

tr:last-child {

border-bottom:none;

}

tr:nth-child(odd) td {

background:#EBEBEB;

}

tr:nth-child(odd):hover td {

background:#4E5066;

}

tr:last-child td:first-child {

border-bottom-left-radius:3px;

}

tr:last-child td:last-child {

border-bottom-right-radius:3px;

}

td {

background:#FFFFFF;

padding:20px;

text-align:left;

vertical-align:middle;

font-weight:300;

font-size:18px;

text-shadow: -1px -1px 1px rgba(0, 0, 0, 0.1);

border-right: 1px solid #C1C3D1;

}

td:last-child {

border-right: 0px;

}

th.text-left {

text-align: left;

}

th.text-center {

text-align: center;

}

th.text-right {

text-align: right;

}

td.text-left {

text-align: left;

}

td.text-center {

text-align: center;

}

td.text-right {

text-align: right;

}

ul {

list-style-type: "🛒";

}

ul.inside {

list-style-position: inside;

}

.flow {

max-width: 40rem;

display: grid;

grid-template-columns: repeat(3, 1fr);

gap: 1rem;

}

.list-container {

margin-top: 0;

}

ul li {

background: #cce5ff;

color: black;

margin: 5px;

}

**Файл «2.css»**

@import url(https://fonts.googleapis.com/css?family=Roboto:400,500,700,300,100);

body {

background-color: #99FFCC;

font-family: "Roboto", helvetica, arial, sans-serif;

font-size: 16px;

font-weight: 400;

text-rendering: optimizeLegibility;

}

div.table-title {

display: block;

margin: auto;

max-width: 600px;

padding:5px;

width: 100%;

}

.table-title h3 {

color: #fafafa;

font-size: 30px;

font-weight: 400;

font-style:normal;

font-family: "Roboto", helvetica, arial, sans-serif;

text-shadow: -1px -1px 1px rgba(0, 0, 0, 0.1);

text-transform:uppercase;

}

/\*\*\* Table Styles \*\*/

.table-fill {

background: white;

border-radius:3px;

border-collapse: collapse;

height: 320px;

margin: auto;

max-width: 600px;

padding:5px;

width: 100%;

box-shadow: 0 5px 10px rgba(0, 0, 0, 0.1);

animation: float 5s infinite;

}

th {

color:#D5DDE5;;

background:#1b1e24;

border-bottom:4px solid #9ea7af;

border-right: 1px solid #343a45;

font-size:23px;

font-weight: 100;

padding:24px;

text-align:left;

text-shadow: 0 1px 1px rgba(0, 0, 0, 0.1);

vertical-align:middle;

}

th:first-child {

border-top-left-radius:3px;

}

th:last-child {

border-top-right-radius:3px;

border-right:none;

}

tr {

border-top: 1px solid #C1C3D1;

border-bottom-: 1px solid #C1C3D1;

color:#666B85;

font-size:16px;

font-weight:normal;

text-shadow: 0 1px 1px rgba(256, 256, 256, 0.1);

}

tr:hover td {

background:#4E5066;

color:#FFFFFF;

border-top: 1px solid #22262e;

}

tr:first-child {

border-top:none;

}

tr:last-child {

border-bottom:none;

}

tr:nth-child(odd) td {

background:#EBEBEB;

}

tr:nth-child(odd):hover td {

background:#4E5066;

}

tr:last-child td:first-child {

border-bottom-left-radius:3px;

}

tr:last-child td:last-child {

border-bottom-right-radius:3px;

}

td {

background:#FFFFFF;

padding:20px;

text-align:left;

vertical-align:middle;

font-weight:300;

font-size:18px;

text-shadow: -1px -1px 1px rgba(0, 0, 0, 0.1);

border-right: 1px solid #C1C3D1;

}

td:last-child {

border-right: 0px;

}

th.text-left {

text-align: left;

}

th.text-center {

text-align: center;

}

th.text-right {

text-align: right;

}

td.text-left {

text-align: left;

}

td.text-center {

text-align: center;

}

td.text-right {

text-align: right;

}

.form-style-8{

font-family: 'Open Sans Condensed', arial, sans;

width: 500px;

padding: 30px;

background: #FFFFFF;

margin: 50px auto;

box-shadow: 0px 0px 15px rgba(0, 0, 0, 0.22);

-moz-box-shadow: 0px 0px 15px rgba(0, 0, 0, 0.22);

-webkit-box-shadow: 0px 0px 15px rgba(0, 0, 0, 0.22);

}

.form-style-8 h2{

background: #4D4D4D;

text-transform: uppercase;

font-family: 'Open Sans Condensed', sans-serif;

color: #797979;

font-size: 18px;

font-weight: 100;

padding: 20px;

margin: -30px -30px 30px -30px;

}

.form-style-8 input[type="text"],

.form-style-8 input[type="date"],

.form-style-8 input[type="datetime"],

.form-style-8 input[type="email"],

.form-style-8 input[type="number"],

.form-style-8 input[type="search"],

.form-style-8 input[type="time"],

.form-style-8 input[type="url"],

.form-style-8 input[type="password"],

.form-style-8 textarea,

.form-style-8 select

{

box-sizing: border-box;

-webkit-box-sizing: border-box;

-moz-box-sizing: border-box;

outline: none;

display: block;

width: 100%;

padding: 7px;

border: none;

border-bottom: 1px solid #ddd;

background: transparent;

margin-bottom: 10px;

font: 16px Arial, Helvetica, sans-serif;

height: 45px;

}

.form-style-8 textarea{

resize:none;

overflow: hidden;

}

.form-style-8 input[type="button"],

.form-style-8 input[type="submit"]{

-moz-box-shadow: inset 0px 1px 0px 0px #45D6D6;

-webkit-box-shadow: inset 0px 1px 0px 0px #45D6D6;

box-shadow: inset 0px 1px 0px 0px #45D6D6;

background-color: #2CBBBB;

border: 1px solid #27A0A0;

display: inline-block;

cursor: pointer;

color: #FFFFFF;

font-family: 'Open Sans Condensed', sans-serif;

font-size: 14px;

padding: 8px 18px;

text-decoration: none;

text-transform: uppercase;

}

.form-style-8 input[type="button"]:hover,

.form-style-8 input[type="submit"]:hover {

background:linear-gradient(to bottom, #34CACA 5%, #30C9C9 100%);

background-color:#34CACA;

}

**Файл «3.css»**

@import url("https://fonts.googleapis.com/css2?family=Poppins&display=swap");

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: "Poppins", sans-serif;

}

body {

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

ul li {

width: 200px;

height: 35px;

display: flex;

align-items: center;

margin: 1.5em;

cursor: pointer;

padding: 1em;

background: rgb(43, 43, 43);

position: relative;

color: white;

border-radius: 5px;

}

ul li::before,

ul li::after {

content: "";

position: absolute;

z-index: -1;

border-radius: 5px;

width: 105%;

transition: all 0.4s;

}

ul li::before {

left: 0%;

height: 130%;

background: linear-gradient(to right, #021B79, #0575E6);

}

ul li::after {

left: -10%;

height: 120%;

background: #ffffff56;

backdrop-filter: blur(10px);

box-shadow: 0 0 20px rgba(0, 0, 0, 0.164);

}

ul li:hover::before {

transform: translateX(-2.5%);

}

ul li:hover::after {

transform: translateX(15%);

}

**Файл «style\_main.css»**

/\* Общие стили \*/

body {

margin: 0;

padding: 15px;

font-family: Arial, sans-serif;

background-image: url('index.jpg');

background-size: cover;

background-position: center;

background-repeat: no-repeat;

height: 90vh;

display: flex;

flex-direction: column;

justify-content: flex-start;

align-items: center;

position: relative; /\* Добавлено для позиционирования текста \*/

}

/\* Стили для текста на фоновом изображении \*/

.welcome-text {

position: absolute;

top: 60%; /\* Расположение текста по вертикали \*/

left: 70%; /\* Расположение текста по горизонтали \*/

transform: translate(-220%, -50%); /\* Центрирование текста \*/

color: white;

font-size: 70px;

font-weight: light;

font-family: 'Montserrat', sans-serif; /\* Добавлен шрифт Montserrat \*/

text-shadow: 3px 10px 30px rgba(0, 0, 0, 0.5);

z-index: 1; /\* Помещаем текст поверх фонового изображения \*/

}

/\* Стили для шапки \*/

header {

position: fixed;

width: 100%;

top: 0;

right: 0;

z-index: 1000;

display: flex;

align-items: center;

justify-content: space-between;

background: var(--bg-color);

padding: 18px 100px;

box-shadow: 0 0.5rem 50rem rgba(0, 0, 0, 0.1);

}

.navbar {

display: flex;

}

.navbar a {

font-size: 1rem;

padding: 10px 20px;

color: var(--text-color);

font-weight: 500;

}

.navbar a:hover {

color: var(--main-color);

}

.left-header h1 {

margin: 0 40px;

}

/\* Стили для модального окна \*/

.modal {

display: none;

position: fixed;

z-index: 3; /\* Помещаем модальное окно поверх всего контента \*/

left: 0;

top: 0;

width: 100%;

height: 100%;

overflow: auto;

background-color: rgba(0, 0, 0, 0.4);

justify-content: center;

align-items: center;

}

.modal1 {

display: none;

position: fixed;

z-index: 3; /\* Помещаем модальное окно поверх всего контента \*/

left: 0;

top: 0;

width: 100%;

height: 100%;

overflow: auto;

background-color: rgba(0, 0, 0, 0.4);

justify-content: center;

align-items: center;

}

.modal2 {

display: none;

position: fixed;

z-index: 3; /\* Помещаем модальное окно поверх всего контента \*/

left: 0;

top: 0;

width: 100%;

height: 100%;

overflow: auto;

background-color: rgba(0, 0, 0, 0.4);

justify-content: center;

align-items: center;

}

.modal-content {

width: 300px;

padding: 40px;

background-color: rgba(255, 255, 255, 0.8);

border-radius: 10px;

box-shadow: 0 0 20px rgba(0, 0, 0, 0.2);

text-align: center;

}

.modal-content1 {

width: 300px;

padding: 40px;

background-color: rgba(255, 255, 255, 0.8);

border-radius: 10px;

box-shadow: 0 0 20px rgba(0, 0, 0, 0.2);

text-align: center;

}

.modal-content2 {

width: 300px;

padding: 40px;

background-color: rgba(255, 255, 255, 0.8);

border-radius: 10px;

box-shadow: 0 0 20px rgba(0, 0, 0, 0.2);

text-align: center;

}

.modal-content p {

font-size: 18px;

margin-bottom: 20px;

}

.modal-content1 p {

font-size: 18px;

margin-bottom: 20px;

}

.modal-content2 p {

font-size: 18px;

margin-bottom: 20px;

}

.modal-content h1 {

margin-bottom: 20px;

}

.modal-content h1 {

margin-bottom: 20px;

}

.modal-content form {

margin-top: 20px;

}

.modal-content1 form {

margin-top: 20px;

}

.modal-content2 form {

margin-top: 20px;

}

.modal-content input[type="text"],

.modal-content2 input[type="text"],

.modal-content input[type="password"] {

width: 90%;

padding: 10px;

margin-bottom: 8px;

border: 1px solid #505050;

border-radius: 3px;

}

.modal-content2 input[type="password"] {

width: 90%;

padding: 10px;

margin-bottom: 8px;

border: 1px solid #505050;

border-radius: 3px;

}

/\* Стили для кнопки \*/

.modal-content input[type="submit"] {

width: 100%;

background-color: #007bff;

color: white;

border: none;

padding: 10px;

cursor: pointer;

border-radius: 5px;

}

.modal-content2 input[type="submit"] {

width: 100%;

background-color: #007bff;

color: white;

border: none;

padding: 10px;

cursor: pointer;

border-radius: 5px;

}

.close {

color: #aaa;

float: right;

font-size: 28px;

font-weight: bold;

cursor: pointer;

}

.close1 {

color: #aaa;

float: right;

font-size: 28px;

font-weight: bold;

cursor: pointer;

}

.close2 {

color: #aaa;

float: right;

font-size: 28px;

font-weight: bold;

cursor: pointer;

}

.home {

width: 100%;

min-height: 100vh;

display: grid;

grid-template-columns: repeat(2, 1fr);

align-items: center;

gap: 1.5rem;

}

/\* Фото чувака \*/

.home-img img {

width: 100%;

}

/\* Доставка пиццы \*/

.home-text h1 {

font-size: var(--big-font);

position: absolute;

top: 23%; /\* Расположение текста по вертикали \*/

left: 77%; /\* Расположение текста по горизонтали \*/

transform: translate(-220%, -50%); /\* Центрирование текста \*/

color: white;

font-size: 100px;

font-weight: light;

font-family: 'NEW YORK', sans-serif; /\* Добавлен шрифт Montserrat \*/

text-shadow: 3px 10px 30px rgba(0, 0, 0, 0.5);

z-index: 1; /\* Помещаем текст поверх фонового изображения \*/

color: var(--main-color);

}

/\* Меню по середине \*/

.heading {

text-align: center;

}

.heading span {

color: var(--main-color);

font-weight: 500;

}

.heading h2 {

font-size: var(--h2-font);

}

.menu-container {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(220px, auto));

gap: 1.5rem;

align-items: center;

}

/\*фото\*/

.box {

position: relative;

margin-top: 2rem;

height: auto;

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

border-radius: 0.5rem;

box-shadow: 0 2px 4px rgb(4 64 54 / 10%);

padding: 10px;

}

.box-img {

width: 200px;

height: 200px;

}

.box-img img {

width: 100%;

height: 100%;

object-fit: contain;

object-position: center;

}

.box h2 {

font-size: 1.2rem;

}

.box h3 {

font-size: 1rem;

font-weight: 400;

margin: 4px 0 10px;

}

/\*контакты\*/

.address {

display: flex;

flex-direction: column;

}

.address i {

margin-bottom: 0.5rem;

font-size: 1rem;

}

.address span {

margin-left: 1rem;

}

.copyright {

padding: 20px;

text-align: center;

}