Министерство образования и науки Республики Башкортостан

Государственное автономное профессиональное образовательное учреждение

Уфимский колледж статистики, информатики и вычислительной техники

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Заместитель директора  по учебной работе  З.З. Курмашева  «» 2024 г. |

ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ И РАЗРАБОТКА ПРОГРАММЫ

ДЛЯ ОФОРМЛЕНИЯ ЗАКАЗОВ НА ПРИОБРЕТЕНИЕ ЦВЕТОВ

Пояснительная записка к курсовому проекту

МДК 11.01 Технологияразработки и защиты баз данных

|  |  |
| --- | --- |
|  | Руководитель проекта  Р.Ф. Каримова  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |
|  | Студент гр. 21П-1  А.И. Фазуллина  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |

2024

Министерство образования и науки Республики Башкортостан

Государственное автономноепрофессиональное образовательное учреждение

Уфимский колледж статистики, информатики и вычислительной техники

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Заместитель директора  по учебной работе  З.З. Курмашева  «» 2024 г. |

ЗАДАНИЕ

на курсовой проект студенту дневного отделения, группы 21П-1, специальности 09.02.07Информационные системы и программирование

Фамилия, имя, отчество: Фазуллина Аделина Ильдаровна

Тема курсового проекта: «Проектирование базыи разработка программы данных для оформления заказов на приобретение цветов».

Текст задания:

при выполнении курсового проекта должны быть решены следующие задачи:

1. спроектирована структура базы данных;
2. разработана структура программы;
3. разработано приложение.

В результате выполнения курсового проекта должны быть представлены:

1. пояснительная записка, состоящая из следующих разделов:

Введение

1 Постановка задачи

2 Экспериментальный раздел

Заключение

Приложения

Список использованных источников

1. электронный носитель, содержащий разработанный программный продукт;
2. презентация курсового проекта в электронном виде.

Список рекомендуемых источников:

1. Култыгин, О. П. Култыгин, О. П. Администрирование баз данных. СУБД MS SQL Server [Текст] : учеб. пособ. / О. П. Култыгин. - М.: МФПА, 2012. - 232 с.
2. Фуфаев, Э.В. Базы данных [Текст]: учеб. пособ. для студ. учрежд. сред. проф. образования / Э.В. Фуфаев, Д.Э. Фуфаев. - 6-е изд., стер. - М.: Издательский центр «Академия», 2012.- 320 с.- (Среднее профессиональное образование)
3. Википедия [Электронный ресурс] // Свободная энциклопедия. – Режим доступа: <http://ru.wikipedia.org/wiki/>, свободный

Задание к выполнению получил «31» января 2024 г.

Студент Фазуллина Аделина Ильдаровна

Срок окончания «31» мая 2024 г.

Руководитель курсового проекта Р.Ф. Каримова

Задание рассмотрено на заседании цикловой комиссии информатики

«11» января 2024 г.

Председатель цикловой комиссии информатики О.В.Фатхулова

Министерство образования и науки Республики Башкортостан

Государственное бюджетное профессиональное образовательное учреждение

Уфимский колледж статистики, информатики и вычислительной техники

ЗАКЛЮЧЕНИЕ

на курсовой проект

Студент Фазуллина Аделина Ильдаровна

Группа 21П-1

Специальность 09.02.07Информационные системы и программирование

Тема Проектирование базы данных и разработка программы для оформления заказов на приобретение цветов

Объем курсового проекта:

количество листов пояснительной записки

количество листов графической части

Заключение о степени соответствия заданию на курсовое проектирование

Характеристика качеств, проявленных студентом при работе над проектом: самостоятельность, дисциплинированность, умение планировать работу и пользоваться литературным материалом и т.д.

Положительные стороны курсового проекта

Недостатки курсового проекта

Характеристика общетехнической и специальной подготовки студента

Заключение и предлагаемая оценка за курсовой проект

Руководитель курсового проектаКаримова Резида Флюновна

«\_\_\_\_» 2024 г.

Подпись

АННОТАЦИЯ

Пояснительная записка к курсовому проекту содержит постановку и программу решения задачи «Проектирование базы данных для оформления заказов на приобретение цветов».

ПрограммаNILSON\_FLOWERS.exe написана на языках html, php, javascript, css в среде программирования VisualStudioCode 2022 с использованием системы управления базой данных phpMyAdminпредназначена для работы в различных браузерах, отлажена на данных контрольного примера.

СОДЕРЖАНИЕ

|  |  |
| --- | --- |
|  | лист |
| Введение | 4 |
| 1 Постановка задачи | 6 |
| 1.1 Описание предметной области | 6 |
| 1.2 Описание входной информации | 8 |
| 1.3 Описание выходной информации | 8 |
| 1.4 Концептуальное моделирование | 9 |
| 1.5 Логическое моделирование | 10 |
| 1.6 Описание структуры базы данных | 11 |
| 1.7 Контрольный пример | 13 |
| 1.8 Общие требования к программному продукту | 13 |
| 2 Экспериментальный раздел | 15 |
| 2.1 Описание программы | 15 |
| 2.2 Руководство пользователя | 16 |
| Заключение | 28 |
| Приложения | 30 |
| Список источников | 80 |

ВВЕДЕНИЕ

Цветочная индустрия сосредоточена на производстве, распространении и продаже цветов для удовольствия человека. Отрасль продолжает диверсифицироваться от производства срезанных цветов до производства и продажи растений и цветочных композиций в самых разных формах. Продажа цветов распространена во многих странах. Многие ухаживают за цветами как в саду, так и дома. Девушки особенно любят срезанные цветы в качестве подарка или знака внимания. Продажа цветов позволяет людям заниматься любимим хобби или украшать и улучшать среду обитания человека.

Цветочный магазин направлен на создание инновационной онлайн-платформы, которая предложит потребителям новый уровень комфорта при покупке цветочных изделий. Используя передовые информационные технологии, проект стремится обеспечить интуитивно понятный интерфейс и широкий спектр услуг, делая выбор и заказ цветов простым и приятным процессом.

Интерактивность и доступность — ключевые характеристики магазина, позволяющие клиентам в любое время и из любого места ознакомиться с ассортиментом и оформить заказ. Это не только экономит время, но и предоставляет возможность тщательного выбора из множества вариантов цветов и букетов.

Цветочный магазин станет незаменимым помощником для тех, кто желает выразить свои чувства через красоту цветов, будь то романтический жест или декоративное украшение для особых событий. Онлайн-формат гарантирует оперативность и качество обслуживания, позволяя каждому клиенту найти идеальное цветочное решение.

С увеличением популярности онлайн торговли, цветочный магазин отвечает современным требованиям рынка, предлагая удобную платформу для быстрого и легкого доступа к широкому спектру цветочных продуктов и услуг. Онлайн-магазин становится местом, где каждый может найти что-то особенное для себя, создавая незабываемые моменты с помощью свежих и ароматных букетов.

Цель курсового проекта — разработка веб-приложения для упрощения взаимодействия между онлайн цветочным магазином и клиентом путём создания каталога товара и реализации механизма заказов.

Задачами курсового проекта являются:

* описать предметную область;
* разработать структуру базы данных;
* разработать приложение;
* провести тестирование приложения.

1. Постановка задачи
   1. Описание предметной области

Требуется разработать информационную систему для автоматизации учета продажи цветов на веб сайте, хранения информации о наличие цветов, данных клиентов, которая будет использоваться компанией, занимающаяся продажей цветов.

Данная информационная система предполагает наличие трех групп пользователей: незарегистрированный пользователь, зарегистрированный пользователь и администратор.

Незарегистрированный пользователь имеет возможность:

* просматривать товары;
* зарегистрироваться.

Зарегистрированный пользователь имеет возможность:

* просматривать товары;
* авторизоваться.

Авторизированный пользователь имеет возможность:

* просматривать товары;
* добавлять товары в корзину;
* удалять товары из корзины;
* оформлять покупки.

Администратор имеет следующие возможности:

* просматривать товары;
* добавлять товары в магазин;
* удалять товары из магазина;
* изменять товары в магазине.

В базе данных должны храниться следующие справочники:пользователи, статус продукта, роль пользователя, продукт, лист заказов, заказ, производитель, категория.

Для создания продукта администратор требуеется ввести наимнование, цену, производителя, категорию и статус, идентификатор продукта назначается автоматически. В таком случае продукт считается созданным, в зависимости от статуса он будет или не будет отображаться в магазине.

В таблице «Продукт» содержится следующая информация:

* идентификатор продукта (primarykey);
* наименование;
* цена;
* описание;
* изображение;
* идентификатор производителя (foreign key);
* идентификатор категории (foreign key);
* идентификатор статуса (foreign key);
* количество.

Пользователь должен войти в систему, для этого ему нужно зарегистироваться. Клиент авторизуется для того, чтобы делать заказы. Администратор авторизуется для изменения товаров. В таблице «Пользователь» содержится следующая информация:

* идентификатор пользователя (primarykey);
* имя;
* фамилия;
* отчество;
* дата рождения;
* номер телефона;
* эмайл;
* пароль;
* идентификатор роли (foreign key).

После оформления заказа клиентом создается заказ. Таблица «Заказ»:

* идентификатор заказа (primary key);
* идентификатор пользователя (foreign key);
* идентификатор статуса (foreign key).

В заказе содержатся продукты и их количества. Таблица «Лист заказов»:

* идентификатор заказа (foreign key);
* идентификатор продукта (foreign key);
* стоимость;
* количество.

Таблица «Поставщики» представляет поставщиков цветов для магазина. В таблице содержится следующая информация:

* идентификатор поставщика (primary key);
* наименование.

Таблица «Роль» показывает роль каждого пользователя. В таблице содержится следующая информация:

* идентификатор роли (primarykey);
* наименование.

Таблица «Категория» раскрывает категорию каждого товара. В таблице содержится следующая информация:

* идентификатор категории (primarykey);
* наименование.

Таблица «Статус»:

* идентификатор статуса (primarykey);
* наименование.

Входными данными являются данные о пользователях и о цветах. Выходными данными является электронное письмо на почту с заказом.

1.2 Описание входной информации

Входной информацией для выполнения задачи являются справочники: пользователи, товары, категории, поставщики, информация о поставкав и заказы пользователей.

1.3 Описание выходной информации

Выходной информацией является электронное письмо, которое формируется при оформлении заказа и поступает на электронную почту клиента. Описание выходного документа представлено в таблице 1.3.1.

Таблица 1.3.1 – Описание выходного документа

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование документов (шифр) | Переодичность выдачи документов | Кол-во экз. | Куда передаются |
| Email\_mess | При оформлении заказа | 1 | Пользователю на почту |

Шаблон выходного документа представлен в приложении А.

1.4. Концептуальное моделирование

Концептуальная модель базы данных –это некая наглядная диаграмма, нарисованная в принятых обозначениях и подробно показывающая связь между объектами и их характеристиками. Создается концептуальная модель для дальнейшего проектирования базы данных и перевод ее, например, в реляционную базу данных. На концептуальной модели в визуально удобном виде прописываются связи между объектами данных и их характеристиками.

В концептуальной модели есть принятые обозначения элементов. Сущность или объект обозначать прямоугольником, отношения обозначать ромбом, атрибуты объектов, обозначаются овалом. Если сущность связана с отношением, то их связь обозначается прямой линией со стрелкой. Необязательная связь обозначается пунктирной линией. Мощная связь обозначается двойной линией.

Концептуальная модель базы данных представлена в рисунке 1.4.1

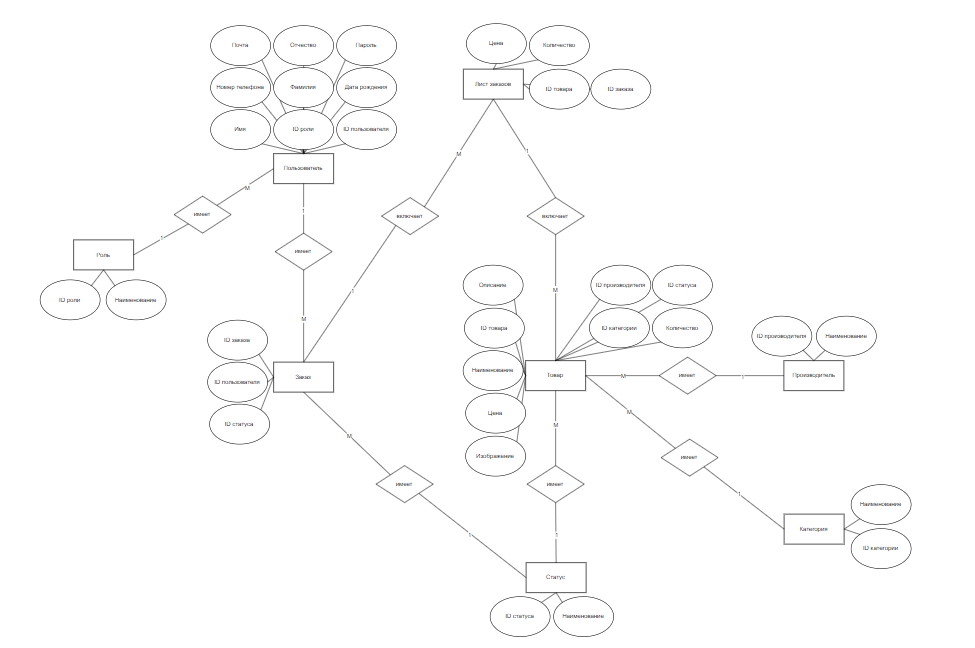


Рисунок 1.4.1 – Концептуальная модель БД

1.5. Логическое моделирование

При логическом моделировании происходит окончательное определение структуры данных, определяются ограничения, накладываемые на эти данные, целью которых является обеспечить целостность данных. Наиболее распространенной моделью данных является реляционная модель. В этой модели данных каждая сущность представляется в виде таблицы.

Логическое моделирование заключается в переходе от концептуальной модели к взаимосвязанным таблицам. Этот переход состоит из следующих шагов:

1. Преобразование сущностей:

- каждая простая сущность становится таблицей.

- каждый атрибут становится столбцом таблицы.

- уникальный идентификатор сущности становится ключом таблицы.

2. Преобразование связи:

- сущности, связанные обязательной связью один кодному, можно объединить в одну таблицу.

- связи типа один к одному возможные и связи типа один ко многим реализуются путем переноса ключевых атрибутов таблиц, соответствующих сущностей, стоящих со стороны один в таблице соответствующих сущностей, стоящих со стороны многие.

- связи типа многие ко многим реализуются при помощи промежуточных таблиц, содержащих ключевые атрибуты связываемых таблиц в качестве внешних ключей.

Схема данных – это структура базы данных, описанная на формальном языке, поддерживаемом СУБД (системой управления базы данных). В реляционных базах данных схема определяет таблицы, поля в каждой таблице и ограничения целостности, такие как первичный и внешний ключи.

Схема данных представлена на рисунке 1.5.1.

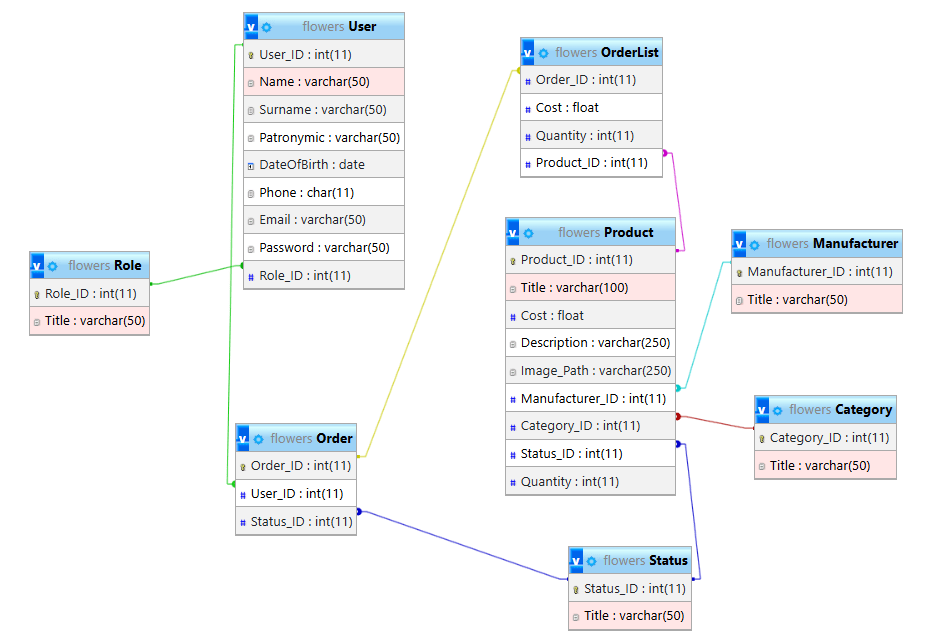


Рисунок 1.5.1 – Схема данных БД

1.6. Описание структуры базы данных

Описание структуры базы данных представлено в таблице 1.6.1.

Таблица 1.6.1 – Описание структуры базы данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описание поля | Тип данных | Размер поля | Тип ключа[[1]](#footnote-2) |
| 1 | 2 | 3 | 4 | 5 |
| User (Пользователь) | | | | |
| User\_ID | IDпользователя | INT | 11 | PK |
| Name | Имя | VARCHAR | 50 |  |
| Surname | Фамилия | VARCHAR | 50 |  |
| Patronymic | Отчество | VARCHAR | 50 |  |
| DateOfBirth | Дата рождения | DATE |  |  |
| Phone | Номер телефона | VARCHAR | 11 |  |
| Email | Почта | VARCHAR | 50 |  |
| Password | Пароль | VARCHAR | 50 |  |
| Role\_ID | ID роли | INT | 11 | FK |
| Status\_ID | ID статуса | INT | 11 | FK |
| Status (Статус) | | | | |
| Status\_ID | ID статуса | INT | 11 | PK |
| Title | Наименование | VARCHAR | 50 |  |
| Role (Роль) | | | | |
| Role\_ID | ID | INT | 11 | PK |
| Title | Наименование | VARCHAR | 50 |  |

Продолжение таблицы 1.6.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| Product (Товар) | | | | |
| Product\_ID | IDтовара | INT | 11 | PK |
| Title | Наименование | VARCHAR | 100 |  |
| Cost | Цена | FLOAT |  |  |
| Description | Описание | VARCHAR | 250 |  |
| Image\_Path | Путь к изображению | VARCHAR | 250 |  |
| Manufacturer\_ID | ID производителя | INT | 11 | FK |
| Category\_ID | ID категории | INT | 11 | FK |
| Status\_ID | ID статуса | INT | 11 | FK |
| Quantity | Количество | INT | 11 |  |
| OrderList (Лист заказов) | | | | |
| Order\_ID | ID заказа | INT | 11 | FK |
| Cost | Цена | FLOAT |  |  |
| Quantity | Количество | INT | 11 |  |
| Product\_ID | IDтовара | INT | 11 | FK |
| Order (Заказ) | | | | |
| Order\_ID | ID заказа | INT | 11 | PK |
| User\_ID | ID пользователя | INT | 11 | FK |
| Status\_ID | ID статуса | INT | 11 | FK |
| Manufacturer (Производитель) | | | | |
| Manufacturer\_ID | ID производителя | INT | 11 | PK |
| Title | Наименование | VARCHAR | 50 |  |
| Category (Категория) | | | | |
| Category\_ID | ID | INT | 11 | PK |

Продолжение таблицы 1.6.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Title | Наименование | VARCHAR | 50 |  |

1.7. Контрольный пример

При построении контрольного примера входные данные и предполагаемые результаты задаются в виде таблиц. Эти результаты в дальнейшем должны совпадать с результатами, полученными при работе программного продукта на соответствующих входных данных. Данные должны быть подобраны таким образом, чтобы на них можно было продемонстрировать работу основных функций задачи, а при получении отчетов однозначно просматривались критерии сортировки, группировки, а также промежуточные и окончательные итоги.

Контрольный пример представлен в таблицах 1.7.1– 1.7.10.

Входные данные контрольного примера

Таблица 1.7.1 - Справочник пользователей

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Email адрес | Пароль | Роль |
| 1 | 2 | 3 | 4 |
| 4 | fluffi@yandex.ru | 1234 | 2 |
| 1 | poli@yandex.ru | poLi11 | 1 |

Таблица 1.7.2 – Данные пользователей для регистраций

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Имя | Фамилия | Отчество | Дата рождения | Телефон |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 4 | Роза | Фламенко | Даниловна | 2000-04-02 | 89176545354 |
| 1 | Полина | Иванова | Витальевна | 2001-12-20 | 89198765454 |

Таблица 1.7.3 - Справочник ролей

|  |  |
| --- | --- |
| ID | Роль |
| 1 | 2 |
| 1 | Администратор |
| 2 | Покупатель |

­Таблица 1.7.4 - Справочник товаров

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Название | Цена | Количество | Производитель | Категория |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 2 | Кенийские розы | 9600 | 6 | 1 | 1 |
| 9 | PinkFlower | 7657 | 7 | 2 | 1 |
| 16 | Гвоздики | 2400 | 10 | 1 | 1 |

Таблица 1.7.4 - Справочник производителей

|  |  |
| --- | --- |
| ID | Производитель |
| 1 | 2 |
| 1 | Англия |
| 2 | Нидерланды |

Таблица 1.7.5 - Справочник категорий

|  |  |
| --- | --- |
| ID | Категория |
| 1 | 2 |
| 1 | Букет |
| 2 | Украшения |

Таблица 1.7.6 – Заказ клиента

|  |  |  |
| --- | --- | --- |
| ID Заказа | Клиент | Статус |
| 1 | 2 | 3 |
| 4 | fluffi@yandex.ru | Оформлен |
| 5 | fluffi@yandex.ru | Оформлен |

Таблица 1.7.7 – Товары заказа

|  |  |  |  |
| --- | --- | --- | --- |
| ID Заказа | Цена | Количество | ID товара |
| 1 | 2 | 3 | 4 |
| 4 | 9060 | 1 | Кенийские розы |
| 4 | 7657 | 1 | PinkFlower |
| 5 | 26440 | 1 | Гвоздики |

Выходные данные контрольного примера

Таблица 1.7.8 – Данные заказа

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID Заказа | Общая сумма | Имя | Фамилия | Телефон |
| 1 | 17257.00 | Роза | Фламенко | 89176545354 |

Таблица 1.7.9 – Товары заказа

|  |  |  |
| --- | --- | --- |
| ID Заказа | Товар | Стоимость |
| 1 | Кенийские розы | 9600 |
| 1 | PinkFlower | 7657 |

Таблица 1.7.10 - Справочник товаров

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Название | Цена | Количество | Производитель | Категория |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 2 | Кенийские розы | 9600 | 6 | 1 | 1 |
| 9 | PinkFlower | 7657 | 5 | 2 | 1 |

Продолжение таблицы 1.7.10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 16 | Гвоздики | 3400 | 4 | 1 | 1 |

1.8 Общие требования к программному продукту

Пользователи должны иметь базовые навыки пользования персональным компьютером.

Минимальные требования к техническому обеспечению программного продукта следующие:

* ОС:  Windows 7 Service Pack1/ Windows 7 64Bit Service Pack1/ Windows 8.1 64Bit / Windows 10 64Bit / Mac OS;
* процессор: 2.40 Ггц (четырехъядерный) / AMDPhenom 9850 (четырехъядерный) @ 2.5 Ггц;
* оперативная память: 512МБ (Win 7/Win8 и выше);
* видеокарта: NVIDIA 9800 GT c 512 МБ видеопамяти/ AMD HD 4870 с 1 Гб видеопамяти (DX 9, 10, 10.1);
* жесткий диск: 10 гигабайт свободного места;
* Microsoft DirectX версия 9.0c;

Функциональные возможности программного продукта:

* приложение должно формировать и отображать выходные данные пользователю;
* в приложении должен быть обеспечен просмотр таблиц базы данных с возможность добавления, редактирования, удаления данных.

Требования к надежности:

* приложение должно обрабатывать ошибочные действия пользователя и сообщать ему об этом;
* приложение должно обеспечивать контроль входной и выходной информации.

Требования к информационной и программной совместимости: обеспечить работу приложения с таблицами СУБД phpMyAdmin.

2 Экспериментальный раздел

2.1 Описание программы

Создана удобная и функциональная платформа для продажи цветов, где пользователи могут просматривать товары, регистрироваться, оформлять заказы, а администраторы управлять ассортиментом.

Основные функции:

1. Просмотр товаров

- Каталог товаров: Пользователи могут просматривать каталог цветов с фотографиями, описаниями и ценами.

2. Регистрация и авторизация

- Регистрация: Новые пользователи могут создать учетную запись, указав свои данные (имя, email, пароль и т.д.).

- Авторизация: Зарегистрированные пользователи могут оформить заказ, используя email и пароль.

- Восстановление пароля: Функция восстановления пароля через email.

3. Оформление заказа

- Корзина: Пользователи могут добавлять товары в корзину, просматривать содержимое корзины.

- Оформление заказа: После авторизации пользователи могут оформить заказ.

4. Административный функционал

- Управление товарами: Администраторы могут добавлять новые товары, редактировать существующие и удалять ненужные.

Технические детали:

- Фронтенд: HTML, CSS, JavaScript.

- Бэкенд: PHP.

- База данных: PhpMyAdminдля хранения данных о пользователях, товарах и заказах.

- Отправка email: PHPMailer для отправки писем (восстановление пароля, уведомления о заказах).

2.2 Протокол тестирования программного продукта

В ходе тестирования программного продукта на корректных и некорректных данных не было обнаружено ошибок, которые влияли бы на работу самого программного продукта и всей системы.

Данный программный продукт удовлетворяет всем предъявленным требованиям, имеет комфортный интерфейс и интуитивно понятный функционал, исключает появления системных ошибок.

В таблицах 2.2.1 -2.2.11 представлены протоколы тестирования.

Таблица 2.2.1 - Протокол тестирования на регистрацию пользователя на корректных данных

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| Поле | Описание |
| 1 | 2 |
| Дата теста | 24.11.2024 |
| Приоритет тестирования | Высокий |
| Наименование тестирования | Проверка добавления пользователя на странице «register.php» с корректными данными |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе корректных данных |
| Этапы тестирования | Ввод корректных данных в текстовые поля;  Нажатие кнопки «Registr». |

Продолжение таблицы 2.2.1

|  |  |
| --- | --- |
| 1 | 2 |
| Тестовые данные | Почта: fluffi@yandex.ru  Пароль: 1234  Имя: Роза  Фамилия: Фламенко  Отчество: Даниловна  Дата: 2000-04-02  Телефон: 89176545354 |
| Ожидаемый результат | Вывод сообщения об успешной регистраций, переход на страницу авторизации |
| Фактический результат | Вывод сообщения «Успешная регистрация», переход на страницу авторизации |

Результат тестирования на регистрацию пользователя представлен на рисунках 2.2.1-2.2.2.

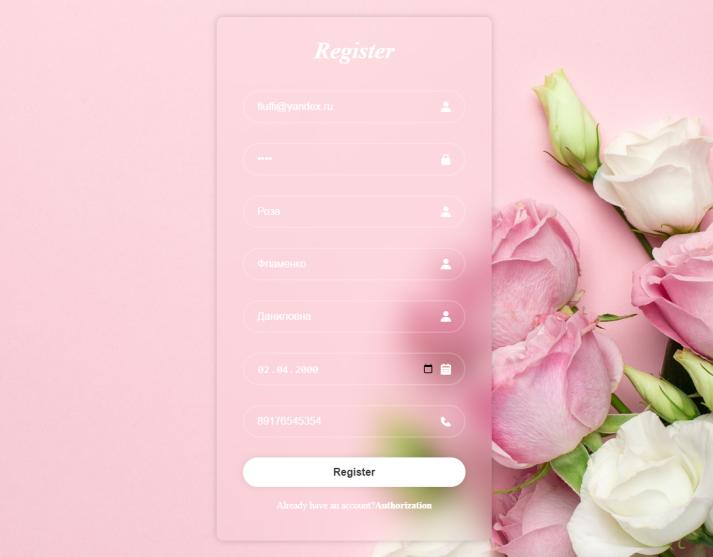


Рисунок 2.2.1 – Регистрация пользователя на корректных данных

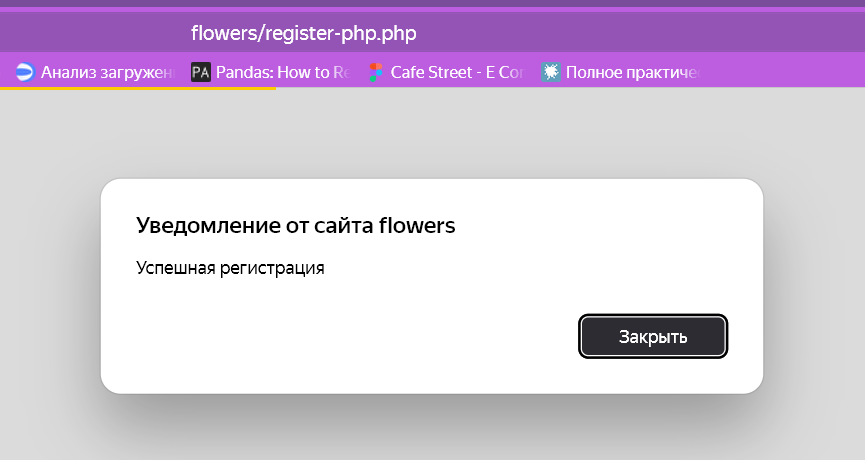


Рисунок 2.2.2 – Сообщение при успешной регистраций

Таблица 2.2.2 - Протокол тестирования на регистрации пользователя на некорректных данных

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| Поле | Описание |
| 1 | 2 |
| Дата теста | 24.11.2024 |
| Приоритет тестирования | Высокий |
| Наименование тестирования | Проверка добавления пользователя на странице «register.php» с некорректными данными |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе некорректных данных |
| Этапы тестирования | Ввод некорректных данных в текстовые поля;  Нажатие кнопки «Registr». |
| Тестовые данные | Почта: poli@yandex.ru  Пароль: poLi11  Имя: Полина  Фамилия: Иванова  Отчество: Витальевна  Дата: 2001-12-20  Телефон: 89198765454 |
| Ожидаемый результат | Вывод сообщения о неверных данных, переход на страницу регистрацию |
| Фактический результат | Вывод сообщения «Пользователь с такой почтой уже существует», переход на страницу регистрацию |

Результат тестирования на регистрацию пользователя на некорректных данных представлен на рисунках 2.2.3 - 2.2.4.

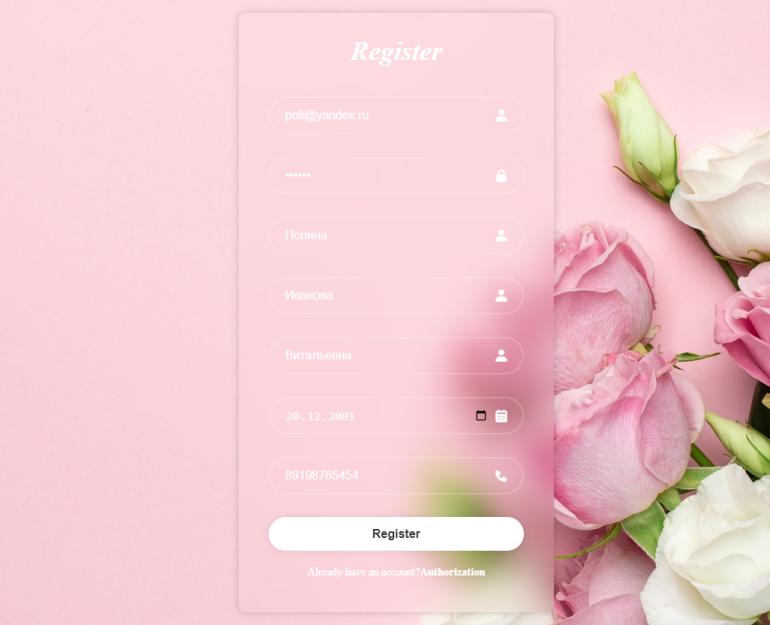


Рисунок 2.2.3 – Регистрация пользователя на некорректных данных

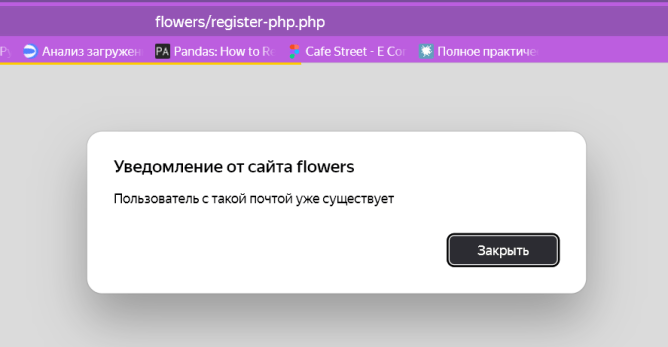


Рисунок 2.2.4 – Сообщение при неверной почте

Таблица 2.2.3 - Протокол тестирования на авторизацию пользователя на корректных данных

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| Поле | Описание |
| 1 | 2 |
| Дата теста | 24.11.2024 |
| Приоритет тестирования | Высокий |
| Наименование тестирования | Проверка авторизации пользователя на странице «authorization.php» с корректными данными |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе корректных данных |
| Этапы тестирования | Ввод корректных данных в текстовые поля;  Нажатие кнопки «Login». |
| Тестовые данные | Почта: fluffi@yandex.ru  Пароль: 1234 |
| Ожидаемый результат | Переход на страницу магазина с авторизированным пользователем |
| Фактический результат | Переход на страницу корзины с профилем с информацией о пользователе |

Результат тестирования на регистрацию пользователя представлен на рисунках 2.2.5 – 2.2.6.

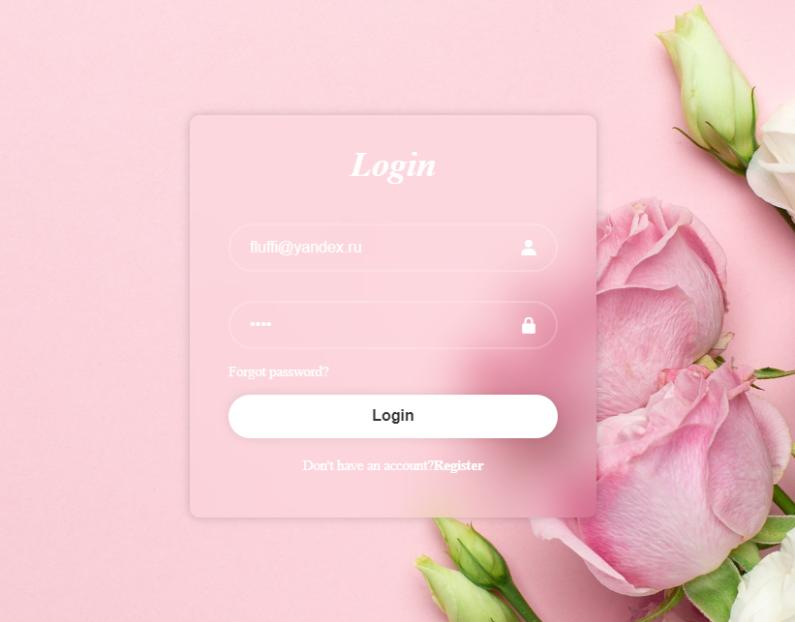


Рисунок 2.2.5 – Авторизация пользователя на корректных данных

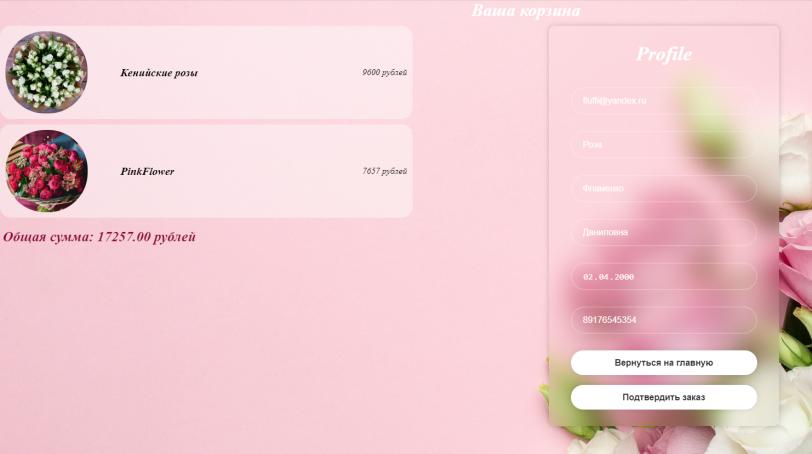


Рисунок 2.2.6 – Профиль авторизированного пользователя

Таблица 2.2.4 - Протокол тестирования на авторизацию пользователя на некорректных данных

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| Поле | Описание |
| 1 | 2 |
| Дата теста | 24.11.2024 |
| Приоритет тестирования | Высокий |
| Наименование тестирования | Проверка авторизации пользователя на странице «authorization.php» с некорректными данными |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе некорректных данных |
| Этапы тестирования | Ввод некорректных данных в текстовые поля;  Нажатие кнопки «Login». |
| Тестовые данные | Почта: fluffi@yandex.ru  Пароль: 12 |
| Ожидаемый результат | Вывод сообщения о неверных данных, переход на страницу авторизации |
| Фактический результат | Вывод сообщения «Неверные данные», переход на страницу авторизации |

Результат тестирования на регистрацию пользователя представлен на рисунках 2.2.7 – 2.2.8.

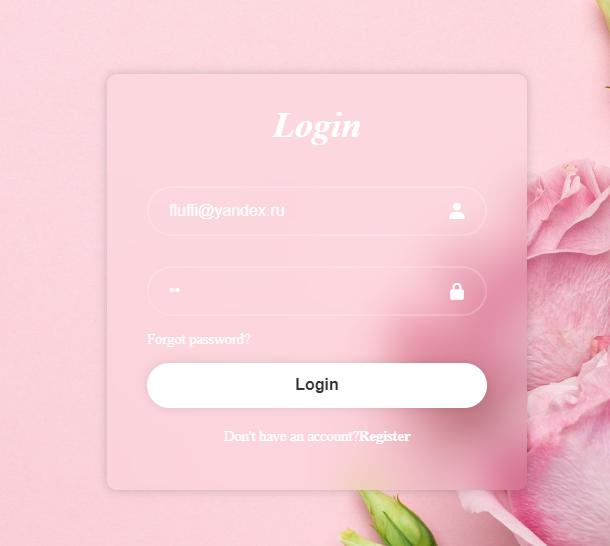


Рисунок 2.2.7 – Авторизация пользователя на некорректных данных

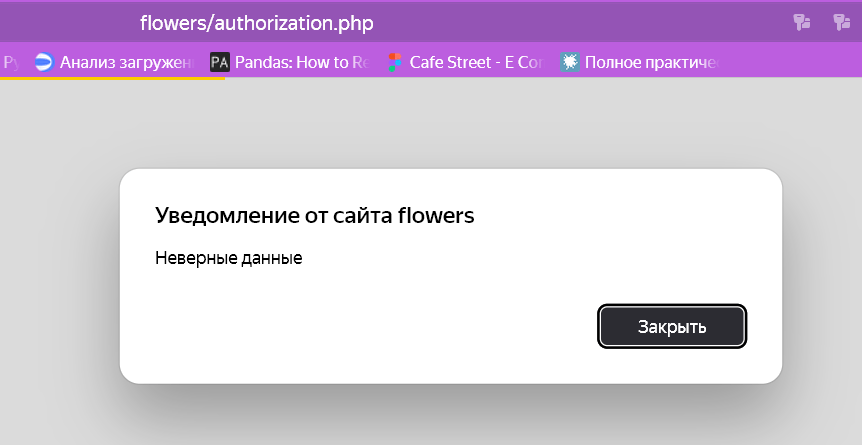


Рисунок 2.2.8 – Сообщение при неверной почте

Таблица 2.2.5 - Протокол тестирования на подтверждения заказа на корректных данных

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| Поле | Описание |
| 1 | 2 |
| Дата теста | 24.11.2024 |
| Приоритет тестирования | Высокий |
| Наименование тестирования | Проверка подтверждения заказа на странице «card.php» с корректными данными |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе корректных данных |
| Этапы тестирования | Добавление имеющихся товаров в корзину;  Нажатие кнопки «Подтвердить заказ». |
| Тестовые данные | Товар: Кенийские розы  Товар: PinkFlower  Авторизированный пользователь: fluffi@yandex.ru |
| Ожидаемый результат | Сообщение об успешном заказе и информации о том, что информация о заказе отправлена на почту. Изменение количества товара в базе данных, сохранение информации о заказе в базе дданых |
| Фактический результат | Вывод сообщения «Заказ успешно отправлен на почту», на почту пришло сообщение с информацией о заказе и пользователе. Отсутсвует запись заказа в базе данных, отсутвует изменение количества товара в базе данных |

Результат тестирования на регистрацию пользователя представлен на рисунках 2.2.9 – 2.2.14.

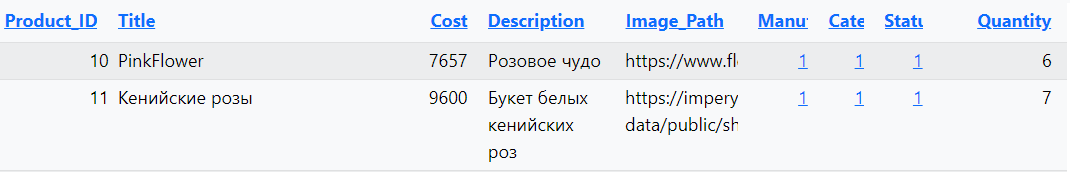


Рисунок 2.2.9 – Данные продуктов до заказа

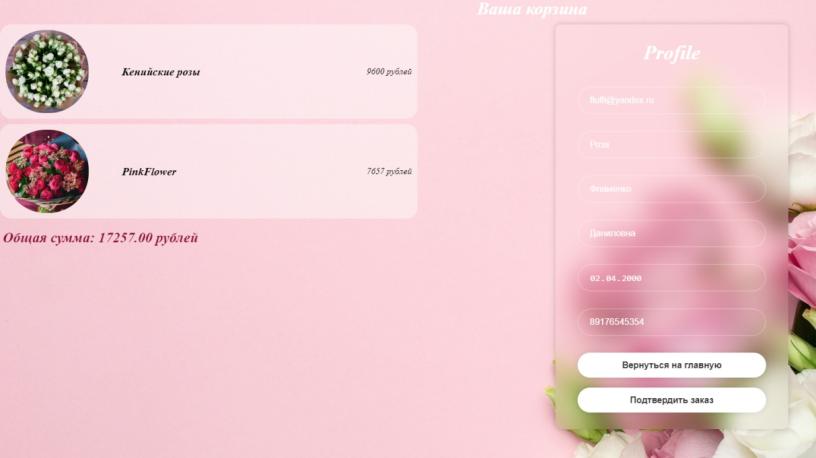


Рисунок 2.2.10 – Корзина пользователя

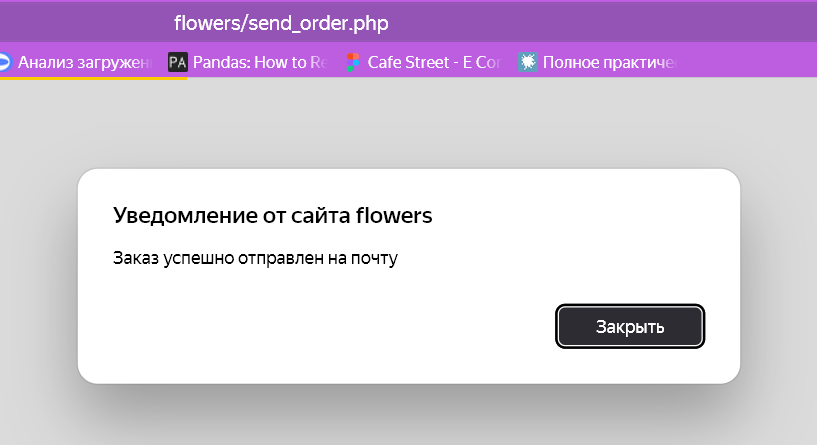


Рисунок 2.2.11 – Сообщение при заказе

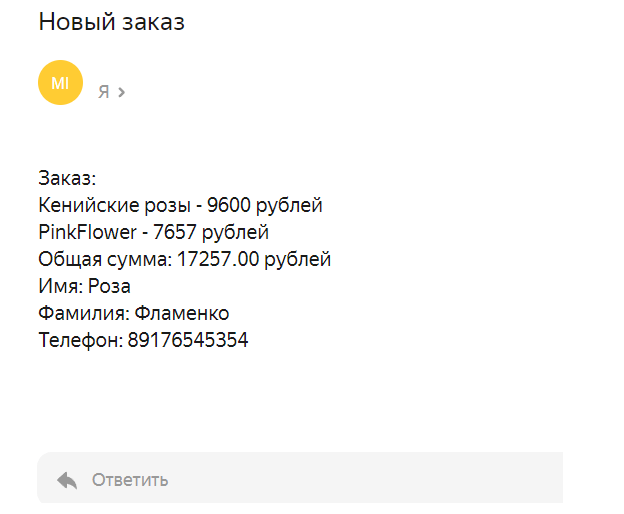


Рисунок 2.2.12 – Письмо, отправленное на почту при заказе

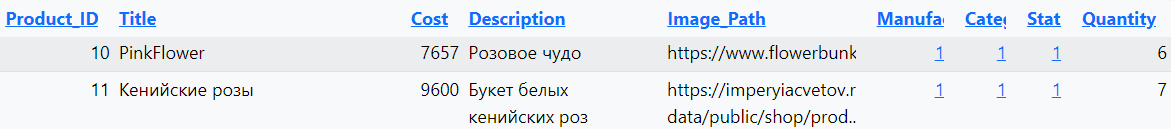


Рисунок 2.2.13 – Данные продуктов после заказа

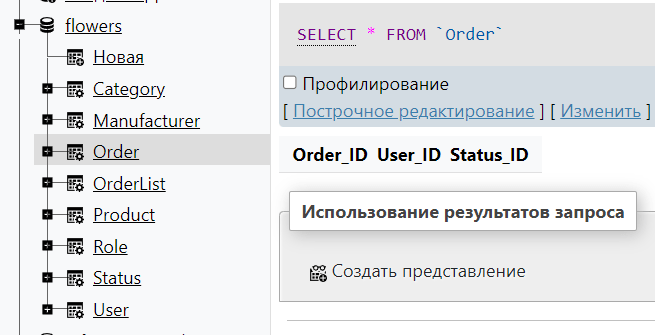


Рисунок 2.2.14 – Данные о заказах в базе данных после оформления заказа

Таблица 2.2.6 - Протокол тестирования на подтверждения заказа на некорректных данных

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| Поле | Описание |
| 1 | 2 |
| Дата теста | 24.11.2024 |
| Приоритет тестирования | Высокий |
| Наименование тестирования | Проверка подтверждения заказа на странице «card.php» с некорректными данными |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе некорректных данных |
| Этапы тестирования | Добавление товаров, которых нет в наличии, в корзину;  Нажатие кнопки «Подтвердить заказ». |
| Тестовые данные | Товар: Мишка из роз на день рождения  Авторизированный пользователь: fluffi@yandex.ru |
| Ожидаемый результат | Сообщение о том, что товара нет в наличии, невозможно оформить заказ |
| Фактический результат | Товар отсутвует в каталоге, если его id статус 2 (Нет в наличии), оформить заказ невозможно |

Результат тестирования на регистрацию пользователя представлен на рисунках 2.2.15 – 2.2.16.

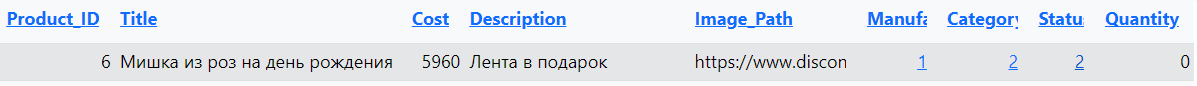


Рисунок 2.2.15 – Данные о товаре в базе данных

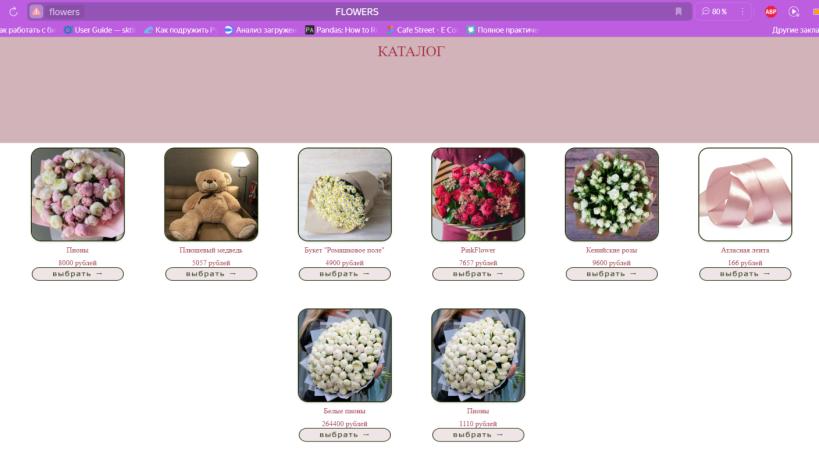


Рисунок 2.2.16 – Каталог в магазине

Таблица 2.2.7 - Протокол тестирования по добавлению товара на корректных данных

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| Поле | Описание |
| 1 | 2 |
| Дата теста | 24.11.2024 |
| Приоритет тестирования | Высокий |
| Наименование тестирования | Проверка добавления товара на странице «add\_Tovar.php» с корректными данными |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе корректных данных |
| Этапы тестирования | Заполнение полей корректными данными;  Нажатие кнопки «Применить». |

Продолжение таблицы 2.2.7

|  |  |
| --- | --- |
| 1 | 2 |
| Тестовые данные | ID: 15  Наименование товара: Гвоздики  Цена: 2400  Описание: Гвоздики розовые 9 шт  Фото: https://flawery.ru/image/thumb\_1220/products/240218/ 1270951\_0.jjp  Поставщик: Англия  Категория: Букет  Статус: В наличии  Количество: 10 |
| Ожидаемый результат | Сообщение о том, что товара добавлен |
| Фактический результат | Вывод сообщение о том, что товара добавлен |

Результат тестирования на добавления товара представлен на рисунках 2.2.17 – 2.2.20.

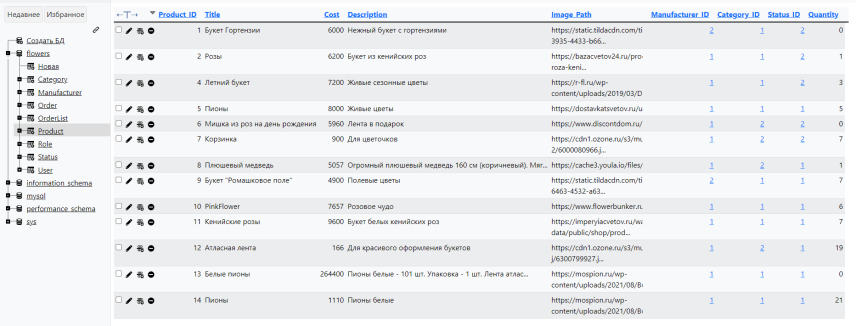


Рисунок 2.2.17 – Данные товаров до добавления товара

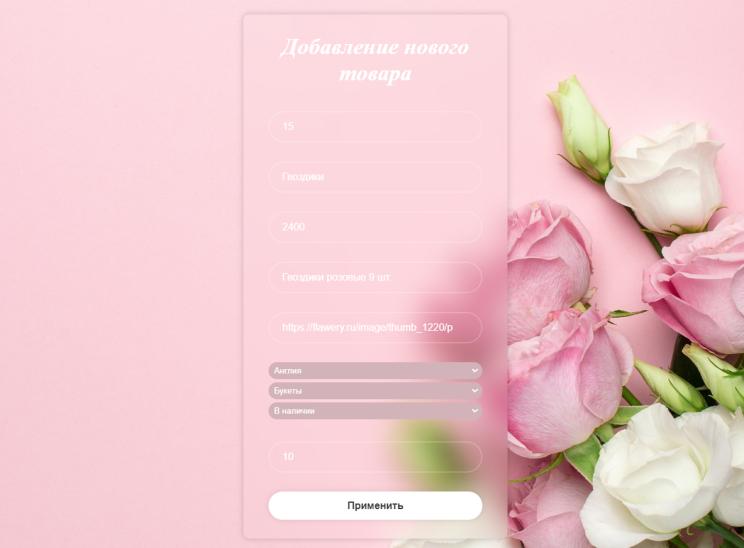


Рисунок 2.2.18 – Добавление товаров на корректных данных



Рисунок 2.2.19 – Вывод товара в каталоге магазина

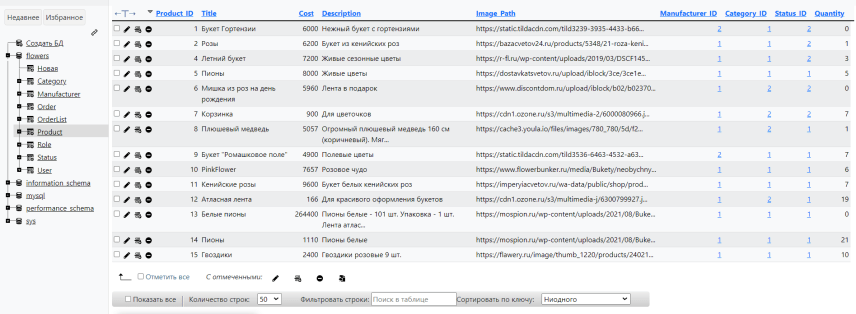


Рисунок 2.2.20 – Данные товаров в базе данных после добавления товара

Таблица 2.2.8 - Протокол тестирования по добавлению товара на некорректных данных

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| Поле | Описание |
| 1 | 2 |
| Дата теста | 24.11.2024 |
| Приоритет тестирования | Высокий |
| Наименование тестирования | Проверка добавления товара на странице «add\_Tovar.php» с некорректными данными |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе некорректных данных |
| Этапы тестирования | Заполнение полей некорректными данными;  Нажатие кнопки «Применить». |
| Тестовые данные | ID: 1  Наименование товара: Гвоздики  Цена: p  Описание: Гвоздики розовые 9 шт  Фото: https://flawery.ru/image/thumb\_1220/products/240218/ 1270951\_0.jjp  Поставщик: Англия  Категория: Букет  Статус: В наличии  Количество: p |
| Ожидаемый результат | Сообщение о том, что введены некорректные данные |
| Фактический результат | Вывод сообщение «Ошибка при добавлении товара» |

Результат тестирования на добавление товара представлен на рисунках 2.2.21 – 2.2.23.

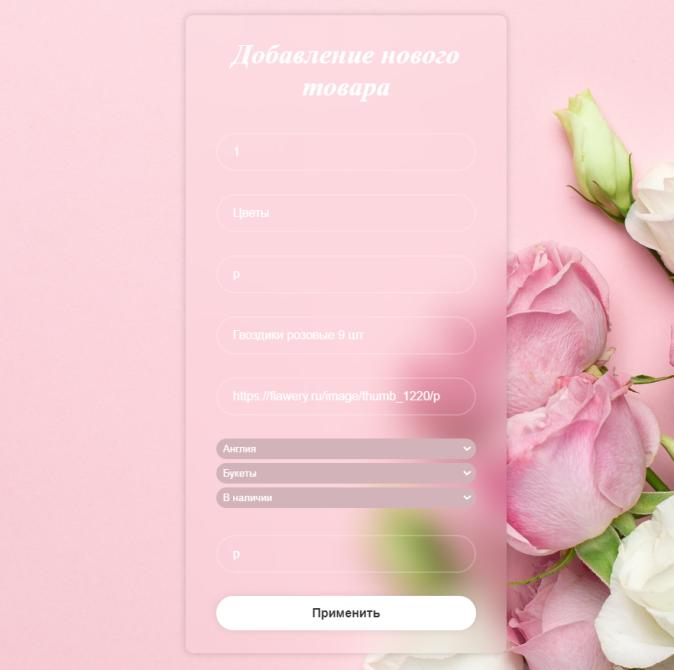


Рисунок 2.2.21 – Добавление товаров на некорректных данных

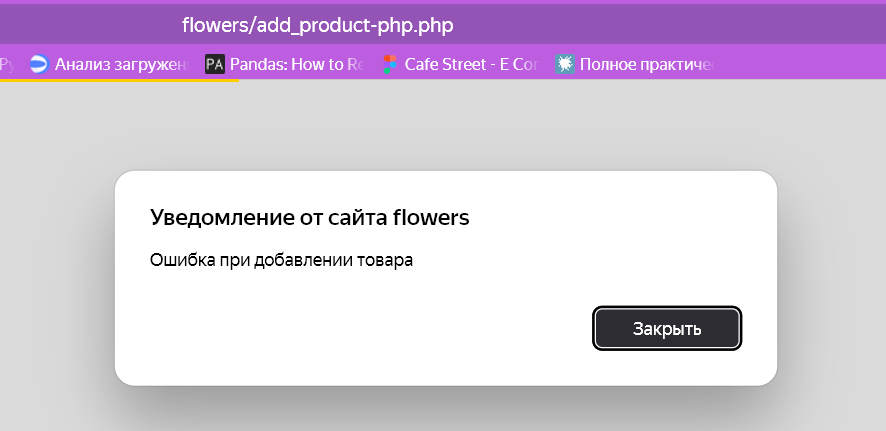


Рисунок 2.2.22 – Сообщение об ошибке добавления товара

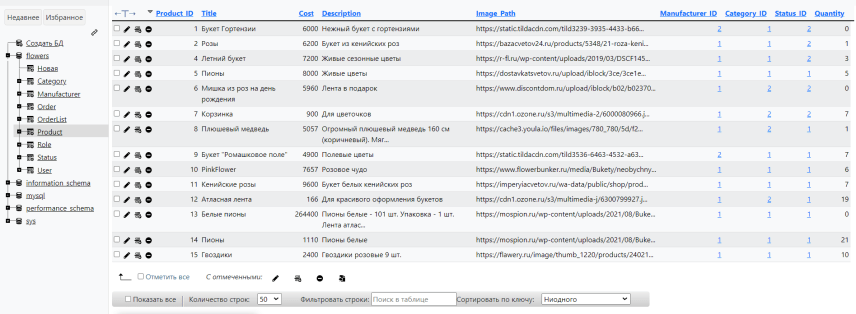


Рисунок 2.2.23 – Данные товаров в базе данных после попытки добавления товара

Таблица 2.2.8 - Протокол тестирования по изменению товара на некорректных данных

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| Поле | Описание |
| 1 | 2 |
| Дата теста | 24.11.2024 |
| Приоритет тестирования | Высокий |

Продолжение таблицы 2.2.8

|  |  |
| --- | --- |
| 1 | 2 |
| Наименование тестирования | Проверка изменения товара на странице «edit-product.php» с некорректными данными |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе некорректных данных |
| Этапы тестирования | Изменение полей некорректными данными;  Нажатие кнопки «Применить». |
| Тестовые данные | ID: 15  Наименование товара: Гвоздики  Цена:  Описание: Гвоздики розовые 9 шт  Фото: https://flawery.ru/image/thumb\_1220/products/240218/ 1270951\_0.jjp  Поставщик: Англия  Категория: Букет  Статус: В наличии  Количество: |
| Ожидаемый результат | Сообщение о том, что введены некорректные данные и переход на окно редактирования |
| Фактический результат | Вывод сообщение «Ошибка при обновлении данных», нет перехода на окно, надо заново авторизироваться. |

Результат тестирования на изменение товара представлен на рисунках 2.2.24 – 2.2.25.

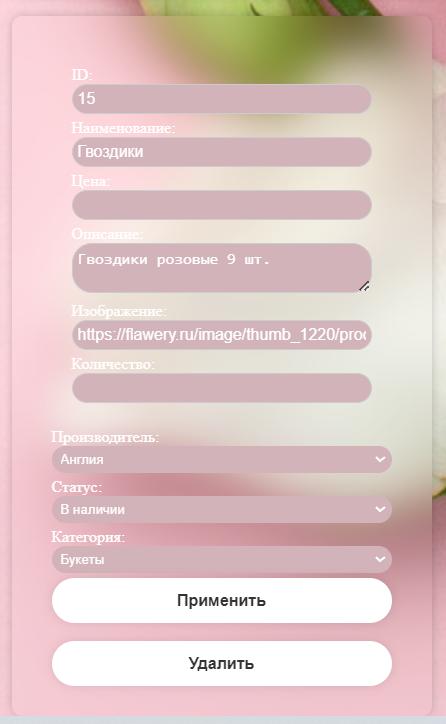


Рисунок 2.2.24 – Изменение товара на некорректных данных

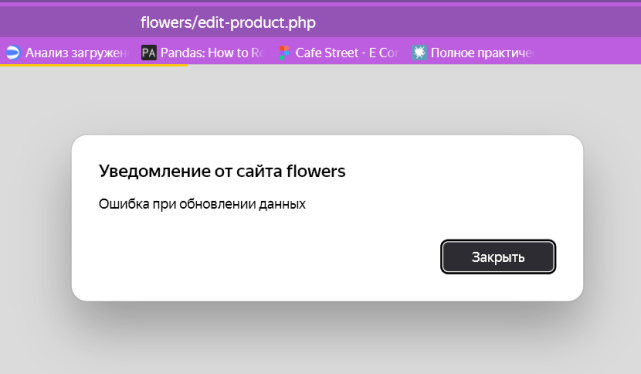


Рисунок 2.2.25 – Сообщение об ошибке изменении товара

Таблица 2.2.9 - Протокол тестирования по изменению товара на корректных данных

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| Поле | Описание |
| 1 | 2 |
| Дата теста | 24.11.2024 |
| Приоритет тестирования | Высокий |
| Наименование тестирования | Проверка изменения товара на странице «edit-product.php» с корректными данными |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе корректных данных |
| Этапы тестирования | Выбор товара;  Изменение полей корректными данными;  Нажатие кнопки «Применить». |
| Тестовые данные | ID: 15  Наименование товара: Гвоздики  Цена: 3400  Описание: Гвоздики розовые 9 шт  Фото:https://flawery.ru/image/thumb\_1220/products/240218/ 1270951\_0.jjp  Поставщик: Англия  Категория: Букет  Статус: В наличии  Количество: 5 |
| Ожидаемый результат | Сообщение о том, что данные изменены, переход на окно редактирования. Данные о товаре изменились в базе данных |

Продолжение таблицы 2.2.9

|  |  |
| --- | --- |
| 1 | 2 |
| Фактический результат | Вывод сообщение «Данные успешно обновлены», переход на окно редактирования. Данные изменились в базе данных |

Результат тестирования на изменения товара представлен на рисунках 2.2.26 – 2.2.29.

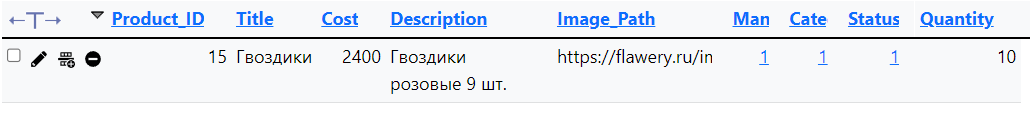


Рисунок 2.2.26 – Данные о товаре в базе данных до изменений

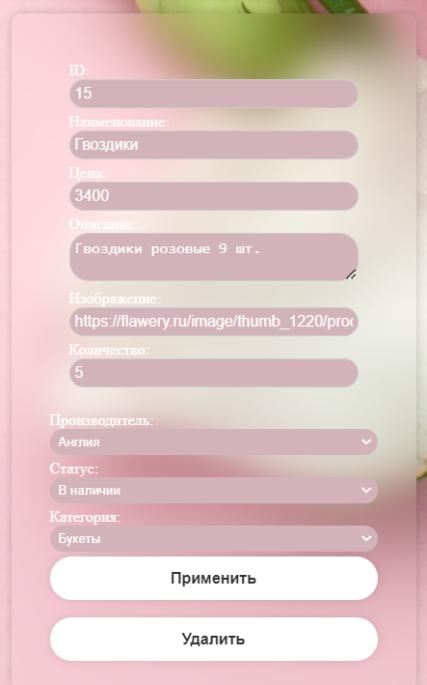


Рисунок 2.2.27 – Изменение товара на корректных данных

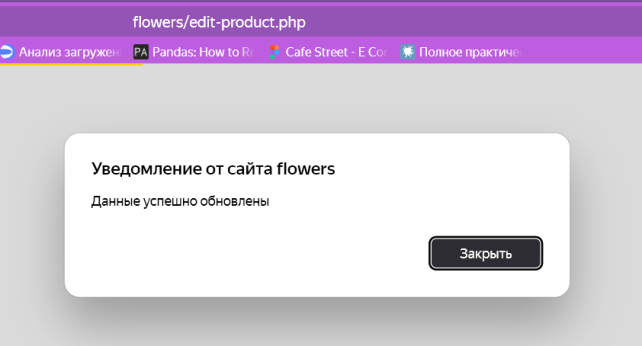


Рисунок 2.2.28 – Сообщение об изменении товара

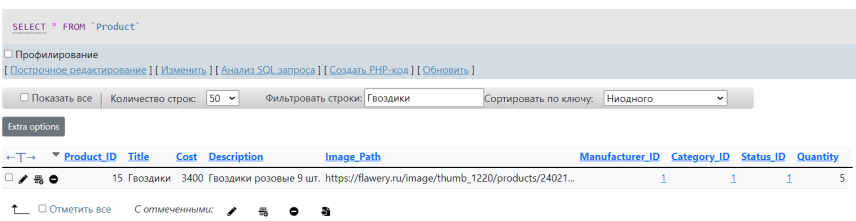


Рисунок 2.2.29 – Данные о товаре в базе данных после изменений

Таблица 2.2.10 - Протокол тестирования по удалению товара

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| Поле | Описание |
| 1 | 2 |
| Дата теста | 24.11.2024 |
| Приоритет тестирования | Высокий |
| Наименование тестирования | Проверка удаления товара на странице «edit-product.php» |
| Резюме испытания | Необходимо добиться корректного поведения программы при удалении товара |

Продолжение таблицы 2.2.10

|  |  |
| --- | --- |
| 1 | 2 |
| Этапы тестирования | Выбор товара;  Нажатие кнопки «Удалить». |
| Тестовые данные | ID: 15  Наименование товара: Гвоздики |
| Ожидаемый результат | Сообщение о том, что данные удалены, переход на окно редактирования, отсутвие товара в панели админа. В базе данных товар отсутствует |
| Фактический результат | Вывод сообщение «Товар успешно удален», переход на окно редактирования, товар не был удалён с панели администратора. Из базы данных товар не был удалён |

Результат тестирования на удаление товара представлен на рисунках 2.2.30 – 2.2.33.

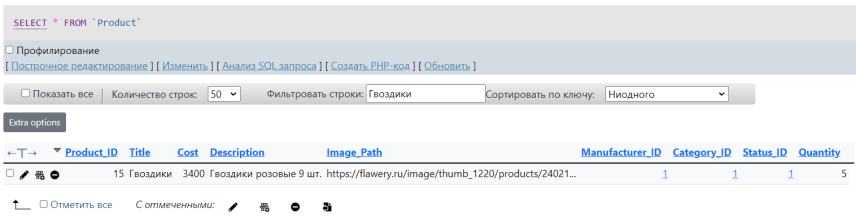


Рисунок 2.2.30 – Данные о товаре в базе данных до удаления

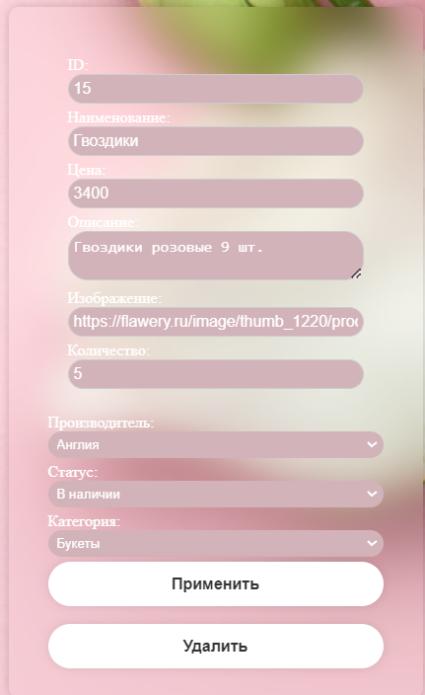


Рисунок 2.2.31 – Удаление товара на корректных данных

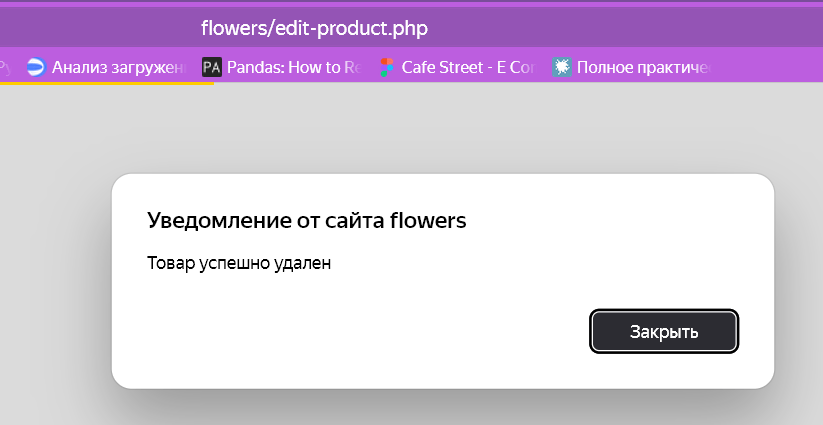


Рисунок 2.2.32 – Сообщение об удалении товара



Рисунок 2.2.33 – Данные о товаре в базе данных после удаления

2.3 Руководство пользователя

Добро пожаловать в "Цветочный магазин"! Это руководство поможет вам разобраться с основными функциями нашего приложения и максимально эффективно использовать его возможности.

1. Просмотр товаров

Каталог:

1. Перейдите на главную страницу, где представлен каталог товаров.



Рисунок 2.2.1 – Каталог товаров

2. Наведите на изображение для просмотра подробной информации.

2. Корзина:

1. Добавьте товары в корзину, нажав кнопку "Выбрать".

2. Перейдите в корзину, прокрутив страницу вниз или нажав на кнопку «Корзина» вверху сайта.

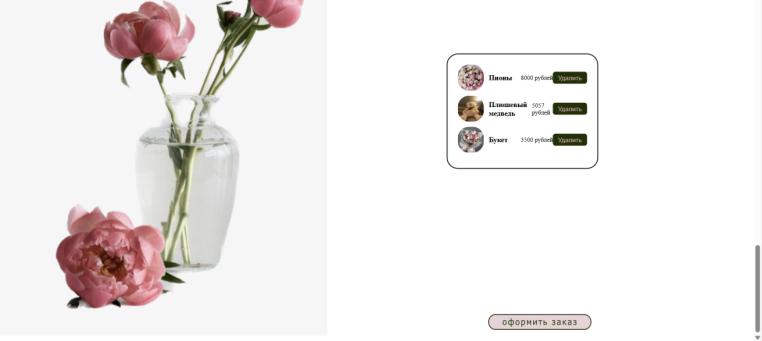


Рисунок 2.2.2 – Корзина

3. Проверьте содержимое корзины, удалите товары при необходимости.

3. Регистрация:

1. Нажмите на кнопку "Оформить заказ" в самом низу главной страницы.

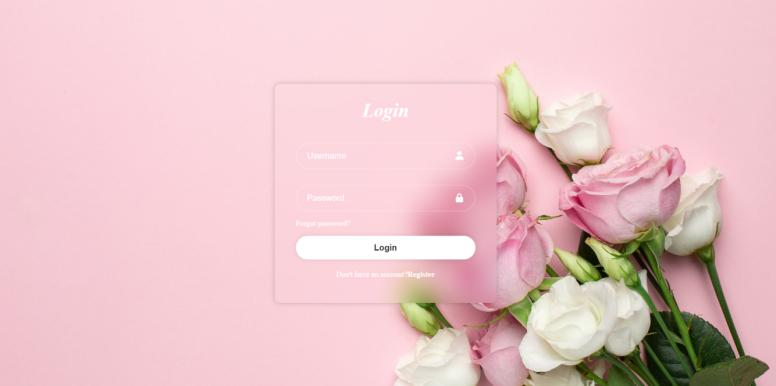


Рисунок 2.2.3 – Авторизация

2. Нажмите на кнопку «Register».

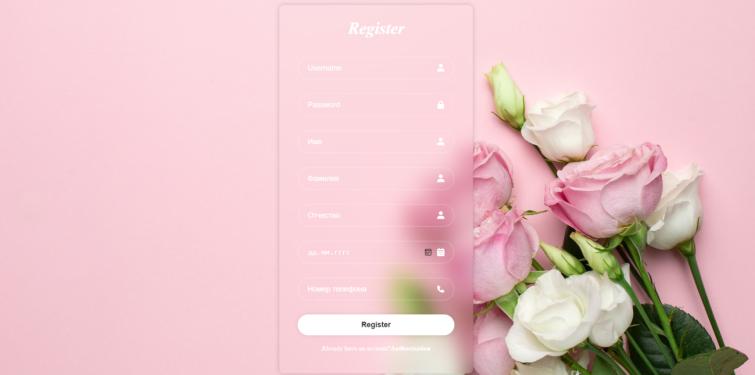


Рисунок 2.2.4 – Регистрация

2. Заполните форму регистрации, указав ваше имя, фамилию, отчество, дату рождения, номер телефона,email и пароль.



Рисунок 2.2.5 – Заполнение формы регистрации

При успешной регистрации выведется сообщение:

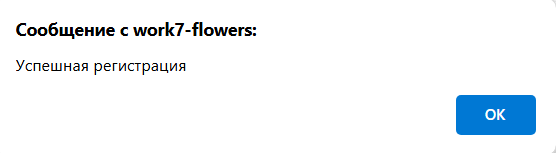


Рисунок 2.2.6 – Сообщение при успешной регистрация

Если пользователь с такой почтой уже существует:

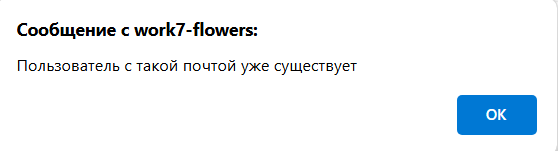


Рисунок 2.2.7 – Сообщение, если пользователь с такой почтой уже зарегистрирован

Если не все поля заполнены:

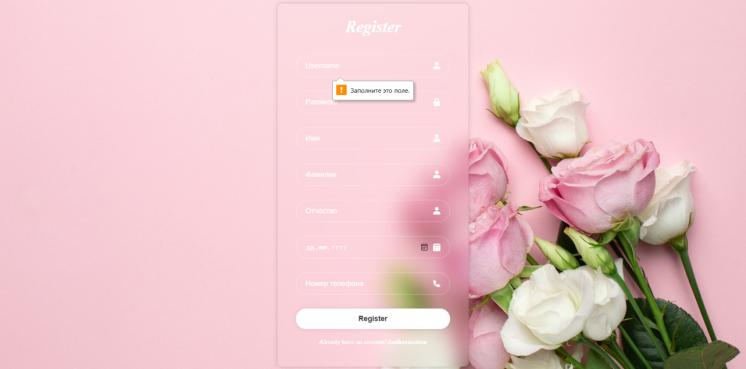


Рисунок 2.2.8 – Подсказка для заполнения всех полей

3. Нажмите "Register".

4. Авторизация:

1. Нажмите на кнопку "Оформить заказ" в самом низу главной страницы.

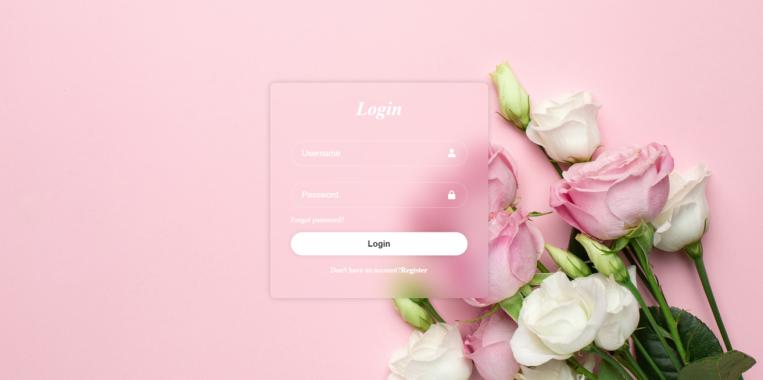


Рисунок 2.2.9 – Авторизация

2. Введите ваш email и пароль.

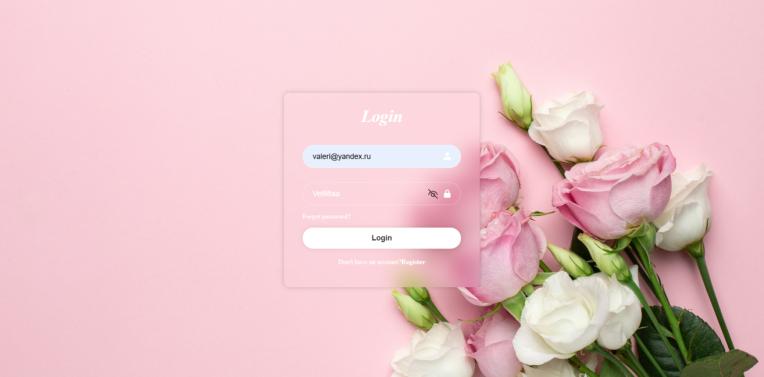


Рисунок 2.2.10 – Заполнение формы авторизации

При успешной авторизации происходит переход на страницу «Корзина».

Если пользователь ввел неверные данные, приходит сообщение:

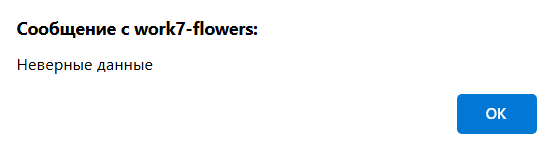


Рисунок 2.2.11– Сообщение при условии, что пользователь с такими данными не зарегистрирован.

3. Нажмите "Login".

Восстановление пароля:

1. Нажмите на ссылку "Forgotpassword?" на странице авторизации.

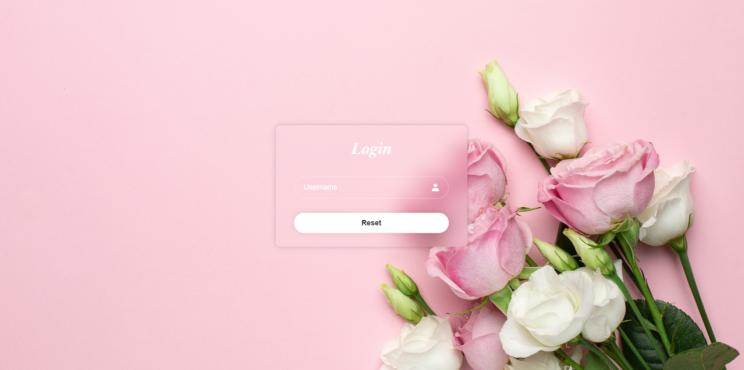


Рисунок 2.2.12 – Форма для сброса пароля

2. Введите ваш email и следуйте инструкциям, отправленным на вашу почту.

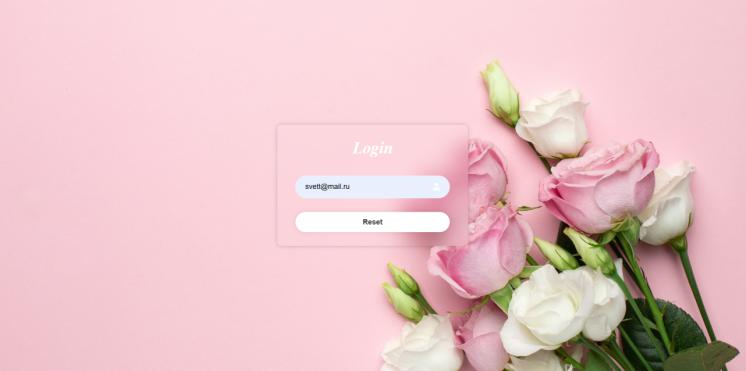


Рисунок 2.2.13 – Заполнение формы для сброса пароля

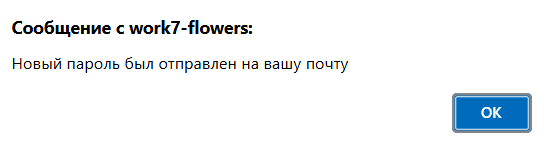


Рисунок 2.2.14 – Сообщение,если введенная почта была ранее зарегистрирована

Если введенная почта не найдена:

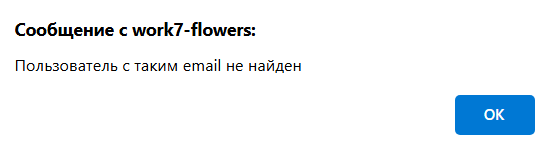


Рисунок 2.2.15 – Сообщение, если введенная почта не была ранее зарегистрирована

При успешном вводе почты новый пароль приходит на электронную почту пользователя и заменяется в базе данных.

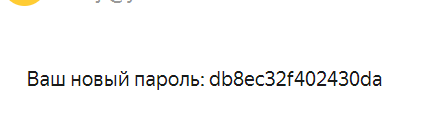


Рисунок 2.2.15 – Обновленный пароль

5. Оформление заказа:

1. Нажмите кнопку "Оформить заказ" в корзине.

2. Авторизуйтесь.



Рисунок 2.2.16 – Авторизация

При успешной авторизации идет перенаправление на корзину пользователя.

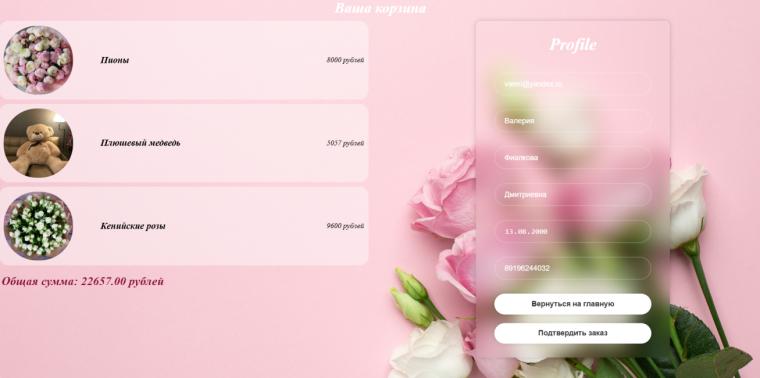


Рисунок 2.2.17 – Корзина пользователя

3. Нажмите "Подтвердить заказ". Вы получите подтверждение заказа на ваш email.

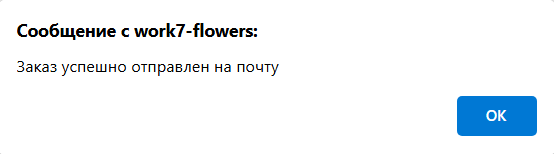


Рисунок 2.2.18 – Сообщение при нажатии на кнопку «Подтвердить заказ»

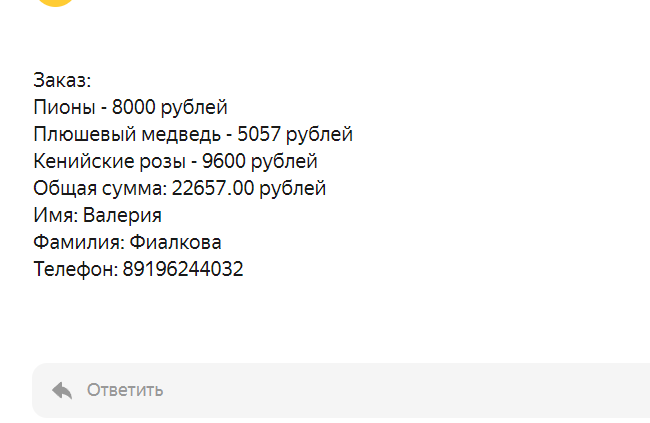


Рисунок 2.2.19 – Отправленный заказ на почту

4. Административный функционал (для администраторов)

Управление товарами:

1. Перейдите в административную панель, нажав на "Admin" в правом верхнем углу.

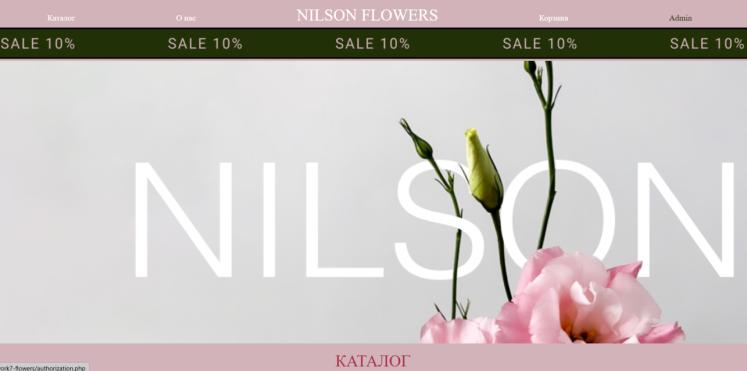


Рисунок 2.2.20 – Главная страница

2. Авторизуйтесь.



Рисунок 2.2.21 – Авторизация

Если пользователь не является администратором, выведется сообщение:

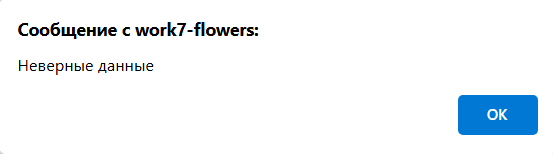


Рисунок 2.2.22 – Сообщение при неверно введеных данных

2. В разделе "Товары" вы можете добавлять новые товары, редактировать существующие и убирать из наличия ненужные.

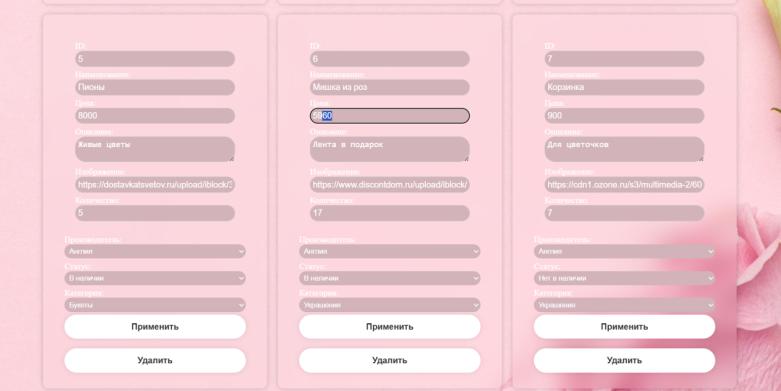


Рисунок 2.2.23 – Редактирование товара

При успешно введенных данных:

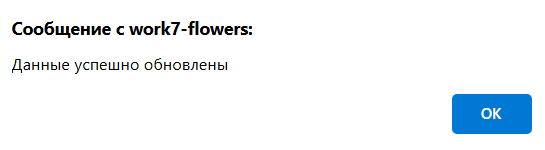


Рисунок 2.2.24 – Сообщение при корректных изменениях

При неккоректном вводе:

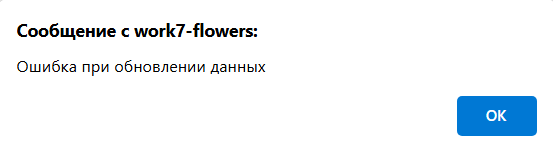


Рисунок 2.2.25 – Сообщение при некорректных изменениях

Для удаления товара необходимо нажать на кнопку удалить:

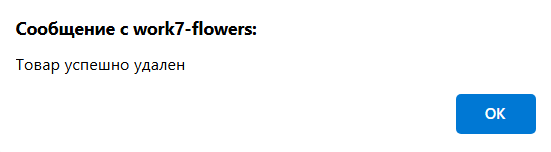


Рисунок 2.2.26 – Удаление товара

Чтобы вернуть товар необходимо изменить статус на «В наличии» и нажать на кнопку «Применить».

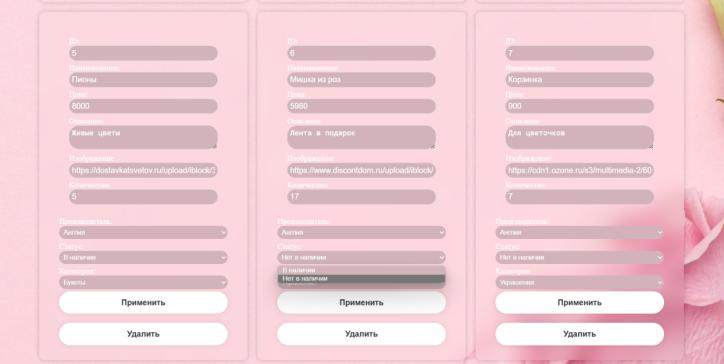


Рисунок 2.2.26 – Изменение статуса товара

3. Для добавления товара нажмите "Добавить", заполните форму и нажмите "Применить".

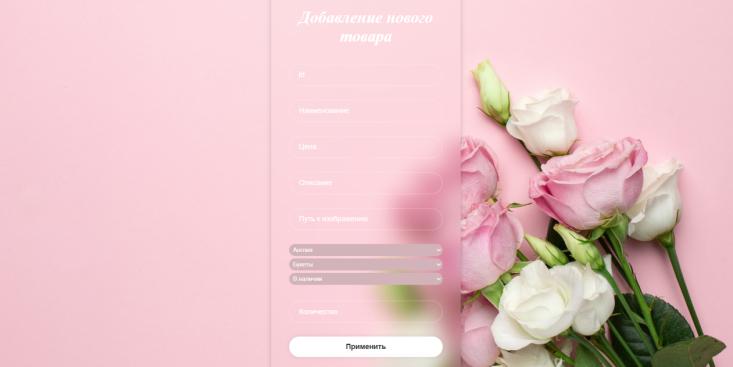


Рисунок 2.2.27 – Форма добавления товара

При успешном добавлении товара:

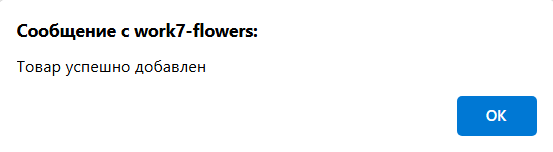


Рисунок 2.2.28 – Сообщение при успешном добавлении товара



Рисунок 2.2.28 – Отображение нового товара на главной странице

Заключение

В ходе выполнения данной курсовой работы была разработана и реализована программа "Цветочный магазин", которая предоставляет пользователям удобный интерфейс для просмотра и заказа товаров, а администраторам — мощные инструменты для управления ассортиментом и заказами.

Реализация проекта позволила углубить знания в области веб-разработки, включая использование современных технологий фронтенда и бэкенда, а также баз данных.

Дальнейшее развитие проекта может включать добавление новых функций, таких как интеграция с платежными системами, улучшение пользовательского интерфейса и расширение возможностей для администраторов.

Таким образом, поставленные цели и задачи курсовой работы были достигнуты, что подтверждает успешность выполненного проекта и его практическую значимость.

Шаблон выходного документа

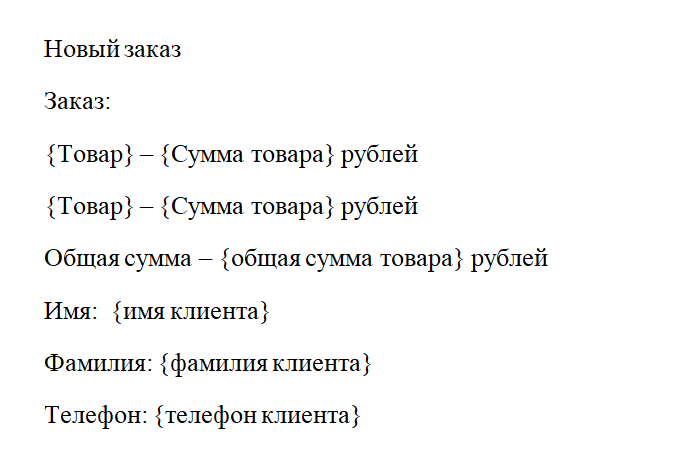


Рисунок А.1 – Шаблон электронного письма при оформлении заказа

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Товар | Стоимость | Общая сумма | Имя | Фамилия | Телефон |
| Кенийские розы | 9600 | 17257.00 | Роза | Фламенко | 89176545354 |
| PinkFlower | 7657 | 17257.00 | Роза | Фламенко | 89176545354 |

Рисунок А.2 - Информация о заказе

Код программы

Форма connectbd.php (Подключение к бд)

<?php

    $conn = mysqli\_connect("localhost", "root", "", "flowers", "3302");

?>

Форма index.php (Главная страница)

<?php

    require "connectbd.php";

    $query = mysqli\_query($conn, "SELECT \* FROM `Product`");

    $list\_product = mysqli\_fetch\_all($query);

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <link rel='stylesheet' href='stylee.css'>

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

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=REM:ital,wght@0,100..900;1,100..900&family=Rubik:ital,wght@0,300..900;1,300..900&display=swap" rel="stylesheet">

    <title>FLOWERS</title>

</head>

<body>

    <div class="header" id="UP">

        <div class="container">

            <div class="header-line">

                <div class="nav">

                    <a class="nav-item" href="#CATALOG">Каталог</a>

                    <a class="nav-item" href="#ABOUT">Онас</a>

                    <a class="nav-item1" href="#UP">NILSON FLOWERS</a>

                    <a class="nav-item" href="#CARD">Корзина</a>

                    <a class="nav-item" href="authorization.php">Admin</a>

                </div>

            </div>

        </div>

        <div class="sale">

                <div class="saleGreen">

                    <imgsrc="image\saleGreen.png" alt="">

                </div>

                <div class="saleTen">

                    <imgsrc="image\sale10.png" alt="">

                </div>

        </div>

        <div class="coverNILSON">

            <imgsrc="image\coverNILSON.png" alt="">

        </div>

        <div class="catalogTitle" id="CATALOG">

            КАТАЛОГ

        </div>

    </div>

    <script>

var cart = {

    items: [],

    addItem: function(item) {

        // Проверяем, есть ли товар с таким названием уже в корзине

        var existingItem = this.items.find(function(existing) {

            return existing.title === item.title;

        });

        if (!existingItem) {

            this.items.push(item);

            this.updateCartDisplay();

        } else {

            console.log("Товаружевкорзине.");

     }

   },

  removeItem: function(itemTitle) {

     // Удаляемтоваризкорзиныпоназванию

    this.items = this.items.filter(function(item) {

            return item.title !== itemTitle;

        });

        this.updateCartDisplay();

    },

    updateCartDisplay: function() {

    var cartDisplay = document.querySelector('.card-delivery');

    if (!cartDisplay) {

        cartDisplay = document.createElement('div');

        cartDisplay.classList.add('card-delivery');

        document.body.appendChild(cartDisplay);

    } else {

        cartDisplay.innerHTML = ''; // Очищаемпредыдущеесодержимое

    }

    cartDisplay.style.transform = 'translate(870px, -550px)';

    cartDisplay.style.position = 'absolute';

    if (this.items.length === 0) {

        // Есликорзинапуста, удаляемрамку

        cartDisplay.style.border = 'none';

        cartDisplay.innerHTML = '';

    } else {

        cartDisplay.style.border = '2px solid black';

    // Добавляемэлементывкорзину

    this.items.forEach(function(item) {

        var itemElement = document.createElement('div');

        itemElement.classList.add('card-delivery-item');

            var img = document.createElement('img');

            img.src = item.image;

            img.classList.add('card-delivery-img');

            img.alt = item.title;

            var title = document.createElement('div');

            title.classList.add('card-delivery-title');

            title.textContent = item.title;

            var price = document.createElement('div');

            price.classList.add('card-delivery-price');

            price.textContent = item.price + ' рублей';

            var removeButton = document.createElement('button');

            removeButton.textContent = 'Удалить';

            removeButton.classList.add('button');

            removeButton.onclick = function() {

            cart.removeItem(item.title);

            };

            itemElement.appendChild(img);

            itemElement.appendChild(title);

            itemElement.appendChild(price);

            itemElement.appendChild(removeButton);

            cartDisplay.appendChild(itemElement);

            localStorage.setItem('selectedItems', JSON.stringify(cart.items));

        });

    }

    }

};

</script>

<div class="cards">

    <div class="container">

        <div class="cards-holder">

            <?phpforeach($list\_product as $item):?>

                <?phpif($item[7] == 1): // Условие для проверки статуса ?>

                    <div class="card">

                        <div class="card-image">

                            <img class="card-img" src="<?=$item[4]?>" title='<?=$item[3]?>'>

                        </div>

                        <div class="card-title">

                            <?=$item[1]?>

                        </div>

                        <div class="card-price">

                            <?=$item[2]?>рублей

                        </div>

                        <div class="card-choose">

                            <img class="card-cho" src="image\choose.png" onclick="cart.addItem({title: '<?=$item[1]?>', price: '<?=$item[2]?>', image: '<?=$item[4]?>'})">

                        </div>

                    </div>

                <?php endif; ?>

            <?phpendforeach;?>

        </div>

    </div>

</div>

    <div class="about" id="ABOUT">

        <div class="about-image">

            <img class=about-imgsrc="image\aboutUs.png">

        </div>

    </div>

    <div class="afterAbout">

            <div class="after">

                <p class="after-item" >Гарантиясвежести</p>

                <p class="after-item" >Доставка по городу и за его пределы</p>

                <p class="after-item" >Открытка в подарок</p>

                <p class="after-item1" >Доставкапосле 00:00</p>

                <p class="after-item1" >Фото и видео перед отправкой</p>

            </div>

    </div>

    <div class="delivery" id="CARD">

        <div class="delivery-image">

            <img class=delivery-imgsrc="image\delivery.png">

        </div>

    </div>

        <div class="card-delivery-buy">

            <a href="authorization.php"><img class="card-delivery-buy" src="image\buy.png"></a>

        </div>

</body>

</html>

Формаauthorization.php (Авторизация)

<?php

require "connectbd.php";

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

    $email = $\_POST['email'];

    $password = $\_POST['password'];

    if (empty($email) || empty($password)){

        echo "Заполните все поля";

    } else {

        $sql = "SELECT \* FROM User WHERE email = '$email' AND password = '$password'";

        $result = $conn->query($sql);

        if ($result->num\_rows > 0) {

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

            setcookie("idUser", $row['User\_ID'], time() + 3600, "/");

            if ($row['Role\_ID'] == 1) {

                header('Location: admin-panel.php');

                exit();

            } elseif ($row['Role\_ID'] == 2) {

                header('Location: card.php');

                exit();

            } else {

                echo "У вас нет доступа к этой странице.";

                exit();

            }

        } else {

            echo "<script>alert(\"Неверныеданные\");</script>";

        }

    }

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <link rel="stylesheet" href="style2.css">

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

    <link href="https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;

    0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap" rel="stylesheet">

    <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='stylesheet'>

    <title>FLOWERS</title>

</head>

<body>

    <div class="wrapper">

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

            <h1>Login</h1>

            <div class="input-box">

                <input type="text" name="email"

                placeholder="Username" required>

                <i class='bx bxs-user'></i>

            </div>

            <div class="input-box">

                <input type="password" name="password"

                 placeholder="Password" required>

                <i class='bx bxs-lock-alt'></i>

            </div>

            <div class="forgot">

                <a href="reset\_password.php">Forgot password?</a>

            </div>

            <button type="submit" class="bth">Login</button>

            <div class="register-link">

                <p>Don't have an account?<a href="register.php">Register</a></p>

            </div>

        </form>

    </div>

</body>

</html>

Формаregister.php (Регистрация)

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <link rel="stylesheet" href="style2.css">

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

    <link href="https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;

    0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap" rel="stylesheet">

    <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='stylesheet'>

    <title>FLOWERS</title>

</head>

<body>

    <div class="wrapper">

        <form action="register-php.php" method="post">

            <h1>Register</h1>

            <div class="input-box">

                <input type="text" name="email" placeholder="Username" required>

                <i class='bx bxs-user'></i>

            </div>

            <div class="input-box">

                <input type="password" name="password" placeholder="Password" required>

                <i class='bx bxs-lock-alt'></i>

            </div>

            <div class="input-box">

                <input type="text" name="name" placeholder="Имя" required>

                <i class='bx bxs-user'></i>

            </div>

            <div class="input-box">

                <input type="text" name="surname" placeholder="Фамилия" required>

                <i class='bx bxs-user'></i>

            </div>

            <div class="input-box">

                <input type="text" name="patronymic" placeholder="Отчество" required>

                <i class='bx bxs-user'></i>

            </div>

            <div class="input-box">

                <input type="date" name="DateOfBirth" placeholder="Датарождения" required>

                <i class='bx bxs-calendar'></i>

            </div>

            <div class="input-box">

                <input type="tel" name="phone" placeholder="Номертелефона" required>

                <i class='bx bxs-phone'></i>

            </div>

            <button type="submit" class="bth">Register</button>

            <div class="register-link">

                <p>Already have an account?<a href="authorization.php">Authorization</a></p>

            </div>

        </form>

    </div>

</body>

</html>

Формаregister-php.php (Логика регистрации)

<?php

require "connectbd.php";

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

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

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

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

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

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

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

if (empty($name) || empty($surname) || empty($patronymic) || empty($DateOfBirth) || empty($phone) || empty($email) || empty($password)) {

    echo "<script>alert(\"Заполнитевсеполя\");</script>";

} else {

    $emailQuery = "SELECT \* FROM User WHERE email = '$email'";

    $emailResult = $conn->query($emailQuery);

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

        echo "<script>alert(\"Пользователь с такой почтой уже существует\");</script>";

    } else {

        $sql = "INSERT INTO User (name, surname, patronymic, DateOfBirth, phone, email, password, Role\_ID)

                VALUES ('$name', '$surname', '$patronymic', '$DateOfBirth', '$phone', '$email', '$password', 2)";

        if ($conn->query($sql) === TRUE) {

            echo "<script>alert(\"Успешнаярегистрация\"); location.href='authorization.php';</script>";

            exit();

        } else {

            echo "Ошибка: " . $conn->error;

        }

    }

}

?>

Формаprofile.php (Профиль)

<?php

require "connectbd.php";

$idUser = $\_COOKIE['idUser'];

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

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

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

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

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

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

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

$query = mysqli\_query($conn, "UPDATE User SET Name='$name', Surname='$surname', Patronymic='$patronymic',

DateOfBirth='$dateBirthday', Phone='$numberPhone', Email='$email', Password='$password' WHERE User\_Id = '$idUser'");

?>

Форма reset\_password.php (Обновление пароля)

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <link rel="stylesheet" href="style2.css">

    <link href="https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;

    0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap" rel="stylesheet">

    <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='stylesheet'>

    <title>FLOWERS</title>

</head>

<body>

<div class="wrapper">

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

            <h1>Login</h1>

            <div class="input-box">

                <input type="text" name="email"

                placeholder="Username" required>

                <i class='bx bxs-user'></i>

            </div>

            <button type="submit" class="bth">Reset</button>

        </form>

    </div>

</body>

</html>

Формаreset\_password-php.php (Логикаобновленияпароля)

<?php

require\_once('connectbd.php');

if ($conn->connect\_error) {

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

}

// Проверкаотправкиформы

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

    $email = $conn->real\_escape\_string($\_POST['email']);

    // Проверка наличия email в базе данных

    $sql = "SELECT \* FROM User WHERE email = '$email'";

    $result = $conn->query($sql);

    if ($result->num\_rows > 0) {

        // Генерацияновогопароля

        $new\_password = bin2hex(random\_bytes(8)); // 16 символов

        // Обновлениепаролявбазеданных

        $update\_sql = "UPDATE User SET password = '$new\_password' WHERE email = '$email'";

        if ($conn->query($update\_sql) === TRUE) {

            // Отправка нового пароля на email

            $to = $email;

            $subject = "Ваш новый пароль";

            $message = "Ваш новый пароль: " .$new\_password;

      $headers = "From: missis.fazullina@yandex.ru";

            if (mail($to, $subject, $message, $headers)) {

                echo "<script>alert(\"Новыйпарольбылотправленнавашупочту\"); location.href='authorization.php';</script>";

            } else {

                echo "Ошибка при отправке пароля.";

            }

        } else {

            echo "Ошибка при обновлении пароля.";

        }

    } else {

        echo "<script>alert(\"Пользователь с таким email не найден\"); location.href='reset\_password.php';</script>";

    }

}

$conn->close();

?>

Форма admin\_panel.php (Админ панель)

<?php

    require "connectbd.php";

    $query = mysqli\_query($conn, "SELECT \* FROM `Product`");

    $list\_product = mysqli\_fetch\_all($query);

    $sql = "SELECT Status\_ID, Title FROM Status";

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

    $list\_status = mysqli\_fetch\_all($result);

    $sql = "SELECT Manufacturer\_ID, Title FROM Manufacturer";

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

    $list\_manufacturer = mysqli\_fetch\_all($result);

    $sql = "SELECT Category\_ID, Title FROM Category";

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

    $list\_category = mysqli\_fetch\_all($result);

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <link rel='stylesheet' href='style4.css'>

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

    <title>FLOWERS</title>

</head>

<body>

    <div class="container">

        <h1 class="h1Products">Продукты</h1>

        <div class="addButContaner">

            <a href="add\_Tovar.php"><button class="addBut">Добавить</button></a>

            <a href="index.php"><button class="addBut">Главная</button></a>

        </div>

    </div>

    <div class="all-products">

        <?phpforeach($list\_product as $item\_product):?>

            <div class="wrapper">

                <form action="edit-product.php" method="POST">

                    <div class="product-form">

                        <p>ID: <input type="text" name="product\_id" value="<?=$item\_product[0]?>" readonly></p>

                        <p>Наименование: <input type="text" name="product\_name"  value="<?=$item\_product[1]?>"></p>

                        <p>Цена: <input type="text" name="product\_price" value="<?=$item\_product[2]?>"></p>

                        <p>Описание: <textarea name="product\_desc"><?=$item\_product[3]?></textarea></p>

                        <p>Изображение: <input type="text" name="product\_image" value="<?=$item\_product[4]?>"></p>

                        <p>Количество: <input type="text" name="product\_quantity" value="<?=$item\_product[8]?>"></p>

                    </div>

                    <p>Производитель:

                        <select name="manufacturer\_id" class="form-select">

                            <?phpforeach($list\_manufacturer as $row):?>

                                <?if($row[0]==$item\_product[5]):?>

                                    <option value="<?=$row[0]?>" selected><?=$row[1]?></option>

                                <?else:?>

                                    <option value="<?=$row[0]?>"><?=$row[1]?></option>

                                <?endif;?>

                            <?endforeach;?>

                        </select>

                    </p>

                    <p>Статус:

                        <select name="status\_id" class="form-select">

                            <?phpforeach($list\_status as $row):?>

                                <?if($row[0]==$item\_product[7]):?>

                                    <option value="<?=$row[0]?>" selected><?=$row[1]?></option>

                                <?else:?>

                                    <option value="<?=$row[0]?>"><?=$row[1]?></option>

                                <?endif;?>

                            <?endforeach;?>

                        </select>

                    </p>

                    <p>Категория:

                        <select name="category\_id" class="form-select">

                            <?phpforeach($list\_category as $row):?>

                                <?if($row[0]==$item\_product[6]):?>

                                    <option value="<?=$row[0]?>" selected><?=$row[1]?></option>

                                <?else:?>

                                    <option value="<?=$row[0]?>"><?=$row[1]?></option>

                                <?endif;?>

                            <?endforeach;?>

                        </select>

                    </p>

                    <button type="submit" class="bth" name="apply\_changes">Применить</button>

                    &nbsp;

                </form>

                <form action="edit-product.php" method="POST">

                    <input type="text" name="product\_id" value="<?=$item\_product[0]?>" hidden>

                    <button type="submit" class="bth" name="delete\_changes">Удалить</button>

                </form>

            </div>

        <?endforeach;?>

    </div>

</body>

</html>

Форма add\_Tovar.php (Добавление товара)

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <link rel='stylesheet' href='style2.css'>

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

    <title>Document</title>

</head>

<body>

    <div class="wrapper">

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

            <h1>Добавление нового товара</h1>

            <div class="input-box">

                <input type="text" name="product\_id" placeholder="id" value="<?=$item[0]?>" required>

                <i class='bx bxs-user'></i>

            </div>

            <div class="input-box">

                <input type="text" name="product\_name" placeholder="Наименование" value="<?=$item[1]?>"  required>

                <i class='bx bxs-user'></i>

            </div>

            <div class="input-box">

                <input type="text" name="product\_price" placeholder="Цена" value="<?=$item[2]?>"  required>

                <i class='bx bxs-user'></i>

            </div>

            <div class="input-box">

                <input type="text" name="product\_desc" placeholder="Описание" value="<?=$item[3]?>"  required>

                <i class='bx bxs-user'></i>

            </div>

            <div class="input-box">

                <input type="text" name="product\_image" placeholder="Путькизображению" value="<?=$item[4]?>"  required>

                <i class='bx bxs-user'></i>

            </div>

            <select name="manufacturer\_id" class="form-select">

            <?php

                require "connectbd.php";

                $sql = "SELECT Manufacturer\_ID, Title FROM Manufacturer";

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

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

                    echo "<option value='" . $row['Manufacturer\_ID'] ."'>" . $row['Title'] . "</option>";

                }

                ?>

            </select>

            <select name="category\_id" class="form-select">

                <?php

                require "connectbd.php";

                $sql = "SELECT Category\_ID, Title FROM Category";

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

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

                    echo "<option value='" . $row['Category\_ID'] ."'>" . $row['Title'] . "</option>";

                }

                ?>

            </select>

            <select name="status\_id" class="form-select">

                <?php

                require "connectbd.php";

                $sql = "SELECT Status\_ID, Title FROM Status";

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

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

                    echo "<option value='" . $row['Status\_ID'] ."'>" . $row['Title'] . "</option>";

                }

                ?>

            </select>

            <div class="input-box">

                <input type="text" name="product\_quantity" placeholder="Количество" value="<?=$item[8]?>"  required>

                <i class='bx bxs-user'></i>

            </div>

            <button type="submit" class="bth" name="apply\_changes">Применить</button>

        </form>

    </div>

</body>

</html>

Формаadd\_product-php.php (Логикадобавлениятовара)

<?php

require "connectbd.php";

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

    $new\_title = $\_POST['product\_name'];

    $new\_cost = $\_POST['product\_price'];

    $new\_desc = $\_POST['product\_desc'];

    $new\_image = $\_POST['product\_image'];

    $new\_manufacturer = $\_POST['manufacturer\_id'];

    $new\_category = $\_POST['category\_id'];

    $new\_status = $\_POST['status\_id'];

    $new\_quantity = $\_POST['product\_quantity'];

    $check\_sql = "SELECT Product\_ID FROM Product WHERE Product\_ID = ?";

    $check\_stmt = $conn->prepare($check\_sql);

    $check\_stmt->bind\_param("s", $new\_title);

    $check\_result = $check\_stmt->get\_result();

    if ($check\_result->num\_rows > 0) {

        echo "Товар с таким ID уже существует.";

    } else {

        $stmt = $conn->prepare("INSERT INTO Product (Title, Cost, Description, Image\_Path, Manufacturer\_ID, Category\_ID, Status\_ID, Quantity)

        VALUES (?, ?, ?, ?, ?, ?, ?, ?)");

        $stmt->bind\_param("ssssssss", $new\_title, $new\_cost, $new\_desc, $new\_image, $new\_manufacturer, $new\_category, $new\_status, $new\_quantity);

        if ($stmt->execute()) {

            echo "<script>alert(\"Товаруспешнодобавлен\"); location.href='admin-panel.php';</script>";

            exit();

        } else {

            echo "<script>alert(\"Ошибкапридобавлениитовара\");</script>";

    }

    }

    $stmt->close();

}

?>

Форма edit\_product.php (Редактирование и удаление товаров)

<?php

// update\_product.php

require "connectbd.php";

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

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

    $new\_title = $\_POST['product\_name'];

    $new\_cost = $\_POST['product\_price'];

    $new\_desc = $\_POST['product\_desc'];

    $new\_image = $\_POST['product\_image'];

    $new\_manufacturer = $\_POST['manufacturer\_id'];

    $new\_category = $\_POST['category\_id'];

    $new\_status = $\_POST['status\_id'];

    $new\_quantity = $\_POST['product\_quantity'];

    $stmt = $conn->prepare("UPDATE Product SET Title = ?, Cost = ?, Description = ?,

    Image\_Path = ?, Manufacturer\_ID = ?, Category\_ID = ?, Status\_ID = ?,

    Quantity = ? WHERE Product\_ID = ?");

    $stmt->bind\_param("sssssssss", $new\_title, $new\_cost, $new\_desc, $new\_image, $new\_manufacturer, $new\_category, $new\_status, $new\_quantity, $product\_id);

    if ($stmt->execute()) {

        echo "<script>alert(\"Данныеуспешнообновлены\"); location.href='admin-panel.php';</script>";

        exit();

    } else {

        echo "<script>alert(\"Ошибкаприобновленииданных\");</script>";

  }

    $stmt->close();

}

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

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

    $stmt = $conn->prepare("UPDATE Product SET `Status\_ID`='2' WHERE Product\_ID = ?");

    $stmt->bind\_param("s", $product\_id);

    if ($stmt->execute()) {

        echo "<script>alert(\"Товаруспешноудален\"); location.href='admin-panel.php';</script>";

        exit();

    } else {

        echo "<script>alert(\"Ошибкаприудалениитовара\");</script>";

   }

   $stmt->close();

}

?>

Формаcard.php (Корзина)

<?php

require "connectbd.php";

$idUser = $\_COOKIE['idUser'];

$query = mysqli\_query($conn, "SELECT \* FROM User WHERE User\_ID = '$idUser'");

$user = mysqli\_fetch\_assoc($query);

?>

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

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

<link rel='stylesheet' href='style3.css'>

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=REM:ital,wght@0,100..900;1,100..900&family=Rubik:ital,wght@0,300..900;1,300..900&display=swap" rel="stylesheet">

<title>FLOWERS</title>

</head>

<body>

    <h1>Вашакорзина</h1>

    <div class="contain">

    <div id="items"></div>

    <div class="wrapper">

        <form action="profile-php.php" method="post">

            <h1>Profile</h1>

            <div class="input-box">

                <input type="text" name='email' readonly value="<?=$user['Email'] ?>">

                <i class='bx bxs-user'></i>

            </div>

            <div class="input-box">

                <input type="text" name='name' readonly value="<?=$user['Name'] ?>" >

                <i class='bx bxs-user'></i>

            </div>

            <div class="input-box">

                <input type="text" name='surname' readonly value="<?=$user['Surname'] ?>">

                <i class='bx bxs-user'></i>

            </div>

            <div class="input-box">

                <input type="text" name='patronymic' readonly value="<?=$user['Patronymic'] ?>">

                <i class='bx bxs-user'></i>

            </div>

            <div class="input-box">

                <input type="date" name='dateBirthday' readonly value="<?=$user['DateOfBirth'] ?>">

                <i class='bx bxs-calendar'></i>

            </div>

            <div class="input-box">

                <input type="tel" name='numberPhone' readonly value="<?=$user['Phone'] ?>">

                <i class='bx bxs-phone'></i>

            </div>

        </form>

        <a href="index.php"><button type="submit" class="bth">Вернутьсянаглавную</button></a>

        &nbsp;

        <form action="send\_order.php" method="POST" id='orderForm'>

            <input type="hidden" name="orderData" id="orderData"></input>

            <button type="submit" class="bth" name="email\_changes">Подтвердитьзаказ</button>

        </form>

    </div>

    </div>

    <script>

  var selectedItems = JSON.parse(localStorage.getItem('selectedItems')) || [];

  vartotalSum = 0; // Переменная для хранения общей суммы

  // Отображаемихнастранице

  var selectedItemsContainer = document.getElementById('items');

  selectedItems.forEach(function(item) {

    var itemElement = document.createElement('div');

    itemElement.classList.add('card-delivery-item');

    var img = document.createElement('img');

    img.src = item.image;

    img.classList.add('card-delivery-img');

    img.alt = item.title;

    var title = document.createElement('div');

    title.classList.add('card-delivery-title');

    title.textContent = item.title;

    var price = document.createElement('div');

    price.classList.add('card-delivery-price');

    price.textContent = item.price + ' рублей';

    // Преобразуем цену в число и добавляем к общей сумме

    totalSum += parseFloat(item.price);

    selectedItemsContainer.appendChild(itemElement);

    itemElement.appendChild(img);

    itemElement.appendChild(title);

    itemElement.appendChild(price);

  });

  // Создаем элемент для отображения общей суммы

  var totalSumElement = document.createElement('div');

  totalSumElement.textContent = 'Общаясумма: ' + totalSum.toFixed(2) + ' рублей'; // Форматируемсуммудодвухзнаковпослезапятой

  selectedItemsContainer.appendChild(totalSumElement); // Добавляемэлементвконтейнер

  totalSumElement.classList.add('total-sum');

</script>

<script>

document.getElementById('orderForm').addEventListener('submit', function(e) {

  e.preventDefault();

  // Формируем строку для отправки на почту

  var orderDetails = 'Заказ:\n';

  selectedItems.forEach(function(item) {

    orderDetails += item.title + ' - ' + item.price + ' рублей\n';

  });

  orderDetails += 'Общаясумма: ' + totalSum.toFixed(2) + ' рублей\n';

  orderDetails += 'Имя: ' + "<?= $user['Name']?>\n";

  orderDetails += 'Фамилия: ' + "<?= $user['Surname']?>\n";

  orderDetails += 'Телефон: ' + "<?= $user['Phone']?>\n";

  // Добавляем подробности заказа в скрытое поле формы

  document.getElementById('orderData').value = orderDetails;

  // Отправляем форму

  e.target.submit();

});

</script>

</body>

</html>

Форма send\_order.php (Отправка заказа на почту)

<?php

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

  $orderData = $\_POST['orderData'];

  $to = 'missis.fazullina@yandex.ru';

  $subject = 'Новыйзаказ';

  $headers = "From: missis.fazullina@yandex.ru\r\n";

  $headers .= "Content-Type: text/plain; charset=utf-8\r\n";

  if (mail($to, $subject, $orderData, $headers)) {

    echo "<script>alert(\"Заказуспешноотправленнапочту\"); location.href='card.php';</script>";

  } else {

    echo "Не удалось отправить заказ.";

  }

}

?>

Стильstylee.css

.body {

    margin: 0;

    font-family: "Rubik", sans-serif;

    font-optical-sizing: auto;

}

.container {

    margin-top: 10px;

}

.header {

    background-color: #D2B3B9;

    height: 100vh;

    background-size: cover;

}

.header-line {

    padding-top: 10px;

    display: flex;

    align-items: center;

}

.nav-item, .nav-item1 {

    color: white;

    text-decoration: none;

    font-weight: 500;

    display: inline-block;

    margin: 5px 96px;

    text-align: center;

    transition: color 0.5s linear;

}

.nav-item1 {

    font-size: 32px;

}

.nav-item:hover {

    color: #223008;

}

.sale {

    position: relative;

    overflow: hidden;

}

.saleTen {

    position: absolute;

    width: 100vh;

    top: 15px;

    left: 5px;

}

.saleGreen {

    width: 100vh;

}

.coverNILSON {

    overflow: hidden;

    width: 100%;

    max-width: 100%;

    height: 550px;

}

.catalogTitle {

    margin-top: 15px;

    font-size: 32px;

    display: flex;

    justify-content: center;

    color: #aa3048;

}

.card-img {

    width: 200px;

    height: 200px;

    border-radius: 22px;

    border: 2px solid #223008;

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

}

.card-cho {

    width: 200px;

    height: 30px;

}

.card-price, .card-image, .card-title, .card-choose {

    display: flex;

    justify-content: center;

}

.card-title, .card-price, .card-id {

    margin-top: 10px;

    color: #aa3048;

}

.card {

    width: 290px;

    height: 350px;

    background-color: #fff;

}

.cards-holder {

    display: flex;

    flex-wrap: wrap;

    justify-content: center ;

}

.about-img {

    max-width: 1100px;

    height: auto;

    display: block;

    margin: 150px auto 0;

}

.afterAbout {

    position: relative;

    text-align: center;

    color: white;

}

    .after {

        position: absolute;

        background-color: #D2B3B9;

        color: #223008;

        top: 50%;

        left: 50%;

        color: #223008;

        transform: translate(-50%, -50%);

        display: grid;

        grid-template-columns: 1fr 1fr;

        grid-gap: 20px;

        width: 100%;

    }

    .after-item, .after-item1 {

        display: block;

        font-size: 20px;

        margin: 10px 0;

        text-align: left;

        padding-left: 150px;

        font-weight: bold;

    }

    .delivery {

        display: flex;

        align-items: center;

        justify-content: space-between;

    }

    .card-delivery {

        display: flex;

        flex-direction: column;

        align-items: center;

        max-width: 250px;

        margin: auto;

        padding: 20px;

        border: 2px solid #223008;

        border-radius: 22px;

        /\* box-shadow: 0 4px 8px rgba(111, 24, 69, 0.997);  \*/

    }

    .card-delivery-item {

        display: flex;

        align-items: center;

        width: 100%;

        margin-bottom: 10px;

    }

    .card-delivery-img {

        border-radius: 22px;

        width: 50px;

        height: 50px;

        margin-right: 10px;

    }

    .card-delivery-title {

        font-weight: bold;

        font-size: 14px;

        flex-grow: 1;

    }

    .card-delivery-price {

        font-size: 12px;

    }

    .card-delivery-buy {

        width: 200px;

        height: 30px;

        margin-left: 475px;

        margin-top: -30px;

    }

    .card-delivery-card {

        width: 60px;

        height: 20px;

    }

    .button {

        background-color: #223008;

        color: #D2B3B9;

        border: none;

        padding: 5px 10px;

        font-size: 12px;

        cursor: pointer;

        border-radius: 5px;

    }

Стиль style2.css

\* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

body {

    display: flex;

    justify-content: center;

    align-items: center;

    min-height: 100vh;

    background: url(image/fon-auth.jpg) no-repeat;

    background-size: cover;

    background-position: center;

}

.wrapper {

    width: 420px;

    background: transparent;

    border-radius: 2px solid rgba(255, 255, 255, .2);

    backdrop-filter: blur(20px);

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

    color: #fff;

    border-radius: 10px;

    padding: 30px 40px;

}

.wrapper h1 {

    font-size: 36px;

    text-align: center;

}

.wrapper .input-box {

    position: relative;

    width: 100%;

    height: 50px;

    margin: 30px 0;

}

.input-box input {

    width: 100%;

    height: 100%;

    background: transparent;

    border: none;

    outline: none;

    border: 2px solid rgba(255, 255, 255, .2);

    border-radius: 40px;

    font-size: 16px;

    color: #fff;

    padding: 20px 45px 20px 20px;

}

.input-box input::placeholder {

    color: #fff;

}

.input-box i {

    position: absolute;

    right: 20px;

    top: 50%;

    transform: translateY(-50%);

    font-size: 20px;

}

.wrapper .forgot {

    display: flex;

    justify-content: space-between;

    font-size: 14.5px;

    margin: -15px 0 15px;

}

.forgot a {

    color: #fff;

    text-decoration: none;

}

.forgot a:hover {

    text-decoration: underline;

}

.wrapper .bth {

    width: 100%;

    height: 45px;

    background: #fff;

    border: none;

    outline: none;

    border-radius: 40px;

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

    cursor: pointer;

    font-size: 16px;

    color: #333;

    font-weight: 600;

}

.wrapper .register-link {

    font-size: 14.5px;

    text-align: center;

    margin: 20px 0 15px;

}

.register-link p a{

    color: #fff;

    text-decoration: none;

    font-weight: 600;

}

.register-link p a:hover {

    text-decoration: underline;

}

.form-select {

    width: 100%;

    padding: 5px;

    margin-bottom: 5px;

    background-color: #D2B3B9;

    color: white;

    border: #D2B3B9;

    border-radius: 20px;

}

.all-products {

    display: flex;

    justify-content: center;

    flex-wrap: wrap;

    gap: 20px;

}

h1 {

    text-align: center;

    color: white;

    font-style: oblique;

    padding-bottom: 10px;

}

.product-form {

    max-width: 600px;

    margin: 0 auto;

    padding: 20px;

    border-radius: 10px;

}

.product-form input[type="text"],

.product-form textarea {

    width: 100%;

    padding: 5px;

    margin-bottom: 5px;

    border: 1px solid #ccc;

    border-radius: 20px;

    font-size: 16px;

    color: #fff;

    background-color: #D2B3B9;

}

.bth[name="delete\_changes"] {

    width: 100%;

    height: 45px;

    background: #fff;

    border: none;

    outline: none;

    border-radius: 40px;

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

    cursor: pointer;

    font-size: 16px;

    color: #333;

    font-weight: 600;

}

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. [ГОСТ 28195-89](https://gost.ruscable.ru/Index/11/11212.htm) Оценка качества программных средств. Общие положения: межгосударственный стандарт: дата введения 1990-07-01 - <https://docs.cntd.ru/document/1200009135?ysclid=lroox3amxp882669222> – Текст: электронный.
2. [ГОСТ 28806-90](https://gost.ruscable.ru/Index/10/10605.htm)Качество программных средств. Термины и определения: межгосударственный стандарт: дата введения 1992-01-01 - <https://docs.cntd.ru/document/1200009077?ysclid=lrq6to4u1z728775522>– Текст: электронный.
3. [ГОСТ 34.321-96](https://gost.ruscable.ru/Index/6/6808.htm)Информационные технологии. Система стандартов по базам данных. Эталонная модель управления данными: межгосударственный стандарт: дата введения 2001-07-01 - <https://docs.cntd.ru/document/1200017662?ysclid=lrq6wn65z5753511468>–Текст: электронный.
4. [ГОСТ Р ИСО/МЭК 12207-99. Информационная технология. Процессы жизненного цикла программных средств](https://docs.cntd.ru/document/1200009075#7D20K3): межгосударственный стандарт: дата введения 2000-07-01 -https://docs.cntd.ru/document/1200009075?ysclid=lrool2t1vo124212660. – Текст: электронный.
5. Ботрос, С. MySQLпо максимуму / C. Ботрос, Д. Тинли., – 4-е изд., перераб. и доп. – Питер, 2023. – 432 с. – ISBN978-5-4461-2261-5. – Текст: непосредственный.
6. Кириченко, А.В. Web на практике. CSS, HTML, JavaScript, MySQL, PHP для fullstack-разработчиков/ А.В. Кириченко, А.П. Никольский, Е.В. Дубовик., перераб. и доп. – Питер, 2021. – 432 с. – ISBN978-5-94-387271-6. –Текст: непосредственный.
7. Макеев, Г.А. Объектно-ориентированное программирование: с нуля к SOLIDи MVC / Г.А. Макеев., перераб. и доп. – БХВ, 2024. – 272 с. – ISBN978-5-9775-1913-7. –Текст: непосредственный.
8. Документация CSS. – 2024. Текст :электронный. – URL: <https://developer.mozilla.org/ru/docs/Web/CSS/Reference>(дата обращения: 26.01.2024).
9. Документация HTML. – 2024. Текст :электронный. – URL: <https://developer.mozilla.org/ru/docs/Web/HTML>(дата обращения: 26.01.2024).
10. Документация JavaScript.–2024. Текст :электронный. – URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript>(дата обращения: 26.01.2024).
11. Документация MySQL. – 2024. Текст :электронный. – URL: <https://dev.mysql.com/doc/>(дата обращения: 26.01.2024).

1. PK-первичный ключ

   FK-внешний ключ [↑](#footnote-ref-2)