**Государственное бюджетное профессиональное образовательное учреждение**

**«Нижегородский Губернский колледж»**

Методическая комиссия «Информатика и вычислительная техника»

Допущен к защите:

Преподаватели

\_\_\_\_\_\_\_\_\_\_\_\_ Л.В.Мухина,

\_\_\_\_\_\_\_\_\_\_\_\_ Е.П. Голубева,

«21» марта 2025 г.

**ОТЧЕТ** **ПО**

**УЧЕБНОЙ ПРАКТИКЕ**

ПМ.01 РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ

Руководитель: \_\_\_\_\_\_\_\_\_\_ Мухина Л.В., Голубева Е.П. 21.03.2025 г.

Студентка: \_\_\_\_\_\_\_\_\_\_\_\_\_ Любивая АО.А. 21.03.2025 г.

Специальность, группа: 09.02.07, 43П.

Нижний Новгород

2025 г.

1. Создание базы данных

Таблица Kategory

Назначение: Эта таблица предназначена для хранения категорий продуктов. Каждая категория может включать в себя несколько продуктов.

Структура:

id SERIAL PRIMARY KEY: Уникальный идентификатор категории, автоматически увеличивается при добавлении новой записи.

name VARCHAR(100) NOT NULL: Название категории, не может быть пустым.

Связи:

Имеет связь с таблицей Product через поле kategory, что позволяет каждому продукту принадлежать к определенной категории.

Таблица Orders

Назначение: Таблица предназначена для хранения информации о заказах. Каждый заказ может содержать информацию о дате заказа, дате доставки, месте получения и статусе заказа.

Структура:

id SERIAL PRIMARY KEY: Уникальный идентификатор заказа.

date\_order DATE NOT NULL: Дата, когда заказ был сделан, не может быть пустым.

date\_delivery DATE NOT NULL: Дата, когда заказ должен быть доставлен, не может быть пустым.

pick\_up INT NOT NULL: Идентификатор пункта выдачи, где заказ может быть забран, не может быть пустым.

FIO VARCHAR(200): ФИО клиента, может быть пустым.

code INT NOT NULL: Код заказа, не может быть пустым.

status INT NOT NULL: Статус заказа, не может быть пустым.

Связи:

FOREIGN KEY (pick\_up) REFERENCES Pick\_up\_point(id): Связь с таблицей Pick\_up\_point, указывающая на пункт выдачи.

FOREIGN KEY (status) REFERENCES OrderStatus(StatusID): Связь с таблицей OrderStatus, указывающая на статус заказа.

Таблица Product

Назначение: Эта таблица предназначена для хранения информации о продуктах, которые могут быть заказаны. Она включает в себя детали о каждом продукте, такие как название, цена и количество на складе.

Структура:

id SERIAL PRIMARY KEY: Уникальный идентификатор продукта.

article\_number VARCHAR(100) NOT NULL: Артикул продукта, не может быть пустым.

name VARCHAR(255) NOT NULL: Название продукта, не может быть пустым.

unit VARCHAR(50) NOT NULL: Единица измерения продукта (например, "кг", "шт."), не может быть пустым.

cost DECIMAL(19, 4) NOT NULL: Цена продукта, не может быть пустой.

max\_discount\_amount INT: Максимальная сумма скидки для продукта, может быть пустой.

production INT NOT NULL: Идентификатор производства, где был произведен продукт, не может быть пустым.

provider INT NOT NULL: Идентификатор поставщика продукта, не может быть пустым.

kategory INT NOT NULL: Идентификатор категории продукта, не может быть пустым.

current\_discount DECIMAL(19, 4): Текущая скидка на продукт, может быть пустой.

quantity\_in\_stock INT NOT NULL: Количество продукта на складе, не может быть пустым.

description TEXT: Описание продукта, может быть пустым.

image VARCHAR(100): Путь к изображению продукта, может быть пустым.

Связи:

FOREIGN KEY (provider) REFERENCES Provider(id): Связь с таблицей Provider, указывающая на поставщика продукта.

FOREIGN KEY (kategory) REFERENCES Kategory(id): Связь с таблицей Kategory, указывающая на категорию продукта.

FOREIGN KEY (production) REFERENCES Production(id): Связь с таблицей Production, указывающая на производство продукта.

Таблица Production

Назначение: Эта таблица предназначена для хранения информации о производителях продуктов. Каждый производитель может производить несколько продуктов.

Структура:

id SERIAL PRIMARY KEY: Уникальный идентификатор производителя, автоматически увеличивается при добавлении новой записи.

name VARCHAR(100) NOT NULL: Название производителя, не может быть пустым.

Связи:

Имеет связь с таблицей Product через поле production, что позволяет каждому продукту быть связанным с определенным производителем.

Таблица Provider

Назначение: Таблица предназначена для хранения информации о поставщиках продуктов. Поставщики могут предоставлять разные продукты.

Структура:

id SERIAL PRIMARY KEY: Уникальный идентификатор поставщика, автоматически увеличивается при добавлении новой записи.

name VARCHAR(100) NOT NULL: Название поставщика, не может быть пустым.

Связи:

Имеет связь с таблицей Product через поле provider, что позволяет каждому продукту быть связанным с определенным поставщиком.

Таблица Role

Назначение: Эта таблица предназначена для хранения ролей пользователей в системе. Роли могут определять права доступа и функциональные возможности пользователей.

Структура:

RoleID SERIAL PRIMARY KEY: Уникальный идентификатор роли, автоматически увеличивается при добавлении новой записи.

RoleName VARCHAR(100) NOT NULL: Название роли, не может быть пустым.

Связи:

Имеет связь с таблицей Users через поле User Role, что позволяет каждому пользователю иметь определенную роль.

Таблица Users

Назначение: Таблица предназначена для хранения информации о пользователях системы. Каждый пользователь имеет свои данные и роль.

Структура:

User ID int primary key: Уникальный идентификатор пользователя.

User Surname varchar(100) not null: Фамилия пользователя, не может быть пустой.

User Name varchar(100) not null: Имя пользователя, не может быть пустым.

User Patronymic varchar(100) not null: Отчество пользователя, не может быть пустым.

User Login text not null: Логин пользователя, не может быть пустым.

User Password text not null: Пароль пользователя, не может быть пустым.

User Role int not null: Идентификатор роли пользователя, не может быть пустым.

Связи:

foreign key (User Role) references Role(RoleID): Связь с таблицей Role, указывающая на роль пользователя.

Таблица OrderProduct

Назначение: Эта таблица предназначена для хранения информации о продуктах, входящих в заказы. Она связывает заказы с конкретными продуктами и их количеством.

Структура:

id SERIAL PRIMARY KEY: Уникальный идентификатор записи о продукте в заказе, автоматически увеличивается при добавлении новой записи.

OrderID INT NOT NULL: Идентификатор заказа, не может быть пустым.

ProductArticleNumber VARCHAR(100) NOT NULL: Артикул продукта, не может быть пустым.

Quantity INT NOT NULL: Количество данного продукта в заказе, не может быть пустым.

Связи:

FOREIGN KEY (OrderID) REFERENCES Orders(id) ON DELETE CASCADE: Связь с таблицей Orders, указывающая на заказ. При удалении заказа все связанные записи в этой таблице также будут удалены.

FOREIGN KEY (ProductArticleNumber) REFERENCES Product(article\_number) ON DELETE CASCADE: Связь с таблицей Product, указывающая на продукт. При удалении продукта все связанные записи в этой таблице также будут удалены.

Предметная область, основанная на описанных таблицах, охватывает управление продуктами, пользователями и заказами в контексте системы, связанной с продажами или поставками товаров. Вот краткий обзор ключевых аспектов этой области: (см. Рисунок 1 – 10, Приложение А)

1. Продукты

Система управляет информацией о товарах, включая их производителей и поставщиков. Каждому продукту присваивается уникальный артикул, что позволяет отслеживать его наличие и характеристики.

2. Производители и Поставщики

Производители отвечают за создание товаров, а поставщики обеспечивают их доставку. Система позволяет связывать продукты с конкретными производителями и поставщиками, что упрощает управление цепочками поставок.

3. Пользователи и Роли

Система включает пользователей, которые могут выполнять различные функции в зависимости от назначенных ролей. Роли определяют права доступа и функциональные возможности, что позволяет организовать работу команды и контролировать действия пользователей.

4. Заказы

Система также управляет процессом заказов, позволяя отслеживать, какие продукты были заказаны, в каком количестве и кем. Это важно для управления запасами и удовлетворения потребностей клиентов.

5. Связи между элементами

Связи между таблицами обеспечивают целостность данных. Например, удаление продукта автоматически удаляет все связанные с ним записи в заказах, что помогает поддерживать актуальность информации.

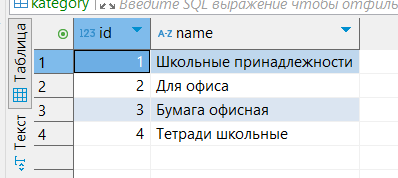


Рисунок №1 – Таблица 1

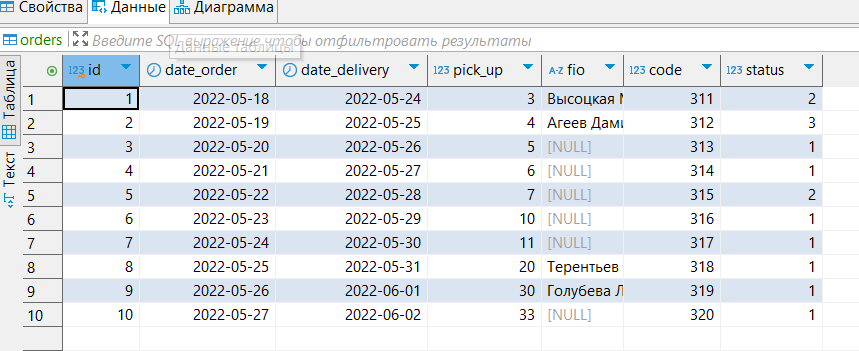


Рисунок №2 – Таблица 2

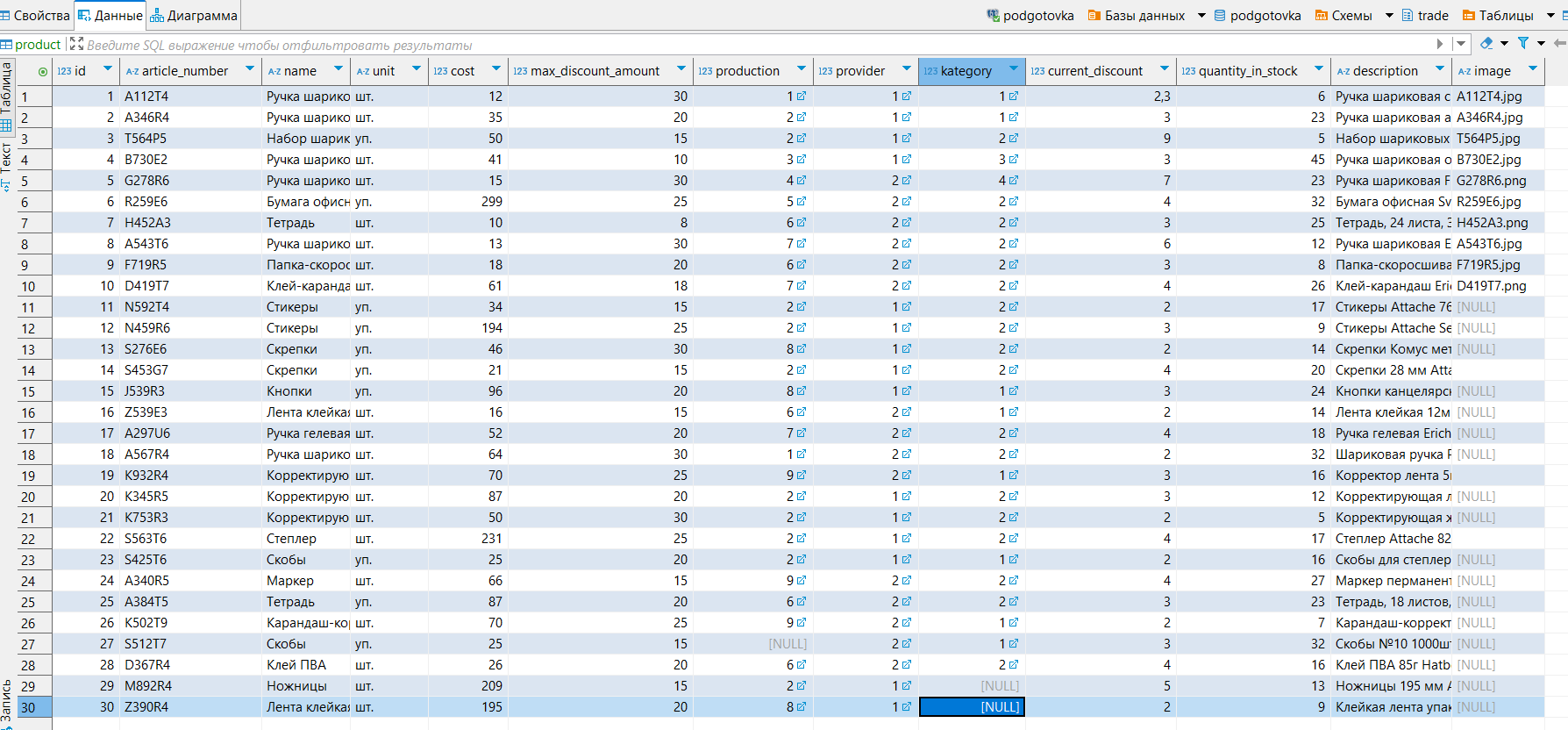


Рисунок №3 – Таблица 3

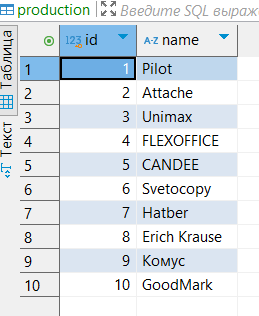


Рисунок №4 – Таблица 4

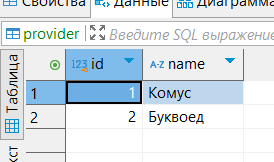


Рисунок №5 – Таблица 5

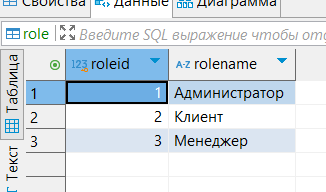


Рисунок №6 – Таблица 6

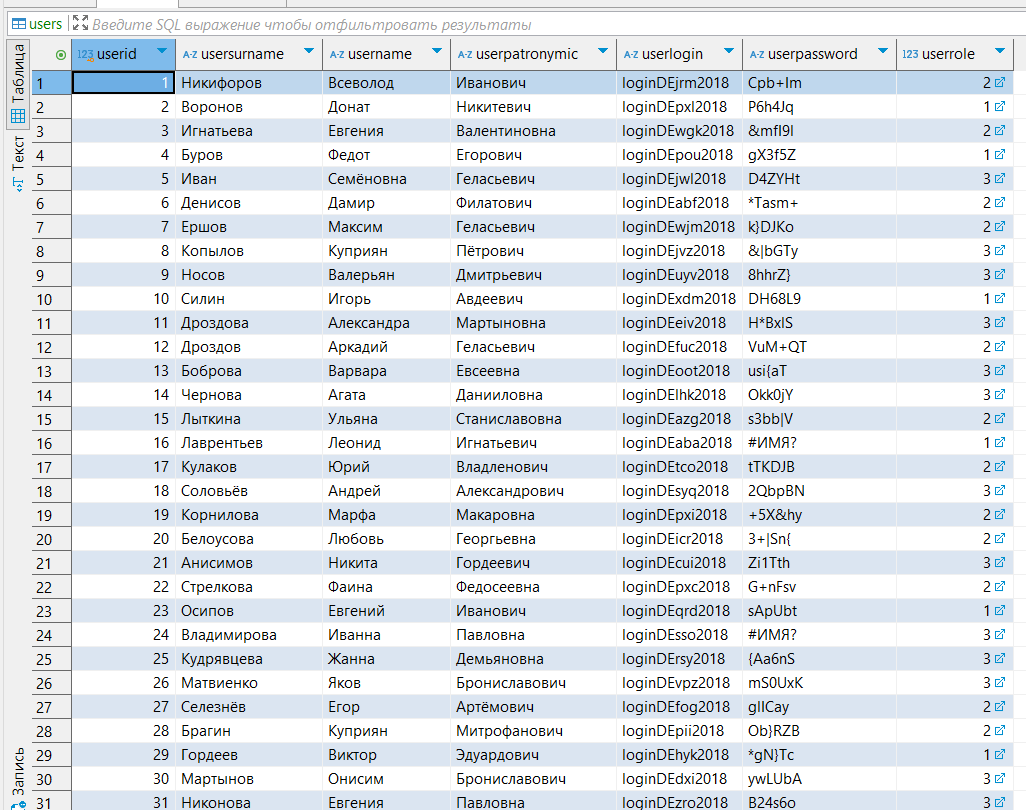


Рисунок №7 – Таблица 7

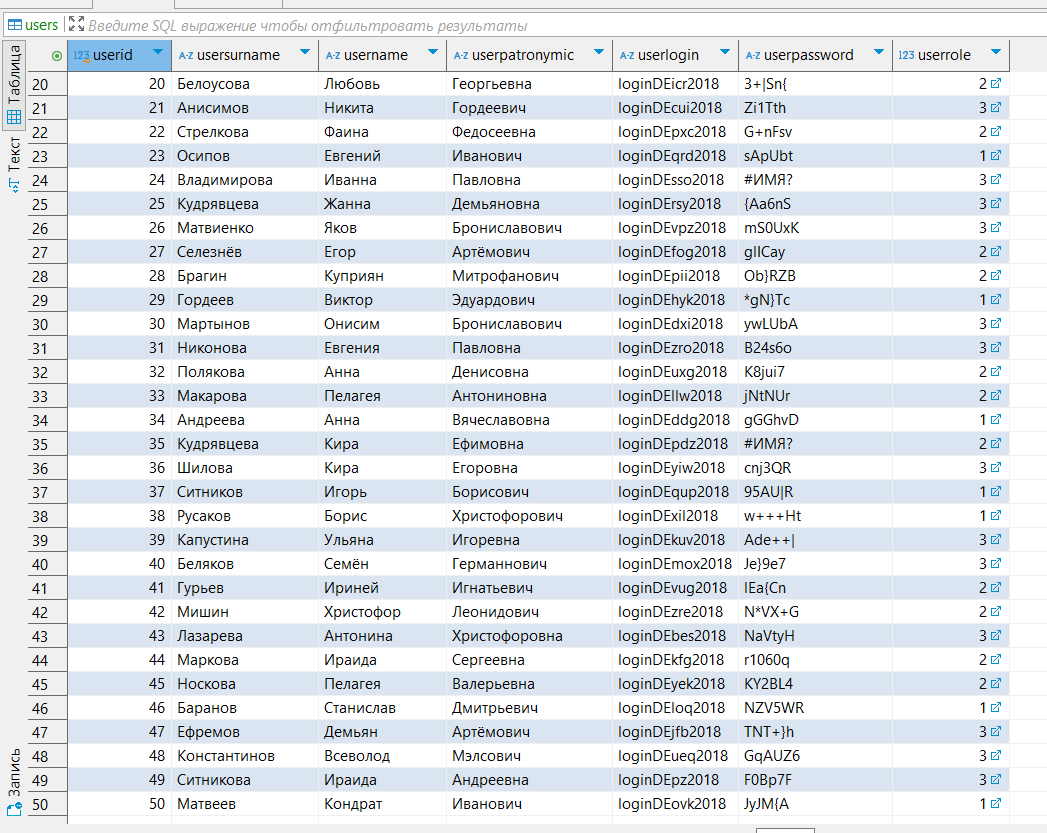


Рисунок №8 – Таблица 8

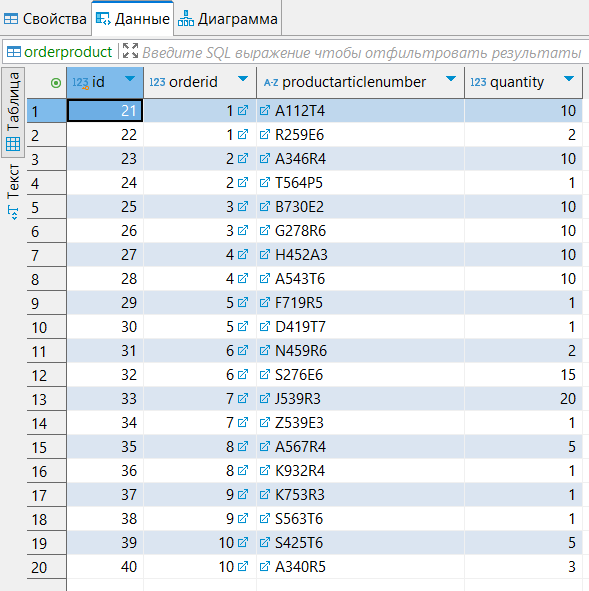


Рисунок №9 – Таблица 9

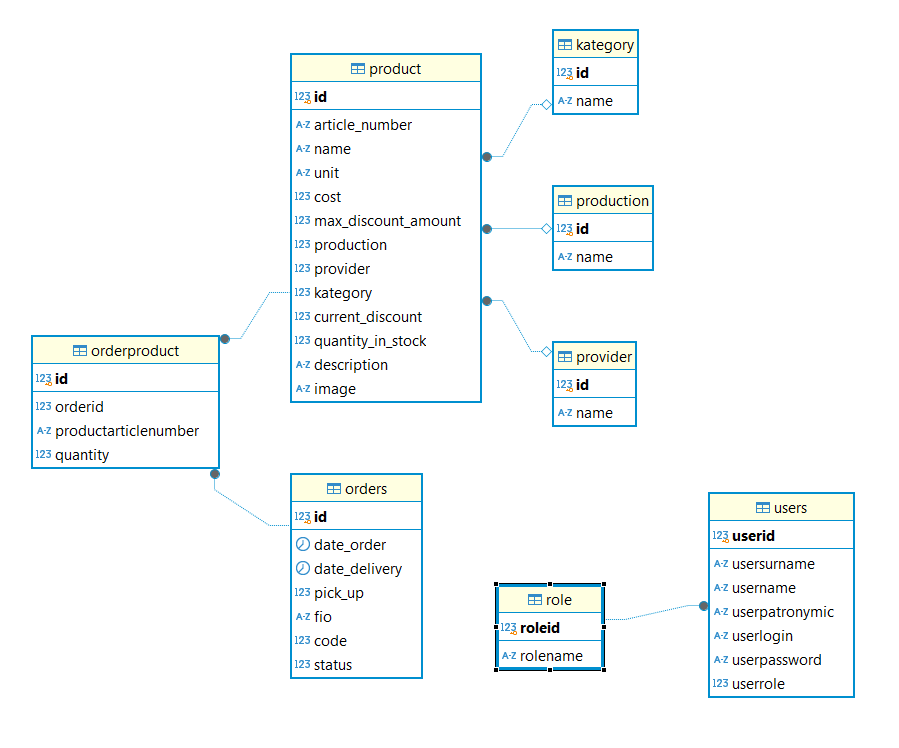


Рисунок №10 – Схема базы данных

1. Разработка информационной системы
   1. Создание проекта

Информационная система разработана на основе фреймворка Avalonia UI, используя архитектурный шаблон MVVM



Рисунок №11 – Название проекта



Рисунок №12 – Иконка

* 1. Создание страницы авторизации

Общая структура

Экран авторизации состоит из двух основных частей:

Заголовок: Отображает текст "Авторизация".

Форма авторизации: Содержит элементы для ввода данных пользователя и кнопки для выполнения действий.

Элементы управления

Заголовок

<TextBlock Grid.Row="0" Text="Авторизация" HorizontalAlignment="Center" FontSize="20" FontWeight="Bold" Margin="10"/>

TextBlock: Отображает заголовок "Авторизация" в центре верхней части экрана. У него задан размер шрифта 20 и жирное начертание.

Основная форма

<ScrollViewer Grid.Row="1" Margin="10" BorderThickness="2" BorderBrush="Green" VerticalScrollBarVisibility="Auto" HorizontalScrollBarVisibility="Disabled">

<Border BorderBrush="Gray" BorderThickness="1" Margin="10" Padding="10" Background="RGB(118, 227, 131)">

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center">

...

</StackPanel>

</Border>

</ScrollViewer>

ScrollViewer: Позволяет прокручивать содержимое, если оно превышает размеры видимой области. У него есть зеленая рамка и возможность вертикальной прокрутки.

Border: Обрамляет содержимое формы, имеет серую рамку и зеленый фон. Это создает визуальный акцент на области авторизации.

StackPanel: Упорядочивает дочерние элементы вертикально и центрирует их.

Элементы ввода

<TextBox x:Name="log" Text="{Binding UserName}" TextWrapping="Wrap" Margin="0,5,0,0" Background="White" Width="200"/>

<TextBox x:Name="par" Text="{Binding Password}" TextWrapping="Wrap" Margin="0,5,0,0" Background="White" Width="200"/>

TextBox для логина (log): Пользователь вводит свое имя пользователя. Связь с User Name из ViewModel позволяет обновлять значение в реальном времени.

TextBox для пароля (par): Пользователь вводит свой пароль. Связь с Password из ViewModel. Для повышения безопасности можно настроить его как PasswordBox.

Кнопки

<Button Command="{Binding ExecuteLogin}" IsEnabled="{Binding EnterButtonEnabled}" Background="RGB(73, 140, 81)" Foreground="White" Width="200" HorizontalContentAlignment="Center">Вход</Button>

<Button Command="{Binding ToShowCommand}" Background="RGB(73, 140, 81)" Foreground="White" Width="200" HorizontalContentAlignment="Center">Войти как гость</Button>

Кнопка "Вход": Выполняет команду ExecuteLogin. Она активна только тогда, когда свойство EnterButtonEnabled равно true.

Кнопка "Войти как гость": Выполняет команду ToShowCommand, позволяя пользователю зайти без авторизации.

Капча

<TextBlock Text="{Binding captcha}" IsVisible="{Binding EnterButtonVisible}" FontWeight="Bold" HorizontalAlignment="Center" Padding="10" />

<TextBox Text="{Binding captchaText}" Watermark="Введите капчу" IsVisible="{Binding EnterButtonVisible}" Background="#E6E6FA" BorderBrush="#9370DB" BorderThickness="2"/>

TextBlock: Отображает текст капчи, который генерируется в ViewModel. Видим только если EnterButtonVisible равно true.

TextBox: Поле для ввода капчи. Пользователь должен ввести текст, чтобы пройти авторизацию. Также видимо только при необходимости.

Логика работы

При загрузке экрана авторизации пользователь видит форму для ввода логина и пароля, а также капчу, если это необходимо.

После ввода данных и нажатия на кнопку "Вход" выполняется проверка введенных данных. Если данные верны и капча введена правильно, происходит переход на следующий экран (например, ShowProduct).

Если авторизация не удалась, отображается сообщение об ошибке, и пользователю предлагается ввести капчу заново.

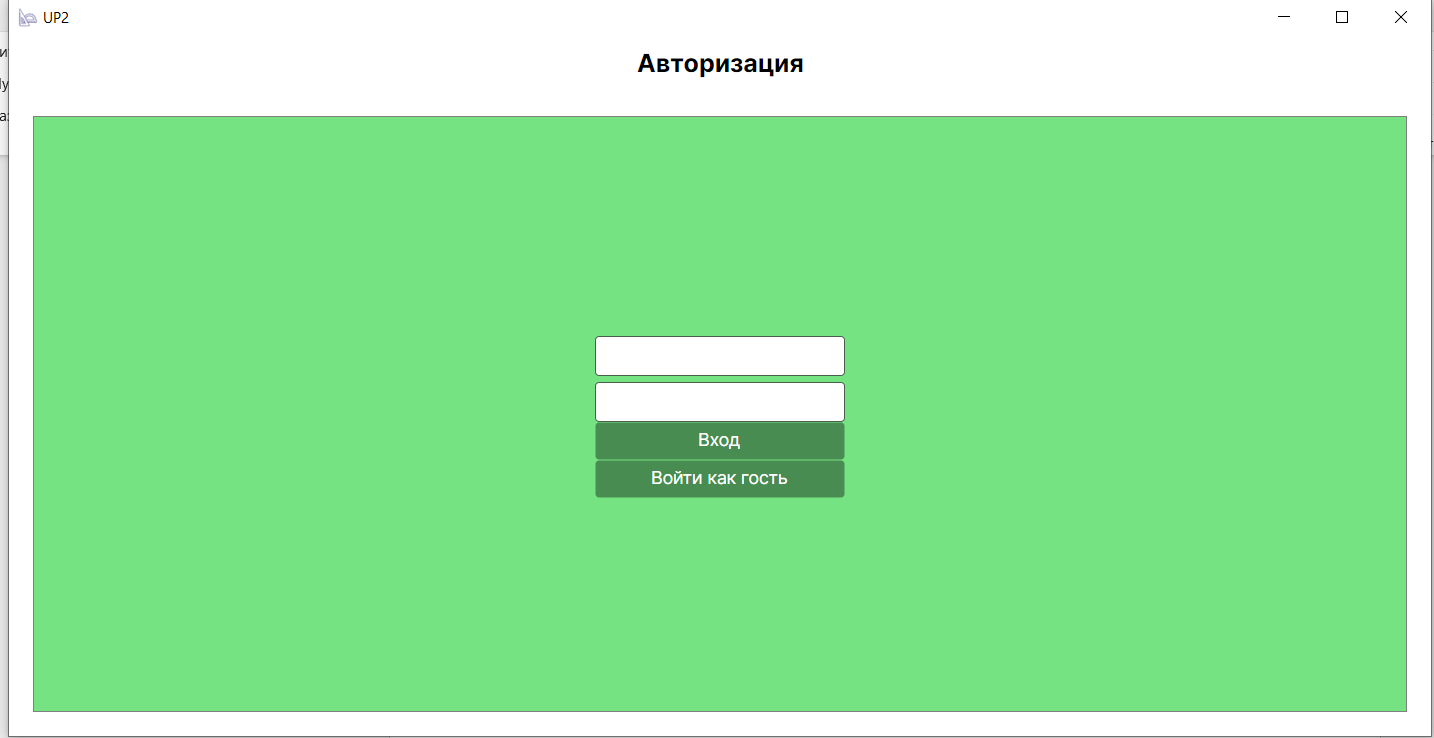


Рисунок №13 – Страница авторизации

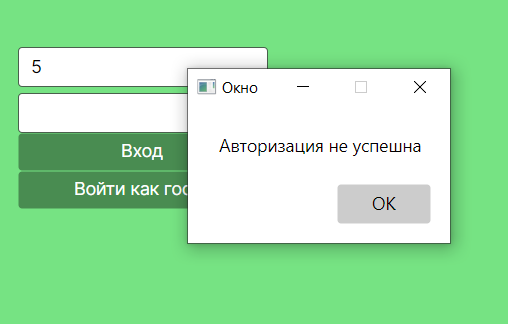


Рисунок №14 – Не успешная авторизация

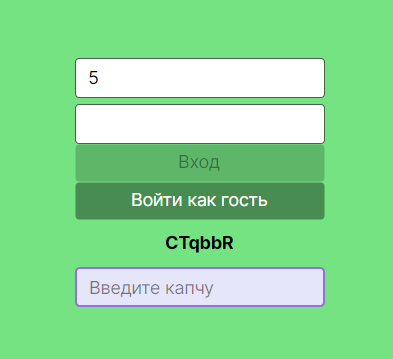


Рисунок №15 – Капча

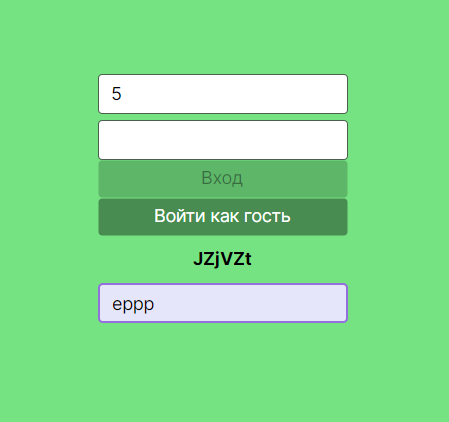


Рисунок №16 – Блокировка кнопки

* 1. Создание страницы вывода продуктов

Приложения

Приложение А

**CREATE** **TABLE** Kategory (

id SERIAL **PRIMARY** **KEY**,

name **VARCHAR**(100) **NOT** **NULL**

);

**CREATE TABLE Orders (**

**id SERIAL PRIMARY KEY,**

**date\_order DATE NOT NULL,**

**date\_delivery DATE NOT NULL,**

**pick\_up INT NOT NULL,**

**FIO VARCHAR(200),**

**code INT NOT NULL,**

**status INT NOT NULL,**

**FOREIGN KEY (pick\_up) REFERENCES Pick\_up\_point(id),**

**FOREIGN KEY (status) REFERENCES OrderStatus(StatusID)**

**);**

**CREATE** **TABLE** Product (

id SERIAL **PRIMARY** **KEY**,

article\_number **VARCHAR**(100) **NOT** **NULL**,

name **VARCHAR**(255) **NOT** **NULL**,

unit **VARCHAR**(50) **NOT** **NULL**,

cost **DECIMAL**(19, 4) **NOT** **NULL**,

max\_discount\_amount **INT**,

production **INT** **NOT** **NULL**,

provider **INT** **NOT** **NULL**,

kategory **INT** **NOT** **NULL**,

current\_discount **DECIMAL**(19, 4),

quantity\_in\_stock **INT** **NOT** **NULL**,

description **TEXT**,

image **VARCHAR**(100),

**FOREIGN** **KEY** (provider) **REFERENCES** Provider(id),

**FOREIGN** **KEY** (kategory) **REFERENCES** Kategory(id),

**FOREIGN** **KEY** (production) **REFERENCES** Production(id)

);

**CREATE** **TABLE** Production (

id SERIAL **PRIMARY** **KEY**,

name **VARCHAR**(100) **NOT** **NULL**

);

**CREATE** **TABLE** Provider (

id SERIAL **PRIMARY** **KEY**,

name **VARCHAR**(100) **NOT** **NULL**

);

**CREATE** **TABLE** Role (

RoleID SERIAL **PRIMARY** **KEY**,

RoleName **VARCHAR**(100) **NOT** **NULL**

);

**create** **table** Users

(

UserID **int** **primary** **key**,

UserSurname **varchar**(100) **not** **null**,

UserName **varchar**(100) **not** **null**,

UserPatronymic **varchar**(100) **not** **null**,

UserLogin **text** **not** **null**,

UserPassword **text** **not** **null**,

UserRole **int** **not** **null**,

**foreign** **key** (UserRole) **references** Role(RoleID)

);

**CREATE** **TABLE** OrderProduct (

id SERIAL **PRIMARY** **KEY**,

OrderID **INT** **NOT** **NULL**,

ProductArticleNumber **VARCHAR**(100) **NOT** **NULL**,

Quantity **INT** **NOT** **NULL**,

**FOREIGN** **KEY** (OrderID) **REFERENCES** Orders(OrderID) **ON** **DELETE** **CASCADE**,

**FOREIGN** **KEY** (ProductArticleNumber) **REFERENCES** Product(article\_number) **ON** **DELETE** **CASCADE**

);