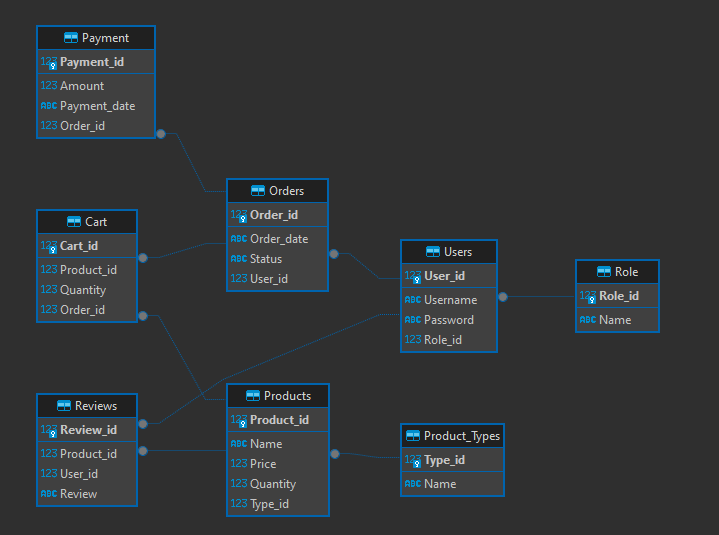
**Приложение А**



**Приложение Б**

CREATE TABLE IF NOT EXISTS Role (

Role\_id INTEGER PRIMARY KEY AUTOINCREMENT,

Name TEXT NOT NULL

);

CREATE TABLE IF NOT EXISTS Product\_Types (

Type\_id INTEGER PRIMARY KEY AUTOINCREMENT,

Name TEXT NOT NULL

);

CREATE TABLE IF NOT EXISTS Users (

User\_id INTEGER PRIMARY KEY AUTOINCREMENT,

Username TEXT NOT NULL,

Password TEXT NOT NULL,

Role\_id INTEGER NOT NULL,

FOREIGN KEY (Role\_id) REFERENCES Role(Role\_id)

);

CREATE TABLE IF NOT EXISTS Products (

Product\_id INTEGER PRIMARY KEY AUTOINCREMENT,

Name TEXT NOT NULL,

Price REAL NOT NULL,

Quantity INTEGER NOT NULL,

Type\_id INTEGER NOT NULL,

FOREIGN KEY (Type\_id) REFERENCES Product\_Types(Type\_id)

);

CREATE TABLE IF NOT EXISTS Orders (

Order\_id INTEGER PRIMARY KEY AUTOINCREMENT,

Order\_date DATE NOT NULL,

Status TEXT NOT NULL,

User\_id INTEGER NOT NULL,

FOREIGN KEY (User\_id) REFERENCES Users(User\_id)

);

CREATE TABLE IF NOT EXISTS Cart (

Cart\_id INTEGER PRIMARY KEY AUTOINCREMENT,

Product\_id INTEGER NOT NULL,

Quantity INTEGER NOT NULL,

Order\_id INTEGER NOT NULL,

FOREIGN KEY (Product\_id) REFERENCES Products(Product\_id),

FOREIGN KEY (Order\_id) REFERENCES Orders(Order\_id)

);

CREATE TABLE IF NOT EXISTS Payment (

Payment\_id INTEGER PRIMARY KEY AUTOINCREMENT,

Amount REAL NOT NULL,

Payment\_date DATE NOT NULL,

Order\_id INTEGER NOT NULL,

FOREIGN KEY (Order\_id) REFERENCES Orders(Order\_id)

);

CREATE TABLE IF NOT EXISTS Reviews (

Review\_id INTEGER PRIMARY KEY AUTOINCREMENT,

Product\_id INTEGER NOT NULL,

User\_id INTEGER NOT NULL,

Review TEXT NOT NULL,

FOREIGN KEY (Product\_id) REFERENCES Products(Product\_id),

FOREIGN KEY (User\_id) REFERENCES Users(User\_id)

);

**Приложение В**

class App:

def \_\_init\_\_(self, root):

self.root = root

self.root.title("Магазин ПК")

self.root.geometry("800x600")

self.create\_login\_page()

def create\_login\_page(self):

self.clear\_frame()

self.login\_frame = ttkb.Frame(self.root, padding="10")

self.login\_frame.pack(expand=True)

ttkb.Label(self.login\_frame, text="Имя пользователя:", anchor="center").pack()

self.username\_entry = ttkb.Entry(self.login\_frame)

self.username\_entry.pack()

ttkb.Label(self.login\_frame, text="Пароль:", anchor="center").pack()

self.password\_entry = ttkb.Entry(self.login\_frame, show="\*")

self.password\_entry.pack()

self.login\_button = ttkb.Button(self.login\_frame, text="Войти", command=self.login)

self.login\_button.pack(pady=10)

self.register\_button = ttkb.Button(self.login\_frame, text="Зарегистрироваться", command=self.create\_register\_page)

self.register\_button.pack()

def create\_register\_page(self):

self.clear\_frame()

self.register\_frame = ttkb.Frame(self.root, padding="10")

self.register\_frame.pack(expand=True)

ttkb.Label(self.register\_frame, text="Имя пользователя:", anchor="center").pack()

self.new\_username\_entry = ttkb.Entry(self.register\_frame)

self.new\_username\_entry.pack()

ttkb.Label(self.register\_frame, text="Пароль:", anchor="center").pack()

self.new\_password\_entry = ttkb.Entry(self.register\_frame, show="\*")

self.new\_password\_entry.pack()

self.register\_button = ttkb.Button(self.register\_frame, text="Зарегистрироваться", command=self.register)

self.register\_button.pack(pady=10)

self.back\_button = ttkb.Button(self.register\_frame, text="Назад", command=self.create\_login\_page)

self.back\_button.pack()

def register(self):

username = self.new\_username\_entry.get()

password = self.new\_password\_entry.get()

cursor.execute("SELECT \* FROM Users WHERE Username = ?", (username,))

if cursor.fetchone():

messagebox.showerror("Ошибка", "Имя пользователя уже существует")

else:

cursor.execute("INSERT INTO Users (Username, Password, Role\_id) VALUES (?, ?, 2)", (username, password))

conn.commit()

messagebox.showinfo("Успех", "Регистрация прошла успешно")

self.create\_login\_page()

def login(self):

username = self.username\_entry.get()

password = self.password\_entry.get()

cursor.execute("SELECT \* FROM Users WHERE Username = ? AND Password = ?", (username, password))

user = cursor.fetchone()

if user:

self.user\_id = user[0]

self.user\_role = user[3]

if self.user\_role == 1:

self.create\_admin\_page()

else:

self.create\_user\_page()

else:

messagebox.showerror("Ошибка", "Неверное имя пользователя или пароль")

def create\_admin\_page(self):

self.clear\_frame()

self.admin\_frame = ttkb.Frame(self.root, padding="10")

self.admin\_frame.pack(expand=True)

self.manage\_products\_button = ttkb.Button(self.admin\_frame, text="Управление товарами", command=self.manage\_products)

self.manage\_products\_button.pack(pady=10)

self.manage\_users\_button = ttkb.Button(self.admin\_frame, text="Управление пользователями", command=self.manage\_users)

self.manage\_users\_button.pack(pady=10)

self.view\_reviews\_button = ttkb.Button(self.admin\_frame, text="Просмотр отзывов", command=self.view\_reviews)

self.view\_reviews\_button.pack(pady=10)

self.logout\_button = ttkb.Button(self.admin\_frame, text="Выйти", command=self.create\_login\_page)

self.logout\_button.pack(pady=10)

def create\_user\_page(self):

self.clear\_frame()

self.user\_frame = ttkb.Frame(self.root, padding="10")

self.user\_frame.pack(expand=True)

self.view\_products\_button = ttkb.Button(self.user\_frame, text="Просмотр товаров", command=self.view\_products)

self.view\_products\_button.pack(pady=10)

self.view\_reviews\_button = ttkb.Button(self.user\_frame, text="Просмотр отзывов", command=self.view\_reviews)

self.view\_reviews\_button.pack(pady=10)

self.leave\_review\_button = ttkb.Button(self.user\_frame, text="Оставить отзыв", command=self.leave\_review)

self.leave\_review\_button.pack(pady=10)

self.logout\_button = ttkb.Button(self.user\_frame, text="Выйти", command=self.create\_login\_page)

self.logout\_button.pack(pady=10)

def manage\_products(self):

self.clear\_frame()

self.manage\_products\_frame = ttkb.Frame(self.root, padding="10")

self.manage\_products\_frame.pack(expand=True)

self.products\_listbox = tk.Listbox(self.manage\_products\_frame)

self.products\_listbox.pack()

cursor.execute("SELECT \* FROM Products")

products = cursor.fetchall()

for product in products:

self.products\_listbox.insert(tk.END, product)

self.add\_product\_button = ttkb.Button(self.manage\_products\_frame, text="Добавить товар", command=self.add\_product)

self.add\_product\_button.pack(pady=10)

self.edit\_product\_button = ttkb.Button(self.manage\_products\_frame, text="Изменить товар", command=self.edit\_product)

self.edit\_product\_button.pack(pady=10)

self.delete\_product\_button = ttkb.Button(self.manage\_products\_frame, text="Удалить товар", command=self.delete\_product)

self.delete\_product\_button.pack(pady=10)

self.back\_button = ttkb.Button(self.manage\_products\_frame, text="Назад", command=self.create\_admin\_page)

self.back\_button.pack(pady=10)

def add\_product(self):

self.clear\_frame()

self.add\_product\_frame = ttkb.Frame(self.root, padding="10")

self.add\_product\_frame.pack(expand=True)

ttkb.Label(self.add\_product\_frame, text="Название:", anchor="center").pack()

self.product\_name\_entry = ttkb.Entry(self.add\_product\_frame)

self.product\_name\_entry.pack()

ttkb.Label(self.add\_product\_frame, text="Цена:", anchor="center").pack()

self.product\_price\_entry = ttkb.Entry(self.add\_product\_frame)

self.product\_price\_entry.pack()

ttkb.Label(self.add\_product\_frame, text="Количество:", anchor="center").pack()

self.product\_quantity\_entry = ttkb.Entry(self.add\_product\_frame)

self.product\_quantity\_entry.pack()

ttkb.Label(self.add\_product\_frame, text="Тип:", anchor="center").pack()

self.product\_type\_entry = ttkb.Entry(self.add\_product\_frame)

self.product\_type\_entry.pack()

self.save\_product\_button = ttkb.Button(self.add\_product\_frame, text="Сохранить", command=self.save\_product)

self.save\_product\_button.pack(pady=10)

self.back\_button = ttkb.Button(self.add\_product\_frame, text="Назад", command=self.manage\_products)

self.back\_button.pack(pady=10)

def save\_product(self):

name = self.product\_name\_entry.get()

price = float(self.product\_price\_entry.get())

quantity = int(self.product\_quantity\_entry.get())

type\_id = int(self.product\_type\_entry.get())

cursor.execute("INSERT INTO Products (Name, Price, Quantity, Type\_id) VALUES (?, ?, ?, ?)", (name, price, quantity, type\_id))

conn.commit()

messagebox.showinfo("Успех", "Товар добавлен успешно")

self.manage\_products()

def edit\_product(self):

selected\_product = self.products\_listbox.curselection()

if selected\_product:

product\_id = self.products\_listbox.get(selected\_product[0])[0]

self.clear\_frame()

self.edit\_product\_frame = ttkb.Frame(self.root, padding="10")

self.edit\_product\_frame.pack(expand=True)

cursor.execute("SELECT \* FROM Products WHERE Product\_id = ?", (product\_id,))

product = cursor.fetchone()

ttkb.Label(self.edit\_product\_frame, text="Название:", anchor="center").pack()

self.product\_name\_entry = ttkb.Entry(self.edit\_product\_frame)

self.product\_name\_entry.insert(0, product[1])

self.product\_name\_entry.pack()

ttkb.Label(self.edit\_product\_frame, text="Цена:", anchor="center").pack()

self.product\_price\_entry = ttkb.Entry(self.edit\_product\_frame)

self.product\_price\_entry.insert(0, product[2])

self.product\_price\_entry.pack()

ttkb.Label(self.edit\_product\_frame, text="Количество:", anchor="center").pack()

self.product\_quantity\_entry = ttkb.Entry(self.edit\_product\_frame)

self.product\_quantity\_entry.insert(0, product[3])

self.product\_quantity\_entry.pack()

ttkb.Label(self.edit\_product\_frame, text="Тип:", anchor="center").pack()

self.product\_type\_entry = ttkb.Entry(self.edit\_product\_frame)

self.product\_type\_entry.insert(0, product[4])

self.product\_type\_entry.pack()

self.save\_changes\_button = ttkb.Button(self.edit\_product\_frame, text="Сохранить изменения", command=lambda: self.save\_changes(product\_id))

self.save\_changes\_button.pack(pady=10)

self.back\_button = ttkb.Button(self.edit\_product\_frame, text="Назад", command=self.manage\_products)

self.back\_button.pack(pady=10)

else:

messagebox.showerror("Ошибка", "Выберите товар для редактирования")

def save\_changes(self, product\_id):

name = self.product\_name\_entry.get()

price = float(self.product\_price\_entry.get())

quantity = int(self.product\_quantity\_entry.get())

type\_id = int(self.product\_type\_entry.get())

cursor.execute("UPDATE Products SET Name = ?, Price = ?, Quantity = ?, Type\_id = ? WHERE Product\_id = ?", (name, price, quantity, type\_id, product\_id))

conn.commit()

messagebox.showinfo("Успех", "Товар изменен успешно")

self.manage\_products()

def delete\_product(self):

selected\_product = self.products\_listbox.curselection()

if selected\_product:

product\_id = self.products\_listbox.get(selected\_product[0])[0]

cursor.execute("DELETE FROM Products WHERE Product\_id = ?", (product\_id,))

conn.commit()

messagebox.showinfo("Успех", "Товар удален успешно")

self.manage\_products()

else:

messagebox.showerror("Ошибка", "Выберите товар для удаления")

def manage\_users(self):

self.clear\_frame()

self.manage\_users\_frame = ttkb.Frame(self.root, padding="10")

self.manage\_users\_frame.pack(expand=True)

self.users\_listbox = tk.Listbox(self.manage\_users\_frame)

self.users\_listbox.pack()

cursor.execute("SELECT \* FROM Users WHERE Role\_id = 2")

users = cursor.fetchall()

for user in users:

self.users\_listbox.insert(tk.END, user)

self.delete\_user\_button = ttkb.Button(self.manage\_users\_frame, text="Удалить пользователя", command=self.delete\_user)

self.delete\_user\_button.pack(pady=10)

self.back\_button = ttkb.Button(self.manage\_users\_frame, text="Назад", command=self.create\_admin\_page)

self.back\_button.pack(pady=10)

def delete\_user(self):

selected\_user = self.users\_listbox.curselection()

if selected\_user:

user\_id = self.users\_listbox.get(selected\_user[0])[0]

cursor.execute("DELETE FROM Users WHERE User\_id = ?", (user\_id,))

conn.commit()

messagebox.showinfo("Успех", "Пользователь удален успешно")

self.manage\_users()

else:

messagebox.showerror("Ошибка", "Выберите пользователя для удаления")

def view\_products(self):

self.clear\_frame()

self.view\_products\_frame = ttkb.Frame(self.root, padding="10")

self.view\_products\_frame.pack(expand=True)

self.products\_listbox = tk.Listbox(self.view\_products\_frame)

self.products\_listbox.pack()

cursor.execute("SELECT \* FROM Products")

products = cursor.fetchall()

for product in products:

self.products\_listbox.insert(tk.END, product)

self.buy\_product\_button = ttkb.Button(self.view\_products\_frame, text="Купить товар", command=self.buy\_product)

self.buy\_product\_button.pack(pady=10)

self.back\_button = ttkb.Button(self.view\_products\_frame, text="Назад", command=self.create\_user\_page)

self.back\_button.pack(pady=10)

def buy\_product(self):

selected\_product = self.products\_listbox.curselection()

if selected\_product:

product = self.products\_listbox.get(selected\_product[0])

self.clear\_frame()

self.buy\_product\_frame = ttkb.Frame(self.root, padding="10")

self.buy\_product\_frame.pack(expand=True)

ttkb.Label(self.buy\_product\_frame, text=f"Товар: {product[1]}", anchor="center").pack()

ttkb.Label(self.buy\_product\_frame, text=f"Цена: {product[2]}", anchor="center").pack()

ttkb.Label(self.buy\_product\_frame, text="Количество:", anchor="center").pack()

self.product\_quantity\_entry = ttkb.Entry(self.buy\_product\_frame)

self.product\_quantity\_entry.pack()

ttkb.Label(self.buy\_product\_frame, text="Данные карты:", anchor="center").pack()

self.card\_data\_entry = ttkb.Entry(self.buy\_product\_frame)

self.card\_data\_entry.pack()

self.confirm\_purchase\_button = ttkb.Button(self.buy\_product\_frame, text="Подтвердить покупку", command=lambda: self.confirm\_purchase(product))

self.confirm\_purchase\_button.pack(pady=10)

self.back\_button = ttkb.Button(self.buy\_product\_frame, text="Назад", command=self.view\_products)

self.back\_button.pack(pady=10)

else:

messagebox.showerror("Ошибка", "Выберите товар для покупки")

def confirm\_purchase(self, product):

quantity = int(self.product\_quantity\_entry.get())

card\_data = self.card\_data\_entry.get()

if quantity <= product[3]:

new\_quantity = product[3] - quantity

cursor.execute("UPDATE Products SET Quantity = ? WHERE Product\_id = ?", (new\_quantity, product[0]))

conn.commit()

cursor.execute("INSERT INTO Orders (Order\_date, Status, User\_id) VALUES (DATE('now'), 'New', ?)", (self.user\_id,))

order\_id = cursor.lastrowid

cursor.execute("INSERT INTO Cart (Product\_id, Quantity, Order\_id) VALUES (?, ?, ?)", (product[0], quantity, order\_id))

cursor.execute("INSERT INTO Payment (Amount, Payment\_date, Order\_id) VALUES (?, DATE('now'), ?)", (product[2] \* quantity, order\_id))

conn.commit()

messagebox.showinfo("Успех", "Покупка успешно завершена")

self.view\_products()

else:

messagebox.showerror("Ошибка", "Недостаточное количество товара")

def view\_reviews(self):

self.clear\_frame()

self.view\_reviews\_frame = ttkb.Frame(self.root, padding="10")

self.view\_reviews\_frame.pack(expand=True)

self.reviews\_listbox = tk.Listbox(self.view\_reviews\_frame)

self.reviews\_listbox.pack()

cursor.execute("SELECT \* FROM Reviews")

reviews = cursor.fetchall()

for review in reviews:

self.reviews\_listbox.insert(tk.END, review)

self.back\_button = ttkb.Button(self.view\_reviews\_frame, text="Назад", command=self.create\_admin\_page if self.user\_role == 1 else self.create\_user\_page)

self.back\_button.pack(pady=10)

if self.user\_role == 1:

self.delete\_review\_button = ttkb.Button(self.view\_reviews\_frame, text="Удалить отзыв", command=self.delete\_review)

self.delete\_review\_button.pack(pady=10)

def delete\_review(self):

selected\_review = self.reviews\_listbox.curselection()

if selected\_review:

review\_id = self.reviews\_listbox.get(selected\_review[0])[0]

cursor.execute("DELETE FROM Reviews WHERE Review\_id = ?", (review\_id,))

conn.commit()

messagebox.showinfo("Успех", "Отзыв удален успешно")

self.view\_reviews()

else:

messagebox.showerror("Ошибка", "Выберите отзыв для удаления")

def leave\_review(self):

self.clear\_frame()

self.leave\_review\_frame = ttkb.Frame(self.root, padding="10")

self.leave\_review\_frame.pack(expand=True)

ttkb.Label(self.leave\_review\_frame, text="ID товара:", anchor="center").pack()

self.product\_id\_entry = ttkb.Entry(self.leave\_review\_frame)

self.product\_id\_entry.pack()

ttkb.Label(self.leave\_review\_frame, text="Отзыв:", anchor="center").pack()

self.review\_entry = ttkb.Entry(self.leave\_review\_frame)

self.review\_entry.pack()

self.submit\_review\_button = ttkb.Button(self.leave\_review\_frame, text="Оставить отзыв", command=self.submit\_review)

self.submit\_review\_button.pack(pady=10)

self.back\_button = ttkb.Button(self.leave\_review\_frame, text="Назад", command=self.create\_user\_page)

self.back\_button.pack(pady=10)

def submit\_review(self):

product\_id = int(self.product\_id\_entry.get())

review = self.review\_entry.get()

cursor.execute("INSERT INTO Reviews (Product\_id, User\_id, Review) VALUES (?, ?, ?)", (product\_id, self.user\_id, review))

conn.commit()

messagebox.showinfo("Успех", "Отзыв оставлен успешно")

self.create\_user\_page()

def clear\_frame(self):

for widget in self.root.winfo\_children():

widget.destroy()

root = ttkb.Window(themename="darkly")

app = App(root)

root.mainloop()

**Приложение Г**

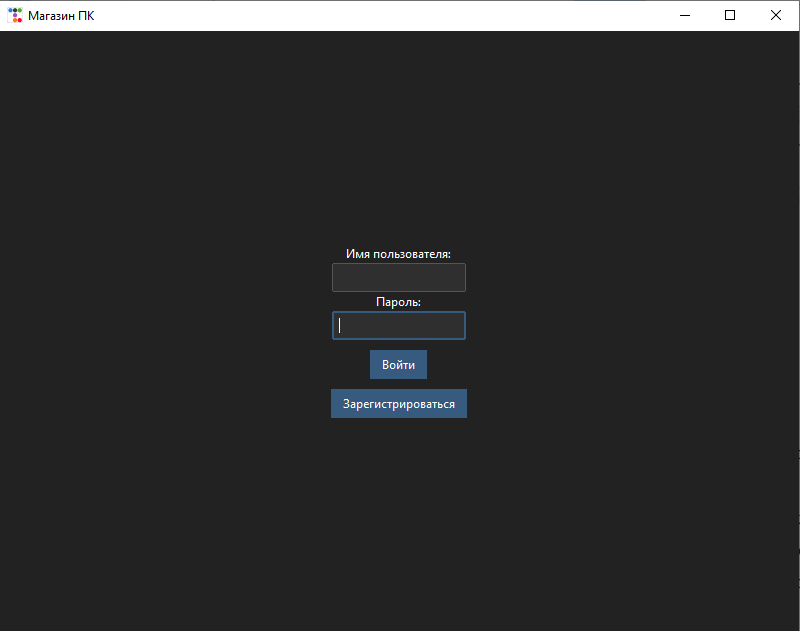


Рисунок 1 – Вид основного окна до авторизации.

Если у пользователя нет акккаунт, то он может зарегистрироваться

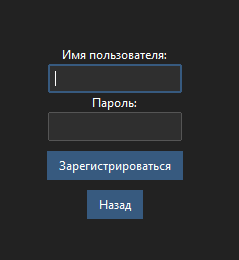


Рисунок 2 – Окно регистрации.

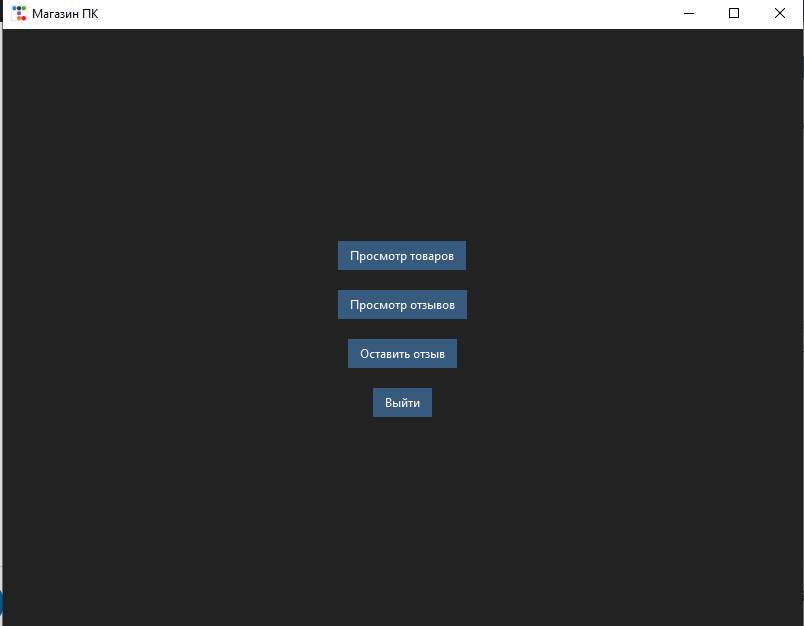


Рисунок 3 – Профиль пользователя

Работа с базой данных

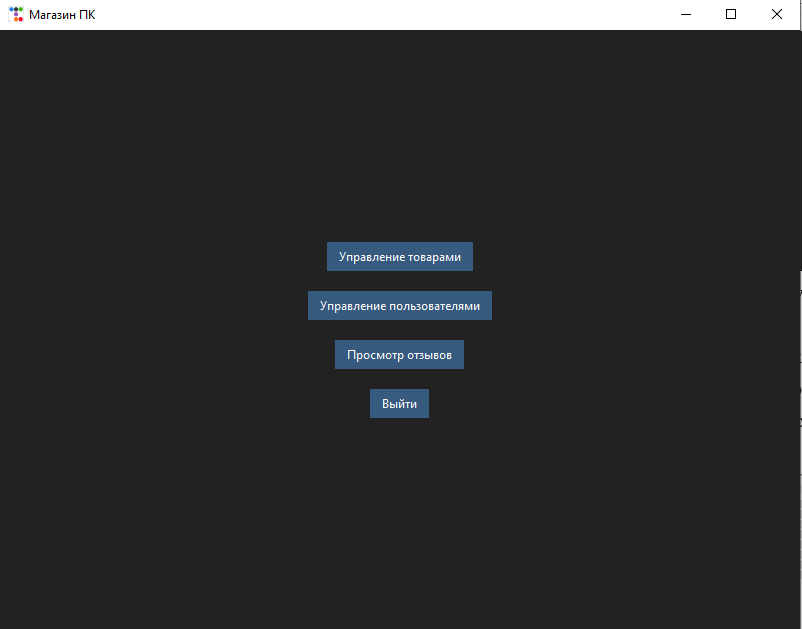


Рисунок 4 – Меню управлений

Можно добавить товар

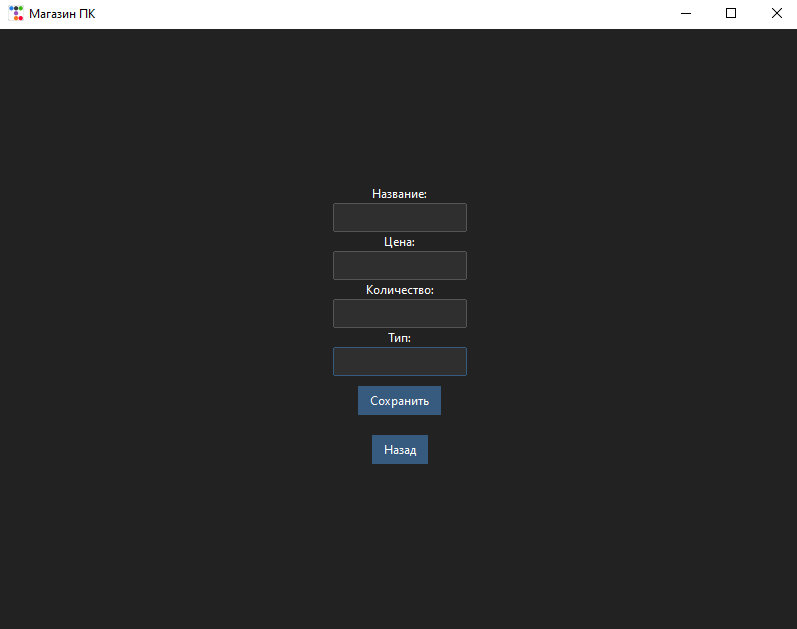


Рисунок 5 – Добавление товара

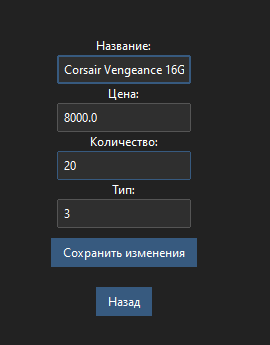


Рисунок 6 – Изменение товара

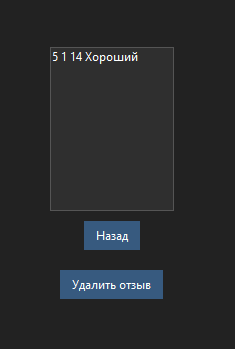


Рисунок 7 – Просмотр отзывов

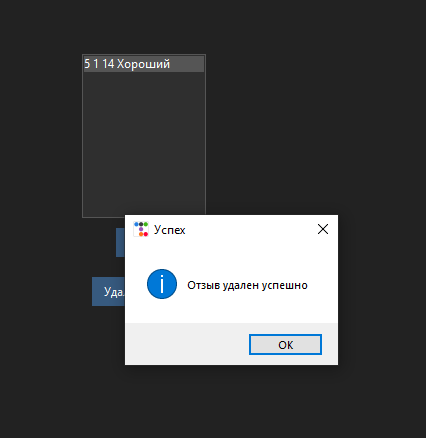


Рисунок 8 – Удаление отзыва

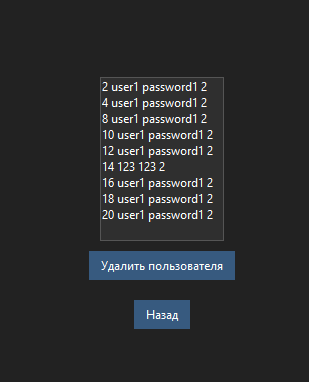


Рисунок 9 – Просмотр зарегистрированных пользователей

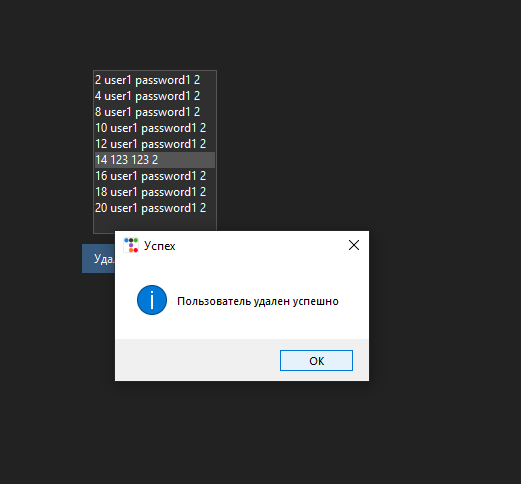


Рисунок 10 – Удаление отзыва