

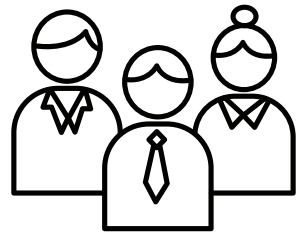
2023 Feb 23



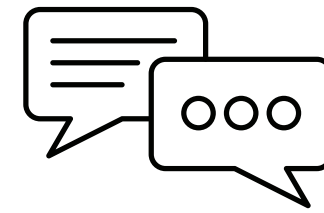
SNEAKER STORE

Nurbek  
Arukhan  
IT-2103

# Aim and goals



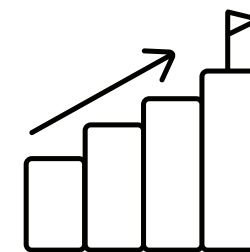
To **showcase** the products that the store sells, including images and descriptions of different sneakers.



To provide customer service, including information about **returns, exchanges, and shipping.**



To allow customers to browse and shop for sneakers **online**. To provide customers with information about **sales and promotions.**



To **build brand awareness** and establish the store as a go-to destination for sneaker enthusiasts.

# Relevance of "WalKing"

## **E-commerce:**

A website allows customers to purchase sneakers online, without having to physically visit the store. This is especially useful for customers who live far away or have busy schedules.

## **Brand presence:**

A website allows a sneaker store to establish a brand presence and showcase its products to a wider audience.

## **Product information:**

A website can provide detailed information about the sneakers available in the store, including their features, sizes, and prices.

## **Social media integration:**

A website can be integrated with social media platforms such as Instagram and Facebook, allowing the store to reach a larger audience and engage with customers.

# Similar applications



## **Youmarket**

Designed for sneaker collectors and enthusiasts in Kazakhstan, allowing them to browse and purchase sneakers from a variety of different stores and brands.

Features:

Provides access to a community of sneaker enthusiasts and allows users to create and share their own sneaker collections.



## **SneakerHead**

The SneakerHead app provides access to the latest sneaker news and releases in Kazakhstan, as well as the ability to purchase sneakers from a variety of different stores and brands.

Features:

It also allows users to create and share their own sneaker collections.



## **SneakerTown**

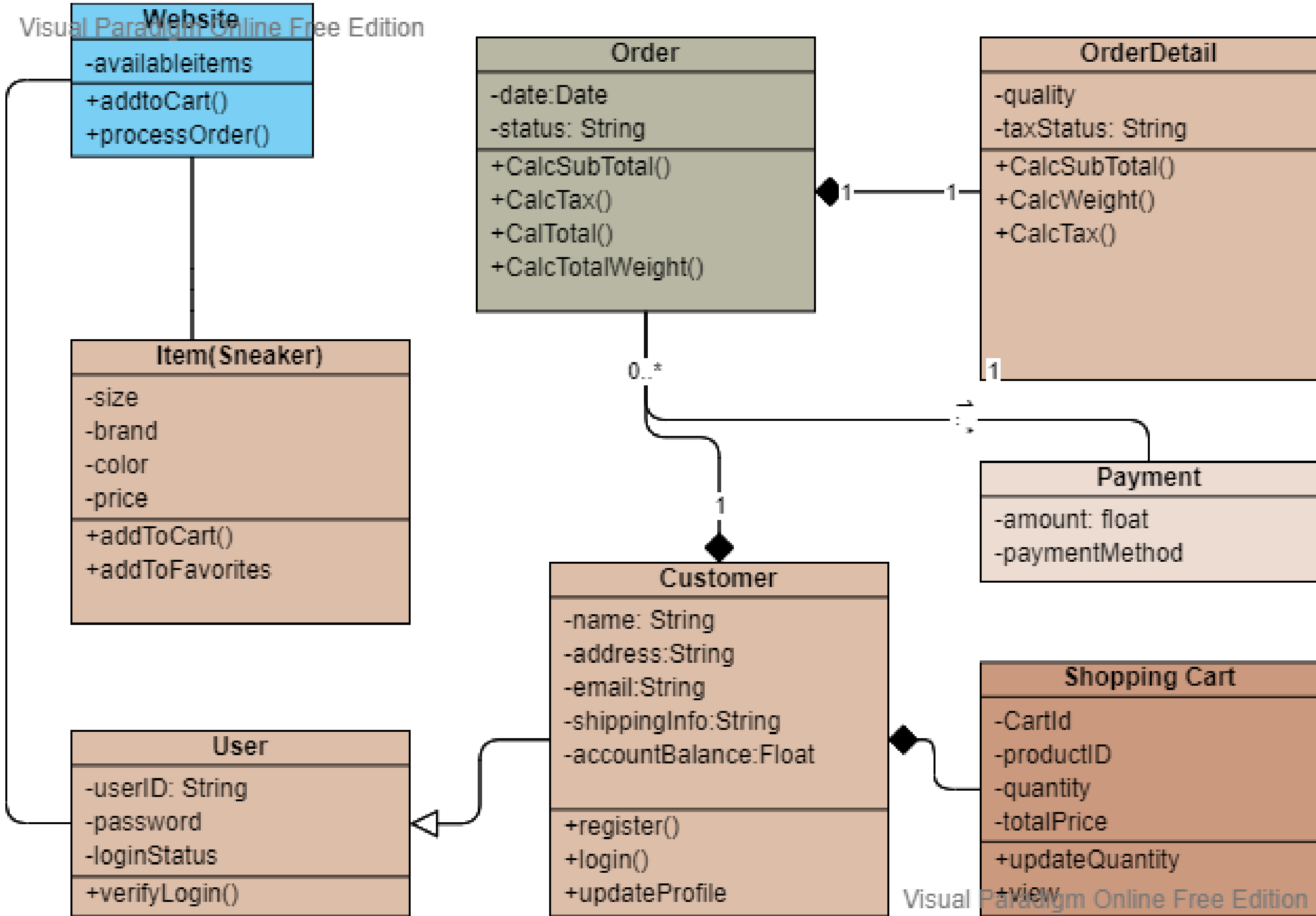
Allows customers in Kazakhstan to browse and purchase sneakers from a variety of different stores and brands.

Features

It also provides access to a community of sneaker enthusiasts and allows users to sell their own sneakers.

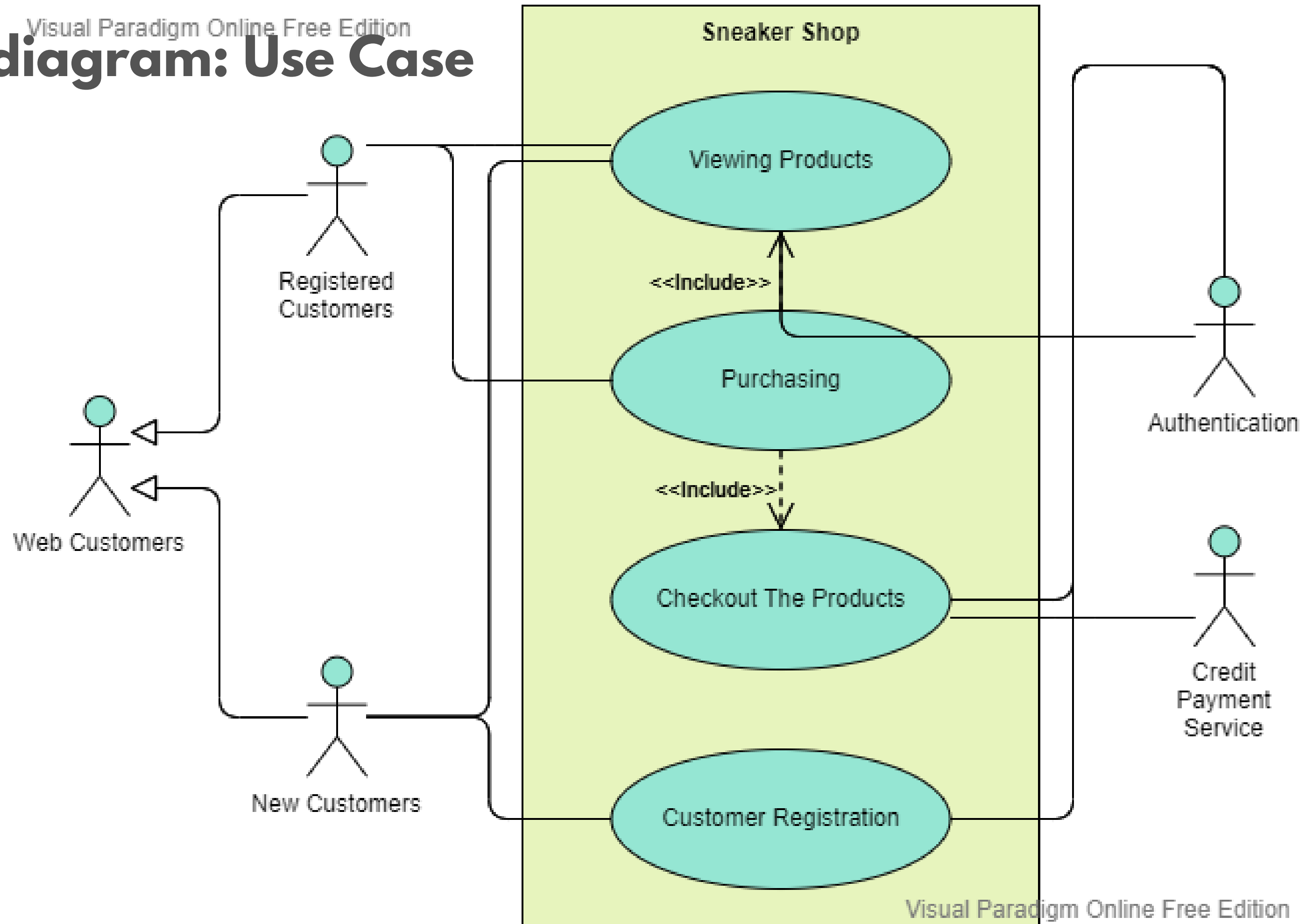


# UML diagram: Class

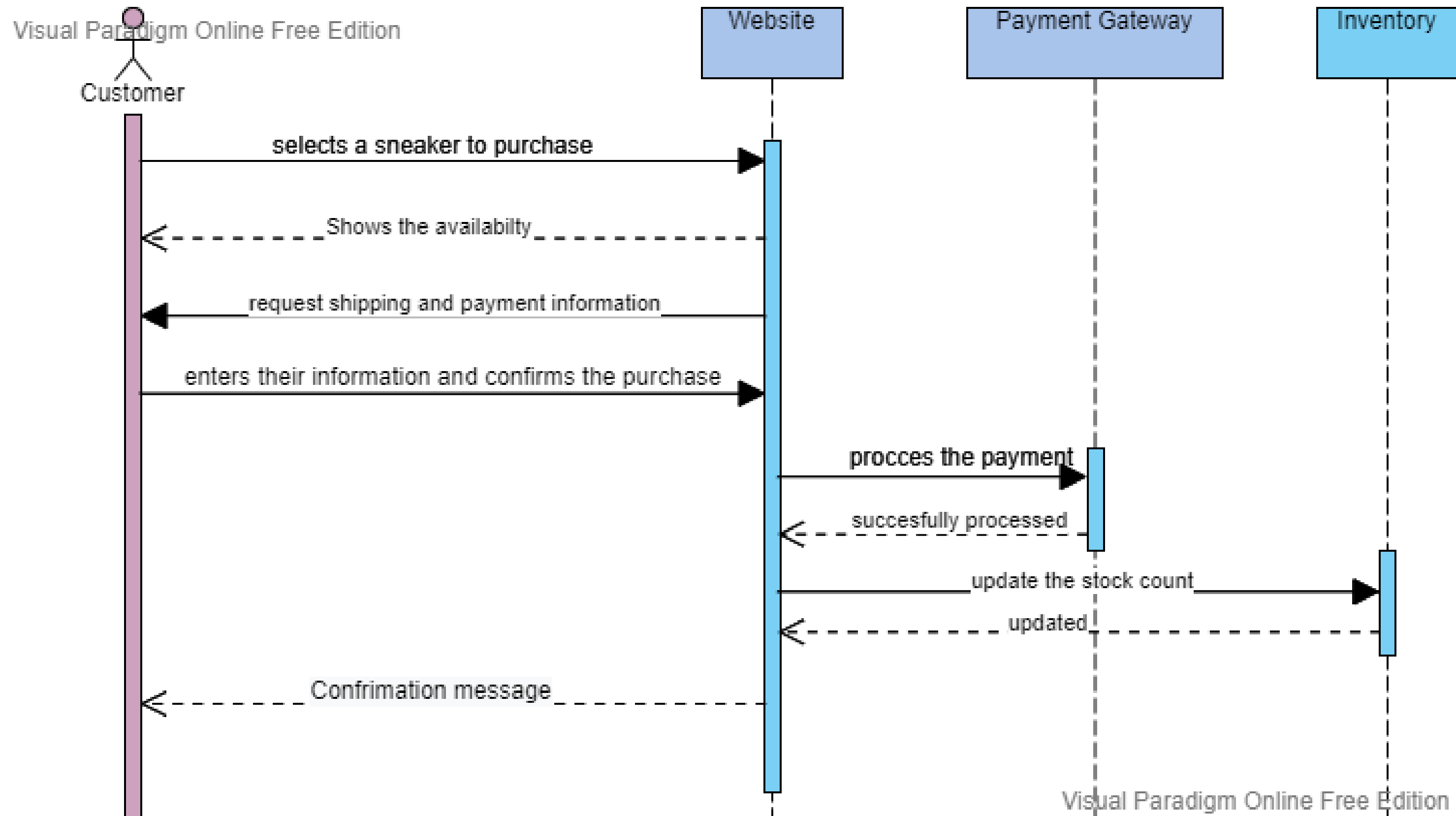


Visual Paradigm Online Free Edition

# UML diagram: Use Case

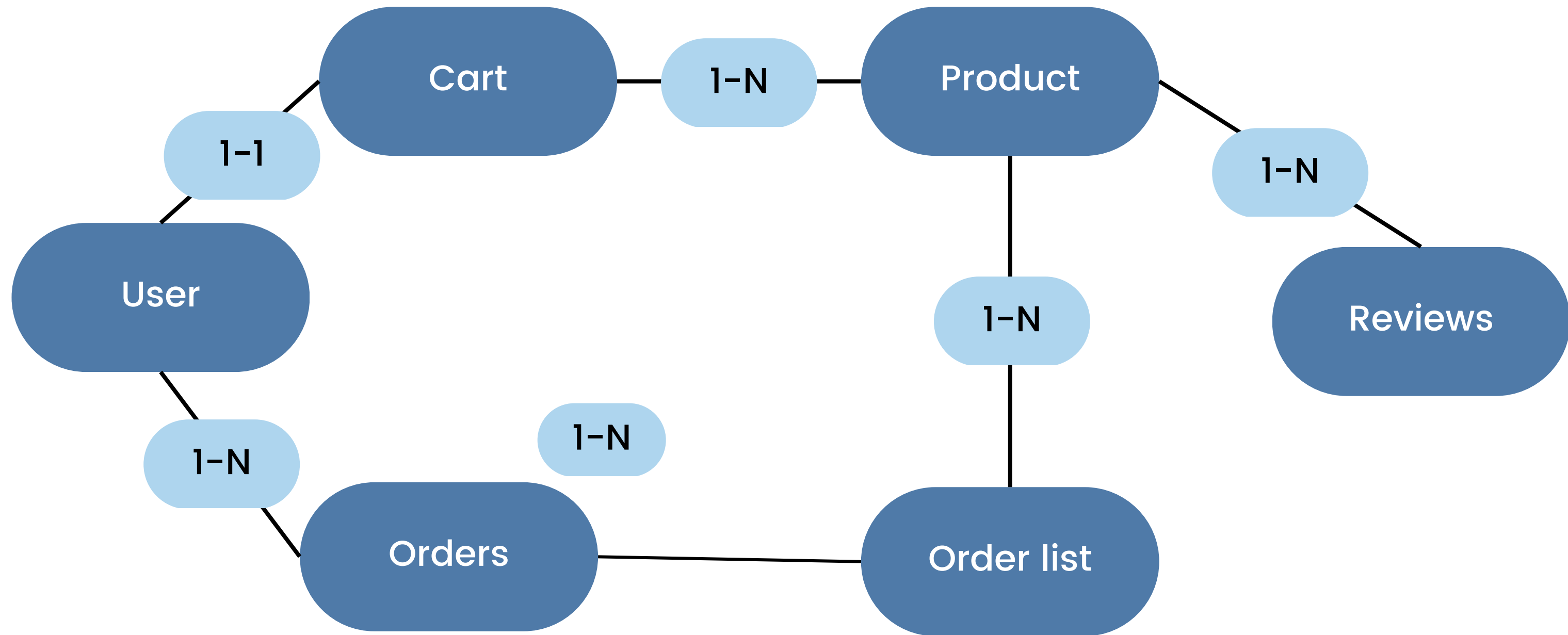


# UML diagram: Sequence



# Data Schemas and modeling

## Entity-Relationship diagram





# Data Schemas and modeling

Collection-Relationship Diagrams



# Data collection

There are a few different types of data that want to collect for a sneaker store website:

## Product data

This includes information about each sneaker that is available for purchase on the website, such as the brand, model, color, size options, price, and any relevant product descriptions or photos.

## Customer data

This includes information about the individuals who visit and make purchases on the website, such as their name, email address, shipping address, and purchase history.

## Sales data

This includes information about the sales that are made on the website, such as the date of the sale, the products that were purchased, and the total amount of the sale.

# Node JS

Here is an example of how you could implement CRUD (Create, Read, Update, Delete) operations for a sneakers store website using Node.js and MongoDB.

To create a new sneaker, you can use the `insertOne` method of the collection object:

```
async function createSneaker(sneaker) {  
  const result = await collection.insertOne(sneaker);  
  return result.insertedId;  
}
```

To read a sneaker, you can use the `findOne` method of the collection object:

```
async function readSneaker(id) {  
  const sneaker = await collection.findOne({ _id: id });  
  return sneaker;  
}
```

# Node JS

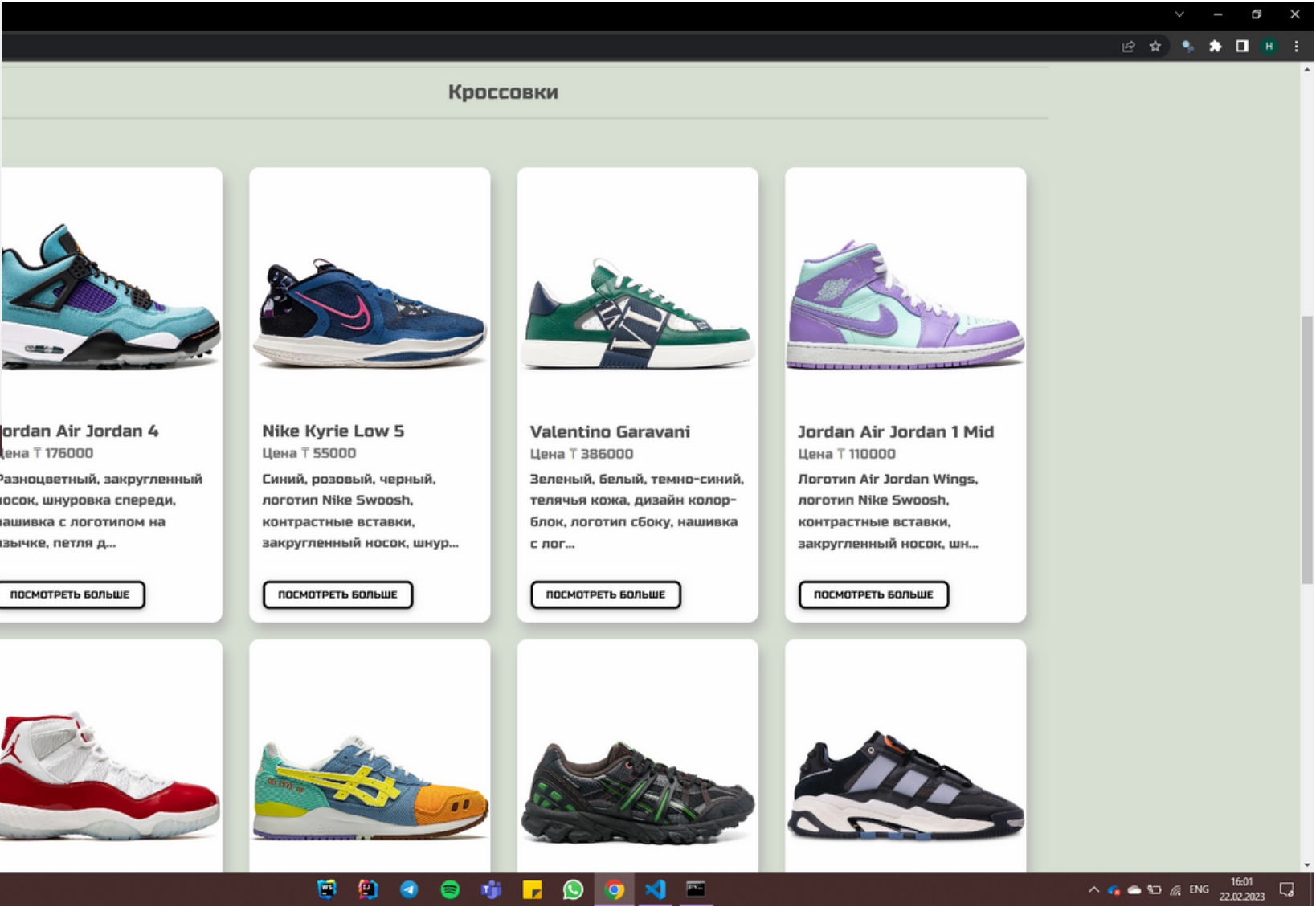
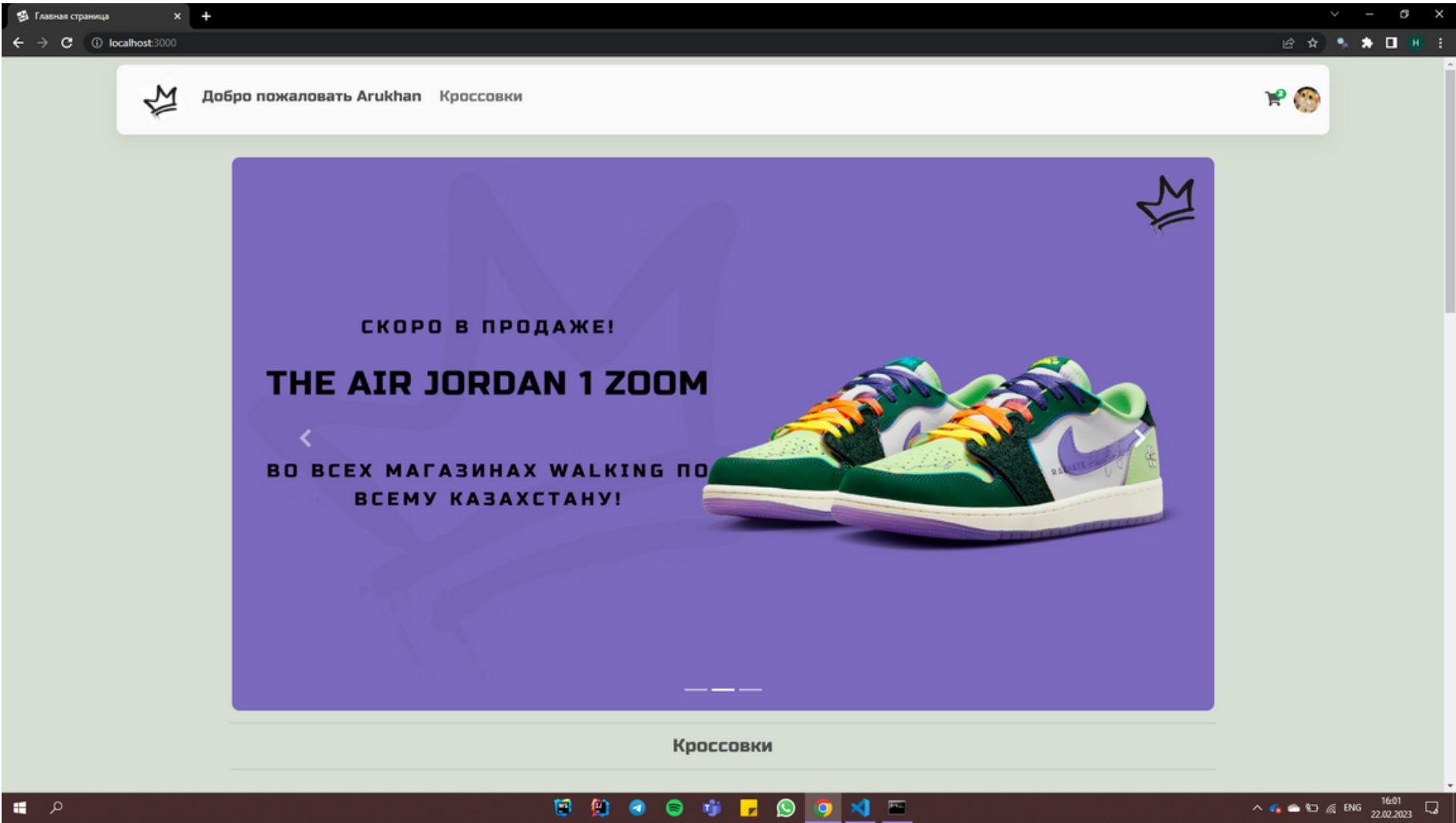
To update a sneaker, you can use the updateOne method of the collection object:

```
async function updateSneaker(id, sneaker) {  
  const result = await collection.updateOne({ _id: id }, { $set:  
sneaker });  
  return result.modifiedCount;  
}
```

To delete a sneaker, you can use the deleteOne method of the collection object:

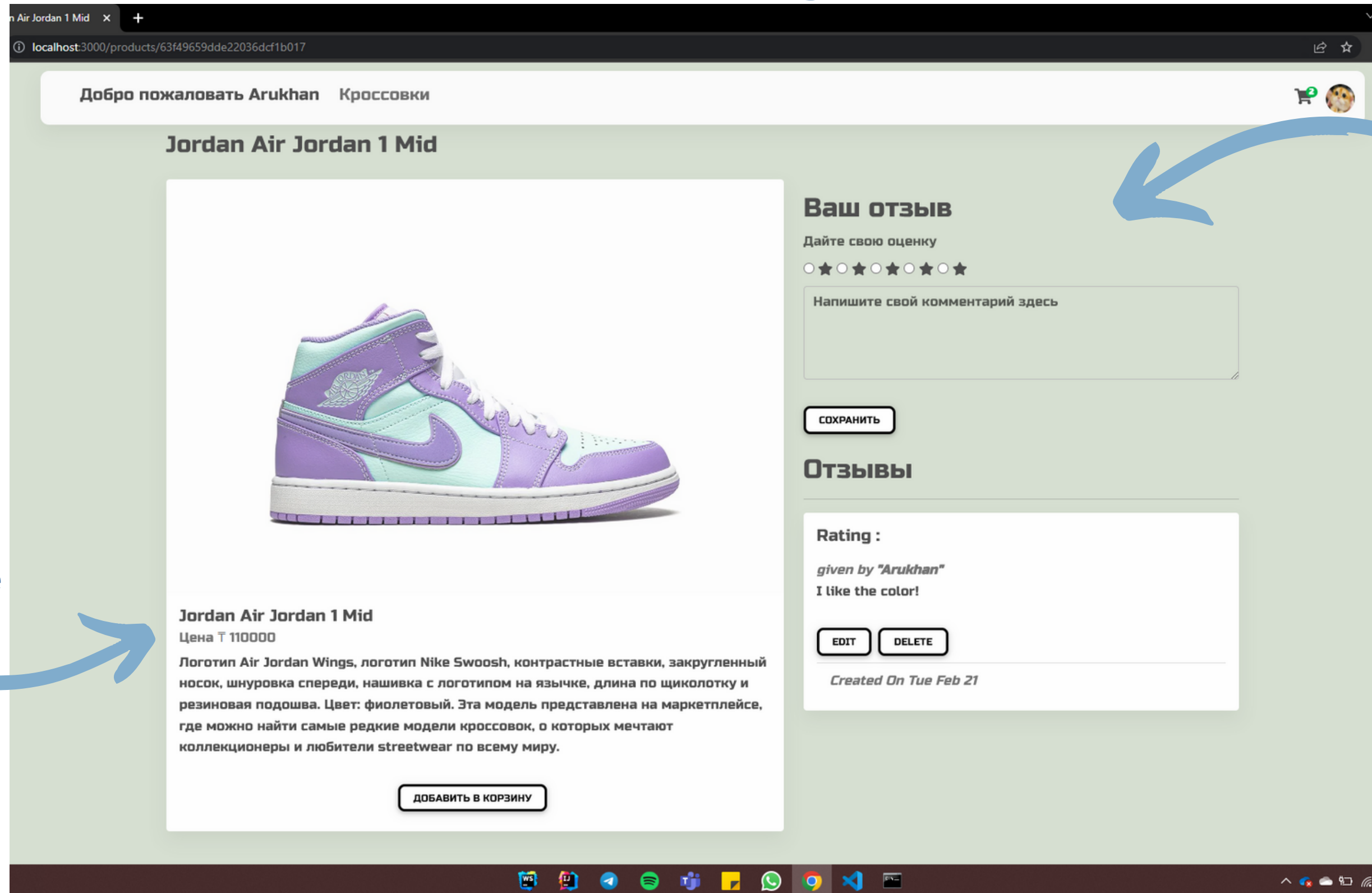
```
async function deleteSneaker(id) {  
  const result = await collection.deleteOne({ _id: id });  
  return result.deletedCount;  
}
```

# Results (Screenshots)



# Home Page

# Item Page



Name and Price

Review

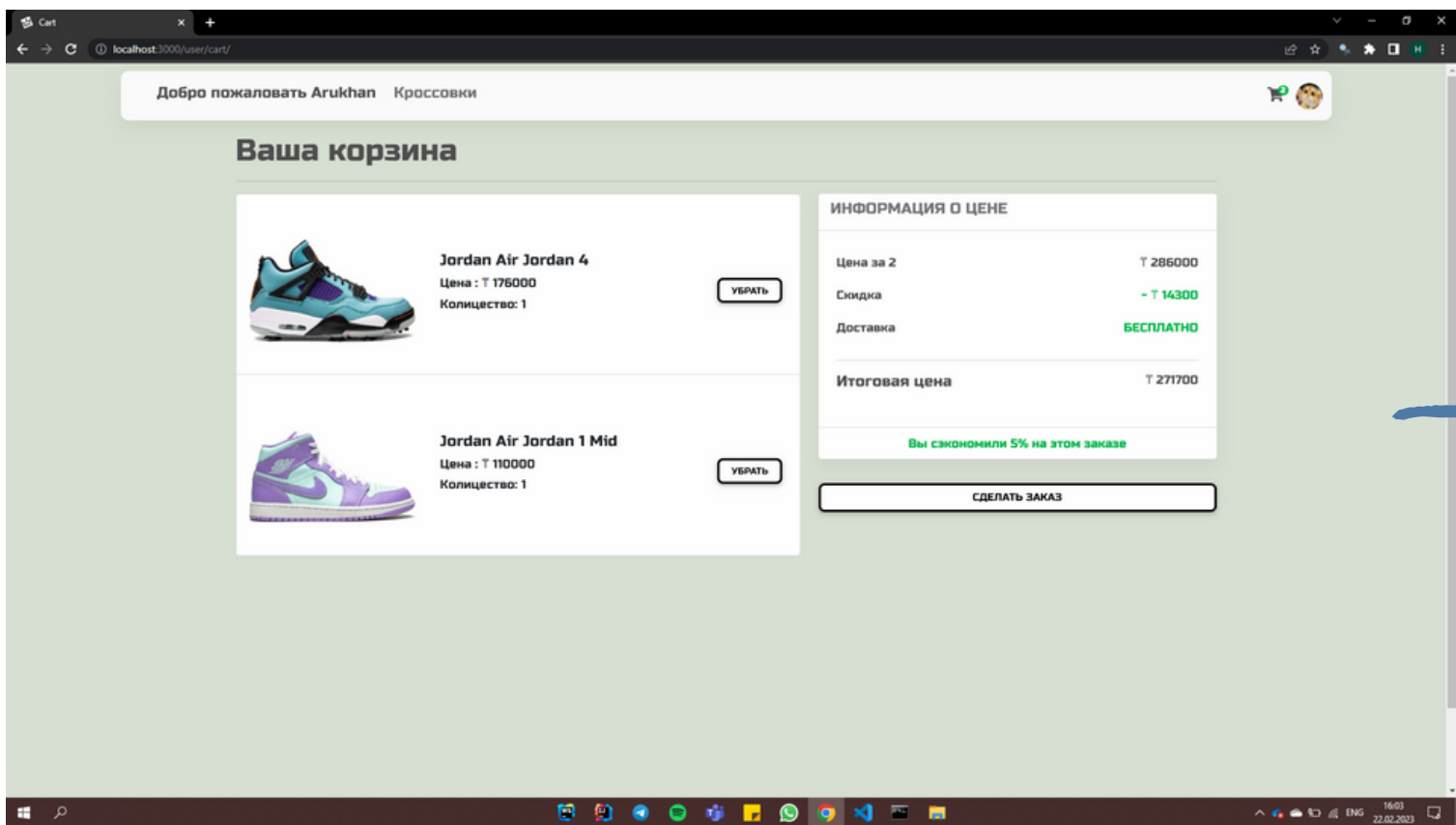
# Login and Register pages

The screenshot shows a web browser window with the address bar displaying 'localhost:3000/login'. The page has a light green background. At the top, there is a white header bar with a crown icon and the text 'Кроссовки' on the left, and a shopping cart icon with a green notification bubble and a user profile icon on the right. The main heading is 'Войдите в свой аккаунт' (Log in to your account). Below it are two input fields: 'Имя пользователя' (Username) and 'Пароль' (Password). A 'войти' (Log in) button is positioned below the password field. At the bottom, there is a link: 'Новый пользователь? [Нажмите здесь, чтобы зарегистрироваться](#)' (New user? Click here to register).

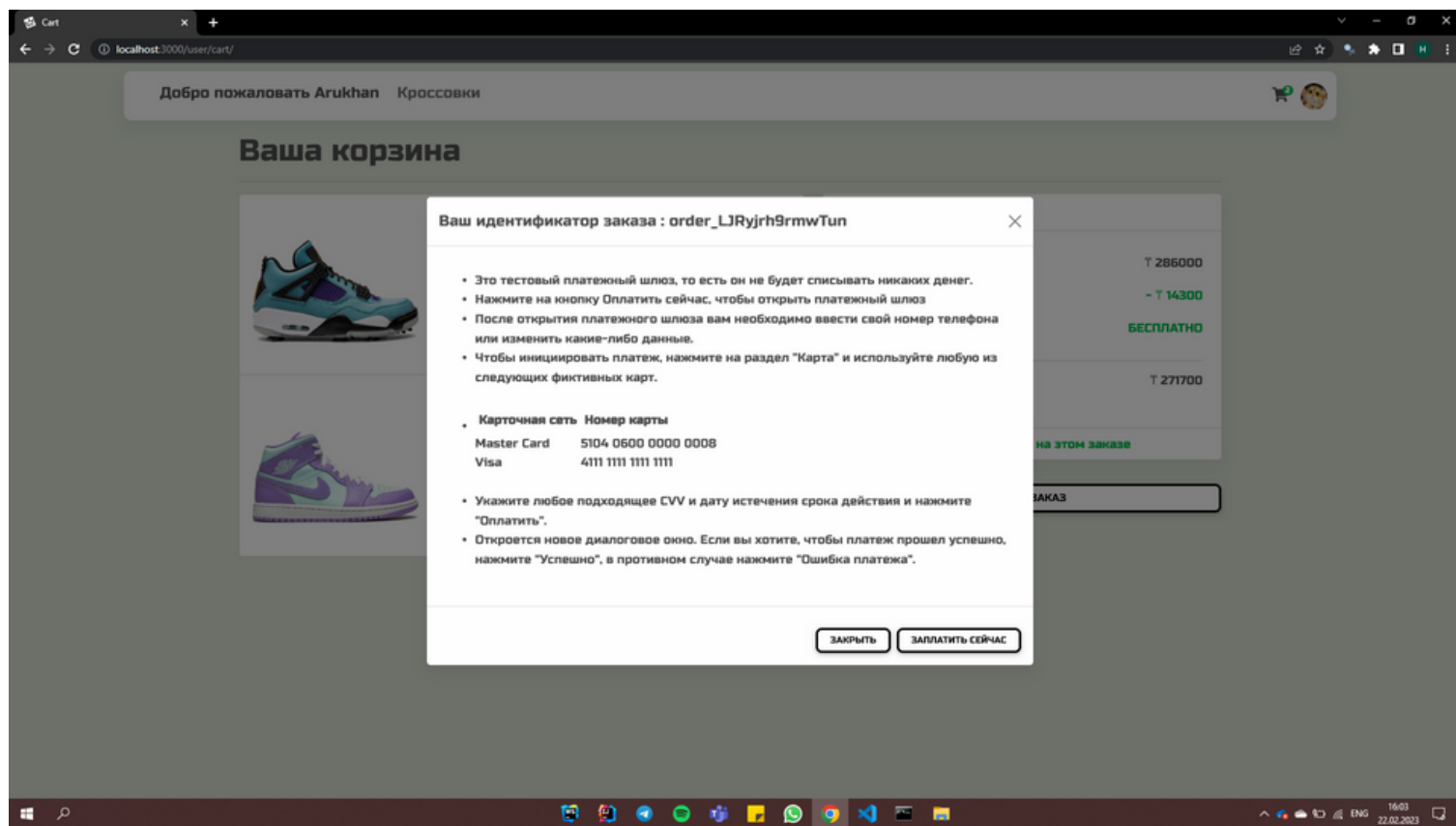
The screenshot shows a web browser window with the address bar displaying 'localhost:3000/register'. The page has a light green background. At the top, there is a white header bar with a crown icon and the text 'Кроссовки' on the left, and a shopping cart icon with a green notification bubble and a user profile icon on the right. The main heading is 'Регистрация' (Registration). Below it is a circular profile picture placeholder. There are two buttons: 'Выберите файл' (Choose file) and 'Файл не выбран' (File not selected). Below these are three input fields: 'Имя пользователя' (Username), 'Ваша почта' (Your email), and 'Пароль' (Password). A dropdown menu for 'Ваша роль:' (Your role) is set to 'Пользователь' (User). A 'РЕГИСТРАЦИЯ' (Registration) button is located below the dropdown. At the bottom, there is a link: 'Уже зарегистрированный пользователь? [Нажмите здесь чтобы войти](#)' (Already registered user? Click here to log in).



# Payment Process



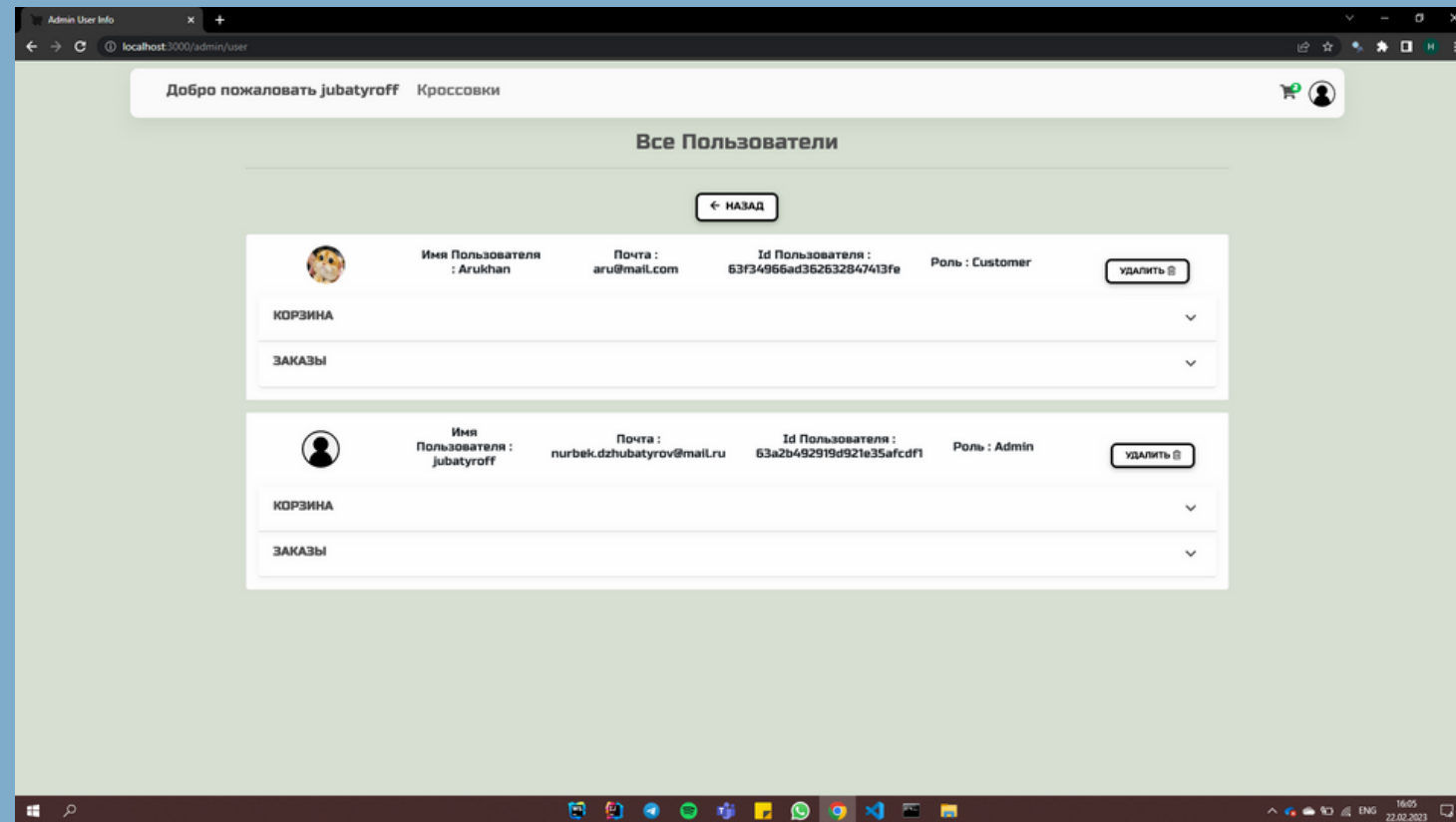
Add to Cart



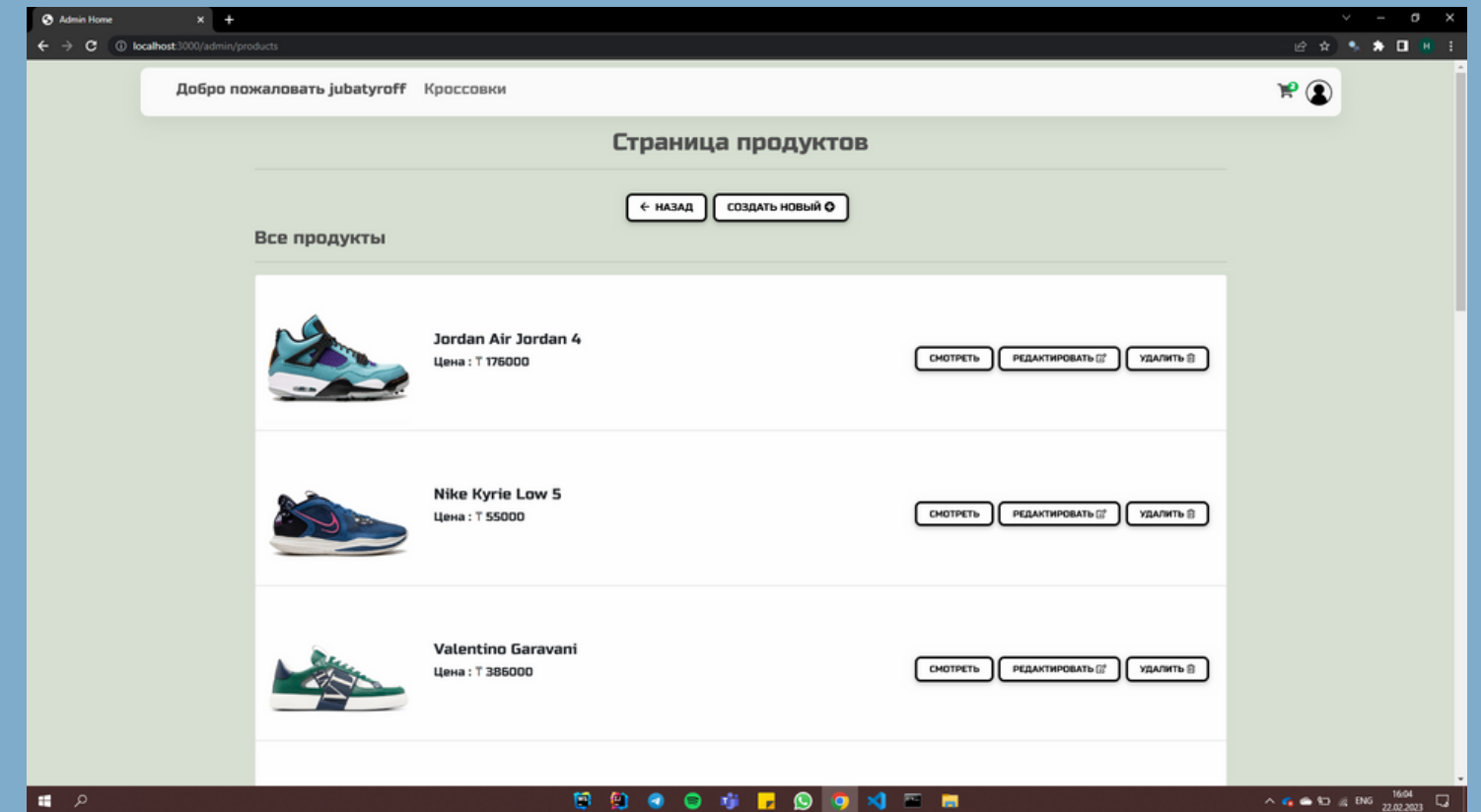
Order Information

# Admin role

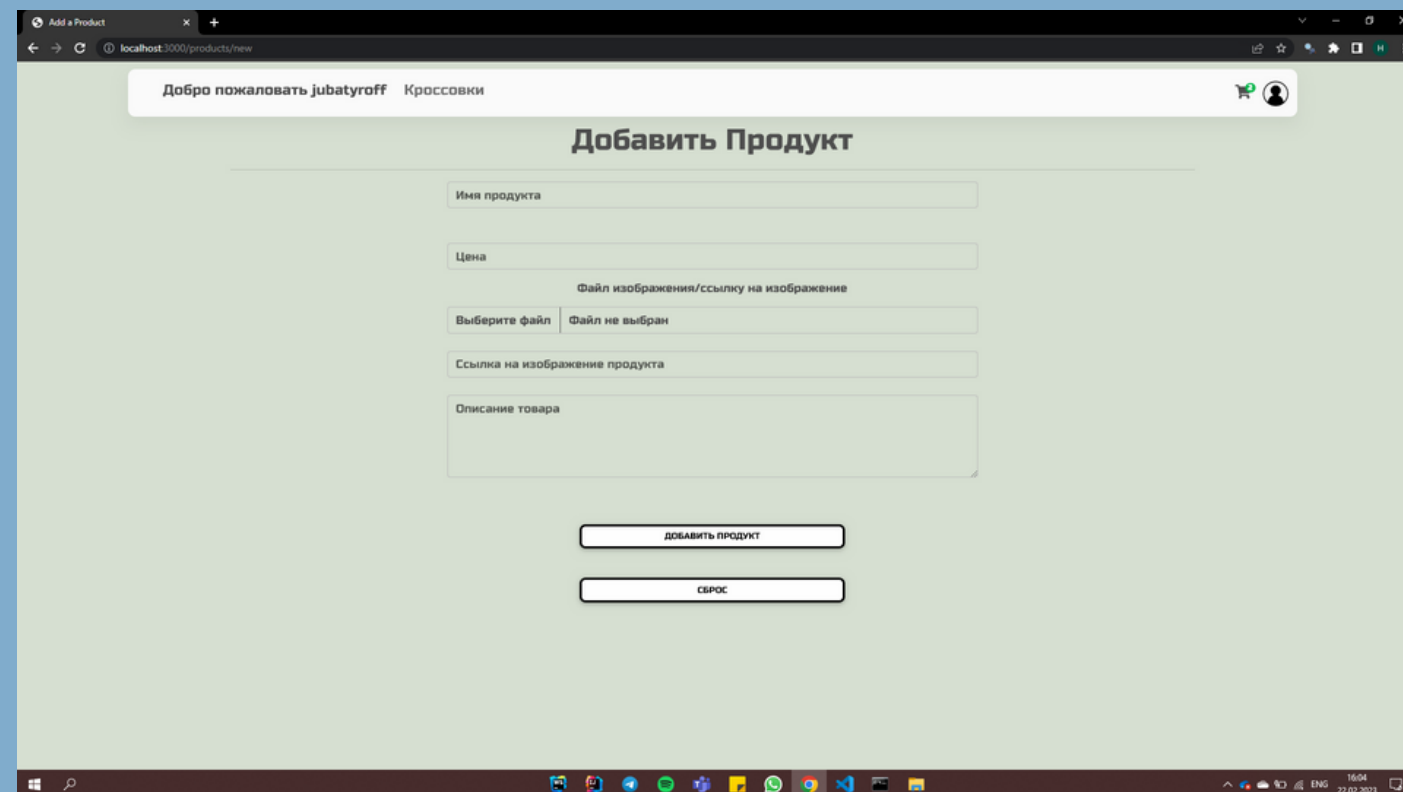
Can edit Users



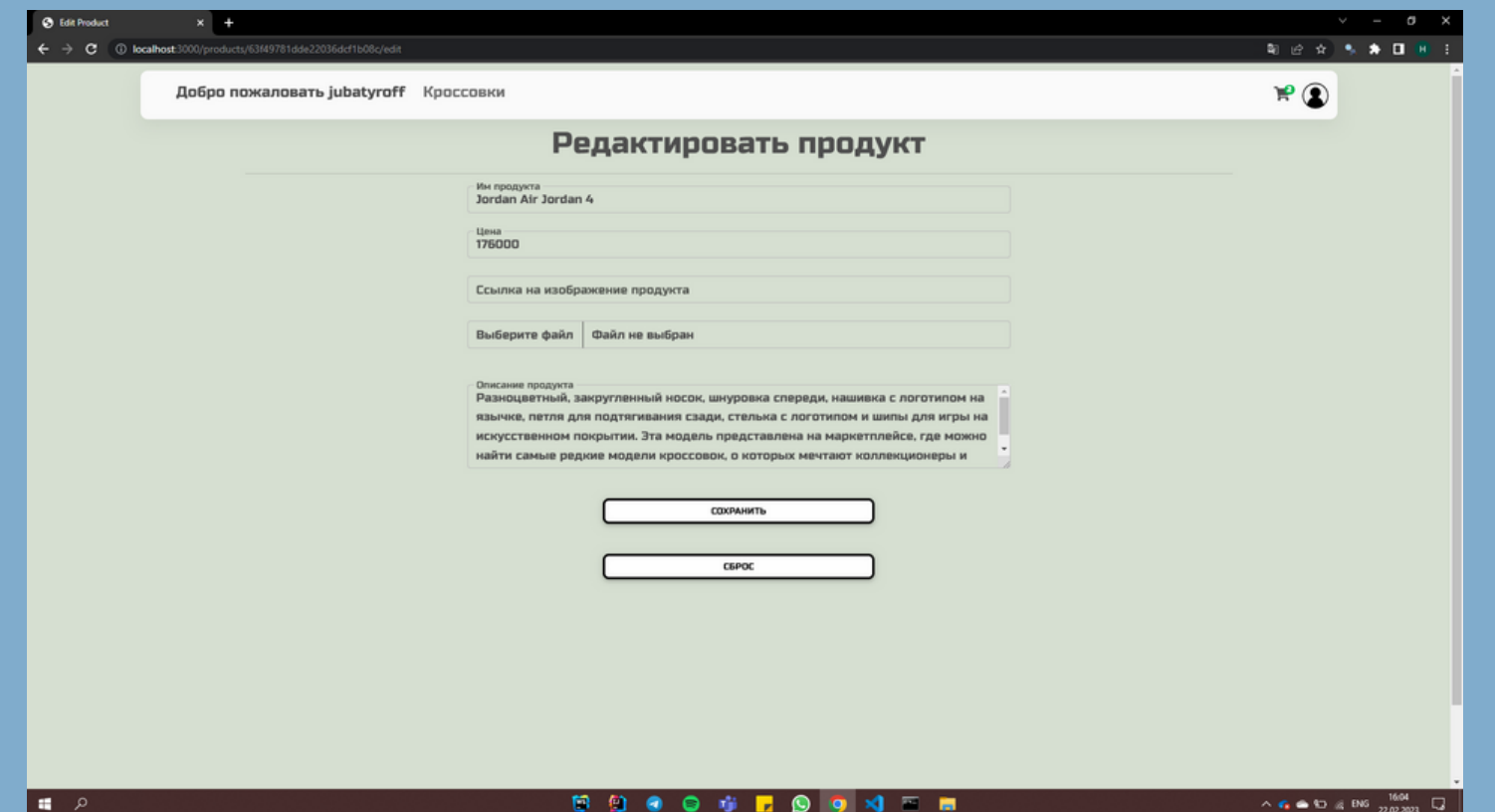
Can edit Products



Adding a new product



Editing existing product



**Thank you  
for listening!**