Q. Add the Remove from cart functionality to the API.

CartServices.cs

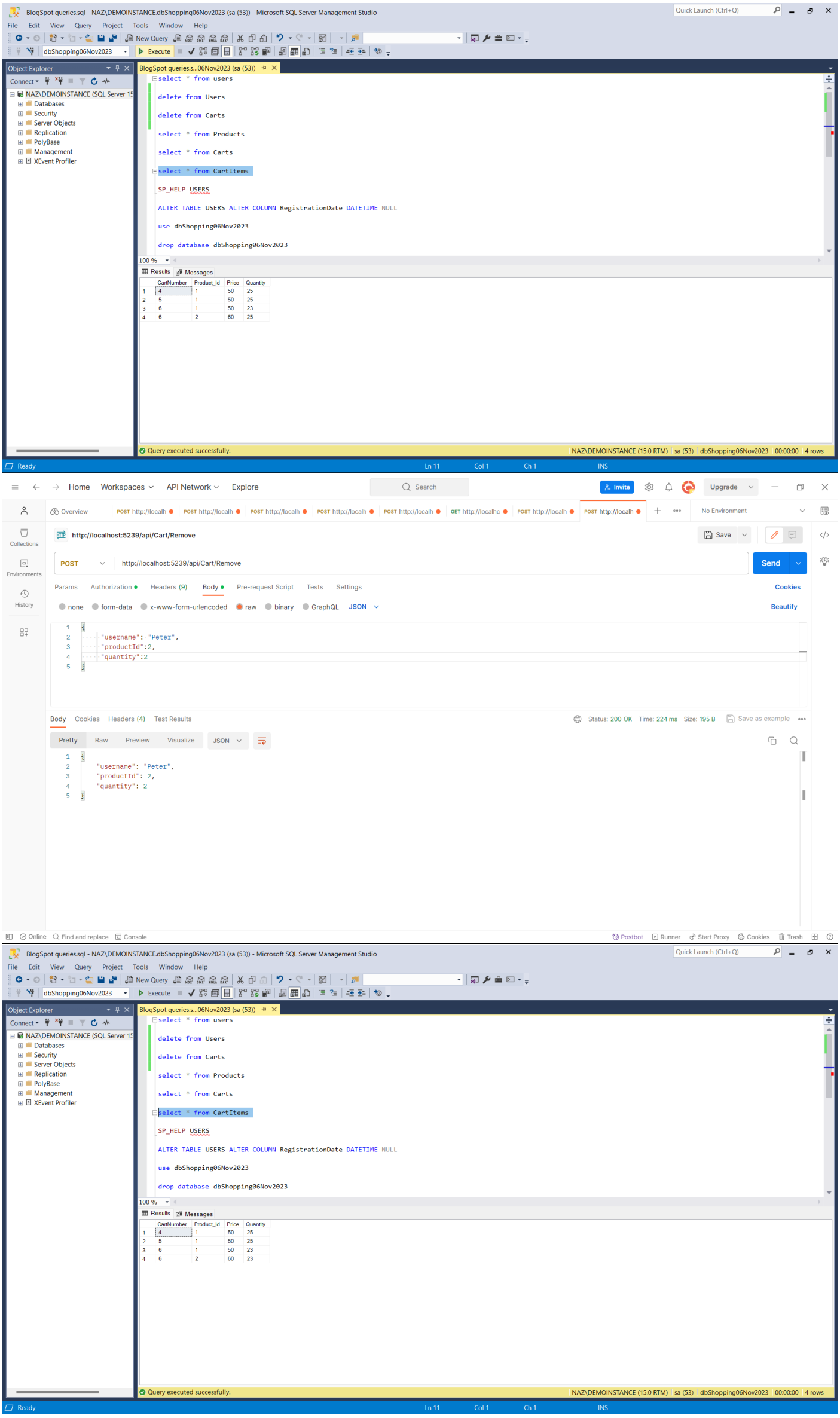
```
using Microsoft.CodeAnalysis;
using ShoppingApp.Interfaces;
using ShoppingApp.Models;
using ShoppingApp.Models.DTOs;
using ShoppingApp.Repositories;

namespace ShoppingApp.Services
{
    public class CartService : ICartService
    {
        private readonly IRepository<int, Cart> _cartRepository;
        private readonly IRepository<int, CartItems> _cartItemRepository;
        private readonly IRepository<int, Product> _productRepository;

        public CartService(IRepository<int, Cart> cartRepository,
            IRepository<int, CartItems> cartItemRepository,
            IRepository<int, Product> productRepository)
        {
            _cartRepository = cartRepository;
            _cartItemRepository = cartItemRepository;
            _productRepository = productRepository;
        }
        CartItems CreateCartItem(int cartNumber, CartDTO cartDTO)
        {
            var product = _productRepository.GetById(cartDTO.ProductId);
            if (product != null)
            {
                var myCartItem = new CartItems
                {
                    CartNumber = cartNumber,
                    Product_Id = cartDTO.ProductId,
                    Quantity = cartDTO.Quantity,
                    Price = product.Price
                };
                return myCartItem;
            }
            return null;
        }
        public bool AddToCart(CartDTO cartDTO)
        {

            var cartCheck = _cartRepository.GetAll().FirstOrDefault(c => c.Username == cartDTO.Username);
            int cartNumber = 0;
            if (cartCheck == null)
            {
                var cart = _cartRepository.Add(new Cart { Username = cartDTO.Username });
                cartNumber = cart.cartNumber;
            }
            else
                cartNumber = cartCheck.cartNumber;
            bool CartItemCheck = CheckIfCartItemAlreadyPresent(cartNumber, cartDTO.ProductId);
            if (CartItemCheck)
            {
                return IncrementQuantityInCart(cartNumber, cartDTO);
            }
            var myCartItem = CreateCartItem(cartNumber, cartDTO);
            if (myCartItem != null)
            {
                var result = _cartItemRepository.Add(myCartItem);
                if (result != null)
                    return true;
            }
            return false;
        }

        private bool IncrementQuantityInCart(int cartNumber, CartDTO cartDTO)
        {
            var cartItem = _cartItemRepository.GetAll()
                .FirstOrDefault(ci => ci.CartNumber == cartNumber && ci.Product_Id == cartDTO.ProductId);
            cartItem.Quantity += cartDTO.Quantity;
            var result = _cartItemRepository.Update(cartItem);
            if (result != null)
                return true;
            return false;
        }

        private bool CheckIfCartItemAlreadyPresent(int cartNumber, int productId)
        {
            var cartItem = _cartItemRepository.GetAll()
                .FirstOrDefault(ci => ci.CartNumber == cartNumber && ci.Product_Id == productId);
            return cartItem != null ? true : false;
        }

        private bool DecrementQuantityInCart(int cartNumber, CartDTO cartDTO)
        {
            var cartItem = _cartItemRepository.GetAll()
                .FirstOrDefault(ci => ci.CartNumber == cartNumber && ci.Product_Id == cartDTO.ProductId);
            cartItem.Quantity -= cartDTO.Quantity;
            int ProductId = cartDTO.ProductId;
            if (cartItem.Quantity ==0)
            {
                var result1 = _cartItemRepository.Delete(ProductId);
                if (result1 != null)
                    return true;
            }
```

```
            else if(cartItem.Quantity < 0)
            {
                return false;
            }
            var result = _cartItemRepository.Update(cartItem);
            if (result != null)
                return true;
            return false;
        }

        public bool RemoveFromCart(CartDTO cartDTO)
        {
            int cartNumber=0;
            var cartCheck = _cartRepository.GetAll().FirstOrDefault(c => c.Username == cartDTO.Username);
            if (cartCheck == null)
            {
                return false;
            }
            else
            {
                cartNumber = cartCheck.cartNumber;
            }
            bool CartItemCheck = CheckIfCartItemAlreadyPresent(cartNumber, cartDTO.ProductId);
            if (CartItemCheck)
            {
                return DecrementQuantityInCart(cartNumber, cartDTO);
            }
            return false;
        }
    }
}
```

CartController.cs

```
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using ShoppingApp.Interfaces;
using ShoppingApp.Models.DTOs;

namespace ShoppingApp.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class CartController : ControllerBase
    {
        private readonly ICartService _cartService;

        public CartController(ICartService cartService)
        {
            _cartService = cartService;
        }

        [Authorize(Roles = "User")]
        [HttpPost("add")]
        public IActionResult AddToCart(CartDTO cartDTO)
        {
            var result = _cartService.AddToCart(cartDTO);
            if (result)
                return Ok(cartDTO);
            return BadRequest("Could not add item to cart");
        }

        [Authorize(Roles = "User")]
        [HttpPost("remove")]
        public IActionResult RemoveFromCart(CartDTO cartDTO)
        {
            var result = _cartService.RemoveFromCart(cartDTO);
            if (result)
                return Ok(cartDTO);
            return BadRequest("Could not remove item from cart");
        }
    }
}
```

Images(Output):-



Home Workspaces API Network Explore Search Invite Upgrade

Overview | POST http://localh | POST http://localh | POST http://localh | POST http://localh | POST http://localh | GET http://localhc | POST http://localh | POST http://localh | No Environment

Collections Environments History

http://localhost:5239/api/Cart/Remove Save

POST http://localhost:5239/api/Cart/Remove Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
{
    "username": "Peter",
    "productId":2,
    "quantity":23
}
```

Body Cookies Headers (4) Test Results Status: 200 OK Time: 246 ms Size: 196 B Save as example

Pretty Raw Preview Visualize JSON

```
{
    "username": "Peter",
    "productId": 2,
    "quantity": 23
}
```

Online Find and replace Console Postbot Runner Start Proxy Cookies Trash

BlogSpot queries.sql - NAZ\DEMOINSTANCE.dbShopping06Nov2023 (sa (53)) - Microsoft SQL Server Management Studio

Quick Launch (Ctrl+Q)

File Edit View Query Project Tools Window Help

New Query | dbShopping06Nov2023 | Execute

Object Explorer

Connect

- NAZ\DEMOINSTANCE (SQL Server 15
  - Databases
  - Security
  - Server Objects
  - Replication
  - PolyBase
  - Management
  - XEvent Profiler

BlogSpot queries.s...06Nov2023 (sa (53))

```
select * from users

delete from Users

delete from Carts

select * from Products

select * from Carts

select * from CartItems

SP_HELP USERS

ALTER TABLE USERS ALTER COLUMN RegistrationDate DATETIME NULL

use dbShopping06Nov2023

drop database dbShopping06Nov2023
```

100 %

Results Messages

	CartNumber	Product_Id	Price	Quantity
1	4	1	50	25
2	5	1	50	25
3	6	1	50	23

Query executed successfully. NAZ\DEMOINSTANCE (15.0 RTM) | sa (53) | dbShopping06Nov2023 | 00:00:00 | 3 rows

Ready Ln 11 Col 1 Ch 1 INS

Home Workspaces API Network Explore Search Invite Upgrade

Overview | POST http://localh | POST http://localh | POST http://localh | POST http://localh | POST http://localh | GET http://localhc | POST http://localh | POST http://localh | No Environment

Collections Environments History

http://localhost:5239/api/Cart/Remove Save

POST http://localhost:5239/api/Cart/Remove Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
{
    "username": "Peter",
    "productId":1,
    "quantity":24
}
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 94 ms Size: 182 B Save as example

Pretty Raw Preview Visualize Text

```
Could not remove item from cart
```

Online Find and replace Console Postbot Runner Start Proxy Cookies Trash

BlogSpot queries.sql - NAZ\DEMOINSTANCE.dbShopping06Nov2023 (sa (53)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query dbShopping06Nov2023 Execute

Object Explorer

Connect

- NAZ\DEMOINSTANCE (SQL Server 15
  - Databases
  - Security
  - Server Objects
  - Replication
  - PolyBase
  - Management
  - XEvent Profiler

BlogSpot queries.s...06Nov2023 (sa (53))

```
select * from users

delete from Users

delete from Carts

select * from Products

select * from Carts

select * from CartItems

SP_HELP USERS

ALTER TABLE USERS ALTER COLUMN RegistrationDate DATETIME NULL

use dbShopping06Nov2023

drop database dbShopping06Nov2023
```

100 %

Results Messages

	CartNumber	Product_Id	Price	Quantity
1	4	1	50	25
2	5	1	50	25
3	6	1	50	23

Query executed successfully. NAZ\DEMOINSTANCE (15.0 RTM) sa (53) dbShopping06Nov2023 00:00:00 3 rows

Ready Ln 3 Col 4 INS