

Visoka škola strukovnih studija za informacione i komunikacione tehnologije

Wine shop

Radila: Nevena Nikolić 174/21
Smer: Internet tehnologije
Modul: Web programiranje
Predmet: Web programiranje ASP

Sadržaj

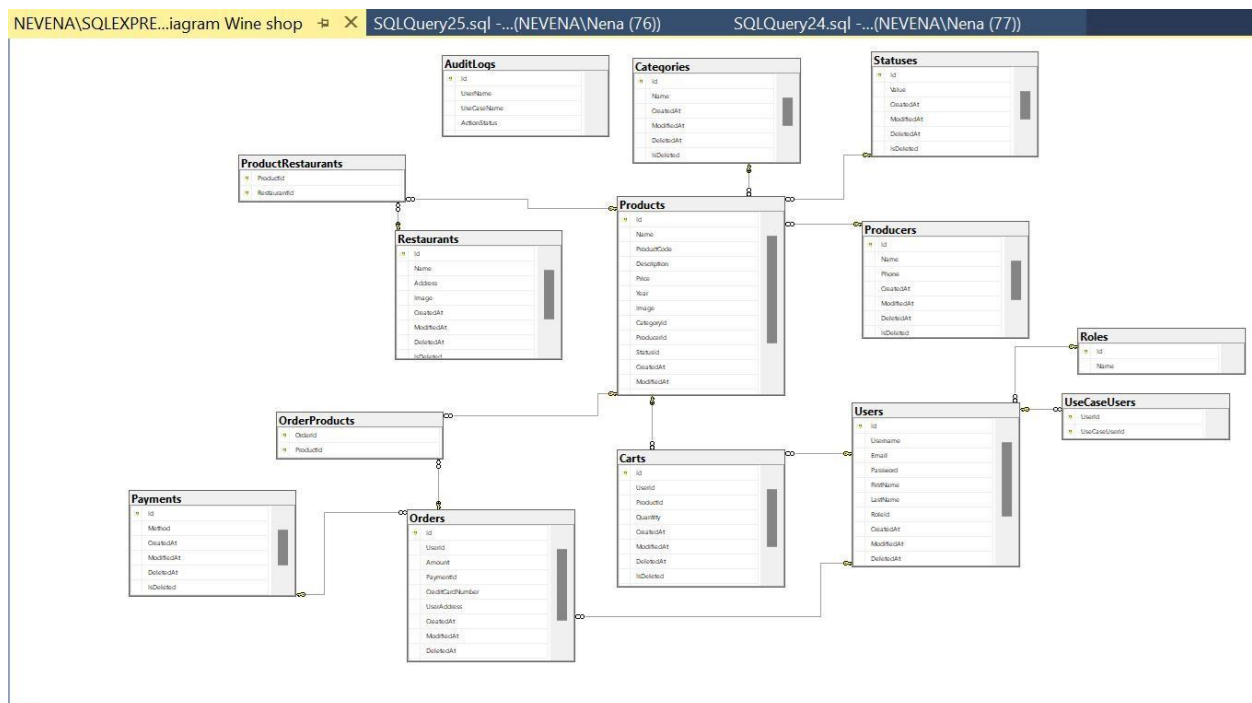
1. Uvod.....	3
Dizajn baze.....	4
2. Controllers.....	5
a) Pristup korisničkoj korpi.....	5
b) Dodavanje proizvoda u korpu.....	5
c) Ažuriranje količine proizvoda u korpi.....	5
d) Uklanjanje proizvoda iz korpe.....	5
e) Naručivanje proizvoda iz korpe.....	5
2.6 Payment Controller.....	8
2.7. Status Controller.....	9
2.8. Producer Controller.....	9
2.9. Restaurant Controller.....	9
2.10. Product Controller.....	10
3. SMTP EMAIL.....	12
4. Swagger.....	13
5. AuditLog.....	13

1. Uvod

Wine shop je online prodavnica specijalizovana za vina i vinske dodatke. U našem sistemu, postoje tri glavne vrste korisnika:

1. **Neautorizovani korisnici** - Mogu pregledati različite vrste vina, detalje o svakom vinu, kao i mogućnost registracije i prijave za pristup dodatnim funkcionalnostima.
2. **Registrovani korisnici** - Imaju mogućnost dodavanja vina u svoju korpu, menjanja količina vina u korpi, brisanja vina iz korpe i finalizovanja narudžbina.
3. **Administrator** - Ima potpunu kontrolu nad celokupnim sadržajem aplikacije, uključujući upravljanje vinskim asortimanom, pregled svih narudžbina, i administraciju korisničkih naloga i ostalih entiteta.

Dizajn baze



OPIS ENDPOINT-A:

Za svaku pretragu, inicijalno će biti prikazana tri rezultata, a korisnici će biti na prvoj stranici rezultata. Korisnici imaju mogućnost da odaberu broj rezultata koje žele da vide po stranici i da prolaze kroz dodatne stranice, ukoliko postoji više rezultata. Pored toga, korisnici mogu pretraživati proizvode koristeći više parametara istovremeno, kao što je naziv proizvoda. Takođe, omogućeno je sortiranje rezultata u opadajućem ili rastućem redosledu prema različitim kriterijumima.

2. Controllers :

2.1. Autorizacija korisnika - **AuthController** + **UserController**

Proces autorizacije korisnika započinje registracijom, gde korisnik dostavlja svoje lične podatke za kreiranje naloga. Nakon uspešne registracije, na postojeću email adresu korisnika se šalje obaveštenje o uspešnoj registraciji.

- Korisniku se dodeljuje inicijalna uloga sa privilegijama osnovnog korisnika (roleId: 1 - user). Da bi pristupio zaštićenim funkcionalnostima sistema, korisnik se mora autentifikovati slanjem tačnih kredencijala (e-mail adrese i lozinke).

- Ukoliko su kredencijali validni, sistem generiše jedinstveni token za korisnika, koji omogućava pristup različitim funkcionalnostima, uključujući:

- a) Pristup korisničkoj korpi
- b) Dodavanje proizvoda u korpu
- c) Ažuriranje količine proizvoda u korpi
- d) Uklanjanje proizvoda iz korpe
- e) Naručivanje proizvoda iz korpe

Autorizacija ADMIN postman

```
{  
  "email": "mika123@gmail.com",  
  "password": "mika123"  
}
```

```
namespace MyProject.Api.DTO  
{  
    public class AuthRequest  
    {  
        public string Email { get; set; }  
        public string Password { get; set; }  
    }  
  
    public class AuthResponse  
    {  
        public string Token { get; set; }  
    }  
}
```

2.2 Cart Controller

Autorizovani korisnici mogu dodavati proizvode u svoju korpu i ažurirati količinu proizvoda. Ovde se proverava da li proizvod prethodno postoji, pa ako postoji, proverava se da li se već nalazi u korpi. Ako je proizvod već prisutan u korpi, njegova količina se povećava za zadatu količinu, a ukupna cena (totalPrice) se proporcionalno prilagođava na osnovu nove količine. Na primer, ako je proizvod već prisutan u korpi s određenom količinom, dodavanje dodatne količine proizvoda uzrokuje da se ukupna cena poveća za iznos koji odgovara novoj količini.

- Primer za dodavanje kroz Postman:

```
{
  "items": [
    {
      "productId": 1,
      "quantity": 2
    },
    {
      "productId": 2,
      "quantity": 1
    }
  ]
}
```

```
{
  public class CartDTO
  {
    public IEnumerable<CartItemDTO> Items { get; set; }
    public double TotalPrice { get; set; }
  }

  public class CartItemDTO
  {
    public int ProductId { get; set; }
    public int Quantity { get; set; }
    public double Price { get; set; }
    public double TotalPrice => Price * Quantity;
  }
}
```

Pretraga :

```
namespace MyProject.Application.DTO.SearchEntities
{
  public class SearchCartItems : PageSearch
  {
    public int UserId { get; set; }
    public int ProductId { get; set; }
  }
}
```

2.3 Order Controller

Kada su spremni za kupovinu, mogu preći na poručivanje.

Primer poručivanja kroz Postman:

```
{
  "PaymentId": 1,
  "CreditCardNumber": 1234567890123456,
  "UserAddress": "Adresa 123",
  "Products": [
    {
      "ProductId": 35
    },
    {
      "ProductId": 36
    }
  ]
}
```

- ProductId mora da bude identifikator onog proizvoda koji se već nalazi u korpi, ne može bilo koji.

- PaymentId je identifikator metoda plaćanja. Ako je PaymentId 1 (kreditna kartica), korisnik mora takođe uneti broj kreditne kartice. Ukoliko nije, ne mora.

2.4 OrderProductController

Kada se uspešno obavi porudžbina, proizvodi se uklanjaju iz korisničke korpe i ažurira se vezivna tabela OrderProduct.

- Ova tabela povezuje svaki orderId sa odgovarajućim productId. Kontroler osigurava da se svi proizvodi iz narudžbine

```
namespace MyProject.Application.DTO
{
    public class OrderDTO : PageSearch
    {
        public int PaymentId { get; set; }
        public long? CreditCardNumber { get; set; }
        public string UserAddress { get; set; }
        public double Amount { get; set; }

        public IEnumerable<OrderProductDTO> Products { get; set; }
    }

    public class OrderProductDTO
    {
        public int ProductId { get; set; }
        public int OrderId { get; set; }
    }
}
```

pravilno povežu sa narudžbinom u sistemu.

2.5 Category Controller

Svaki endpoint zahteva da korisnik bude prijavljen i da ima administratorske privilegije za izvršavanje bilo koje akcije.

Funkcionalnosti uključuju:

- 1) Dodavanje nove kategorije
- 2) Brisanje postojeće kategorije
- 3) Pretraga kategorije prema imenu.
- 4) Sortiranje: po imenu kategorije - ASC ili DESC

```
namespace MyProject.Application.DTO.SearchEntities
{
    public class SearchByPaymentMethodDTO : PageSearch
    {
        public string Method { get; set; }
        public OrderByFunction? OrderByMethod { get; set; }
    }

    public enum OrderByFunction
    {
        ASC,
        DESC
    }
}
```

2.6 Payment Controller :

Svaki endpoint zahteva da korisnik bude prijavljen i da ima administratorske privilegije za obavljanje bilo koje akcije.

Funkcionalnosti uključuju:

- 1) Dodavanje nove metode plaćanja
- 2) Izmena postojeće metode plaćanja
- 3) Pretraga metoda plaćanja prema nazivu metode.
- 4) Brisanje metode plaćanja

```
namespace MyProject.Application.DTO.SearchEntities
{
    public class SearchByNameDTO : PageSearch
    {
        public string Name { get; set; }
        public OrderByFunction? OrderByName { get; set; }
    }

    public enum OrderByFunction
    {
        ASC,
        DESC
    }
}
```


2.7. Status Controller :

Svaki endpoint zahteva da korisnik bude prijavljen i da ima administratorske privilegije za obavljanje bilo koje akcije.

Funkcionalnosti uključuju:

Dodavanje novog statusa
Izmena postojećeg statusa
Pretraga statusa
Brisanje statusa

```
namespace MyProject.Application.DTO.SearchEntities
{
    public class SearchByValueDTO : PageSearch
    {
        public string Value { get; set; }
        public OrderByStatus? OrderByValue { get; set; }
    }

    public enum OrderByStatus
    {
        ASC,
        DESC
    }
}
```

2.8. Producer Controller :

Svaki endpoint zahteva da korisnik bude prijavljen i da ima administratorske privilegije za obavljanje bilo koje akcije.

Funkcionalnosti uključuju:

- 1) Dodavanje novog proizvođača
- 2) Izmena postojećeg proizvođača
- 3) Pretraga proizvođača po imenu
- 4) Brisanje proizvođača

2.9. Restaurant Controller:

Svaki endpoint zahteva da korisnik bude prijavljen i da ima administratorske privilegije za obavljanje bilo koje akcije.

Funkcionalnosti uključuju:

1. Dodavanje novog restorana
2. Izmena postojećeg restorana
3. Pretraga restorana po imenu
4. Brisanje restorana

2.10. Product Controller:

Svaki endpoint zahteva da korisnik bude prijavljen i da ima administratorske privilegije za obavljanje bilo koje akcije.

Funkcionalnosti uključuju:

- 1) Dodavanje novog proizvoda - *parametar za sliku ImageFile. Slika se šalje kao deo multipart/form-data zahteva. Slike se čuvaju u direktorijumu wwwroot/images, a naziv slike se generiše dodavanjem time() kako bi se osigurala jedinstvenost, posebno ako su iste slike ponovo dodate.
- 2) Izmena proizvoda
- 3) Pretraga proizvoda omogućava pretragu proizvoda prema različitim kriterijumima kao što su naziv, minimalna i maksimalna cena, te sortiranje po imenu, ceni i godini proizvodnje.
- 4) Brisanje proizvoda

```
namespace MyProject.Application.DTO.SearchEntities
{
    public class SearchProductsDTO : PageSearch
    {
        public string Name { get; set; }
        public double? PriceMin { get; set; }
        public double? PriceMax { get; set; }
        public OrderByFunc? OrderByName { get; set; }
        public OrderByFunc? OrderByPrice { get; set; }
        public OrderByFunc? OrderByYear { get; set; }
    }

    public enum OrderByFunc
    {
        ASC,
        DESC
    }
}
```

2.11 ProductRestaurant Controller

Kada su uspešno dodati proizvodi i restorani, ovaj kontroler omogućava povezivanje proizvoda i restorana.

Zahtevi za validaciju:


- a) Naziv proizvoda mora postojati u tabeli Products i ne sme biti prazan.
- b) Naziv restorana mora postojati u tabeli Restaurants i ne sme biti prazan.
- c) Svaki par proizvoda i restorana mora biti jedinstven. Kombinacija proizvoda i restorana ne sme već postojati u tabeli ProductRestaurants.

3. SMTP EMAIL

Omogućena je funkcionalnost slanja e-mail obaveštenja korisnicima prilikom uspešne registracije. Ako korisnik unese važeću e-mail adresu, obaveštenje će biti uspešno poslato na navedenu adresu. U slučaju neuspešnog slanja e-maila, administratori će primiti obaveštenje da je registracija uspešno završena, ali da slanje e-maila nije uspelo.


Validan mejl korisnika :

Welcome :) Примљено x



nevena.nikolic.174.21@ict.edu.rs
коме ја ▼

пет 30. авг 09:40 (

 Преведи на: српски ×

Registration Successful!


Hi Mika,

Thank you for signing up. Your registration is complete.


If you need assistance, contact us at nevena.nikolic.174.21@ict.edu.rs.

Nevalidan mejl korisnika :

Delivery Status Notification (Failure) Примљено x



Mail Delivery Subsystem <mailer-daemon@googlemail.com>
коме ја ▼

**Адреса није пронађена**
Порука није испоручена на **admin123@gmail.com** зато што адреса није пронађена или не може да прима имејлове.
[САЗНАЈТЕ ВИШЕ](#)

Одговор је био:
550 5.2.1 The email account that you tried to reach is inactive. For more information, go to <https://support.google.com/mail/?c=DisabledUser> 5b1f17b1804b1-e2ac51999d6e0r42602155e9.11 - gmail

4. Swagger

Swagger je implementiran u aplikaciji i možete mu pristupiti putem sledeće URL adrese: **<http://localhost:5000/swagger/index.html>**.

5. AuditLog :

Omogućava administratoru da pretražuje sve aktivnosti korisnika u aplikaciji.

Funkcionalnosti uključuju filtriranje prema

- 1) imenu i/ili prezimenu korisnika,
- 2) nazivu slučaja upotrebe i opsegu datuma (od - do)
- 3) sortiranje prema datumu izvršenja akcije