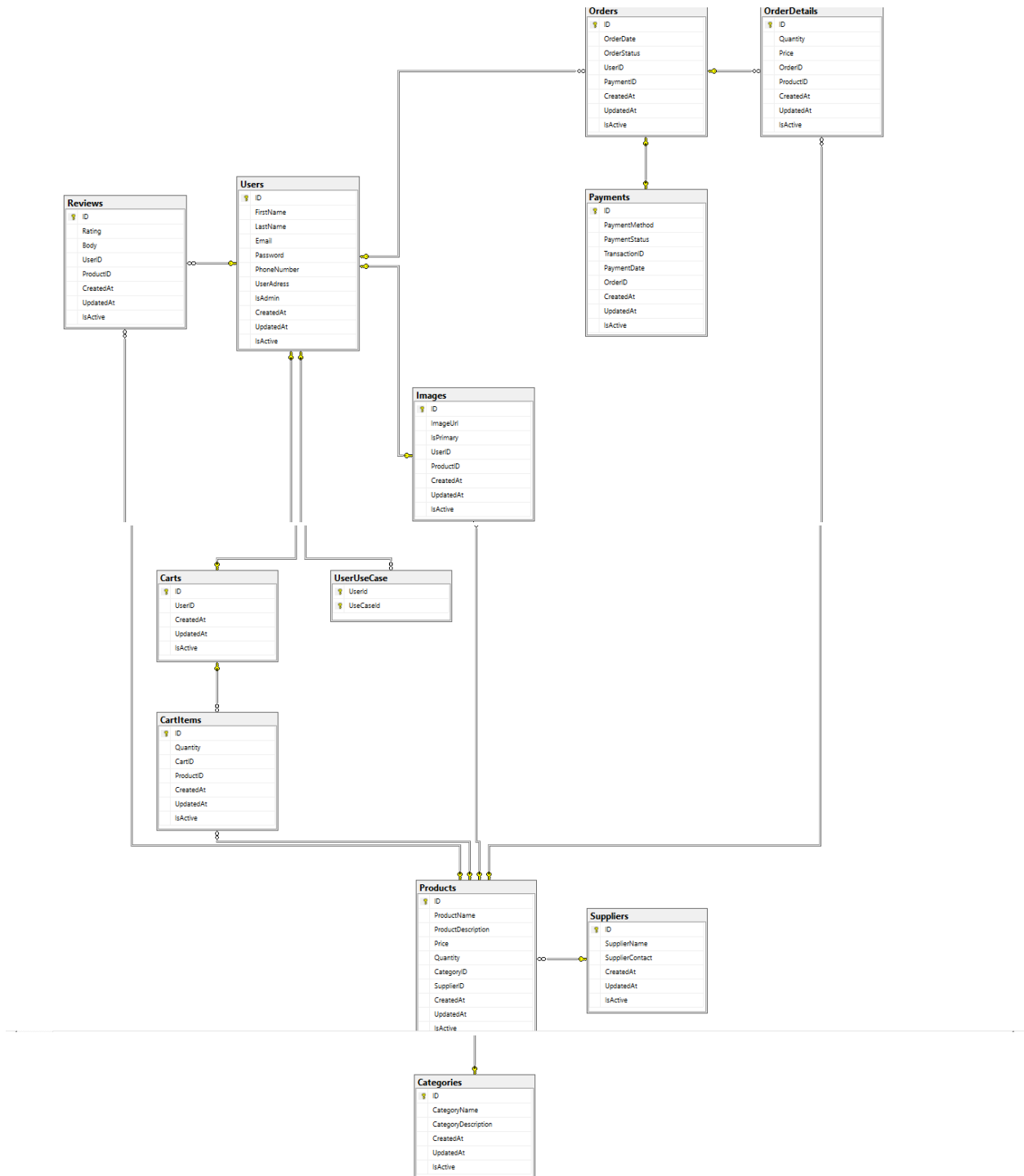


Dokumentacija

Milivojević Nemanja 270/19

ASP .NET

Dizajn Baze



Opis Teme

Projekat je koncipiran kao sofisticirana online prodavnica, koja nudi širok asortiman proizvoda različitih kategorija. Ova platforma je opremljena intuitivnim korisničkim interfejsom, koji omogućava kupcima da jednostavno dodaju proizvode u korpu i kreiraju porudžbine. Dodatno, prodavnica poseduje napredne funkcionalnosti za upravljanje sadržajem, omogućavajući kreiranje, ažuriranje i brisanje informacija na svim endpointovima. Ovaj dinamički pristup garantuje aktuelnost i tačnost prikazanih informacija, istovremeno obezbeđujući fleksibilnu i bezbrižnu kupovinu za sve korisnike.

Na sledećoj strani će biti objašnjene funkcionalnosti endpointova jedna po jedna.

Endpointovi

1.User

User		▼
GET	/api/User/search	
POST	/api/User/register	
PUT	/api/User/update	
DELETE	/api/User/delete	
POST	/api/User/login	
POST	/api/User/change-password	

Skoro svi endpointovi imaju zaštitu od neulogovanih korisnika.
Tako da se mora uzeti token iz /api/User/login endpointa.

Parametri za logovanje su:

{

"username": "aaa@gmail.com",

"password": "aaa12345"

}

JWT se koristi za autorizaciju.

User

GET /api/User/search

Parameters Try it out

Name	Description
Email string (query)	<input type="text" value="Email"/>
Keyword string (query)	<input type="text" value="Keyword"/>
PageNumber integer(\$int32) (query)	<input type="text" value="PageNumber"/>
PageSize integer(\$int32) (query)	<input type="text" value="PageSize"/>

Responses

Code	Description	Links
200	Success	No links

Pretraga je omogućena na skoro svim endpointovima i na svim pretragama je omogućena paginacija.

POST /api/User/register

Parameters

Try it out

No parameters

Request body

application/json

Example Value | Schema

```
{
  "id": 0,
  "firstName": "string",
  "lastName": "string",
  "email": "string",
  "password": "string",
  "phoneNumber": "string",
  "userAddress": "string"
}
```

Responses

Code	Description	Links
200	Success	No links

Registracija je jedini endpoint na kome je dozvoljena interakcija anonimnih korisnika.

PUT /api/User/update

Parameters

Try it out

No parameters

Request body

application/json

Example Value | Schema

```
{
  "id": 0,
  "firstName": "string",
  "lastName": "string",
  "email": "string",
  "password": "string",
  "phoneNumber": "string",
  "userAddress": "string",
  "isAdmin": true,
  "isActive": true
}
```

Responses

Code	Description	Links
200	Success	No links

DELETE /api/User/delete

Parameters

Try it out

No parameters

Request body

application/json

Example Value | Schema

```
{
  "id": 0
}
```

Responses

Code	Description	Links
200	Success	No links

Svi endpointovi koji menjaju stanje aplikacije jasno pokazuju koji podatak zahtevaju u Swaggeru.

POST /api/User/change-password

Parameters

Try it out

No parameters

Request body

application/json

Example Value | Schema

```
{
  "email": "string",
  "oldPassword": "string",
  "newPassword": "string"
}
```

Responses

Code	Description	Links
200	Success	No links

Endpoint za promenu lozinke šalje mejl korisniku ukoliko je uspešno poslat.

```
{
  "errorMessage": "Email not found in database.",
  "propertyName": "Email"
},
{
  "errorMessage": "Old Password cannot be lower than 8 characters.",
  "propertyName": "OldPassword"
},
{
  "errorMessage": "New Password cannot be lower than 8 characters.",
  "propertyName": "NewPassword"
}
```

Validaciona pravila su umetnuta na svaki endpoint koji menja stanje aplikacije.

2. Kategorije, Snabdevači, Proizvodi

The screenshot shows a REST client interface with a 'Code' tab selected. The status is 200. The response body is a JSON array of category objects. The response headers are also visible.

```
Code    Details
200
Response body
[
  {
    "categoryName": "Category 1",
    "categoryDescription": "Category 1",
    "id": 1
  },
  {
    "categoryName": "Electronics",
    "categoryDescription": "Electronic devices",
    "id": 2
  },
  {
    "categoryName": "Books",
    "categoryDescription": "Books and e-books",
    "id": 3
  },
  {
    "categoryName": "Clothing",
    "categoryDescription": "Men and women clothing",
    "id": 4
  },
  {
    "categoryName": "Grocery",
    "categoryDescription": "Grocery items",
    "id": 5
  },
  {
    "categoryName": "Toys",
    "id": 6
  }
]
Response headers
content-type: application/json; charset=utf-8
date: Fri 09 Jun 2023 07:25:55 GMT
server: Kestrel
transfer-encoding: chunked
```

Pretraga sa paginacijom postoji na svakom endpointu, kao i kreiranje, update i delete.

3. Images

Images	
POST	/api/Images/create
PUT	/api/Images/update
DELETE	/api/Images/delete

Images deo aplikacije koja se bavi uploadovanjem fajlova.

Tabela ima korisnički ID i proizvod ID.

Oba ta podatka su nullabilna jer slika može da se uploaduje kao profilna slika korisnika (1:1) ili galerija za proizvode (1:m).

Slika se uploaduje na `wwwroot/images`, i briše se delete endpointom.

4.Orders, Order Detail

OrderDetail		▼
GET	/api/OrderDetail/search	
POST	/api/OrderDetail/create	
PUT	/api/OrderDetail/update	
DELETE	/api/OrderDetail/delete	
Orders		▼
GET	/api/Orders/search	
POST	/api/Orders/create	
PUT	/api/Orders/update	
DELETE	/api/Orders/delete	

Orders tabela je vezana za korisnika koji je napravio narudžbinu.

OrderDetails je vezana za Orders i Proizvod koji je naručen i predstavlja stavke u narudžbini.

5. Cart, CartItems

CartItems		▼
GET	/api/CartItems/search	
POST	/api/CartItems/create	
PUT	/api/CartItems/update	
DELETE	/api/CartItems/delete	
Carts		▼
GET	/api/Carts/byUser/{userId}	
POST	/api/Carts/create	
PUT	/api/Carts/update	
DELETE	/api/Carts/delete	

Korpa je vezana za korisnika i samo jedan korisnik može imati jednu korpu.

Može se pretražiti po Identifikatoru korisnika.

CartItems ima stavke proizvoda koji se stavljaju u korpu, slične logici za OrderItems.

6.Payments

Payments		▼
GET	/api/Payments/search	
POST	/api/Payments/create	
PUT	/api/Payments/update	
DELETE	/api/Payments/delete	

Payments je tabela predviđena da bude integrisana sa nekim 3rd party sistemom za uplate.

Nešto kao Paypal, Stripe i sl.

Osmišljeno je tako da ulogovani korisnik submituje porudžbinu sa svim stavkama iz korpe, platio bi preko kartice čiji se detalji ne pamte na aplikaciji.

7.Reviews

Reviews		▼
GET	/api/Reviews/search	
POST	/api/Reviews/create	
PUT	/api/Reviews/update	
DELETE	/api/Reviews/delete	

Kao dodatna stvar na aplikaciji, korisnici mogu da ocene proizvod koji su naručili.

8.Audit Log

AuditLog		▼
GET	/api/AuditLog/search	
Parameters		Try it out
Name	Description	
DateFrom string(datetime) (query)	<input type="text" value="DateFrom"/>	
DateTo string(datetime) (query)	<input type="text" value="DateTo"/>	
UseCaseName string(query)	<input type="text" value="UseCaseName"/>	
Username string(query)	<input type="text" value="Username"/>	

Na žalost, naišao sam na problem u kodu što se tiče audit loga.

Potrudicu se da popravim pre nego što krene pregledanje.

Ali kao što se vidi, napravljeno je tako da se logovi pretražuju po datumima, slučaju korišćenja i korisničkom imenu korisnika koji je uradio logovani slučaj korišćenja.