



موجودیت‌ها و ویژگی‌های آن‌ها

۱. کاربر (User)

این موجودیت نمایانگر کاربران سیستم است و شامل ویژگی‌های زیر می‌باشد:

- id: شناسه یکتای کاربر
- fullName: نام کامل کاربر
- phone number: شماره تلفن کاربر
- email: ایمیل کاربر
- username: نام کاربری
- password: رمز عبور
- role: نقش کاربر در سیستم (مدیر، مشتری و غیره)
- city: شهر محل سکونت
- province: استان محل سکونت
- description: توضیحاتی در مورد کاربر
- sign-up date: تاریخ ثبت‌نام کاربر
- last login date: تاریخ آخرین ورود کاربر

۲. احراز هویت (Authentication)

این موجودیت فرآیند احراز هویت کاربران را مدیریت می‌کند و شامل ویژگی‌های زیر است:

- auth_id: شناسه یکتای احراز هویت
- user_id: شناسه یکتای کاربر
- token: توکن احراز هویت

- expires_at تاریخ انقضای توکن

- created_at تاریخ ایجاد توکن

۳. محصول (Product)

این موجودیت محصولات موجود در سیستم را ذخیره می‌کند و ویژگی‌های زیر را دارد:

- id: شناسه یکتای محصول

- name: نام محصول

- product name: نام دقیق‌تر محصول

- information: اطلاعات محصول

- price: قیمت محصول

- in-store: موجودی محصول در انبار

- width: عرض محصول

- height: ارتفاع محصول

- discount: میزان تخفیف محصول

- category id: شناسه دسته‌بندی محصول

- color: رنگ محصول

۴. دسته‌بندی (Categories)

این موجودیت دسته‌بندی محصولات را مدیریت می‌کند و دارای ویژگی‌های زیر است:

- category id: شناسه یکتای دسته‌بندی

- name: نام دسته‌بندی

۵. سبد خرید (Cart)

سبد خرید نشان‌دهنده محصولاتی است که کاربران به سبد خرید خود اضافه کرده‌اند. ویژگی‌های این موجودیت عبارتند از:

- cart id: شناسه یکتای سبد خرید

- user id: شناسه کاربر صاحب سبد خرید
- product id: شناسه محصول افزوده شده به سبد خرید
- quantity: تعداد محصول
- discount: میزان تخفیف

۶. آیتم سفارش (Order Item)

این موجودیت آیتم‌های سفارش را ذخیره می‌کند و شامل ویژگی‌های زیر است:

- id: شناسه یکتای آیتم سفارش
- order id: شناسه سفارش
- product id: شناسه محصول سفارش داده شده
- price: قیمت محصول
- discount: تخفیف اعمال شده بر محصول

۷. سفارش (Order)

سفارش‌ها توسط کاربران ایجاد می‌شوند و ویژگی‌های زیر را شامل می‌شوند:

- order id: شناسه یکتای سفارش
- user id: شناسه کاربری که سفارش را ثبت کرده است
- total price: قیمت کل سفارش
- payment method: روش پرداخت
- created at: تاریخ ایجاد سفارش

ارتباطات بین موجودیت‌ها

۱. کاربر و احراز هویت: هر کاربر می‌تواند چندین توکن احراز هویت داشته باشد (ارتباط یک به چند). ۲. کاربر و سبد خرید: هر کاربر می‌تواند یک یا چند سبد خرید داشته باشد (ارتباط یک به چند). ۳. کاربر و سفارش: هر کاربر می‌تواند چندین سفارش ثبت کند (ارتباط یک به چند). ۴. سبد خرید و محصول: هر سبد خرید شامل چندین محصول است (ارتباط چند به چند). ۵. سفارش و آیتم‌های سفارش: هر سفارش می‌تواند شامل چندین آیتم باشد (ارتباط یک به چند). ۶. محصول و دسته‌بندی: هر محصول متعلق به یک دسته‌بندی است (ارتباط یک به چند).

1. فرایند ثبت نام و ورود کاربر

- کاربر ابتدا در سایت ثبت نام می کند و اطلاعات خود مانند نام، شماره تلفن، ایمیل، و سایر جزئیات شخصی را وارد می کند.
- پس از ثبت نام، کاربر یک توکن احراز هویت دریافت می کند. این توکن برای هر بار ورود به سیستم استفاده می شود تا کاربر شناسایی و مجاز به استفاده از امکانات سایت شود.
- توکن های احراز هویت برای هر کاربر ایجاد شده و به تاریخ انقضای مشخصی می رسند، بنابراین کاربر باید پس از مدت زمان مشخص، دوباره وارد سیستم شود.
- پس از وارد شدن به سیستم، کاربر می تواند وضعیت حساب کاربری خود، مانند آخرین ورود و اطلاعات دیگر را مشاهده کند.

2. جستجو و مشاهده محصولات

- محصولات مختلف چرم در دسته بندی های مختلف نمایش داده می شوند. برای مثال، محصولات می توانند شامل کیف چرم، کفش چرم، دستبند چرم و غیره باشند.
- هر محصول اطلاعاتی مانند نام، توضیحات، قیمت، رنگ و اندازه، تخفیف و موجودی در انبار دارد. کاربر می تواند از میان این محصولات جستجو کند و آن ها را مشاهده کند.
- دسته بندی محصولات به کمک فیلترهایی مانند نوع محصول، رنگ و قیمت به کاربران این امکان را می دهد که محصولات مورد نظر خود را سریع تر پیدا کنند.

3. اضافه کردن به سبد خرید

- هنگامی که کاربر محصولی را پسندید، می‌تواند آن را به سبد خرید خود اضافه کند.
- سبد خرید شامل شناسه کاربر و محصولات مختلف است که کاربر می‌خواهد خریداری کند. در این مرحله، کاربر می‌تواند تعداد هر محصول را تغییر دهد و تخفیف‌ها را نیز مشاهده کند.
- این ارتباط بین سبد خرید و محصولات در واقع یک رابطه چند به چند است، زیرا هر سبد خرید ممکن است شامل چندین محصول باشد و هر محصول ممکن است در سبد خرید چندین کاربر قرار گیرد.

4. ثبت سفارش و خرید

- پس از انتخاب محصولات مورد نظر، کاربر وارد مرحله خرید می‌شود. در این مرحله، کاربر اطلاعات سفارش خود مانند آدرس ارسال، روش پرداخت، و غیره را وارد می‌کند.
- پس از نهایی کردن انتخاب‌ها، سفارش ثبت می‌شود و آیتم‌های سفارش (Order Items) برای هر محصول که در سفارش گنجانده شده است، ثبت می‌شود.
- هر آیتم سفارش شامل محصول، قیمت و تخفیف مربوطه است که به سفارش اضافه می‌شود.

5. پرداخت و تکمیل خرید

- پرداخت می‌تواند از روش‌های مختلفی مانند کارت بانکی، درگاه‌های پرداخت آنلاین یا پرداخت نقدی در محل انجام شود.
- پس از پرداخت، سفارش به عنوان کامل شده ثبت می‌شود و کاربر می‌تواند وضعیت سفارش خود را پیگیری کند.
- سفارش‌ها اطلاعات مربوط به تاریخ ایجاد، مبلغ نهایی، و روش پرداخت را شامل می‌شوند.

6. مدیریت موجودی و دسته‌بندی‌ها

- مدیر سایت می‌تواند موجودی محصولات را به‌روزرسانی کند. به عنوان مثال، پس از هر خرید، موجودی انبار به‌طور خودکار کاهش می‌یابد.
- همچنین، محصولات در دسته‌بندی‌های مختلف به نمایش گذاشته می‌شوند تا کاربران بتوانند محصولات مشابه را به راحتی پیدا کنند.

7. رابطه‌ها و جریان داده‌ها

- کاربر می‌تواند چندین سبد خرید داشته باشد (برای مثال، برای خریدهای مختلف)، ولی تنها یک سبد خرید فعال می‌تواند در هر زمان وجود داشته باشد.

- هر کاربر می‌تواند چندین سفارش ثبت کند. در نتیجه، جریان داده‌های کاربر → سبد خرید → سفارش به راحتی مدیریت می‌شود.
- علاوه بر این، هر سفارش شامل چندین آیتم سفارش است که به آن سفارش تعلق دارند.
- محصولات به دسته‌بندی‌های مختلف تعلق دارند و اطلاعات دسته‌بندی‌ها به مدیران کمک می‌کند تا محصولات را بهتر سازماندهی کنند.

در نتیجه:

فرایند کلی سایت فروشگاهی محصولات چرم به شرح زیر است:

۱. ثبت‌نام/ورود کاربر: کاربر اطلاعات خود را وارد می‌کند و با دریافت توکن احراز هویت وارد سایت می‌شود.
۲. مرور محصولات: کاربر می‌تواند از میان محصولات مختلف، چرم‌های مورد نظر خود را جستجو و مشاهده کند.
۳. اضافه به سبد خرید: کاربر محصولات مورد نظر خود را به سبد خرید اضافه می‌کند.
۴. ثبت سفارش و خرید: پس از تایید سبد خرید، کاربر سفارش خود را ثبت و پرداخت می‌کند.
۵. پرداخت و تکمیل سفارش: سفارش پردازش و پرداخت انجام می‌شود و وضعیت سفارش نهایی ثبت می‌شود.
۶. مدیریت موجودی و دسته‌بندی‌ها: مدیر سایت می‌تواند موجودی‌ها را به‌روزرسانی کرده و محصولات را در دسته‌بندی‌های مختلف سازماندهی کند.

set search_path to nik

CREATE TABLE "user" "

id SERIAL PRIMARY KEY,

fullname VARCHAR(100) NOT NULL,

username VARCHAR(50) UNIQUE NOT NULL,

password VARCHAR(255) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

phone_number VARCHAR(12),

city VARCHAR(50)

province VARCHAR(50)

description TEXT,

" role" VARCHAR(20) NOT NULL,

signup_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP

;

CREATE TABLE product)

id SERIAL PRIMARY KEY,

" name" VARCHAR(100) NOT NULL,

description TEXT,

price DECIMAL(10, 2) NOT NULL,

discount DECIMAL(5, 2) DEFAULT 0,

category_id INTEGER,

color VARCHAR(50)


```
width DECIMAL(10,2)
height DECIMAL(10,2)
in_store BOOLEAN DEFAULT TRUE,
product_information TEXT
;
```

```
CREATE TABLE categories
(
  id SERIAL PRIMARY KEY,
  "name" VARCHAR(100) NOT NULL
);
```

```
ALTER TABLE product
ADD CONSTRAINT fk_product_category
FOREIGN KEY (category_id) REFERENCES categories(id);
```

```
CREATE TABLE cart
(
  id SERIAL PRIMARY KEY,
  user_id INTEGER NOT NULL,
  product_id INTEGER,
  quantity INTEGER DEFAULT 1,
  discount DECIMAL(5, 2) DEFAULT 0,
  FOREIGN KEY (user_id) REFERENCES "user"(id),
  FOREIGN KEY (product_id) REFERENCES product(id)
);
```

```
;(
```

```
CREATE TABLE "order) "
```

```
id SERIAL PRIMARY KEY,
```

```
user_id INTEGER NOT NULL,
```

```
total_price DECIMAL(10, 2) NOT NULL,
```

```
payment_method VARCHAR(50)
```

```
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
```

```
FOREIGN KEY (user_id) REFERENCES "user"(id)
```

```
;(
```

```
CREATE TABLE order_item)
```

```
id SERIAL PRIMARY KEY,
```

```
order_id INTEGER NOT NULL,
```

```
product_id INTEGER NOT NULL,
```

```
price DECIMAL(10, 2) NOT NULL,
```

```
discount DECIMAL(5, 2) DEFAULT 0,
```

```
FOREIGN KEY (order_id) REFERENCES "order"(id),
```

```
FOREIGN KEY (product_id) REFERENCES product(id)
```

```
;(
```

```
CREATE TABLE "authentication) "
```

```
auth_id SERIAL PRIMARY KEY,
```

```
user_id INT REFERENCES "user" (id) ON DELETE CASCADE,  
token VARCHAR(255) NOT NULL,  
expires_at TIMESTAMP NOT NULL,  
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
FOREIGN KEY (user_id) REFERENCES "user" (id)  
;
```

