DOCUMENTAȚIA PROIECTULUI
PcGear - Magazin Online de Componente PC

Porfireanu Constantin Laurențiu
Stanga George

23 iunie 2025

Cuprins

1 Prezentarea proiectului

1.1 Descrierea generală

PcGear este o aplicație web de tip magazin online specializată în vânzarea de componente și accesorii pentru calculatoare personale. Proiectul implementează o arhitectură Clean Architecture cu separarea responsabilităților pe mai multe straturi.

1.2 Problemele rezolvate

Aplicația rezolvă următoarele probleme:

- **Gestionarea inventarului** - Administrarea produselor, categoriilor și producătorilor
- **Autentificare și autorizare** - Sistem securizat de logare cu roluri diferite (Admin/User)
- **Sistem de recenzii** - Clienții pot lăsa feedback pentru produse
- **Căutare avansată** - Filtrare, sortare și paginare pentru o experiență îmbunătățită
- **Management stoc** - Actualizarea în timp real a disponibilității produselor

2 Tehnologiile folosite

Proiectul utilizează următoarele tehnologii moderne:

- **Backend Framework**: ASP.NET Core 9.0
- **Bază de date**: SQL Server cu Entity Framework Core 9.0.5
- **Autentificare**: JWT (JSON Web Tokens)
- **Documentare API**: Swagger/OpenAPI 7.2.0
- **Arhitectură**: Clean Architecture cu separarea în straturi
- **Pattern-uri**: Repository Pattern, Dependency Injection, DTO Pattern
- **Securitate**: Hashing cu salt pentru parole, autorizare bazată pe roluri

3 Baza de date

3.1 Diagrama bazei de date

Baza de date implementează următoarele entități principale:
Entități și relații:

- **User** - Utilizatori ai sistemului (clienți și administratori)
- **Category** - Categorii de produse (ex: Plăci video, Procesoare)
- **Manufacturer** - Producători (ex: NVIDIA, AMD, Intel)

- **Product** - Produsele disponibile în magazin
- **ProductReview** - Recenziile lăsate de utilizatori pentru produse

Relațiile dintre tabele:

- Product **Many-to-One** Category (un produs aparține unei categorii)
- Product **Many-to-One** Manufacturer (un produs are un producător)
- ProductReview **Many-to-One** Product (mai multe recenzii pentru un produs)
- ProductReview **Many-to-One** User (un utilizator poate avea mai multe recenzii)

Toate entitățile moștenesc din `BaseEntity` care conține câmpurile comune: `Id`, `CreatedAt`, `ModifiedAt`, `DeletedAt` pentru implementarea soft delete.

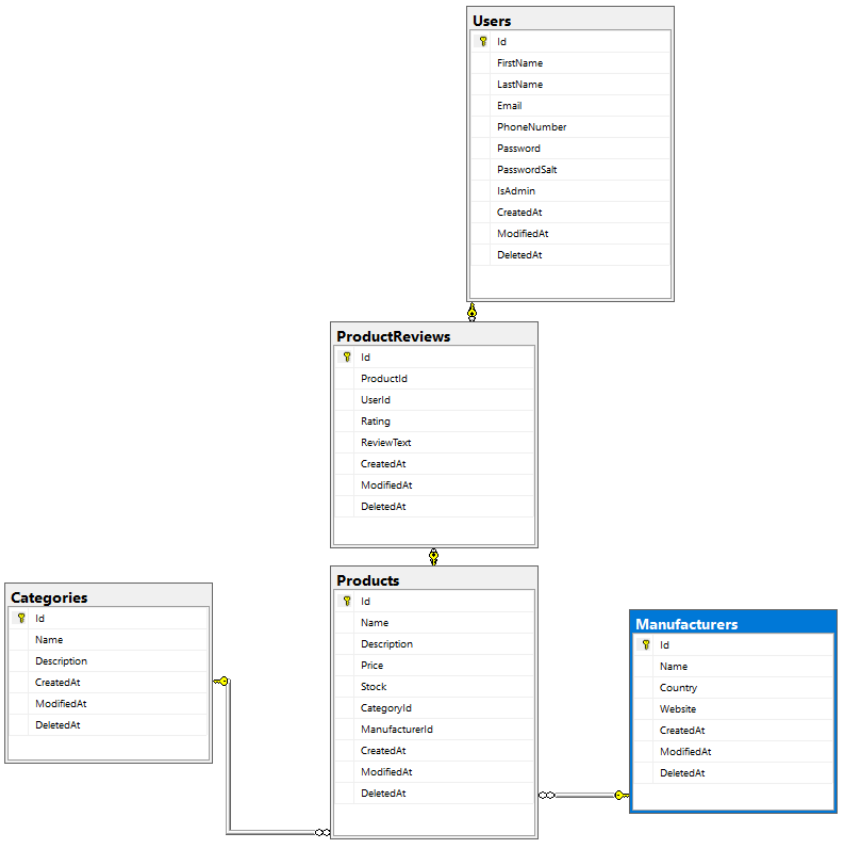

Figura 1: Diagrama entităților și relațiilor din baza de date PcGear

4 Prezentarea API-ului

API-ul este documentat complet cu Swagger și expune următoarele endpoint-uri:

4.1 Autentificare (AuthController)

- `POST /api/auth/Login` - Autentificare utilizator
- `POST /api/auth/Register` - Înregistrare utilizator nou
- `GET /api/auth/profile` - Obținerea profilului utilizatorului autentificat

4.2 Produse (ProductsController)

- `POST /api/products/Add_product` - Adăugare produs nou
- `GET /api/products/Get_products` - Listarea tuturor produselor
- `GET /api/products/Get_products_by_id:{id}` - Obținere produs după ID
- `GET /api/products/Get_producs_with_reviews{id}` - Produs cu recenzii
- `PUT /api/products/Update_product{id}` - Actualizare produs
- `PATCH /api/products/Update_product_stock{id}` - Actualizare stoc
- `DELETE /api/products/Delete_product{id}` - Ștergere produs
- `GET /api/products/Get_paged_and_filter` - Căutare avansată cu filtre

4.3 Categorii și Producători

Endpoint-uri similare pentru gestionarea categoriilor (`/api/categories`) și producătorilor (`/api/manufacturers`) cu operații CRUD complete.

4.4 Recenzii (ProductReviewsController)

- `POST /api/reviews/Add_review` - Adăugare recenzie
- `GET /api/reviews/Get_Reviews` - Listarea recenziilor
- `DELETE /api/reviews/Delete_by_id{id}` - Ștergere recenzie

4.5 Documentarea API-ului cu Swagger

5 Utilizarea aplicației

5.1 Tipuri de utilizatori

- **Utilizatori obișnuiți**: Pot vizualiza produsele, lăsa recenzii, se pot înregistra și autentifica
- **Utilizatori cu drepturi de administrator**: Orice utilizator poate avea și rolul de administrator (câmpul `IsAdmin`), având astfel acces complet la toate funcționalitățile - gestionarea produselor, categoriilor, producătorilor și utilizatorilor

5.2 Autentificare

Sistemul folosește JWT tokens pentru autentificare. După logare, utilizatorul primește un token valid pentru 525600 minute. Toate endpoint-urile (exceptând login/register) necesită autentificare.

5.3 Funcționalități avansate

- **Filtrare**: După nume, preț, categorie, producător, disponibilitate
- **Sortare**: După anumite câmpuri (nume, preț, stoc, dată creării)
- **Paginare**: Rezultate împărțite pe pagini pentru performanță optimă
- **Recenzii cu rating**: Sistem de evaluare de la 1 la 5 stele

6 Concluzii și contribuții

6.1 Împărțirea task-urilor

Deși am împărțit responsabilitățile după cum urmează, am lucrat în echipă pe tot parcursul dezvoltării:

Porfireanu Constantin Laurențiu:

- Implementarea elementelor de bază (entități, repository-uri)
- Operațiile CRUD pentru toate entitățile
- Configurarea Entity Framework și migrărilor
- Implementarea pattern-urilor (Repository, DTO mapping)

Stanga George:

- Sistem de autentificare și autorizare JWT
- Implementarea filtrării, sortării și paginării avansate
- Middleware-urile pentru logging și exception handling
- Configurarea Swagger și documentarea API-ului

6.2 Ce am învățat

În urma acestui proiect am dobândit experiență în:

- Arhitectura Clean Architecture și separarea responsabilităților
- Implementarea sistemelor de autentificare securizate cu JWT
- Lucrul cu Entity Framework Core și SQL Server
- Dezvoltarea API-urilor RESTful cu documentare Swagger

- Pattern-urile Repository și Dependency Injection
- Implementarea funcționalităților avansate (filtrare, sortare, paginare)
- Gestionarea excepțiilor și logging-ul în aplicații web

6.3 Link GitHub

Codul complet al proiectului este disponibil la:PcGear

Figura 2: Screenshot al documentării API-ului în Swagger - toate endpoint-urile disponibile