

# Online Skincare Shop

Proiect realizat de Alexandrescu Andra si Nitu Teodora

Link catre proiect:

[https://github.com/andialalexandrescu/Skincare\\_Online\\_Shop\\_.NET](https://github.com/andialalexandrescu/Skincare_Online_Shop_.NET) (varianta de Windows)

<https://github.com/tteeeoooo/project2> (varianta de Mac)

## Etapa 1 - 07 – 14 Noiembrie

---

### Sprint 1 (saptamanile 6, 7)

OVERALL:

Inceputul proiectului a pornit de la trei tabele din diagrama asociata aplicatiei (Products, Categories, Reviews, 2 relatii one-to-many configurate la nivel de ApplicationDbContext), impreuna cu views-urile pentru Products si Categories.

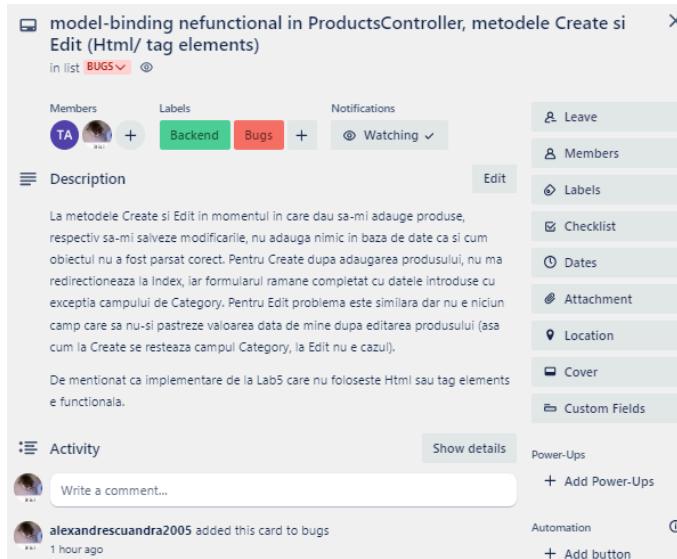
Etape imediat urmatoare:

De terminat CRUD-ul pentru Categories si Reviews (controller + views).

Probleme intalnite/ Bugs:

[X] Merge conflicts rezolvate din cauza diferentelor de configuratii de proiect pentru Mac/ Windows => vom incerca folosind Docker (sapt 7)

[X] Model-binding nefunctional pentru ambii membri ai echipei

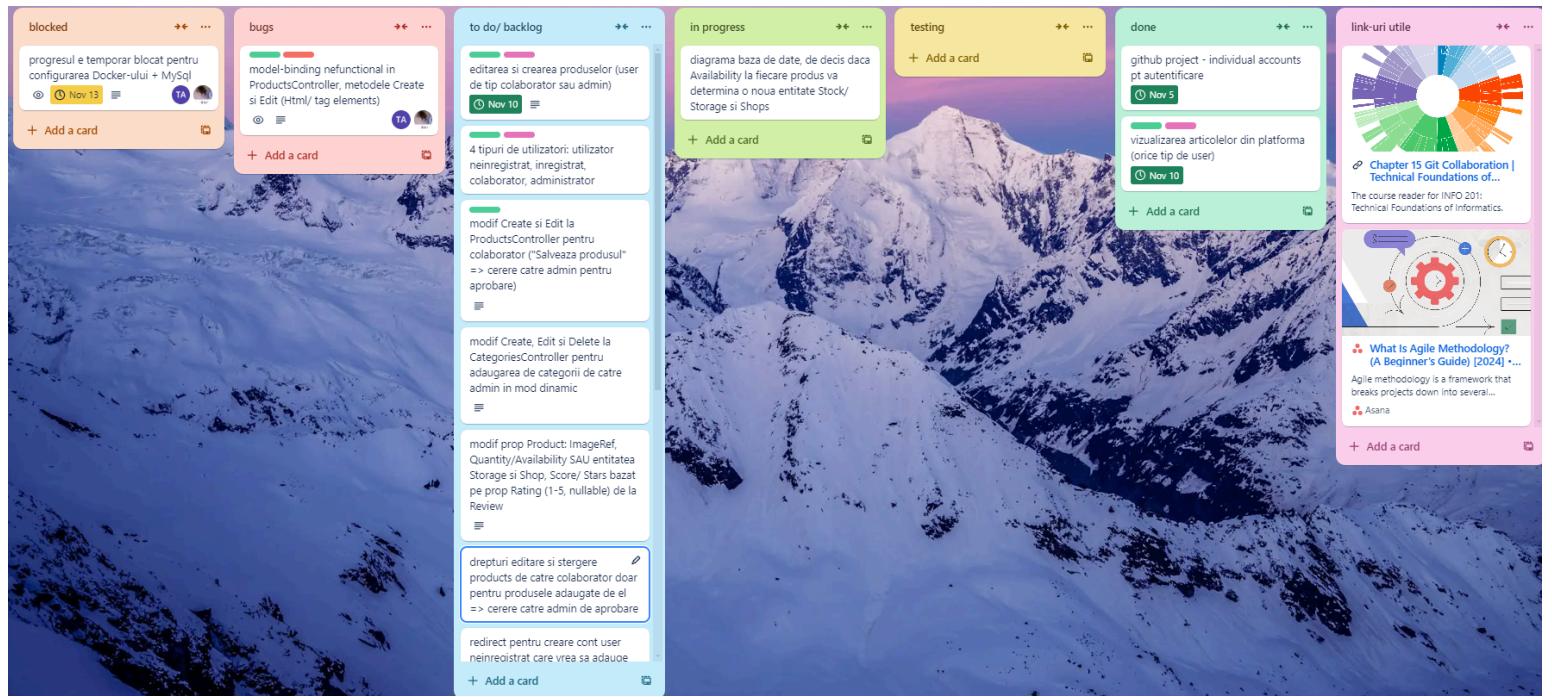


[X] Schimbarea unor proprietati la nivel de Products (Andi <- Teo: Name <- Description, Description <- Content, (+)Brand, (+)Ingredients) => modificare ProductsController, Products' views

## Trello Set-Up - Planificare task-uri si sprinturi

The Trello board has a header 'to do/ backlog' with three columns: 'in progress' (green), 'ready to review' (pink), and 'done' (blue). Below are the tasks:

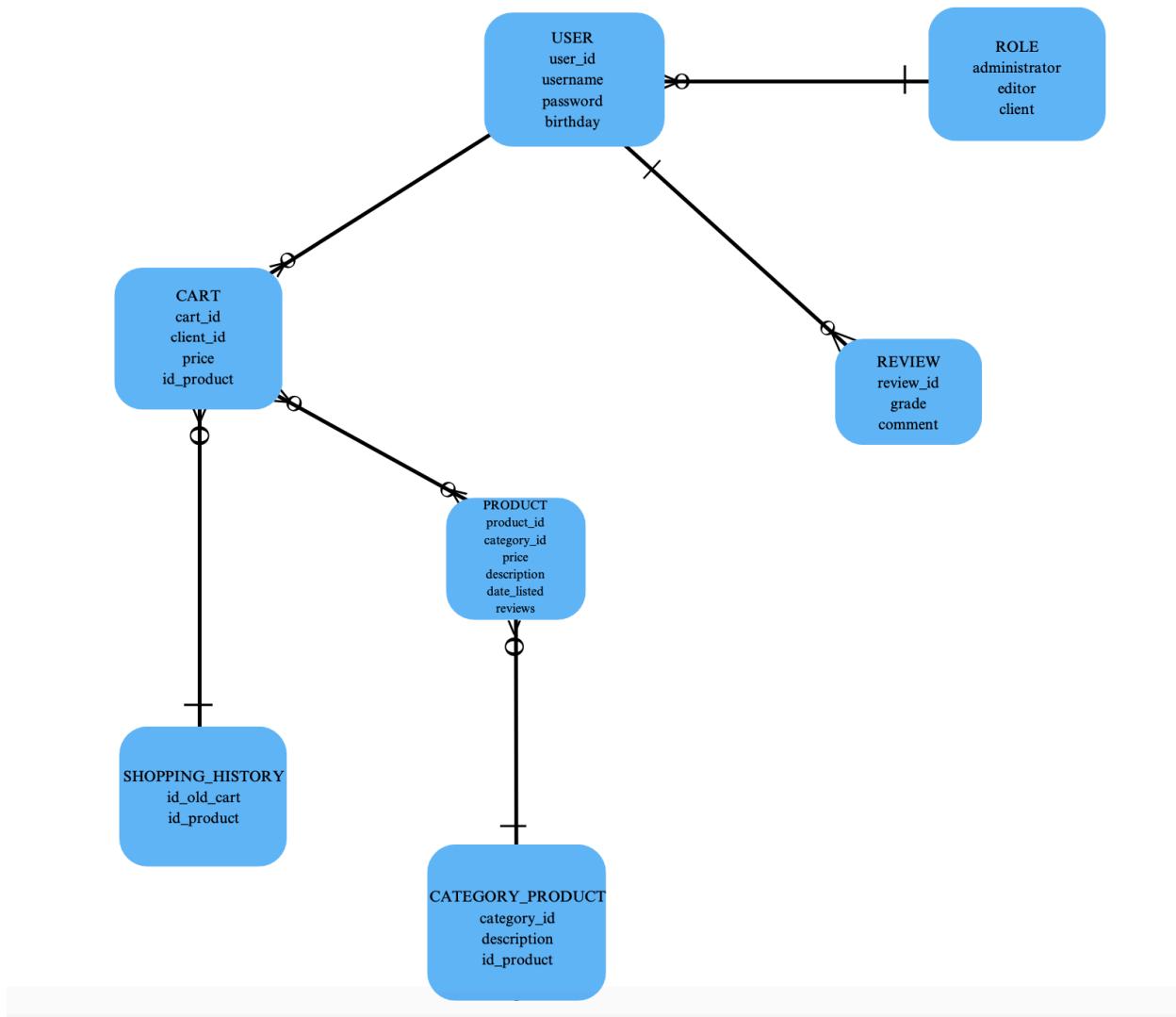
- editarea si crearea produselor (user de tip colaborator sau admin)**  
Due Nov 10
- 4 tipuri de utilizatori: utilizator neinregistrat, inregistrat, colaborator, administrator**
- modif Create si Edit la ProductsController pentru colaborator ("Salveaza produsul" => cerere catre admin pentru aprobare)**
- modif Create, Edit si Delete la CategoriesController pentru adaugarea de categorii de catre admin in mod dinamic**
- modif prop Product: ImageRef, Quantity/Availability SAU entitatea Storage si Shop, Score/ Stars bazat pe prop Rating (1-5, nullable) de la Review**
- drepturi editare si stergere products de catre colaborator doar pentru produsele adaugate de el => cerere catre admin de aprobare**
- redirect pentru creare cont user neinregistrat care vrea sa adauge un produs in cos si NU atunci cand vrea sa plaseze o comanda**
- diferenta de vizualizare product Details => registered user poate plasa in cos si lasa review-uri fata de unregistered user care nu are aceste privilegii**
- adaugare in cos => selectare cantitate per product**
- trigger pt a reduce cantitatea disponibila dintr-un produs la nivel de stoc cand e plasata o comanda**
- motor de cautare - cautare partiala dupa nume pt Products**
- rezultate cautare => filtre de cautare in functie de Price (asc sau desc; gte, lte in functie de o valoare introd de user) si de Score/ Stars**
- drepturi de editare, stergere admin la Products si Reviews**
- drepturi de revocare tip privilegiu de user pt admin**



## Modul de lucru

Sprint-uri de cate două săptămâni, ce înseamnă că task-urile din acel sprint vor fi finalizate la finalul sprint-ului, existând date intermediare de finalizare (semnalizate pentru fiecare card din workspace-ul de pe Trello) în funcție de prioritatea lor. Am decis să folosim metodologia Agile. Întâlniri pentru a scrie cod și discuta vor avea loc o dată pe săptămână, în rest lucrând în paralel în funcție de task-ul fiecăreia.

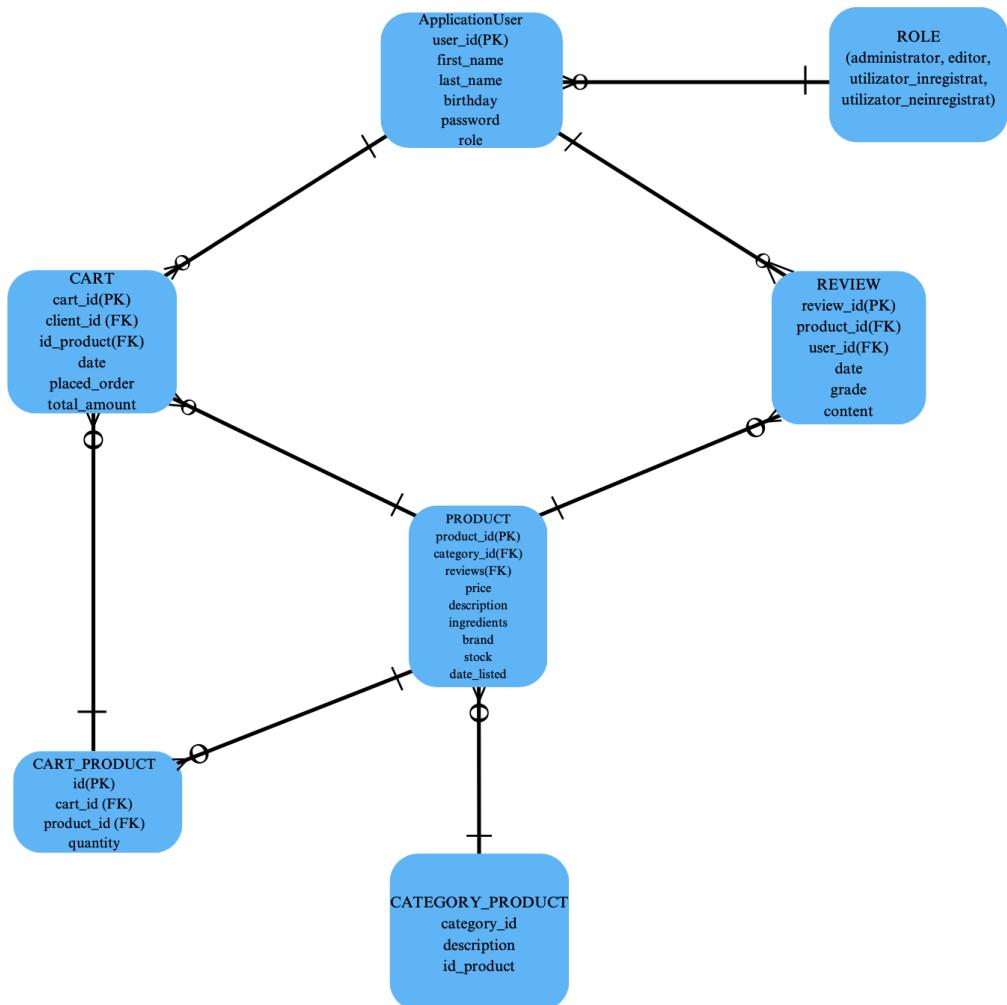
Diagrama bazei de date (varianta initială, în continuarea documentului se va găsi și varianta finală):



## Etapa 2 - 15 – 19 Noiembrie

OVERALL:

Am inceput implementarea in detaliu a claselor, dar si definirea bazei de date (migratii etc). Momentan suntem in etapa de definire a modului de lucru al aplicatiei. Inca sunt mici diferente dintre arhitecturile pe care lucreaza fiecare, dar vom incerca rezolvarea lor prin Docker. In final, am convenit la o implementare comună pentru Products, Categories și Reviews, ca după să avem o bază pentru a dezvolta Carts, CartProducts (care rezolvă relația many-to-many dintre produse și cosuri de cumpărături), care implica existența atributului de Quantity în CartProducts și alte chei straine împreună cu priorități virtuale pentru asocierea tabelelor cărora le aparțin acele foreign keys, având posibilitatea accesării altor proprietăți. Rezolvarea relației many-to-many se vede în ApplicationDbContext. În plus, am finalizat diagrama proiectului.



## Etapa 3 - 20 Noiembrie – 16 Decembrie

OVERALL:

Am incercat in repetate randuri folosirea Docker-ului, insa am avut diferite probleme tehnice pe parcurs, asa ca am revenit la folosirea unui proiect colaborativ pe Github. Vom incerca sa rezolvam individual diferențele de arhitectură (prin configurarea separată a fisierelor pentru diferențele tipuri de roluri pentru utilizatori, a fisierului appsettings.json etc.). Varianta principală ramane cea de Windows, insa pe Mac incercam dezvoltarea altor aspecte care ar trebui să meargă pe ambele arhitecturi, și anume modelele, view-urile și controller-ele. Singurele diferențe dintre cele două fiind că pe Mac lipsesc motorul de căutare, afisarea paginată și implementarea a metodelor Create și Edit pentru imaginea unui produs, din moment ce un membru s-a ocupat de aceste aspecte pentru Windows, aceea fiind varianta pe care am decis să o prezintăm.

In aceasta perioada am dezvoltat ideea de a avea 4 utilizatori, însemnând că fiecare acțiune publică din fiecare controller a primit atribuțile de Authorize(Roles="") corespunzătoare și am stabilit că Admin-ul să nu aibă restricții aproape deloc asupra datelor pe care le poate sterge sau edita. De retinut că utilizatorul neînregistrat trebuie numai să își creeze un cont, devenind implicit User și putând deveni Partner (acel utilizator care colaborează cu magazinul și adaugă produse) sau chiar Admin dacă unul dintre administratori dorește acest lucru (prin intermediul UsersController).

In continuare, am modificat puțin logica de a cunoaște exact cine a postat produse și review-uri, astfel încât folosind metoda privată SetAccessRights de la nivelul fiecarui controller (ProductsController și ReviewsController), să se afiseze butoane în Views pentru stergere și editare dacă produsul/ review-ul respectiv aparțin utilizatorului (Partner sau User, din moment ce Admin nu are nicio restricție).

Urmatorul pas urmat în această etapă a fost să manipulam în sfârșit atributul Image din clasa Product, de tip IFormFile, ce se poate observa în acțiunile Create și Edit din ProductsController.

In plus, odată ce am avut destule inserări de produse în baza de date locală, am implementat paginarea, filtrarea rezultatelor după preț și medie a rating-urilor de la nivelul tuturor review-urilor asociate fiecarui produs în parte, împreună cu un motor de căutare. Legat de paginare și cum funcționează alături de filtrare, dacă nu este selectat niciun tip de sortare crescătoare/ descrescătoare, atunci nu se va aplica niciun filtru, nefiind afectată paginarea. Este specificat în cerință că acest motor de căutare trebuie să aibă o căutare parțială pentru atributul Name de la Product numai după ce a fost introdus în search bar (de retinut că ar putea fi adăugată o implementare asemanătoare d.p.d.v. al căutării parțiale pentru Content din Review). De altfel, media de Rating a review-urilor postează pentru un produs anume va fi calculată dinamic, adică o dată cu adăugarea fiecarui review, fiind numai o afișare în view-ul partajat din Shared, numit ProductInfo, care calculează media în secțiunea de Overall Rating (am menționat asta pentru că inițial nu găsesc că Product să aibă atributul de Rating și că să fie modificat de fiecare dată o dată cu crearea, editarea sau stergerea unui review, dar ar fi însemnat să ne complicăm, deci am găsit o soluție destul de bună care permite o sortare după criteriile rating\_asc și rating\_desc folosind funcții lambda).

Dupa, am decis să facem parte cea mai importantă de logica a magazinului online, anume să existe acele cereri de aprobare, respingere a produsului propus să fie vândut la magazin de către un partener

al magazinului. A fost exprimata ideea asta la nivel de cod folosind un enum cu trei posibile valori: Approved, Rejected si Unverified. Acest lucru inseamna ca o data ce a fost creat un produs, cererea e trimisa in metoda Create din ProductsController, iar pana nu e aprobată/ respinsă de un Admin care intra in View-ul PendingApproval (ce afiseaza numai acele produse pentru care atributul RequestStatus este Unverified), produsul pur si simplu nu va fi afisat pe pagina principala a site-ului cu o conditie de afisare doar pentru produsele Accepted in Index din ProductsController (ne referim la ce a fost setat ca si cale URL default in Program.cs in MapControllerRoute, adica View-ul Index din Products). De sigur, ca intregul proces a insemnat ca si metoda Edit, cu toate ca produsul sa zicem ca a fost aprobat prima oara sa apara in magazinul online, va modifica atributul produsului editat RequestStatus din Accepted in Unverified. De aceea, apar noi metode la care doar Admin-ul are acces: PendingApproval care are un View corespunzator si Approve si Reject care nu au View-uri separate (logica e continua tot in View-ul PendingApproval).

De asemenea, cand vrea sa fie adaugat un produs in cos, acel user trebuie sa fie logat, daca nu e logat nu are acces la prea multe informatii de pe site, numai metodele Index, Details din ProductsController via AllowAnonymous, de aceea o data ce se apasa pe butonul Add To Cart din pagina de Details a unui produs, utilizatorul e prompted sa se logheze.

## Concluzie:

Cu toate dificultatile pe care le-am avut din cauza diferentelor arhitecturale, am reusit completarea partii de backend, alaturi de un frontend reusit, pe care insa il vom imbunatatii pe parcurs prin folosirea Bootstrap-ului. Proiectul este in final functional, are o baza de date care imita foarte bine una folosita de magazinele online reale (de produse de infumusetare: makeup, skincare etc). In screenshot-urile de mai jos, din Trello-ul folosit pe parcursul proiectului, am inclus tot ceea ce am vrut sa realizam si am si reusit, cuprinzand toate cerintele de backend ale proiectului.

done

github project - individual accounts pt autentificare

Nov 5

vizualizarea articolelor din platforma (orice tip de user)

Nov 10

modif Create, Edit si Delete la CategoriesController pentru adaugarea de categorii de catre admin in mod dinamic

Nov 13

progresul e temporar blocat pentru configurarea Docker-ului + MySql, am renuntat la Docker pentru echipa, lucram in paralel si perioade succesive, punand fiecare implementarea ei pe GitHub

Nov 13

model-binding nefunctional in ProductsController, metodele Create si Edit (Html/ tag elements) - implementare windows e corecta acum

Nov 10

editarea si crearea produselor (user de tip colaborator sau admin)

Nov 10

drepturi de editare, stergere admin la Products si Reviews

done

Views/Users, UsersController - 4 tipuri de utilizatori: utilizator neinregistrat, inregistrat, colaborator, administrator (pt cel neinregistrat se creaza un ont pur si simplu, ulterior adminul ii da un rol)

Dec 6

orice produs/ review postat de un editor/ colaborator poate fi editat sau sters de el insusi (cel care l-a creat)

Dec 5

Image cu IFormFile (atacat create cat si edit)

Dec 7

atributul Stock pt Product, functionare? - diagrama baza de date, de decis daca Availability la fiecare produs va determina o noua entitate Stock/ Storage si Shops - nu, fiecare produs are Stock care este scazut cand e adaugat in cos, altfel creste cand e scos din cos

Dec 7

paginare care sa fie combinata cu o sortare (daca nu e selectata nicio sortare nu se aplica niciun filtru)

Dec 9

motor de cautare - cautare partiala dupa nume pt Products

Dec 10

done

modif Create si Edit la ProductsController pentru colaborator ("Salveaza produsul" => cerere catre admin pentru aprobatie)

① Dec 13

request (de 3 tipuri, bazate pe un enum: Approved, Rejected si Unverified) catre admin impreuna cu view-ul PendingApproval (de sigur modificare logica si pentru Index - afiseaza doar produsele Approved, Create si Edit - admin va fi acceptat si nu mai trece prin acelasi proces ca si un produs editat sau creat de un partener)

drepturi editare si stergere products de catre colaborator doar pentru produsele adaugate de el => cerere catre admin de aprobatie

① Dec 13

Cart, CartProduct relatii si metode controllere

adaugare in cos => selectare cantitate per product in metodele din CartsController

① Dec 13

diferenta de vizualizare product Details => registered user poate plasa in cos si lasa review-uri fata de unregistered user care nu are aceste privilegii

redirect pentru creare cont user neinregistrat care vrea sa adauge un produs in cos si NU atunci cand vrea sa plaseze o comanda - [AllowAnonymous] in loc de authorize pt metodele Index si Details din Products

① Dec 14

rezultate cautare => filtre de cautare in functie de Price (asc sau desc; gte, lte in functie de o valoare introdusa de user) si de Rating (calculat dinamic)

① Dec 14

drepturi de revocare tip privilegiu de user pt admin in UsersController

① Dec 15

int sortedCount =  
products.Count(); e 0 pentru sortare rating\_asc si rating\_desc daca folosesc functia CalculateAverageRating

② Dec 15

modif prop Product: Rating bazat pe prop Rating (1-5, nullable) de la Review - de fapt Rating e calculat dinamic o data cu adaugarea unei noi review, nu e un atribut al lui Product

≡

Aici sunt cateva screenshot-uri cu proiectul finalizat:

This screenshot shows a product detail page from a web application. The URL in the browser is `localhost:7089/Products/Details/10010`. The product name is "Botanical Blend". The description includes ingredients like Water/Eau/Aqua, Opuntia Tuna Fruit Extract, Arnica Montana Flower Extract, Aloe Barbadensis Leaf Extract, Cypripedium Pubescens Extract, Opuntia Vulgaris Leaf Extract, Agave Tequilana Leaf Extract, Saccharomyces Cerevisiae (Yeast) Extract, Leuconostoc/Radish Root Ferment Filtrate, Caprylic/Capric Triglyceride, and 85% Sunday Riley. The overall rating is 2.3. Below the product information, there are buttons for "Add to MyCart" and "Show MyCart". The product is categorized under "Treatments". The email address associated with the product is alexandrescuandra2005@gmail.com, and it was last updated on 12/15/2024 at 7:19:46 PM. There is also a "Delete product" button.

## Reviews

Grade	Content	Date	Editing Options
3	I used about 2 bottles worth once at night, didn't notice much of a difference, but it did make my face smooth. I already have good skin, so it didn't get worse.	12/15/2024	No editing options
2		12/15/2024	No editing options
2		12/16/2024	<a href="#">Edit review</a> <a href="#">Delete review</a>

This screenshot shows the main page of the skincare shop. The title bar says "Skincare\_Online\_Shop\_.NET". The top navigation bar includes links for "Display products" and "Add new product", and a user greeting "Hello alexandrescuandra2005@gmail.com!" with a "Logout" link. The main content area is titled "Displaying skincare products". It features a search bar with "Search by keyword" and a "Search" button. Below the search bar, there is a product card for "SENTÉ Daily Soothing Cleanser". The card shows an image of the product tube, which is white with "Sente" and "Daily Soothing Cleanser" printed on it. To the right of the image, the product name "CLEANSER" is written vertically, followed by "Daily Soothing Cleanser". A descriptive text below states: "Award winning, gentle, daily cleanser ideal for sensitive skin prone to acne, eczema, and rosacea." There is also a small circular icon with a camera symbol in the bottom right corner of the card.

- Skincare\_Online\_Shop\_.NET x +

localhost:7089/Carts/Details/6002

Skincare\_Online\_Shop\_.NET Display products Add new product Hello alexandrescuandra2005@gmail.com! Logout

Shopping Cart created at: 12/16/2024 8:07:44 AM

Content of shopping cart:

Sunday Riley GOOD GENES All-In-One Lactic Acid Treatment



8:08 AM Monday 12/16/2024

- Skincare\_Online\_Shop\_.NET x +

localhost:7089/Carts/ShoppingHistory

Skincare\_Online\_Shop\_.NET Display products Add new product Display categories Display Users Pending Approval Products Hello admin@test.com! Logout

12/15/2024 7:47:24 PM Created by: partner@test.com <a href="#">Display shopping cart contents</a>	12/15/2024 7:48:47 PM Created by: partner@test.com <a href="#">Display shopping cart contents</a>	12/15/2024 7:49:45 PM Created by: user@test.com <a href="#">Display shopping cart contents</a>
12/15/2024 7:50:47 PM Created by: admin@test.com <a href="#">Display shopping cart contents</a>	12/16/2024 8:07:44 AM Created by: alexandrescuandra2005@gmail.com <a href="#">Display shopping cart contents</a>	

8:08 AM Monday 12/16/2024

© 2024 - Skincare\_Online\_Shop\_.NET - [Privacy](#)