

MINISTERUL EDUCAȚIEI



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE INGINERIE
SPECIALIZAREA CALCULATOARE

ARTICOLE SPORTIVE

Absolvent: **Crișan Antonel Gabriel**

2025

Cuprins

Capitolul 1. Introducere	Error! Bookmark not defined.
Capitolul 2. Tehnologii utilizate.....	Error! Bookmark not defined.3
Capitolul 3. Funcționalități	4
Capitolul 4. Arhitectura aplicației	9
Capitolul 5. Posibile îmbunătățiri	12
Capitolul 6. Concluzii	14
Bibliografie	15

Capitolul 1. Introducere

1.1. Descriere generală

Comerțul electronic a devenit una dintre cele mai importante modalități de a conecta consumatorii cu produsele și serviciile de care au nevoie. Cu un număr tot mai mare de utilizatori care preferă să facă cumpărături online datorită confortului și eficienței, platformele e-commerce au evoluat într-un ritm accelerat. Proiectul **ArticoleSportive** reprezintă o astfel de soluție modernă, concepută pentru a transforma experiența de achiziție a articolelor sportive într-una intuitivă, rapidă și personalizată.

ArticoleSportive este o aplicație web care își propune să creeze o legătură solidă între pasionații de sport și comercianții de articole sportive. Cu o gamă diversificată de produse, filtre avansate pentru selecție, notificări în timp real și recomandări bazate pe preferințele utilizatorilor, aplicația răspunde atât nevoilor clienților finali, cât și celor ale administratorilor care gestionează platforma.

1.2. Contextul proiectului

Lansarea și gestionarea unui magazin online pot fi provocatoare, mai ales în cazul unei nișe competitive precum articolele sportive. **ArticoleSportive** s-a născut din dorința de a oferi o soluție completă, care să includă atât funcționalități moderne pentru utilizatori, cât și un panou de administrare prietenos pentru comercianți. În acest context, aplicația a fost proiectată să răspundă următoarelor cerințe:

- Să faciliteze procesul de cumpărare printr-o interfață modernă și intuitivă.
- Să integreze tehnologii de ultimă generație pentru performanță ridicată și siguranța datelor.
- Să ofere o experiență personalizată fiecărui utilizator, utilizând algoritmi avansați de recomandare.
- Să permită comercianților să gestioneze eficient produsele, comenzile și stocurile.

1.3. Scopul aplicației

Scopul principal al **ArticoleSportive** este de a crea o platformă de e-commerce care să îmbunătățească experiența utilizatorilor, atât din punct de vedere al ușurinței de navigare, cât și al accesului rapid la produse. Pentru a realiza acest obiectiv, proiectul include o serie de funcționalități esențiale:

1. **Catalog de produse detaliat:** Toate produsele sunt organizate în categorii bine definite, fiind prezentate cu imagini, descrieri și specificații clare.
2. **Filtrare avansată:** Utilizatorii pot căuta produsele dorite în funcție de criterii precum preț, brand, categorie sau specificații tehnice.
3. **Notificări în timp real:** Clienții primesc alerte despre oferte speciale, reduceri sau actualizări legate de comenzi.

4. **Recomandări personalizate:** Pe baza istoricului de navigare și a produselor adăugate în coș, aplicația sugerează articole care ar putea fi de interes.
5. **Panou de administrare:** Administratorii pot adăuga, modifica sau șterge produse, monitoriza comenzile și analiza datele vânzărilor.

1.4. Public țintă

ArticoleSportive este dedicat unei game largi de utilizatori. Pe de o parte, se adresează clienților individuali care caută articole sportive de calitate, fie că sunt amatori sau profesioniști. Pe de altă parte, platforma este ideală pentru comercianții care doresc să gestioneze un magazin online fără a avea nevoie de cunoștințe tehnice avansate.

De asemenea, aplicația este adaptată pentru a oferi suport utilizatorilor de pe dispozitive mobile și desktop, astfel încât cumpărăturile să poată fi efectuate de oriunde, în orice moment.

Capitolul 2. Tehnologii utilizate

Această platformă a fost construită folosind o combinație de tehnologii moderne care garantează performanță, scalabilitate și o experiență de utilizator excelentă.

Frontend:

- **React:** React este o bibliotecă JavaScript pentru construirea interfețelor de utilizator. Este utilizată pentru crearea unei aplicații dinamice, ușor de întreținut și care răspunde rapid la interacțiunile utilizatorului.
- **Material-UI:** O bibliotecă de componente React care oferă un set de interfețe elegante și ușor de utilizat. Material-UI este folosită pentru a construi o interfață modernă și pentru a asigura un design consistent și responsive pe toate dispozitivele.

Backend:

- **Spring Boot:** Un framework Java care permite crearea de aplicații backend robuste și scalabile. Spring Boot este folosit pentru a expune API-uri RESTful pentru gestionarea utilizatorilor, produselor și comenzilor.
- **WebSocket:** Tehnologie utilizată pentru a trimite notificări în timp real utilizatorilor, fără a fi nevoie ca aceștia să actualizeze pagina.
- **Spring Security:** Este folosit pentru gestionarea autentificării și autorizării utilizatorilor, asigurându-se că doar utilizatorii autorizați pot accesa anumite funcționalități.

Baze de date:

- **MySQL:** Folosit pentru gestionarea datelor despre utilizatori și comenzi, fiind o bază de date relațională fiabilă și scalabilă.

Capitolul 3. Funcționalități

Must-Have (Trebuie Să Aibă) Autentificare Utilizatori: Funcționalități sigure de înregistrare, logare și delogare. Listare Produse: Pagină pentru afișarea articolelor sportive, cu categorii, filtre și opțiuni de sortare. Pagină Detalii Produs: Fiecare articol trebuie să aibă o vizualizare detaliată cu imagini, descrieri, preț și disponibilitate. Adăugare în Coș: Posibilitatea de a adăuga produse în coșul de cumpărături și de a vizualiza conținutul coșului. Proces de Finalizare a Comenzii: Un proces simplu și sigur de checkout, care include sumarul comenzii și opțiuni de plată. Integrarea API-ului Backend: Comunicare eficientă între frontend-ul React și backend-ul Spring Boot pentru gestionarea datelor.

Should-Have (Ar Trebui Să Aibă) Funcționalitate de Căutare: Bară de căutare pentru găsirea rapidă a produselor. Recenzii și Evaluări: Posibilitatea pentru utilizatori de a lăsa recenzii și de a evalua produsele. Wishlist: Opțiune de a salva produse preferate pentru cumpărături viitoare. Notificări prin E-mail: Confirmări pentru comenzi și actualizări despre livrări.

Could-Have (Ar Putea Să Aibă) Recomandări de Produse: Sugestii personalizate pe baza istoricului de cumpărături. Blog sau Secțiune de Știri: Articole despre echipamente sportive, noutăți și sfaturi. Opțiuni de Livrare Variate: Livrare standard și rapidă, cu estimări ale timpului de livrare.

Won't-Have (Nu Va A avea) Integrare cu Platforme de Social Media: Deși utilă, nu este esențială pentru versiunea inițială. Realitate Augmentată pentru Vizualizare Produse: Funcționalitate avansată care nu va fi implementată în această etapă.

Această aplicație este concepută pentru a fi ușor de utilizat. Pașii pe care utilizatorii trebuie să-i urmeze pentru a beneficia de toate funcționalitățile platformei:

3.1. Creare cont și autentificare

Înregistrare: Utilizatorii trebuie să completeze un formular de înregistrare cu informațiile necesare (nume, email, parolă) și să trimită formularul.

The image shows a registration form titled "Creare cont" (Create account). It is centered on a light gray background. The form itself is a white rounded rectangle. It contains three input fields stacked vertically: "Nume *" (Name), "Email *" (Email), and "Parola *" (Password). Below these fields is a prominent orange button with the text "INREGISTREAZA-TE" (REGISTER) in white. At the bottom of the form, there is a link that says "Ai deja cont?" (Do you already have an account?).

Fig 3.1

Autentificare: După înregistrare, utilizatorii se pot autentifica cu credențialele lor pentru a accesa contul personal.

The image shows a login form titled "Autentificare". It contains two input fields: "Email *" and "Parola *". Below the password field is a link "Ai uitat parola?". A large orange button labeled "AUTENTIFICA-TE" is positioned below the link. At the bottom, there is another link "Ai nevoie de cont?".

Fig 3.2

3.2. Căutare produse

Utilizatorii pot căuta produse prin intermediul barei de căutare, care permite căutări simple sau avansate (după categorie, preț, etc.).

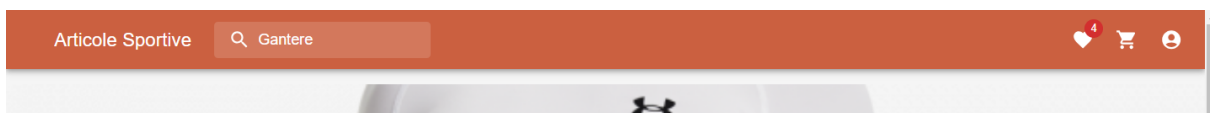


Fig 3.3

3.3. Vizualizare detalii produs

Fiecare produs are o pagină de detalii, care include informații complete (descriere, imagini, preț) și opțiuni pentru a-l adăuga în coșul de cumpărături.



Fig 3.4

3.4. Coș de cumpărături și finalizare comandă

Produsele adăugate în coș pot fi vizualizate oricând. La finalizarea comenzii, utilizatorii completează datele necesare pentru livrare și plătesc.

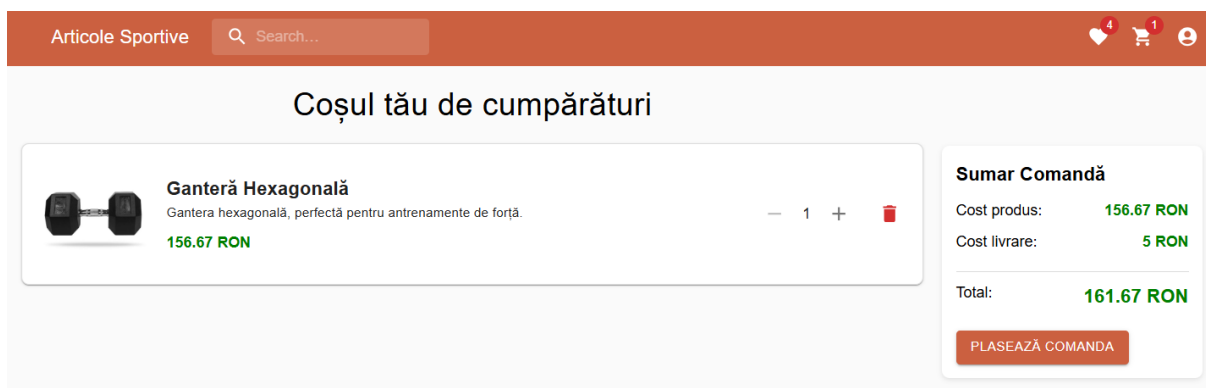


Fig 3.5

3.5. Notificări în timp real

Dacă utilizatorul finalizează o comandă, va primi o notificare în timp real prin WebSocket, confirmând comanda.

Capitolul 3. Funcționalități

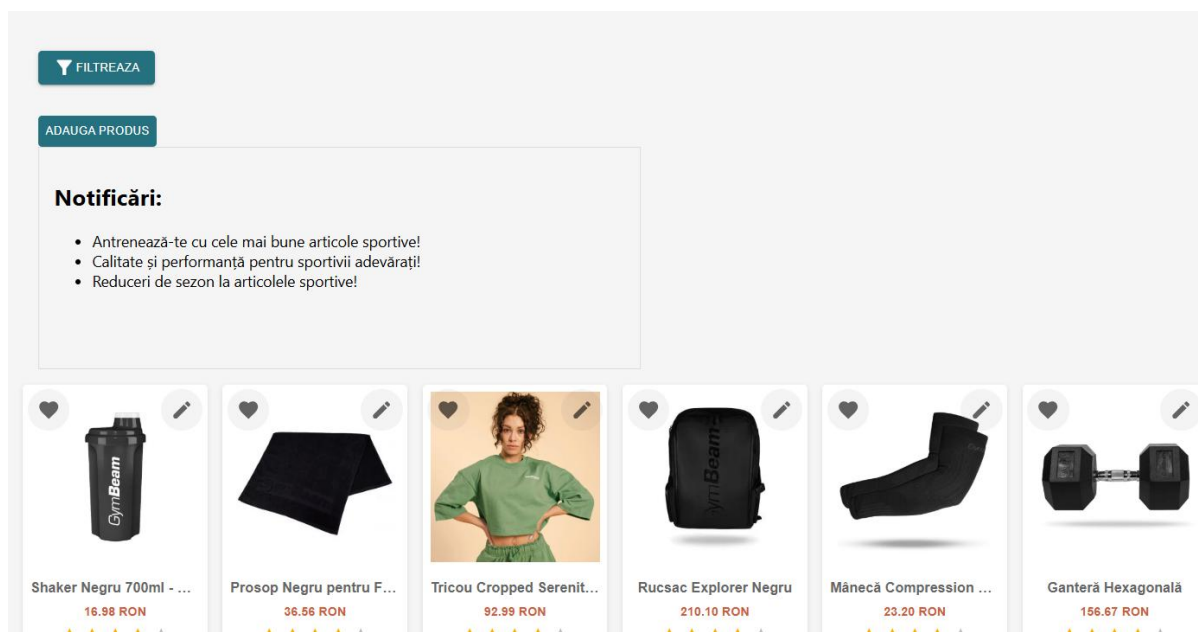


Fig.3.6

Capitolul 4. Arhitectura aplicației

Arhitectura aplicației web pentru gestionarea articolelor sportive este bazată pe un model de tip **n-tier (multi-layered)**, organizat în trei straturi principale: **frontend**, **backend** și **baza de date**. Fiecare strat este responsabil pentru o parte distinctă a funcționalităților aplicației, ceea ce facilitează dezvoltarea, întreținerea și scalabilitatea sistemului.

4.1. Structura generală a aplicației

1. Frontend (Prezentare):

- Dezvoltat folosind **React**, acesta este responsabil pentru interfața utilizatorului și interacțiunile cu sistemul.
- Frontend-ul utilizează componente reutilizabile, precum butoane, carduri de produse și bare de navigare, pentru a asigura o experiență de utilizare coerentă și plăcută.
- Comunicarea cu backend-ul se realizează prin intermediul cererilor HTTP folosind protocolul REST sau prin WebSockets pentru notificări în timp real.

2. Backend (Logică de business):

- Implementat în **Spring Boot**, backend-ul gestionează logica aplicației, procesarea datelor și integrarea cu baza de date.
- Structura backend-ului este modulară, fiind organizată în pachete care corespund diverselor responsabilități:
 - **Controller:** Preia cererile de la frontend și returnează răspunsuri adecvate.
 - **Service:** Conține logica de afaceri și este responsabil pentru procesarea datelor înainte de a fi transmise către frontend sau persistate în baza de date.
 - **Repository:** Realizează operațiunile CRUD cu baza de date folosind JPA.
- Backend-ul include și funcționalități de autentificare și autorizare pentru utilizatori, utilizând **Spring Security**.

3. Baza de date:

- Aplicația folosește două tipuri de baze de date:
 - **MySQL** pentru gestionarea datelor tranzacționale.
- Datele sunt organizate în tabele și colecții, conform unui model relațional și non-relațional, pentru a optimiza performanța și scalabilitatea.

4.2. Interacțiunile între componente

1. Frontend - Backend:

- Comunicarea între frontend și backend se face prin API-uri RESTful. Exemple de endpoint-uri includ:
 - /api/produse pentru gestionarea produselor.
 - /api/utilizatori pentru gestionarea utilizatorilor.
- Pentru notificări în timp real, frontend-ul se conectează la backend folosind **WebSockets**, ceea ce permite actualizarea automată a interfeței fără a fi necesar un refresh manual.

2. Backend - Baza de date:

- Backend-ul folosește Hibernate pentru a realiza maparea obiect-relatională (ORM) cu baza de date MySQL.
- Operațiunile CRUD sunt gestionate prin repository-uri specifice, de exemplu:
 - ProdusRepository pentru produse.
 - NotificareRepository pentru notificări.
- MongoDB este accesată prin intermediul unui driver specific și a unor clase de service dedicate, care oferă flexibilitate în stocarea și regăsirea documentelor JSON-like.

3. Backend - Servicii auxiliare:

- Aplicația utilizează WebSockets pentru notificările în timp real. Acestea sunt gestionate de clasa NotificationScheduler, care programează și trimite notificări către utilizatori.
- Backend-ul include și validări suplimentare pentru date, prin intermediul anotărilor precum @Valid și @NotNull.

4.3. Interacțiunile între componente

1. Separarea responsabilităților:

- Fiecare strat are un rol bine definit, ceea ce facilitează dezvoltarea și depanarea.

2. Scalabilitate:

- Atât frontend-ul, cât și backend-ul pot fi extinse independent pentru a adăuga noi funcționalități.

3. Performanță optimizată:

- Utilizarea WebSockets pentru notificări reduce încărcarea serverului prin eliminarea cererilor constante de tip polling.

4. Flexibilitate:

- Folosirea a două baze de date diferite permite gestionarea eficientă a diverselor tipuri de date, fiecare fiind optimizată pentru cerințe specifice.

5. Experiență utilizator îmbunătățită:

- Interfața dinamică React asigură o experiență de utilizare interactivă și intuitivă.

4.4. Fluxul datelor în aplicație

1. Utilizatorul accesează aplicația și inițiază o acțiune, cum ar fi vizualizarea produselor.
2. Frontend-ul trimite o cerere către backend printr-un API REST.
3. Backend-ul procesează cererea și interacționează cu baza de date pentru a obține informațiile solicitate.
4. Răspunsul este returnat către frontend sub formă de date JSON, care sunt apoi afișate utilizatorului.
5. În cazul notificărilor, backend-ul inițiază o conexiune WebSocket către frontend, trimițând datele necesare pentru afișarea notificării în timp real.

Această arhitectură modulară și bine structurată asigură atât performanța aplicației, cât și o experiență superioară pentru utilizatori.

Capitolul 5. Posibile îmbunătățiri

Dezvoltarea aplicației web pentru gestionarea articolelor sportive deschide numeroase oportunități pentru îmbunătățiri și extindere în viitor. Unele dintre acestea includ:

5.1. Funcționalități adiționale pentru utilizatori

1. Recomandări personalizate de produse:

- Integrarea unui sistem bazat pe machine learning care să analizeze preferințele utilizatorilor și să ofere sugestii personalizate de produse.

2. Recenzii și evaluări:

- Permite utilizatorilor să lase recenzii și să acorde evaluări produselor achiziționate, sporind încrederea altor clienți.

3. Wishlist:

- Implementarea unei funcționalități prin care utilizatorii pot salva produse pentru a le achiziționa ulterior.

5.2. Optimizări pentru backend

1. Caching:

- Utilizarea unui sistem de caching (de exemplu, Redis) pentru a reduce timpul de răspuns al aplicației, mai ales pentru cererile frecvente.

2. Migrarea către microservicii:

- Extinderea arhitecturii pentru a include microservicii independente, care să gestioneze diferite funcționalități ale aplicației (de exemplu, gestionarea utilizatorilor, procesarea plăților).

3. Scalabilitate verticală și orizontală:

- Configurarea aplicației pentru a suporta o creștere semnificativă a numărului de utilizatori și a volumului de date, utilizând instrumente precum Kubernetes pentru orchestrarea containerelor.

5.3. Optimizări pentru frontend

1. Responsive design avansat:

- Perfecționarea interfeței pentru a oferi o experiență optimă pe toate dispozitivele, inclusiv tablete și telefoane mobile.

2. Performanță îmbunătățită:

- Reducerea dimensiunii resurselor (imagini, scripturi) și utilizarea tehnicilor precum lazy loading pentru a accelera încărcarea paginilor.

3. Dark mode:

- Adăugarea unui mod întunecat pentru a spori confortul utilizatorilor care navighează pe site în condiții de iluminare redusă.

5.4. Extinderea notificărilor

1. Notificări prin email:

- Implementarea unui sistem care să trimită notificări prin email utilizatorilor pentru actualizări importante, oferte speciale sau confirmări de comenzi.

2. Notificări push:

- Integrarea notificărilor push pentru utilizatorii care accesează aplicația pe dispozitive mobile sau desktop.

Capitolul 6. Concluzii

Proiectul web pentru gestionarea articolelor sportive reprezintă o soluție modernă, eficientă și scalabilă, capabilă să răspundă cerințelor variate ale utilizatorilor. Dezvoltat pe o arhitectură solidă și utilizând tehnologii de ultimă generație precum React, Spring Boot, MySQL, aplicația oferă o interfață intuitivă, o performanță optimizată. Deși există posibilități de îmbunătățire, cum ar fi recomandări personalizate de produse, recenzii și evaluări, wishlist, aplicația reprezintă o bază solidă pentru extindere și evoluție în viitor.

Bibliografie

“React” - <https://react.dev>

“Spring Boot” - <https://spring.io/projects/spring-boot>

“MySQL” - <https://dev.mysql.com/doc/>

“Material-UI” - <https://mui.com>

“WebSocket API Specification” - https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API

Bibliografie