

PROIECT

AFACERI ELECTRONICE

Student: Ifrim Ana-Maria

BDSA

Url Github: <https://github.com/AnaM229/AE-project>

1. Introducere

Proiectul dezvoltat are ca scop realizarea unei aplicatii web de tip comert electronic, care sa implementeze un flux complet de gestionare a produselor, autentificare a utilizatorilor si administrare a unui cos de cumparaturi. Aplicatia urmarest sa ofere o experienta moderna si interactiva, fiind construita pe baza unei arhitecturi RESTful, in care front-end-ul comunica cu back-end-ul prin intermediul unui API.

Prin intermediul acestei aplicatii, utilizatorii se pot inregistra si autentifica in sistem, pot vizualiza lista produselor disponibile si pot adauga produse in cosul de cumparaturi. Administratorul are posibilitatea de a adauga, edita sau sterge produse, asigurand astfel o administrare completa a catalogului. Colectarea si persistenta datelor se realizeaza intr-o baza de date relationala, gestionata prin intermediul unui ORM, iar comunicarea dintre client si server este securizata prin token-uri de autentificare.

2. Arhitectura generala a aplicatiei

Aplicatia este formata din doua componente majore: partea de back-end, responsabila cu logica de server si persistenta datelor, si partea de front-end, care reprezinta interfata interactiva cu utilizatorul. Cele doua componente comunica exclusiv prin intermediul unor rute API, ceea ce asigura o separare clara intre logica de business si interfata grafica.

Back-end-ul a fost implementat folosind Node.js si Express.js, tehnologii care permit crearea rapida a unui server REST. Pentru maparea obiect-relationala s-a folosit Sequelize, un ORM care simplifica interactiunea cu baza de date SQLite. Aceasta asigura o gestionare sigura a tabelelor, relatiilor si operatiilor CRUD pentru toate entitatile definite in aplicatie.

Front-end-ul a fost realizat in React, folosind TypeScript pentru tipare stricte si siguranta sporita in timpul dezvoltarii. Gestionarea starii aplicatiei s-a facut prin Redux Toolkit, iar interfata grafica utilizeaza biblioteca Material UI, care ofera componente moderne, responsive si usor de personalizat. Comunicarea intre front-end si back-end este intermediata de biblioteca Axios, care gestioneaza apelurile HTTP catre server.

3. Tehnologii folosite si modul de integrare

Pentru partea de server, Node.js a fost ales datorita flexibilitatii si compatibilitatii cu Express, un framework minimalist care permite gestionarea rutelor si a middleware-urilor intr-un mod clar si modular. Sequelize a fost utilizat pentru a crea modelele aferente celor trei entitati principale ale aplicatiei: User, Product si CartItem. Aceste modele sunt interconectate prin relatii

de tipul "unul la mai multi", astfel incat fiecare utilizator poate avea mai multe produse in cos, iar fiecare produs poate aparea in mai multe cosuri.

Baza de date folosita este SQLite, aleasa datorita simplitatii de configurare si portabilitatii sale, fiind ideală pentru aplicatii educationale si proiecte locale.

Pentru securitatea datelor, autentificarea utilizatorilor s-a realizat prin intermediul mecanismului JWT (JSON Web Token). In momentul autentificarii, serverul genereaza un token semnat, care este stocat in localStorage pe partea de client si este transmis in antetul fiecarei cereri catre server. Astfel, accesul la rutele protejate este permis doar utilizatorilor autentificati.

Pe partea de client, React asigura structura componentizata a aplicatiei. Interfata a fost gandita in mod responsive si intuitiv, cu ajutorul Material UI, astfel incat sa ofere o experienta moderna de utilizare. Redux Toolkit gestioneaza starea aplicatiei, coordonand datele utilizatorului, produsele si elementele din cosul de cumparaturi.

Prin intermediul Axios, datele sunt obtinute si actualizate prin apele catre API-ul RESTful definit in back-end, garantand sincronizarea permanenta dintre interfata si baza de date.

4. Instructiuni de pornire a proiectului

Pentru a rula proiectul, este necesar ca ambele componente, front-end si back-end, sa fie instalate si configurate corespunzator.

Pe partea de server, se instaleaza dependintele folosind comanda npm install, apoi se creeaza fisierul .env care contine cheia secreta pentru JWT si portul de rulare. Serverul se porneste cu npm run dev, moment in care aplicatia devine disponibila la adresa <http://localhost:4000>.

Pe partea de front-end, se instaleaza dependintele React si se porneste aplicatia prin comanda npm start, accesand ulterior adresa <http://localhost:3000>. Cele doua componente comunica automat prin rutele definite in Axios, utilizand acelasi port de backend pentru accesul la date.

5. Functionalitati implementate

Aplicatia permite autentificarea si inregistrarea utilizatorilor, cu validare prin token JWT. Odata autentificat, utilizatorul are acces la lista de produse, pe care le poate adauga in cos. Cantitatea fiecarui produs din cos poate fi modificata, iar totalul se actualizeaza automat. De asemenea, utilizatorul poate sterge un produs individual sau poate golii complet cosul de cumparaturi, operatiile fiind reflectate si in baza de date.

Administratorul are rol special in sistem, avand acces la panoul de administrare produse, unde poate adauga, edita sau sterge produse din catalog. Astfel, aplicatia implementeaza atat partea de client, cat si cea de management al datelor, oferind un flux complet de tip e-commerce.



Pagina Register permite crearea unui cont nou in cadrul aplicatiei AE Shop. Utilizatorul introduce adresa de email si o parola, care trebuie confirmata pentru validare. Datele sunt trimise catre server, unde sunt verificate si inregistrate in baza de date, iar dupa succesul procesului de inregistrare, utilizatorul este redirectionat catre pagina de autentificare.

Pagina Login permite autentificarea utilizatorilor existenti pe baza adresei de email si a parolei introduse. Dupa validarea credentialelor pe server, utilizatorul primeste un token de acces care ii permite utilizarea functiilor protejate din aplicatie, cum ar fi adaugarea produselor in cos sau accesarea panoului de administrare. De asemenea, pagina contine un link rapid catre zona de inregistrare pentru utilizatorii noi.

AE Shop

ADMIN PANEL CART LOGOUT (ADMIN@TEST.COM)

Products

iPhone
1200 RON
In stock: 19

[ADD TO CART](#)

Laptop
3600 RON
In stock: 10

[ADD TO CART](#)

Pagina AE Shop reprezinta zona principala de interactiune a utilizatorului cu catalogul magazinului electronic. Aceasta afiseaza produsele disponibile in baza de date, fiecare fiind prezentat sub forma unui card ce contine denumirea produsului, pretul exprimat in RON, si stocul disponibil. Utilizatorul autentificat poate adauga un produs in cosul de cumparaturi prin intermediul butonului Add to Cart.

Interfata este simpla si intuitiva, realizata cu ajutorul componentelor Material UI, asigurand o experienta de navigare moderna si responsive. Bara de navigare din partea superioara permite accesul rapid catre panoul de administrare, cosul de cumparaturi si optiunea de delogare, afisand in acelasi timp adresa de email a utilizatorului curent.

AE Shop

ADMIN PANEL CART LOGOUT (ADMIN@TEST.COM)

Admin Panel

Add Product

Title

Price

Stock

Currency

RON

[CREATE](#)

Existing Products

iPhone — 1200 RON (Stock: 19)

[EDIT](#) [DELETE](#)

Laptop — 3600 RON (Stock: 10)

[EDIT](#) [DELETE](#)

Pagina Admin Panel este dedicata administratorului aplicatiei si permite gestionarea completa a catalogului de produse. In aceasta sectiune, administratorul poate adauga un produs nou completand campurile pentru titlu, pret, stoc si moneda, urmand sa confirme prin butonul Create. Sub formularul de adaugare este afisata lista produselor existente in baza de date, fiecare insotit de butoanele Edit si Delete, care permit modificarea sau stergerea imediata a unui produs.

Aceasta interfata asigura un mod simplu si rapid de actualizare a ofertei din magazin, menținând consistența datelor între client și server.

The screenshot shows a user interface for a shopping cart. At the top, there's a blue header bar with the text "AE Shop" on the left and "ADMIN PANEL CART LOGOUT (ADMIN@TEST.COM)" on the right. Below the header is a table titled "My Cart". The table has columns for "Product", "Quantity", "Price", "Subtotal", and "Actions". There are two rows in the table: one for an iPhone (quantity 1, price 1200.00 RON, subtotal 1200.00 RON) and one for a Laptop (quantity 1, price 3600.00 RON, subtotal 3600.00 RON). Each row has a minus sign, a plus sign, and a trash can icon in the "Actions" column. Below the table, the total amount "Total: 4800.00 RON" is displayed, followed by a red "CLEAR CART" button.

Product	Quantity	Price	Subtotal	Actions
iPhone	- 1 +	1200.00 RON	1200.00 RON	
Laptop	- 1 +	3600.00 RON	3600.00 RON	

Total: 4800.00 RON

CLEAR CART

Pagina My Cart centralizeaza produsele selectate de utilizator si afiseaza informatii detaliate despre acestea, precum denumirea, pretul unitar, cantitatea si subtotalul calculat automat. Utilizatorul poate ajusta cantitatea fiecarui produs direct din interfata, prin butoanele „+” si „-”, iar totalul general se actualizeaza in timp real. De asemenea, fiecare produs poate fi eliminat individual din cos prin iconita de stergere, sau intregul cos poate fi golit instant prin butonul Clear Cart.

6. Concluzii

Proiectul AE Shop a avut ca scop realizarea unei aplicatii web de tip magazin online, care sa permita inregistrarea utilizatorilor, administrarea produselor si gestionarea unui cos de cumparaturi. Aplicatia functioneaza corect, oferind o interfata usoara de utilizat si o comunicare sigura intre partea de client si server.

Dezvoltarea proiectului a ajutat la intelegherea modului in care diferite tehnologii web pot fi folosite impreuna pentru a crea o aplicatie completa si functionala.