

ACADEMIA DE STUDII ECONOMICE DIN BUCUREȘTI  
FACULTATEA DE CIBERNETICĂ, STATISTICĂ ȘI  
INFORMATICĂ ECONOMICĂ  
SPECIALIZAREA INFORMATICĂ ECONOMICĂ

LUCRARE DE LICENȚĂ

COORDONATOR ȘTIINȚIFIC:

Prof. univ. dr. Ghilic-Micu Bogdan

ABSOLVENT:

Catalina Aurora-Elena

BUCUREȘTI

2025

ACADEMIA DE STUDII ECONOMICE DIN BUCUREȘTI  
FACULTATEA DE CIBERNETICĂ, STATISTICĂ ȘI  
INFORMATICĂ ECONOMICĂ  
SPECIALIZAREA INFORMATICĂ ECONOMICĂ

# REALIZAREA UNEI APLICAȚII WEB PENTRU GESTIONAREA UNEI BIBLIOTECI

COORDONATOR ȘTIINȚIFIC:

Prof. univ. dr. Ghilic-Micu Bogdan

ABSOLVENT:

Catalina Aurora-Elena

BUCUREȘTI

2025

## CONȚINUT

<b>Introducere .....</b>	<b>4</b>
<b>Motivarea alegerii temei .....</b>	<b>4</b>
<b>Obiectivele lucrării.....</b>	<b>6</b>
<b>1. Descrierea problemei economice .....</b>	<b>7</b>
<b>1.1. Prezentarea organizației sau a domeniului abordat .....</b>	<b>7</b>
<b>1.2. Prezentarea activității care va fi informatizată.....</b>	<b>8</b>
<b>1.3. Comparăția cu alte produse/aplicații software existente în domeniul abordat.....</b>	<b>9</b>
<b>2. Analiza și proiectarea aplicației informatice.....</b>	<b>13</b>
<b>2.1. Specificarea cerințelor aplicației informatice .....</b>	<b>13</b>
<b>2.1.1 Funcționalități pentru utilizatori .....</b>	<b>13</b>
<b>2.1.2 Funcționalități pentru administrator.....</b>	<b>15</b>
<b>2.1.3 Cerințe privind datele de intrare și ieșire.....</b>	<b>15</b>
<b>2.2. Analiza aplicației .....</b>	<b>16</b>
<b>2.2.1 Diagrama cazurilor de utilizare .....</b>	<b>17</b>
<b>2.2.2 Diagrama de clase.....</b>	<b>18</b>
<b>2.2.3 Diagrama de activitate .....</b>	<b>19</b>
<b>2.3. Proiectarea aplicației.....</b>	<b>20</b>
<b>2.3.1 Stabilirea arhitecturii sistemului.....</b>	<b>20</b>
<b>2.3.2 Alegerea unui model de date.....</b>	<b>21</b>
<b>2.3.3 Modelarea diagramei de interacțiune.....</b>	<b>23</b>
<b>3. Implementare aplicației informatice.....</b>	<b>25</b>
<b>3.1. Prezentarea tehnologiilor .....</b>	<b>25</b>
<b>3.1.1 Front-End .....</b>	<b>26</b>
<b>3.1.1 Back-End .....</b>	<b>29</b>
<b>3.2. Implementarea aplicației .....</b>	<b>30</b>
<b>3.2.1 Implementarea funcționalității de autentificare/înregistrare utilizatori.....</b>	<b>32</b>
<b>3.2.2 Implementarea panoului administrativ.....</b>	<b>32</b>
<b>3.2.3 Implementarea catalogului digital .....</b>	<b>34</b>

3.2.4 Gestionarea coșului de cumpărături și efectuarea comenzilor .....	37
3.3. <i>Prezentarea aplicației</i> .....	38
Concluzii .....	54
Bibliografie .....	56
Anexe .....	58

## Introducere

În ultimii ani, sectorul bibliotecilor și al serviciilor informaționale a trecut prin transformări semnificative în ceea ce privește modul de gestionare al colecțiilor, de organizare și furnizare a serviciilor. Astăzi, bibliotecile sunt sisteme dinamice, care au puterea de a interacționa cu cititorii fără restricțiile limitelor geografice. Noțiunea de spațiu fizic, deși încă valabilă, coexistă cu noțiunea de spațiu electronic. Utilizatorul nu mai trebuie să meargă la rafturi pentru a căuta cartea dorită, cu șansa de a o găsi împrumutată sau folosită de altcineva. În schimb, din confortul propriei case, trebuie doar să acceseze catalogul online al bibliotecii și să citească varianta electronică, fie să verifice disponibilitatea documentului căutat ca mai apoi să-l împrumute.

Procesul de a digitaliza o bibliotecă se referă la implementarea de tehnologii cu scopul de a îmbunătăți modul în care aceste instituții furnizează informații și servicii. Acesta poate include transferarea informațiilor într-o bază de date, automatizarea serviciilor, dar și oferirea accesului la resurse digitale, precum publicații electronice. Numeroase organizații optează pentru digitalizare cu scopul de a îmbunătăți experiența utilizatorilor, crescând astfel nivelul de implicare al acestora. Prin oferirea de resurse multimedia interactive sunt satisfăcute nevoi ale clientului, iar prin automatizarea sistemului, performanța angajatului este optimizată.

Există o multitudine de factori care au motivat dezvoltarea bibliotecilor digitale. Un exemplu îl reprezintă necesitatea de a face informațiile mai ușor accesibile, mai ales în cadrul unei comunități academice. Există însă provocări uriașe în ceea ce privește achiziționarea, păstrarea și accesul la colecții digitale. Cu cât numărul de documente electronice crește, apar probleme din punct de vedere al catalogării, căutării și conservării. Pentru utilizatori, valoarea bibliotecilor constă nu numai în propriile resurse, precum colecțiile de cărți deținute, ci și în managementul conținutului, anume catalogarea și căutarea eficientă și fluidă a resurselor.

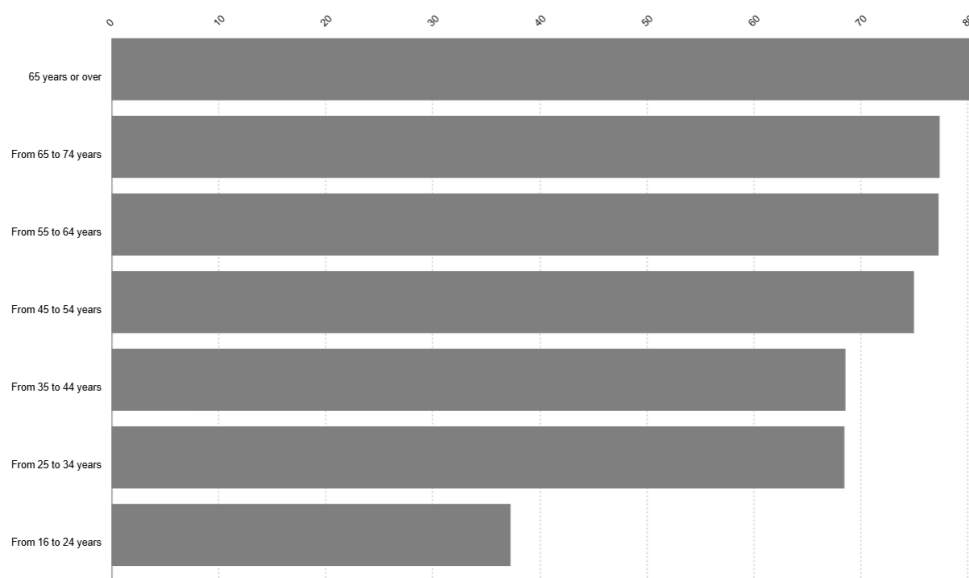
## Motivarea alegerii temei

Tehnologia informației și a comunicării reprezintă unul dintre numeroasele avantaje ale secolului 21. În momentul de față, utilizatorii înclină către resurse electronice și biblioteci accesibile 24/7/365. Astfel, aceștia pot naviga, căuta și accesa materiale într-un mod convenabil, oricând și oriunde doresc, spre deosebire de bibliotecile fizice cu ore de funcționare stabilite. Când ne gândim la biblioteci online ne gândim la scalabilitate, datorită spațiului de găzduire al unei colecții în continuă creștere, care nu prezintă constrângeri fizice și spațiale. În consecință, o eră digitală determină nevoia unui site web de gestiune al unei biblioteci, care să ofere cele mai bune

servicii în ceea ce privește accesarea resurselor cu ușurință cât și optimizarea experienței utilizatorilor.

Utilizarea tehnologiei în spațiul unei instituții de informare și lectură extinde accesul la resurse dincolo de locația fizică. Volumele digitale, fie că vorbim de cărți, jurnale sau manuscrise, pot fi accesate de la distanță, satisfăcând astfel nevoile indivizilor cu probleme de mobilitate sau care nu dispun de mijloacele necesare pentru a ajunge într-un asemenea spațiu.

De asemenea, digitalizarea reprezintă o modalitate de personalizare a spațiului online. Prin crearea unui design modern, website-ul poate deveni țința unui public nou, și anume tineri cu vârsta cuprinsă între 16 și 24 de ani. În figura următoare se prezintă procentul persoanelor din România care au citit cel puțin o carte în ultimele 12 luni, grupate pe categorii de vârstă, în anul 2024.



Procentul persoanelor din România care au citit cel puțin o carte în ultimele 12 luni

Sursa: Eurostat (2024)

Conform datelor, doar 37% din tineri au citit cel puțin o carte în ultimul an. În altă ordine de idei, aproximativ 6 din 10 tineri nu citesc nicio carte timp de 12 luni. Acest lucru sugerează un dezinteres pentru lectură în rândul persoanelor cu o vârstă fragedă. Adolescenții și tinerii adulți petrec cea mai mare parte din timp pe rețele de socializare precum TikTok, YouTube sau Instagram, unde consumă short-form content sau conținut de scurtă durată. Acest fenomen nu se datorează doar conținutului ușor de digerat, ci este cauzat și de accesul limitat la cărțile dorite.

Prețul cărților este în continuă creștere, iar bibliotecile sunt percepute ca spații demodate. Astfel, crearea unei platforme atractive pentru tineri, cu design intuitiv, o interfață modernă cu elemente vizuale dinamice precum animații, și folosirea unei palete de culori vii ar putea aduce o creștere semnificativă al utilizatorilor cu vârsta cuprinsă între 16 și 24 de ani.

## Obiectivele lucrării

Obiectivul principal al proiectului este de a realiza o aplicație web ce poate fi folosită pentru gestionarea unei biblioteci. Mi-am propus ca portalul web să aibă o funcționalitate ușor de înțeles, să fie accesibil pentru oricine, iar design-ul să fie fluid, nefăcând utilizatorul să se simtă confuz. Populația țintă a site-ului este reprezentată, în principal, de populația de cititori dintr-o comunitate, la care se adaugă și administratorii bibliotecii. Așadar, biblioteca electronică are ca scop oferirea de informații cu privire la colecția de cărți și publicații, accesul la resurse de încredere, dar și economisirea de timp al utilizatorilor cât și al angajaților, făcând procesul de căutare și gestionare mult mai ușor. De asemenea, pentru a aduce un element de diferențiere față de alte website-uri din aceeași arie, am integrat animații și tranziții, folosind diverse librării precum framer-motion sau react-toastify.

# 1. Descrierea problemei economice

Bibliotecile sunt esențiale pentru păstrarea și diseminarea cunoașterii, având o istorie îndelungată ce se întinde încă din antichitate. Rolul lor a evoluat odată cu societatea, dar obiectivul fundamental rămâne neschimbat: facilitarea accesului liber și egal la informație pentru toate categoriile sociale. În acest context, înțelegerea modului în care bibliotecile, în special cele publice, își gestionează colecțiile și serviciile este esențială pentru a identifica nevoile de modernizare și informatizare.

## 1.1. Prezentarea organizației sau a domeniului abordat

Bibliotecile sunt instituții care au ca atribuții principale gestionarea, conservarea și dezvoltarea unor colecții de materiale, incluzând cărți, manuscrise și documente istorice, reviste și publicații periodice, cât și resurse digitale.

Luând naștere încă din antichitate, acestea au constituit un loc unde informația a putut fi păstrată de-a lungul timpului și transmisă către generații viitoare. Odată cu trecerea timpului, ele și-au păstrat structura de bază, însă au îmbrăcat diverse veșminte. Fie că discutăm despre biblioteci universitare, publice sau private, scopul rămâne mereu același, anume de a oferi acces la informații și cunoștințe. Cu toate acestea, ceea ce diferențiază o bibliotecă publică de alte tipuri de biblioteci este diversitatea. O bibliotecă publică are ca obiectiv satisfacerea nevoilor unui individ indiferent de rasă, naționalitate, vârstă, gen sau existența unor studii superioare, asigurând acces egal, neluând în considerare statutul social sau economic. În trecut, astfel de biblioteci au fost construite cu ajutorul donațiilor oferite de filantropi care considerau că dreptul de a citi nu poate fi rezervat doar pentru persoane înstărite. Conform unui articol postat pe Wikipedia („Biblioteca Batthyaneum”, 2025), biblioteca Batthyaneum poate fi considerată un exemplu potrivit în acest sens, lucrările de amenajare din perioada 1792-1798 fiind finanțate de episcopul Ignățiu Batthyány. Fiind formată, la înființare, din doar 18000 de volume care includeau manuscrise sau cărți de tip bibliografic, la ora actuală biblioteca păstrează peste 70000 de lucrări. Cu toate acestea, deși donațiile reprezentau temelia acestor instituții, în prezent finanțările pentru astfel de biblioteci provin din fonduri publice, precum bugetul local sau național.

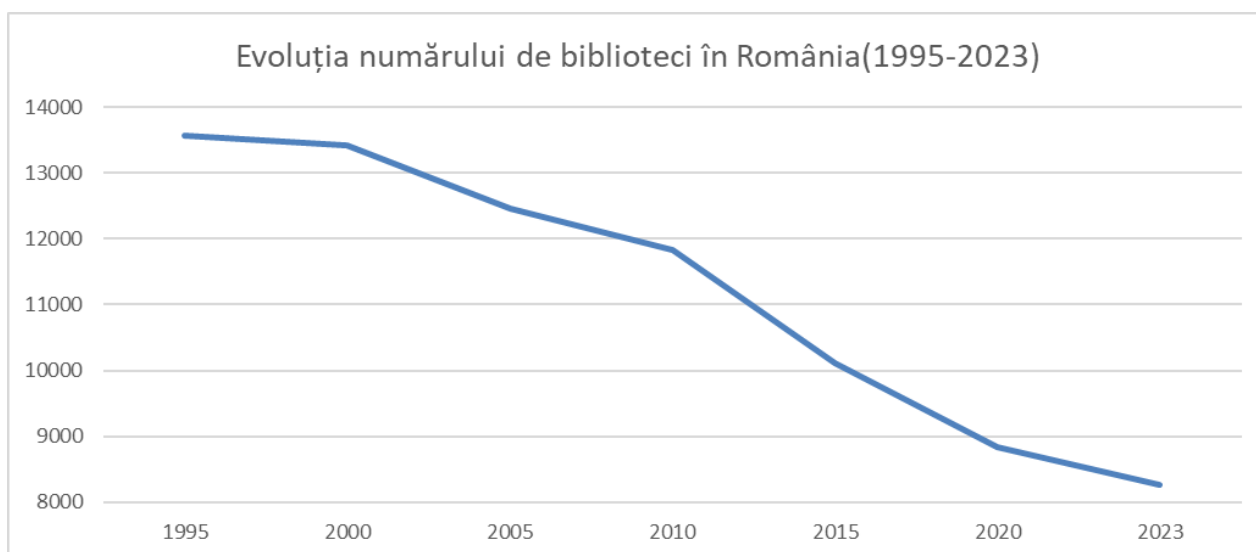
Considerând aportul pozitiv pe care îl are o bibliotecă publică pentru comunitate, am considerat ca organizația Studysphere să facă parte din aceeași categorie. Unul din motivele care au stat la baza acestei decizii îl reprezintă tratamentul diferențiat între membrii comunității academice și restul cetățenilor, promovat în cadrul bibliotecilor private sau universitare. Deși acest comportament nu reprezintă un act de discriminare, creează o barieră între utilizator și instituție, perpetuând conceptul de inechitate în ceea ce privește accesul la educație. Astfel, adulții care doresc să se reconvertească profesional prin autoeducare sunt descurajați, anumite categorii sociale se vor simți excluse, iar în jurul bibliotecii se va crea o percepție de elitism.



## 1.2. Prezentarea activității care va fi informatizată

În teorie, avantajele aduse de existența unei biblioteci într-o comunitate sunt numeroase, iar impactul generat de aceasta ar trebui să fie unul semnificativ. Schimbările rapide atât pe frontul tehnologiei, cât și în contextul socio-cultural generează o realitate diferită, o realitate care include un număr de volume împrumutate în continuă scădere, din ce în ce mai puține permise eliberate și, în final, desființarea acestor instituții.

Conform datelor furnizate de Institutului Național De Statistică, din 1995 și până în prezent peste 5000 de biblioteci din România au fost desființate, iar numărul de cititori este în declin. Există mai mulți factori care au contribuit la acest rezultat, însă creșterea numărului de resurse digitale și stilul de viață actual au avut cel mai mare impact asupra bibliotecilor.



Figură 1.1 Evoluția numărului de biblioteci în România (1995-2023)

Sursă: Institutul Național de Statistică, 2023

Tabel 1: Cititori activi la biblioteci, grupați pe județe și localități

Rezultatele cautarii - Cititori activi la biblioteci pe județe și localități			
Județe	Localități	Perioade	
		Anul 2013	Anul 2023
		UM: Numar persoane	
		Numar persoane	Numar persoane
<b>TOTAL</b>	<b>TOTAL</b>	4004366	2419179

Sursă: Institutul Național de Statistică, 2025

Dacă în trecut bibliotecile reprezentau inima unei comunități, un loc liniștit, potrivit atât pentru cercetare, cât și pentru a petrece timp alături de prieteni, în momentul de față sunt privite ca o ultimă opțiune atunci când vine vorba de alegerea unui loc de studiu. În comparație cu telefoanele mobile, social media și accesul rapid la informații, bibliotecile par concepte demodate. Foarte des, oamenii asociază aceste instituții de informare și lectură cu procese de împrumut-retur care durează mult, cu sisteme vechi care se blochează la fiecare click și bibliotecari care sunt depășiți de tehnologie, însă aceste circumstanțe nu sunt mereu valabile. Procesul de digitalizare a luat amploare în ultimii ani și continuă să se extindă în diverse domenii, inclusiv cel al bibliotecilor.

Pentru a stimula creșterea numărului de persoane care aleg să acceseze informații prin intermediul unei biblioteci, este necesar un proces de conversie al resurselor fizice cât și al serviciilor operaționale în formate digitale. Astfel, o bibliotecă publică în curs de digitalizare trebuie să investească în crearea de cataloage online și servicii de împrumut online sau e-lending, crescând astfel nivelul de confort al utilizatorului.

Acest proiect dorește să exploreze procesul de digitalizare al unei biblioteci prin prisma creării unei platforme interactive pentru un centru public de documentare și implementării de funcționalități specifice. Studysphere, în esență, reprezintă un plan de creștere al numărului de cetățeni care aleg să interacționeze cu o bibliotecă pentru a avea acces la informații. Informatizarea va viza diverse activități precum adăugarea informațiilor despre utilizatori și cărți într-o bază de date, crearea unui catalog care să reflecte colecția gestionată de bibliotecă, posibilitatea de a căuta cărți după titlu sau autor, filtrarea acestora în funcție de categorie, gestionarea împrumuturilor și a returnărilor precum și oferirea de resurse digitale dacă acestea sunt disponibile. Cu toate acestea, un mare impediment pe care îl au cititorii atunci când vine vorba de împrumutul unei cărți fizice îl constituie lipsa timpului. Astfel, o călătorie de doar 15 minute contribuie la decizia finală de a nu merge la bibliotecă. Am considerat că soluția optimă pentru această problemă este integrarea unei opțiuni de livrare la domiciliu, așadar, atunci când o persoană ajunge la pasul de finalizare al unei comenzi, ea poate opta fie pentru ridicare personală de la sediul fizic al instituției, fie pentru livrare la domiciliu contra cost. În acest fel, plăcerea de a răsfoi paginile unei cărți este prezentă, cititorul nu este nevoit să își modifice programul, iar profitul obținut poate fi folosit pentru menținerea și dezvoltarea colecției de cărți.

### **1.3. Comparația cu alte produse/aplicații software existente în domeniul abordat**

Digitalizarea unei biblioteci implică o investiție uriașă, atât din punct de vedere al timpului, cât și al costului. Așa cum se poate observa în figura următoare (Kumar & Kumar, 2020), un astfel de proces implică numeroase probleme care trebuie abordate în timpul dezvoltării programului software. Din acest punct de vedere, profesioniștii în domeniu trebuie să analizeze riguros și să

identifice care sunt obiectivele principale atunci când vine vorba de crearea unei platforme pentru o bibliotecă. De asemenea, trebuie să identifice toți pașii necesari pentru a finaliza cu succes acest proiect, astfel trebuie să răspundă la întrebări precum „Care este cel mai potrivit software?”, „Ce resurse trebuie digitizate?” sau „Cum se poate securiza conținutul?”.



*Figură 2.2 Digital Library Challenges and Opportunities: An Overview*

*Sursă: Kumar & Kumar, 2020*

Una dintre cele mai mari provocări în crearea unui website pentru o bibliotecă este accesibilitatea. De cele mai multe ori, utilizatorii se confruntă cu obstacole în timp ce vizitează portalul online precum supraîncărcarea cu informații, o interfață cu un design complicat sau imposibilitatea de a accesa resursele digitale.

Navigând pe mai multe website-uri ale unor biblioteci din România am observat un element comun: prezența unui număr mare de casete de căutare. Deși inițial nu pare a fi un defect de design, după câteva minute de interacțiune se dovedește a fi un impediment în căutare. O multitudine de câmpuri de căutare, filtrare și sortare nu numai că poate fi copleșitoare, în special pentru utilizatorii care nu sunt familiarizați cu astfel de sisteme, ci și reduc calitatea estetică a site-ului. Într-un secol al vitezei se dorește obținerea cât mai rapidă a unui rezultat și parcurgerea a cât mai puțini pași. Un exemplu concludent poate fi observat în figura următoare, reprezentând interfața catalogului online pentru Biblioteca Națională a României, câteva din probleme fiind afișarea simultană a unui

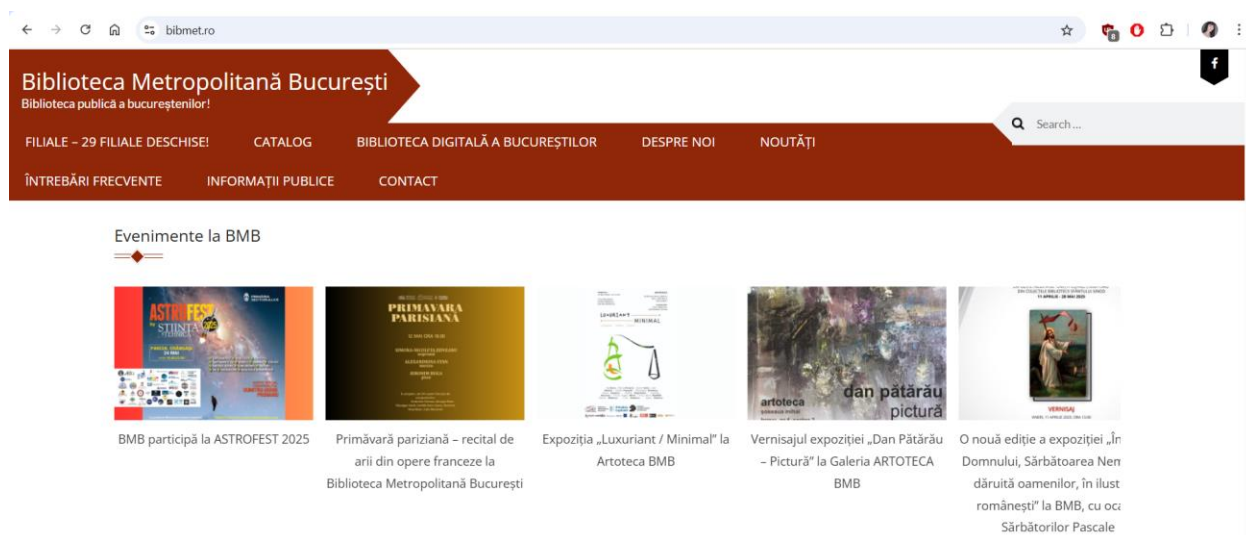
număr mare de filtre, fără explicații suplimentare sau ierarhizare, cât și designul vizual aglomerat care generează oboseală vizuală pentru vizitatorii site-ului.

Figură 3.3 Interfața catalogului digital pentru Biblioteca Națională a României

Sursă: Screenshot realizat pe website-ul : <https://aleph.bibnat.ro/F?RN=108383291>

Când vine vorba de design web, unul dintre principiile fundamentale este legea lui Hick, numită și legea lui Hick-Hyman. Conform acesteia, creșterea numărului de opțiuni disponibile determină creșterea timpului și dificultății de a lua o decizie. Fenomenele de supraîncărcare cognitivă (cognitive overload) apar atunci când un individ primește mai multe informații decât poate procesa, astfel efortul mental este unul mult mai mare și poate conduce la sentimente de frustrare și pierderea interesului din partea utilizatorului. Pentru a evita astfel de scenarii, interfața unui website trebuie să fie cât mai simplificată, iar în ceea ce privește filtrele de căutare, se pot aplica metode precum filtrare progresivă, unde se adaugă treptat diferite criterii de sortare și afișare.

Cel de-al doilea site de referință pentru elaborarea acestui proiect este <https://bibmet.ro/>, dedicat pentru Biblioteca Metropolitană din București. După cum se poate vedea în imaginea următoare, putem observa un design greu de urmărit în ceea ce privește secțiunile platformei. Deși acestea sunt evidențiate printr-o culoare îndrăzneță, menită să le scoată în evidență, forma finală denotă dezordine și aglomerare de elemente. De asemenea, se poate observa lipsa unui element central pe pagina principală care să atragă atenția utilizatorului. În schimb, există foarte multă informație secundară, iar valoarea interfeței este redusă drastic.



Figură 4.4 Interfața principală a site-ului pentru Biblioteca Metropolitană din București

Sursă: Screenshot realizat pe website-ul : <https://bibmet.ro/>

Următorul pas a fost să analizez și să testez funcționalitatea de căutare, așadar am introdus în bara de căutare prezentă pe pagina principală numele „Mihai Eminescu”, având intenția de a găsi titluri de cărți scrise de acest autor. În schimb, sistemul a returnat o listă de anunțuri, fără niciun fel de referință către o listă de volume disponibile scrise de poet. O astfel de funcționalitate poate cauza frustrare, confuzie cât și o pierdere a încrederii, mai ales pentru utilizatorii care nu au experiență cu mediul digital.

Din analiza celor două platforme a rezultat o serie de probleme și posibile riscuri. Fie că e vorba de puncte slabe în ceea ce privește interfața site-ului sau un motor de căutare neoptimizat, rezultatul este același, anume reducerea accesibilității și gradului de implicare al utilizatorilor. Cu toate acestea, existența aspectelor pozitive nu poate fi ignorată. Ambele site-uri oferă resurse esențiale și se preocupă cu adaptarea la nevoile publicului prin postarea de informații despre evenimente și servicii. În concluzie, scopul acestei lucrări este de a analiza opțiunile deja existente pe piață, de a identifica nevoile nesatisfăcute de acestea, și de a încerca implementarea lor într-un mod cât mai eficient. Astfel, deși există loc pentru îmbunătățire, platformele de mai sus pavează drumul către destinația finală, aceea a accesului liber la resursele tuturor bibliotecilor din România pentru persoane de toate vârstele, indiferent de gradul de experiență avut cu tehnologia.

## 2. Analiza și proiectarea aplicației informatice

### 2.1. Specificarea cerințelor aplicației informatice

Această lucrare are ca scop crearea unui website pentru o librărie publică denumită Studysphere și prezentarea pașilor necesari implementării. Aplicația va permite gestionarea colecției de volume fizice deținute de Studysphere într-o formă digitală, ușor de accesat și navigat, automatizând, de asemenea, procese precum împrumutul, rezervarea și consultarea cărților. Design-ul platformei va fi unul modern dar cu funcționalități ușor de utilizat, atât de către publicul general, cât și de către administratorul bibliotecii.

#### 2.1.1 Funcționalități pentru utilizatori

În ceea ce privește modul de organizare al unei biblioteci digitale, un actor principal îl reprezintă utilizatorul general, un cititor din publicul larg. Astfel, instituția are ca scop oferirea de acces la colecțiile de cărți și publicații, fie că sunt fizice sau digitale. Cu toate acestea, pentru a duce la împlinirea acestei sarcini, trebuie formulat un plan cât mai adecvat, care să ia în considerație toate variabilele care contribuie la experiența unui beneficiar. Printre acestea se numără designul aplicației, implementarea acțiunilor care au loc într-o bibliotecă fizică în formă digitală cât și asigurarea securității și confidențialității.

- **Crearea unui cont de utilizator și gestionarea acestuia**

Unul dintre aspectele principale ale tranzacțiilor dintr-o bibliotecă îl constituie obținerea de informații despre beneficiari. Din motive administrative, organizatorice dar și legale, bibliotecile au nevoie de date despre utilizatorii care doresc să împrumute cărți. Aceste date contribuie la evitarea fraudelor, astfel, odată ce o identitate a fost creată în sistem pentru o persoană, scade riscul ca aceasta să nu mai returneze cărți.

Crearea unui cont de utilizator a fost dintre funcționalitățile de bază în procesul de proiectare al website-ului pentru Studysphere pentru mai multe motive. Posibilitatea creării unui cont online oferă utilizatorilor posibilitatea de a stoca informații personale precum liste de dorințe sau istoricul de comenzi. De asemenea, oferirea datelor cu caracter personal contribuie la menținerea responsabilității legale și a securității, atât pentru beneficiar cât și pentru personalul bibliotecii, dar asigură și o legătură unică bazată pe încredere între public și instituție.

Interfața aplicației este una intuitivă, așadar pune la dispoziție o secțiune dedicată conectării în header-ul paginii principale. Procesul de înregistrare este unul simplu și presupune completarea unui formular cu un număr scăzut de câmpuri obligatorii. Pentru a crea un profil digital, persoana care folosește site-ul trebuie să furnizeze numele complet, o adresă de e-mail validă, o parolă și adresa de domiciliu, care va fi folosită pentru comenzile la domiciliu. După

înregistrare, utilizatorul va putea beneficia de funcționalități precum rezervarea de cărți, livrarea acestora la domiciliu, adăugarea lor la secțiunea de favorite etc. De asemenea, în contul principal acesta va putea accesa istoricul de comenzi și coșul de cumpărături.

- **Catalogul digital al bibliotecii**

Cataloagele digitale constituie colecții de produse, ușor accesibile, structurate și ordonate în baza a diverse criterii. Cel mai des, acestea sunt folosite în e-commerce, sau comerț electronic, însă au luat amploare în mai multe arii, fiind adesea încorporate și în procesul de digitalizare al bibliotecilor. Existența unui astfel de catalog aduce numeroase beneficii. Printre acestea se numără căutarea eficientă și rapidă, o mai bună gestionare a colecției de volume, posibilitatea de a rezerva direct din catalogul online, și nu în ultimul rând, oportunitatea de a atrage mai mulți utilizatori.

Aplicația oferă vizitatorilor website-ului Studysphere un catalog online completat cu toate volumele disponibile în colecția instituției. Fiecare carte afișată în listă este prezentată sub forma unui chenar dreptunghiular, care conține informațiile principale precum titlul și autorul volumului. În interiorul chenarului, există de asemenea un buton care poate face posibilă adăugarea cărții în coșul de cumpărături. Imediat ce utilizatorul dă click pe acest chenar, va fi redirecționat către o pagină nouă care va afișa toate informațiile referitoare la cartea respectivă, precum o scurtă descriere, o galerie cu imagini (coperta anterioară, coperta posterioară, și eventual, câteva pagini), editura, genul literar, numărul de exemplare disponibile pentru împrumut, și, dacă se află în baza de date, versiunea PDF. De asemenea, utilizatorul poate opta pentru căutarea cu ajutorul filtrelor de categorie sau poate introduce în bara de căutare titlul, autorul, sau categoria din care face parte cartea dorită.

- **Rezervarea cărților**

Dacă în perioada incipientă a bibliotecilor, acestea aveau scopul doar de a conserva și proteja colecțiile de cărți pe care le aveau în posesie, cu timpul, conceptul de a oferi acces și publicului larg a luat amploare, consolidându-se în prima jumătate a secolului al 18-lea. Deși modelul inițial de împrumut era pe bază de subscripție, unde indivizii aveau obligația de a plăti o taxă pentru a deveni membri și de a avea acces la publicații, acesta a pavat drumul către bibliotecile publice, unde accesul este gratuit și garantat pentru orice persoană, indiferent de vârstă, ocupație, gen sau religie.

În cadrul site-ului Studysphere, utilizatorii au opțiunea de a rezerva cărți doar în cazul în care acestea sunt disponibile în format fizic. După selectarea cărților dorite, acesta poate opta pentru diverse modalități de obținere, fie prin ridicarea personală de la sediul instituției, fie prin livrare la domiciliu contra-cost, către adresa selectată de client.

Sistemul înregistrează datele referitoare la comandă, le adaugă în baza de date și emite un mesaj de confirmare. În cazul ridicării personale, cartea va fi reținută pe numele beneficiarului

pentru o perioada de timp limitată, anume 3 zile lucrătoare, iar pentru livrările la domiciliu, se generează o dată estimativă, de aproximativ 3-5 zile lucrătoare.

### **2.1.2 Funcționalități pentru administrator**

Cele mai multe website-uri au nevoie de roluri de administrator pentru un management de informații și resurse cât mai eficient. Aceste roluri furnizează permisiuni necesare pentru acțiuni precum adăugarea de informații în baza de date a sistemului, modificarea sau ștergerea acestora, gestionarea datelor despre utilizatori și produse, și asigurarea securității.

- **Panou de administrare pentru utilizatori**

Panourile de administrare sunt elemente cruciale în cadrul bibliotecilor digitale, necesare pentru un management eficient de resurse, informații și tranzacții. Existența unei secțiuni separate dedicată utilizatorilor permite gestionarea conturilor celor ce folosesc sistemul respectiv, și este esențială în cadrul aplicațiilor web care se bazează pe interacțiunea dintre vizitatori și servicii.

Pentru această aplicație, administratorul printre atribuții controlul accesului și al permisiunilor. Acesta se ocupă de atribuirea de roluri, având ca opțiuni „General” sau „Admin”. Un utilizator general nu poate accesa informații despre alți utilizatori, despre informații cu privire la conturile acestora, precum istoric de comenzi sau cărți favorite. Un admin are permisiunea de a accesa astfel de informații, și are posibilitatea de a le edita.

- **Panou de administrare pentru produse/cărți**

Scopul unei secțiuni de produse în cadrul unui panou administrativ este de a permite o gestionare completă și eficientă a colecției de produse disponibile. Aplicația web pentru biblioteca Studysphere pune la dispoziția administratorului o secțiune dedicată managementului de cărți. Printre funcționalități se numără adăugarea de produse noi prin intermediul unei ferestre modale, unde bibliotecarul va completa diverse câmpuri cu informație utilă. De asemenea, datele deja existente în sistem pot fi modificate, incluzând numărul de cărți disponibile pentru împrumut.

### **2.1.3 Cerințe privind datele de intrare și ieșire**

În programare, datele de intrare reprezintă informațiile pe care un program le primește din mediul exterior, având ca scop procesarea acestora, rezultatul fiind datele de ieșire.

Datele de intrare pot varia, având diverse surse: input-ul de la utilizatori, anume informații introduse cu ajutorul tastaturii, mouse-ului sau a altor dispozitive, fișiere stocate pe calculator, conexiuni de rețea sau senzori.



Datele de ieșire sunt informații pe care un program le produce după procesarea datelor de intrare. Acestea pot fi afișate pe ecran, printate sau trimise către alte dispozitive. Câteva exemple sunt: afișarea rezultatelor sub formă de text în consolă sau integrarea lor într-o formă grafică în interfață, extragerea datelor procesate în fișiere, trimiterea unor semnale către alte dispozitive sau sisteme, crearea unor grafice, diagrame etc.

- **Date de intrare**

În ceea ce privește aplicația web Studysphere, datele de intrare se împart în două categorii:

a) Date introduse de utilizator

- Date cu caracter personal, introduse de utilizator la tastatură, în formularul de înregistrare: nume complet, e-mail, parolă, adresă;
- Date care au legătură cu procesul de căutare al cărților: titlu, autor, categorie/gen literar;
- Date care au legătură cu procesul de rezervare al cărților: id-ul cărților împrumutate, id-ul utilizatorului care împrumută cărțile, metoda de livrare;

b) Date provenite din surse externe

- Bază de date gestionată dinamic care conține informații în timp real cu privire la colecția de cărți, stocul disponibil, utilizatorii cu cont digital;

- **Date de ieșire**

Website-ul generează următoarele date de ieșire:

- Notificări toastify se eroare sau succes cu privire la efectuarea acțiunilor specifice, spre exemplu: „Înregistrare efectuată cu succes!” sau „Înregistrare eșuată! Parola nu se potrivește!”;
- Informații despre fiecare carte: autor, titlu, imagini specifice, descriere, disponibilitate, editură, variantă PDF(dacă există);
- Rezultate pentru căutarea cărților în catalogul digital;
- Confirmări în legătură cu procesul de rezervare;

## 2.2. Analiza aplicației

Fiecare website are la bază conceptul de sistem sau sistem informatic. Un sistem reprezintă un ansamblu de componente care sunt intercorelate funcțional și conlucrează pentru a realiza obiective prestabilite. Folosindu-se aceeași logică, un sistem informatic combină elemente

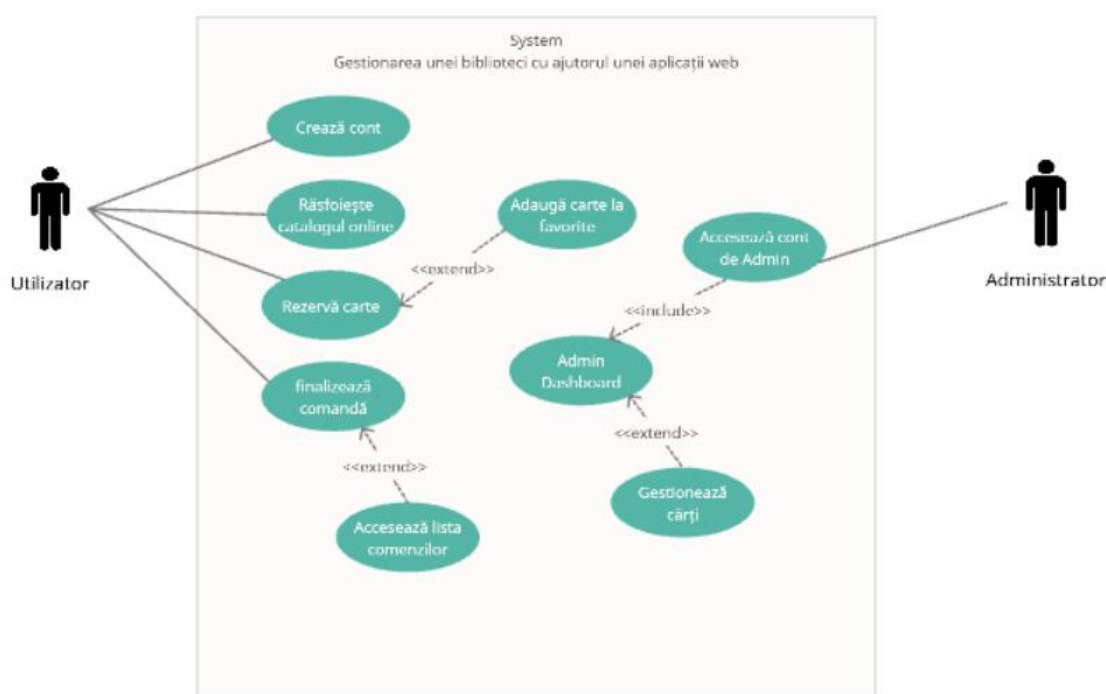
software, hardware cât și rețele de comunicare, având ca scop colectarea de date utile și realizarea și gestionarea unor operațiuni specifice organizației de care aparține.

## 2.2.1 Diagrama cazurilor de utilizare

Pentru proiectarea cât mai eficientă și adecvată a unui sistem informatic, este necesară elaborarea unei diagrame de cazuri de utilizare. Aceasta are rolul de a prezenta funcționalitățile unui website într-o formă grafică și ușor de înțeles. Diagrama este folosită pentru a exprima atât așteptările clienților cât și ale beneficiarilor.

Având scopul de a conferi o imagine cât mai clară a proiectului, diagramele de cazuri de utilizare sau „use case” sunt formate din următoarele elemente: actorii, care au scopul de a răspunde la întrebarea „cine interacționează cu sistemul?”, cazurile de utilizare, menite să exemplifice ce acțiuni pot efectua actorii, și sistemul, care reprezintă ce descrie întreg proiectul.

După cum se poate identifica în cadrul figurii 2.1, în cazul aplicației web pentru gestiunea unei biblioteci există 2 actori principali care interacționează cu sistemul, anume utilizatorii și administratorul. Aceștia efectuează diferite acțiuni, prezentate în diagrama de mai jos sub forma unor cazuri de utilizare. Spre exemplu utilizatorul își poate crea un cont, poate răsfoi catalogul online, poate rezerva sau adăuga la secțiunea de „favorite” una sau mai multe cărți iar în cele din urmă poate efectua o comandă care să finalizeze procesul de împrumut. Administratorul are ca acțiuni principale gestionarea platformei și a colecției de cărți prin intermediul dashboard-ului.



Figură 2.1 - Diagrama cazurilor de utilizare a unei platforme online de gestionare a unei biblioteci

Sursă: Generată prin intermediul website-ului Creatly

### 2.2.2 Diagrama de clase

Structura statică a sistemului poate fi proiectată cu ajutorul unei diagrame de clase. Acest tip de diagramă este un element fundamental al UML(Unified Modeling Language) care are scopul de a evidenția atributele, operațiile și asocierile unui sistem. În esență, o diagramă de clase reprezintă o schiță a designului orientat pe obiecte, fiind utilizată pentru a înțelege modul în care clasele relaționează între ele. Acestea sunt cruciale în documentarea structurilor software sau altor sisteme.

Componentele cheie în realizarea unei astfel de diagrame sunt, implicit, clasele, atributele, operațiile și relațiile dintre clase. O clasă reprezintă un schelet pentru formarea unui obiect. Obiectele și clasele au o relație de dependență bidirecțională, definindu-se reciproc. Nu putem vorbi de un concept fără să îl aducem în discuție și pe celălalt. În programarea orientată obiect (POO), clasele reprezintă nucleul central, fiind folosite mai apoi pentru crearea obiectelor. Astfel, o clasă descrie un obiect, dar nu este unul.

În cazul diagramelor UML, o clasă încorporează stări, denumite atribute, și comportamente, denumite operații. Un atribut reprezintă o proprietate a unei clase și descrie caracteristici ale obiectelor care aparțin de clasa respectivă. În general, acestea sunt scrise sub numele clasei și sunt definite prin nume, tip(exemplu:Integer, String etc.) și vizibilitate. Operațiile, numite și metode, constituie acțiuni pe care instanțele unei clase le efectuează. Ele sunt cruciale pentru modelarea funcționalităților unui sistem, și formează o imagine de ansamblu asupra comportamentului clasei.



Figură 2.2 - Diagrama de clase a unei platforme online de gestionare a unei biblioteci

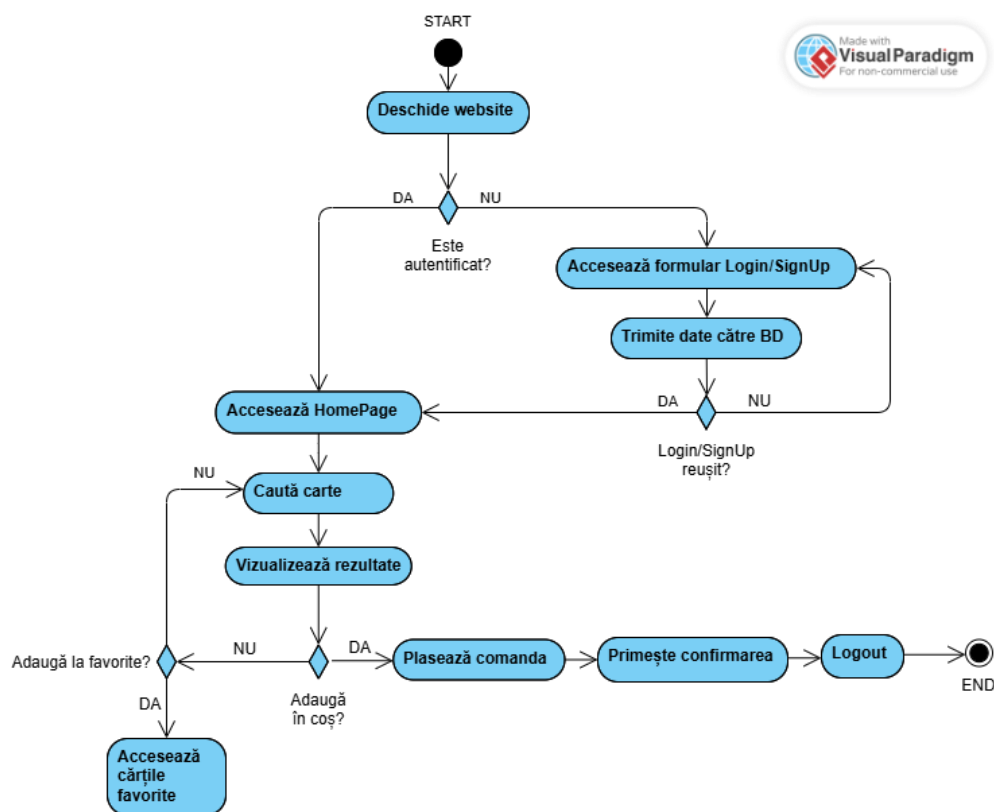
Sursă: Generată prin intermediul website-ului Visual Paradigm

După cum se poate observa în diagrama din figura 2.2, se disting 4 clase principale în cadrul aplicației web Studysphere, și anume: User, Admin, Book și Order. Între acestea, există mai multe tipuri de asocieri:

- Relație de generalizare între clasele Admin—User: un admin este un tip special de utilizator
- Relație de asociere cu multiplicitate între clasele User↔Book(un utilizator poate rezerva mai multe cărți), Order↔Book(o comandă poate conține mai multe cărți) și Order↔User(un utilizator poate plasa mai multe comenzi);
- Relație de agregare între clasele Order și Book: Order agregă Book, așadar cărțile dintr-o comandă nu dispar odată cu comanda, ci pot fi folosite și în alte comenzi;

### 2.2.3 Diagrama de activitate

Pentru a crea o reprezentare vizuală a fluxului de activități dintr-un punct de plecare până într-un punct de terminare în cadrul unui sistem se utilizează diagrama de activitate. Aceasta descrie secvența de acțiuni sau pași necesari finalizării unei sarcini, evidențiind ordinea execuției cât și punctele de decizie care determină fluxul.



Figură 2.3 - Diagrama de activitatea unei platforme online de gestionare a unei biblioteci

Sursă: Generată prin intermediul website-ului Visual Paradigm

## 2.3. Proiectarea aplicației

Proiectarea unei aplicații web este etapa în care programatorul analizează cum vor fi realizate și implementate funcționalitățile necesare identificate în planificare. Acest stadiu este crucial pentru obținerea unui produs final care să îndeplinească toate criteriile impuse, și implică definirea structurii, a componentelor și a arhitecturii. Se referă, de asemenea, la crearea unui plan de navigare al site-ului, cât și la analiza interacțiunii dintre diverse componente. Întregul proces are scopul de a crea o fundație solidă pentru o experiență cât mai plăcută pentru utilizatori.

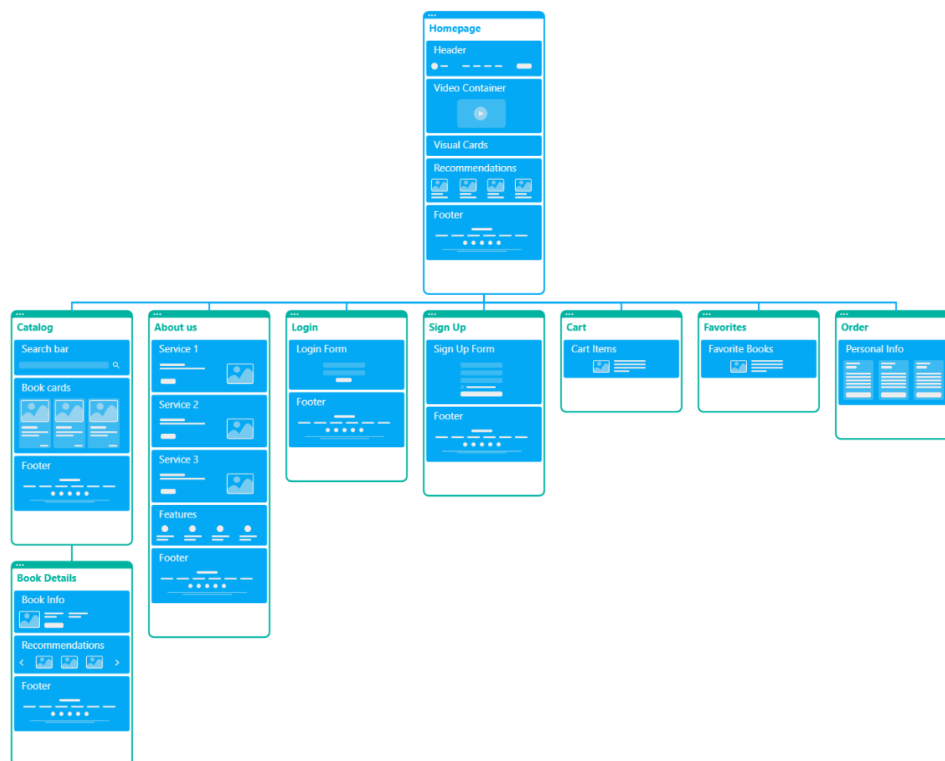
### 2.3.1 Stabilirea arhitecturii sistemului

În medie, utilizatorii petrec doar 54 de secunde pe un website. Astfel, în mai puțin de 1 minut, este important ca aceștia să găsească un punct de interes relevant pentru a-și concentra atenția asupra lui. Arhitectura aplicației web joacă un rol esențial în această privință și are capacitatea de a influența în mod decisiv eșecul sau succesul unui site.

Structura de bază a unui website constituie arhitectura acestuia, și include modul în care paginile grupate, organizate și interconectate. Câteva elemente de bază sunt: ierarhia conținutului, navigarea, existența link-urilor interne și structura URL(Uniform Resource Locator).

Ierarhia unei aplicații web se referă la structura și organizarea paginilor, vizualizate în general sub formă de arbore care are ca rădăcină pagina inițială, numită și homepage. În cazul site-ului creat pentru Studysphere (fig. 2.4), ierarhia urmează regula de bază, iar descendenții sunt reprezentați de următoarele pagini:

- „Login” și „Sign Up” pentru conectare sau creare cont;
- „Despre noi” pentru detalii despre instituție și serviciile oferite;
- „Catalog” pentru lista cu toate cărțile disponibile pentru împrumut; Această pagină are la rândul său un alt descendent, „Detalii carte”, care prezintă informații suplimentare despre cartea accesată;
- „Coș de cumpărături”, unde pot fi vizualizate sub formă de listă toate cărțile ce urmează a fi împrumutate;
- „Favorite”, unde pot fi vizualizate sub formă de listă toate cărțile ce vor fi împrumutate în viitor, în funcție de disponibilitate;
- „Comandă”, unde este finalizat procesul de împrumut al cărților selectate;



Generated with app.writemaps.com

Figură 2.4 - Ierarhia unei platforme online de gestionare a unei biblioteci

Sursă: Generată prin intermediul website-ului WriteMaps

### 2.3.2 Alegerea unui model de date

Crearea unui model de date este procesul de reprezentare vizuală al datelor și a relațiilor dintre acestea, creând astfel o schiță pentru structurarea și stocarea informațiilor în interiorul unui sistem. Scopul final este de a ilustra tipurile de date utilizate în cadrul unei aplicații, modalitățile în care pot fi grupate și organizate și relațiile pe care le formează.

Există numeroase avantaje pentru proiectarea unui model de date anterior implementării unui website. Printre acestea se numără reducerea timpului de dezvoltare, îmbunătățirea acurateții sistemului de date prin crearea unor structuri și relații solide și optimizarea performanței aplicației.

În funcție de nivelul de abstracție, modelele de date se clasifică în trei categorii: modele conceptuale, modele logice și modele fizice. Pentru o idee cât mai generală asupra sistemului, sunt folosite modelele conceptuale. Acestea includ clasele principale alături de câteva atribute importante, constrângerile și relațiile dintre acestea. Cu un nivel mai mic de abstractizare, modelele logice conțin un număr mai mare de detalii. Acestea cuprind totalitatea atributelor unei clase, cu constrângeri pentru fiecare acolo unde este cazul, și relații bine descrise între componente. Spre deosebire de aceste două categorii, modelul fizic oferă un design final care urmează a fi implementat sub forma unei baze de date relaționale. Acesta include tabele complete, atribute

definite prin tip și constrângeri, și relații bine definite și menținute prin caracteristici precum „Primary Key” sau „Foreign Key”.

O bază de date relațională organizează informațiile pe rânduri (care conțin înregistrări) și coloane (care definesc tipuri), formând astfel o tabelă. Într-o astfel de bază de date există mai multe tabele care sunt legate între ele prin chei primare și chei străine. O înregistrare poate fi considerată cheie primară dacă este un identificator unic, astfel nu pot exista mai multe rânduri cu aceeași valoare, ci doar unul. Pe de altă parte, cheile străine asigură conexiunea dintre mai multe tabele ale bazei de date.

În cazul aplicației web Studysphere, baza de date are scopul de a face posibile activitățile unei biblioteci. Astfel, tabelele principale sunt:

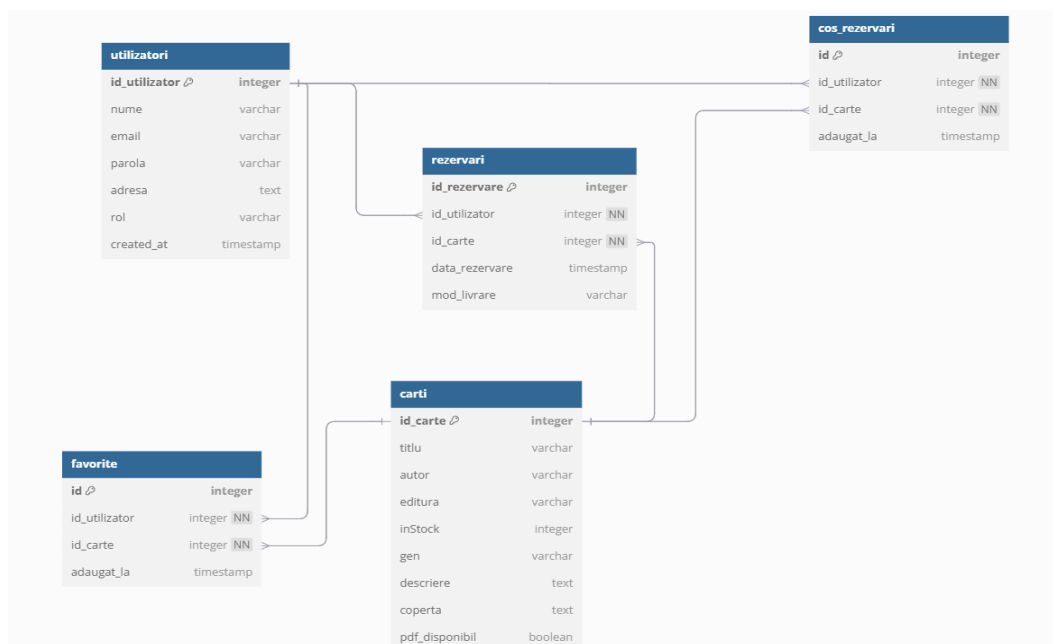
- Utilizatori - această tabelă conține informații despre utilizatori: id\_utilizator, care este unic, generat automat la fiecare înregistrare, fiind astfel cheia primară, nume, email și parolă completate în formularul de înregistrare, adresă, rol (GENERAL sau ADMIN) și data la care a fost creat contul.
- Cărți - această tabelă conține informații despre cărțile gestionate de bibliotecă: id\_carte, unic pentru fiecare înregistrare, astfel cheie primară, titlu, autor, editură, gen, descriere, copertă, pdf\_disponibil, număr de exemplare disponibile.
- Rezervări - această tabelă conține informații despre fiecare comandă efectuată de utilizatori: id\_rezervare, unic și cheie primară, id-ul utilizatorului care a finalizat rezervarea, o listă cu id-urile cărților care au fost rezervate, data rezervării și modul de livrare.
- Coș\_rezervări - conține cărțile adăugate de utilizator în coșul de cumpărături: id unic, id-ul utilizatorului conectat, id-ul cărții adăugate.
- Favorite - conține cărțile adăugate de utilizator la secțiunea de favorite: id unic, id-ul utilizatorului conectat, id-ul cărții adăugate.

În ceea ce privește relațiile dintre tabele, există diferite tipuri:

- Many-to-One (M:1) între rezervări și utilizatori. Explicația este următoarea: un utilizator poate finaliza mai multe rezervări/comenzi, dar o rezervare aparține unui singur utilizator.
- Many-to-Many (N:M) între utilizatori și cărți. Pentru implementarea acestei relații, este nevoie de tabelele intermediare coș\_rezervări și favorite. Relația funcționează astfel: un utilizator are posibilitatea de a adăuga mai multe cărți în coș sau la favorite, iar o carte poate fi adăugată în coș sau la favorite de mai mulți utilizatori.

În contextul bazei de date pentru Studysphere, fiecare tabelă are o cheie primară: id\_utilizator (tabela Utilizatori), id\_carte (tabela Cărți), id\_rezervare (tabela Rezervări). Aceste chei primare sunt folosite ca referințe în diverse tabele, prin intermediul cheilor străine. Câteva exemple de chei străine din baza de date ar fi: id\_utilizator din tabela Rezervări, care face legătura cu tabela Utilizatori, id\_carte din tabela Rezervări, care face legătura cu tabela Cărți.

Conform acestor proprietăți și constrângeri, schema bazei de date pentru Studysphere este prezentată în figura 2.5:



Figură 2.5 - Schema bazei de date relaționale unei platforme online de gestionare a unei biblioteci

Sursă: Generată prin intermediul website-ului DB Diagram

Baza de date relațională de mai sus a constituit fundația pentru manipularea eficientă a datelor în cadrul acestei aplicații, fiind un instrument util pentru implementarea tuturor funcționalităților.

### 2.3.3 Modelarea diagramei de interacțiune

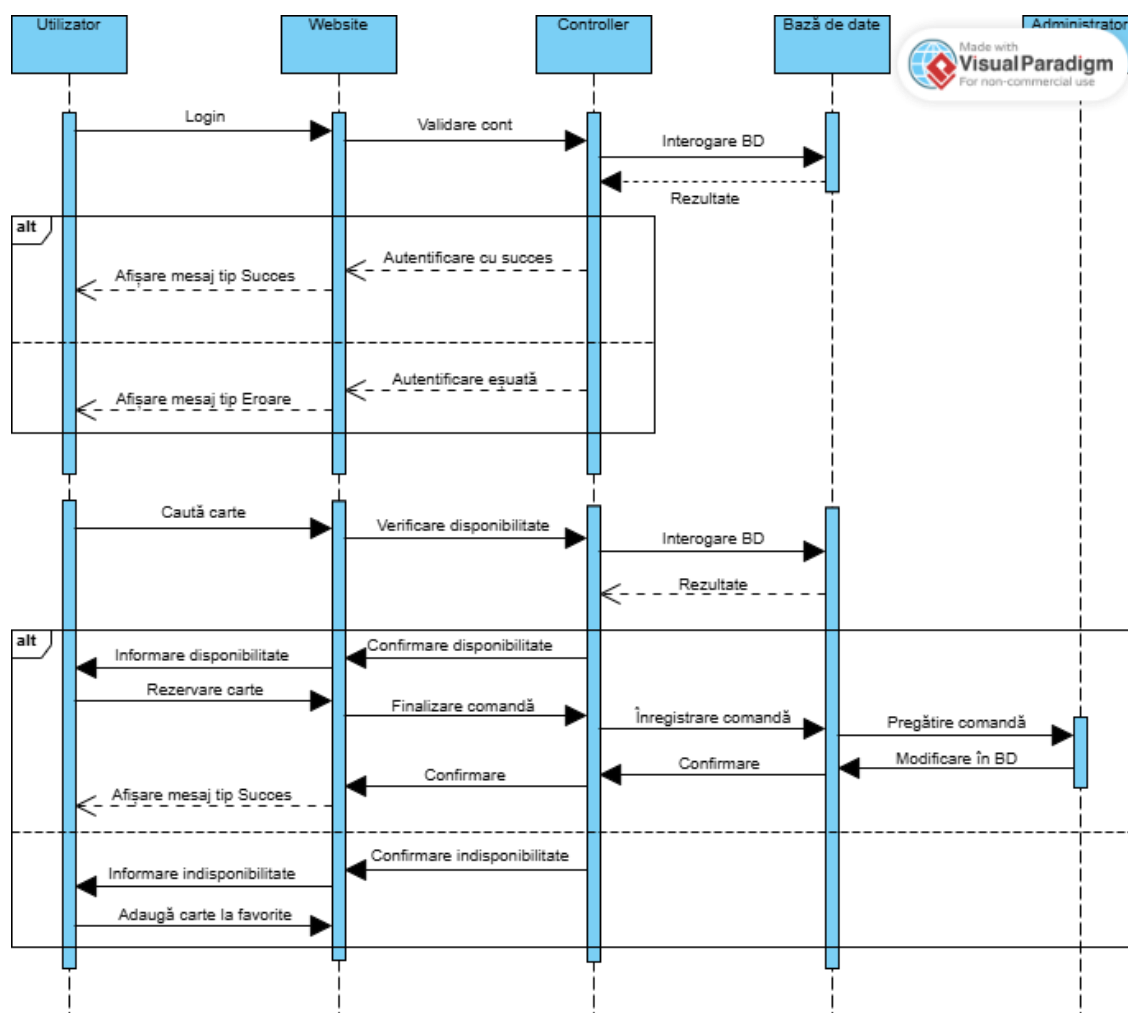
Diagramele de interacțiune reprezintă o soluție în ceea ce privește modelarea dinamică a comportamentului unui sistem. Acestea capturează interacțiunea dintre obiectele prezente într-un sistem, concentrându-se atât pe procesele care există în mod simultan, cât și pe mesajele schimbate între ele cu scopul de a îndeplini o funcție. O diagramă de interacțiune este întotdeauna bidimensională, axa orizontală ilustrând obiectele și actorii care sunt incluși în interacțiune, iar axa verticală indicând ordinea cronologică a secvențelor unor evenimente.

Un caz de utilizare cât mai descriptiv pentru funcționalitățile website-ului Studysphere îl reprezintă acțiunea de a rezerva o carte. Așadar, participanții acestei interacțiuni sunt: utilizatorul care dorește să împrumute cărți, interfața aplicației, componenta logică a sistemului denumită controller, baza de date unde sunt stocate toate datele necesare, și nu în ultimul rând administratorul bibliotecii. Toți acești actori formează linii de viață (lifelines), construite, în



general, prin forme dreptunghiulare care conțin nume de rol, la care se adaugă o linie verticală punctată, asociată cu durata de viață a obiectului aferent. Pentru a vizualiza fereastra de timp în care unul dintre actori execută o acțiune anume, se folosesc dreptunghiuri subțiri plasate de-a lungul liniei de viață, iar pentru a schița o cronologie a evenimentelor se folosesc mesaje, afișate ca săgeți.

Aplicând aceste concepte, diagrama realizată în figura 2.6 oferă cu acuratețe o perspectivă cât mai comprehensibilă asupra secvențelor de evenimente și a rolului fiecărui actor implicat.



Figură 2.6 - Diagrama de interacțiune pentru o platformă online de gestionare a unei biblioteci

Sursă: Generată prin intermediul website-ului Visual Paradigm

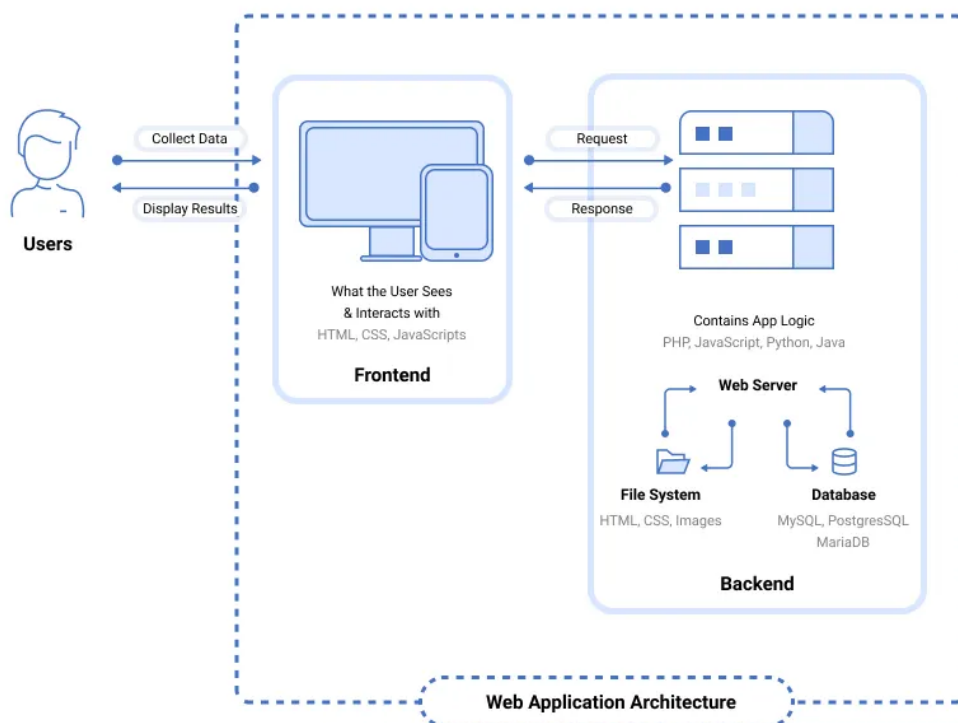
## 3. Implementare aplicației informatice

### 3.1. Prezentarea tehnologiilor

Un site web este un instrument important, utilizat în mod constant de majoritatea oamenilor. Toate site-urile web concurează între ele prin design, funcționalitate sau eficiență. Modul de prezentare al informațiilor este, de asemenea, foarte important. În loc să caute pentru o perioadă îndelungată de timp pe internet, o persoană poate accesa un site pentru a primi toate informațiile necesare prin doar câteva click-uri. Rolul unei aplicații web de gestionare a unei biblioteci este de a pune o „față” tuturor acestor informații, astfel încât utilizatorul nu trebuie să aibă contact cu o mulțime de alte interfețe web. Mai mult decât atât, biblioteca electronică a unei universități poate conține diverse resurse educaționale, reviste electronice, teze de licență sau dizertație. Prin crearea unui portal web și încărcarea acestor resurse într-un server, putem crește nivelul de utilizare și funcționare la maxim. Utilizatorii pot economisi astfel timp și costuri prin localizarea diverselor informații necesare din vasta colecție de resurse disponibile de pe internet.

Istoria creării de website-uri este interesantă și începe cu World Wide Web, care a intrat în funcțiune în Noiembrie 1992. Primul website cu care au interacționat oamenii a fost 100% text-based, singura pată de culoare prezentă fiind albastrul atașat hypertext-ului. În perioada incipientă a designului web, singura modalitate de a adăuga structură și formă unei pagini web a fost folosirea unor tag-uri specifice HTML(Hypertext Markup Language), printre care amintim faimosul ‘<table>’. Deși crearea unui website a fost un eveniment inovator, existau numeroase limitări atât în ceea ce privește funcționalitatea, cât și aspectul. Cu toate acestea, cu trecerea timpului, procesul de creare web a evoluat semnificativ, astfel că simplele fișiere HTML au fost înlocuite cu instrumente complexe de dezvoltare mult mai rapide și eficiente.

Din punct de vedere al arhitecturii, orice website este format din 2 componente majore: partea de front-end cu care utilizatorii interacționează, și partea de backend, care este reprezentată de serviciile serverului care procesează cereri și furnizează informațiile necesare interfeței.



Figură 3.1 - Arhitectura unei aplicații web

Sursă: MindInventory / Rajput, M. (2021)

### 3.1.1 Front-End

Dezvoltarea front-end se concentrează pe interacțiunea utilizatorului cu website-ul, asigurând vizitatorilor navigarea fără probleme. Aceasta utilizează limbaje de programare precum JavaScript, HyperText Markup Language (HTML) și Cascading Style Sheets (CSS) pentru a crea designuri unice.

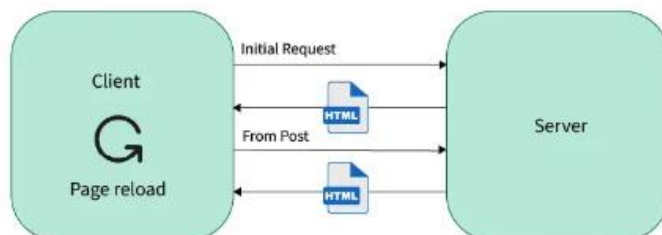
Având un număr mare de avantaje, am ales ca instrumentul de bază în crearea propriului website-ului să fie Vite + React. Printre beneficiile Vite + React se numără pornirea aproape instantanee a serverului, observarea în timp real a modificării componentelor și integrarea perfectă a fișierelor JSX, CSS sau TypeScript.

Interfața a fost construită în React, folosindu-mă de JavaScript XML(JSX). Fișierele cu acest tip de extensie permit integrarea limbajului HTML direct în sintaxa de cod JavaScript, oferind astfel o abordare mai intuitivă și făcând procesul de creare mult mai simplu. Câteva avantaje pe care le oferă JSX includ lizibilitate, performanță ridicată și un nivel mai mare de siguranță.

În ceea ce privește routing-ul, navigația a fost efectuată prin intermediul librăriei React Router. Aceasta permite crearea de rute dinamice sau imbricate, și oferă utilizatorului tranziții aproape instantanee între componente (Single Page Application).

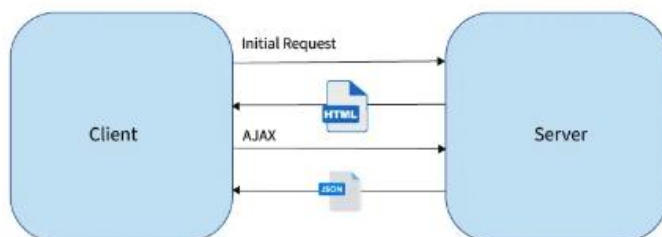
### Traditional Page Lifecycle

Once the user accesses the page and performs any kind of action on that page, the page gets reloaded with the changes that were done on the server-side.



### Single Page Lifecycle

Once the user accesses the page and performs any kind of action on that page, they get an almost-instant reaction from the page (think of Facebook, when you comment on someone's post.)



Figură 3.2 - What is Single Page Application?

Sursă: GeeksforGeeks (fără dată)

Pentru rutele dinamice am definit parametrii „:id” și „:categoryName” (fig. 3.3), ceea ce mi-a permis să reutilizez componente, să reduc codul redundant și să scurtez timpul de dezvoltare.

```
<Route path='book-category/:categoryName' element={<CategoryBooks />} />
<Route path='book/:id' element={<BookDetails />} />
```

Figură 3.3 - Fragment de cod pentru gestionarea rutelor dinamice

Sursă: cod sursă propriu

Pentru a integra o navigare ierarhică am folosit rutele imbricate, numite și nested routes (fig. 3.4). Acestea permit randarea mai multor componente în interiorul unei pagini principale. Implementarea lor se face prin definirea unor rute pentru componentele „copii” (child components) în interiorul rutei „părinte” (parent component) și utilizarea unui <Outlet> pentru randarea sub-componentei corespunzătoare.

```

<Route path='admin-panel' element={<AdminPanel />}>
  <Route path="all-users" element={<AllUsers />} />
  <Route path="all-products" element={<AllProducts />} />
</Route>

```

Figură 3.4 - Fragment de cod pentru gestionarea rutelor imbricate

Sursă: cod sursă propriu

În ceea ce privește stilizarea și experiența vizuală, am folosit Tailwind CSS. Acest framework simplifică dezvoltarea web prin furnizarea unei colecții de clase predefinite (fig. 3.5), care permit construirea unor designuri personalizate fără scrierea unei cantități mari de cod CSS.

Tailwind prezintă numeroase avantaje în comparație cu CSS clasic, precum eliminarea denumirilor complexe de clase, minimizarea codului scris și personalizarea ușoară a designurilor.

```

src > pages > Success.jsx > ...
1  import React from 'react'
2  import SUCCESSIMAGE from '../assets/success.gif'
3  import { Link } from 'react-router'
4
5  const Success = () => {
6    return (
7      <div className="bg-slate-200 w-full max-w-md mx-auto flex justify-center items-center flex-col p-4 m-2 rounded">
8        <img
9          src={SUCCESSIMAGE}
10         width={150}
11         height={150}
12       />
13       <p className="text-green-600 font-bold text-xl">Plată efectuată cu succes!</p>
14       <Link to={"/order"} className="p-2 px-3 mt-5 border-2 border-green-600 rounded font-semibold text-green-600 hover:bg-green-600 hover:text-white">Vezi comandă</Link>
15     </div>
16   )
17 }
18
19 export default Success

```

Figură 3.5 - Fragment de cod pentru stilizare, folosind Tailwind CSS

Sursă: cod sursă propriu

Adițional Tailwind, am folosit animații de micro-interacțiune realizate cu Framer Motion iar iconografia provine din „react-icons”. De asemenea, pentru a conferi utilizatorului un sentiment de siguranță și încredere, am adăgat notificări non-intruzive după acțiuni de importanță ridicată precum confirmarea creării unui cont, sau finalizarea unei comenzi. Acestea au fost implementate cu ajutorul librăriei „react-toastify” (Fahim, 2024) și au fost configurate astfel încât feedbackul să fie instant și să ofere o experiență plăcută vizitatorilor.



Figură 3.6 - Notificări Toastify afișate la o conectare/deconectare cu succes

Sursă: Captură de ecran din aplicația proprie

### 3.1.1 Back-End

Dezvoltarea back-end se concentrează pe partea de server a unui website și include o aplicație, un server și o bază de date. De cele mai multe ori aceste elemente nu sunt vizibile pentru utilizator deoarece nu interacționează în mod direct cu ele. Cele mai folosite limbaje de programare folosite de către dezvoltatorii de back-end sunt Java, PHP, ASP.NET etc., la care se adaugă instrumente precum SQL Server sau Oracle pentru a stoca, organiza și modifica datele.

Deși selecția de tehnologii necesare pentru crearea unui server este una vastă, am considerat că cel mai potrivit mediu de execuție pentru realizarea acestui proiect este Node.js.

Node.js permite rularea de cod JavaScript în afara browserului, făcându-l potrivit pentru construirea de aplicații scalabile în ceea ce privește partea de server și rețea. Acesta permite, de asemenea, crearea de API-uri cu arhitecturi REST și oferă o bibliotecă bogată în module prin intermediul NPM(Node Package Manager).

Pentru mecanisme cerere-răspuns (request-response) am utilizat HTTP (Hypertext Transfer Protocol), asigurând comunicarea dintre client și server. Acest protocol a fost creat în anul 1989 în cadrul organizației CERN('European Council for Nuclear Research' sau în formatul original 'Conseil Européen pour la Recherche Nucléaire') de către Tim Berners-Lee, formând fundația pentru comunicare pe internet. În cadrul acestui protocol, clientul trimite o cerere către server, iar serverul răspunde cu datele cerute. O cerere tipică HTTP conține informații precum versiunea protocolului, un URL, o metodă, antete(headers) și, opțional, un corp. Metodele, numite și verbe, indică tipul de acțiune pe care cererea o așteaptă de la server.

În completarea acestor tehnologii, am ales să folosesc și Express, un micro-framework pentru Node.js, al cărui scop este să ușureze procesul de creare al aplicațiilor web. Acesta simplifică dezvoltarea aplicațiilor pe partea de server, permițând implementarea de site-uri cu pagini multiple, cât și folosirea de servicii REST(REpresentational State Transfer).

Serviciile de tip REST au fost prezentate pentru prima oară în anul 2000, în disertația despre arhitectura software scrisă de T. Fielding. REST nu reprezintă un protocol precum HTTP, ci este un set de principii de proiectare care definesc modul în care componentele unui sistem trebuie să interacționeze. Într-un astfel de sistem, clientul și serverul pot comunica doar trimițându-și mesaje care urmează structura unui protocol anume.

În cele mai multe cazuri, acest protocol este HTTP. Cele mai comune tipuri de mesaje care pot fi trimise sunt:

- GET - folosit pentru a obține o resursă anume de la un server.
- DELETE - folosit pentru ștergerea sau distrugerea unei resurse de pe un server.
- POST - folosit pentru a trimite date către un server cu scopul de a crea o resursă nouă.
- PUT - folosit pentru actualizarea unei resurse deja existente.

Starea unei resurse la un moment dat este cunoscută sub denumirea de reprezentare a resursei, și poate fi trimisă către client în diferite formate, precum JSON (JavaScript Object Notation), HTML, Python, XLT, PHP sau simplu text. Cu toate acestea, cel mai des utilizat este JSON deoarece este ușor de citit, și este independent de limbajul de programare. Apelurile către API-uri de tip REST conțin, de asemenea, antete (headers) și parametrii. Acestea includ informații de identificare precum metadate, URI-uri (Uniform Resource Identifier), setări de cache sau cookie-uri.

În urma instalării „express”, am optat și pentru instalarea pachetului CORS (Cross-Origin Resource Sharing), adăugând astfel un mecanism de securitate implementat de browsere.

Pentru a face posibilă livrarea la domiciliu contra-cost, am ales să folosesc Stripe, o companie de procesare a plăților, care poate accepta diferite tipuri de achitare, precum carduri de credit sau debit. Aceasta oferă diverse instrumente de facturare, gestionează financiar încasările beneficiarului, și previne fraude, fiind astfel o soluție adecvată pentru orice platformă digitală.

Una dintre cele mai importante componente ale oricărui website este baza de date la care este conectat, care conține toate informațiile necesare pentru interacțiunile dintre utilizator și server. În contextul acestei aplicații, am optat pentru varianta MongoDB, fiind una dintre cele mai populare baze de date. Aceasta are numeroase avantaje precum:

- Este un proiect open-source, așadar codul sursă este disponibil întregului public și poate fi vizualizat și modificat de oricine.
- Asigură flexibilitate prin scheme dinamice. Acest lucru asigură o stocare ușoară a datelor și permite modificări multiple și iterații rapide.
- Conferă scalabilitate și poate stoca o cantitate mare de informații.
- Este eficientă din punct de vedere al costului.

În vederea asigurării autentificării și autorizării în cadrul aplicației, am optat pentru implementarea unui mecanism bazat pe JSON Web Token. În ceea ce privește persistența și verificarea credențialelor, am decis ca parolele să fie procesate cu ajutorul librăriei Bcrypt.

### 3.2. Implementarea aplicației

Dezvoltarea acestei aplicații a urmat structura clasică a creării unui website, fiind respectați următorii pași:

- Planificarea sistemului - În cadrul acestei etape se stabilește necesitatea și fezabilitatea implementării proiectului. Se evaluează valoarea adăugată de sistem prin comparația cu alte produse asemănătoare de pe piață și se definesc cerințe globale, fără să se intre în detalii tehnice. Se caută răspuns pentru întrebarea „Merită să fie acest sistem dezvoltat?”.
- Analiza sistemului - În această etapă se identifică utilizatorii principali, funcționalitățile sistemului, și datele care trebuie prelucrate. Se caută răspuns

pentru întrebările „Ce trebuie să facă sistemul?” și „Ce persoane vor utiliza sistemul?”.

- Proiectarea sistemului - Este etapa în care se decide modalitatea prin care vor fi implementate cerințele identificate în cadrul analizei sistemului. Se aleg tehnologii software, componente hardware și se clarifică aspecte tehnice precum arhitectura sistemului, modelul de date și proiectarea algoritmilor pentru prelucrări.
- Implementarea sistemului - În acest punct, se realizează aplicațiile software în mod efectiv prin scriere de cod.
- Testarea sistemului - După ce funcționalitățile dorite au fost implementate, acestea vor fi testate separat pentru a verifica dacă mecanismul este adecvat. La final, se testează sistemul în ansamblu.
- Punerea în funcțiune a sistemului - În cazul în care sistemul satisface toate cerințele inițiale, acesta este pus în funcțiune și este disponibil pentru utilizatori.

În ceea ce privește implementarea website-ului pentru Studysphere, am ales abordarea clasică modulară, proiectul fiind compus din două subfoldere principale : frontend și backend. Primul pas a fost să creez un fișier principal numit index.js. Într-un terminal separat, am instalat și configurat framework-urile utilizate folosind comenzile: npm init, npm install express, npm install cors etc., construind astfel baza pentru server. Ulterior, am declarat o variabilă PORT și am inițiat-o cu valoarea 8080 (fig. 3.6), aceasta reprezentând punctul de acces prin care serverul ascultă și primește cereri HTTP de la client. Serverul pornește odată ce se efectuează și conectarea la baza de date MongoDB prin intermediul funcției connectDB() și a bibliotecii Mongoose.

```
app.use(cors(corsOptions));
app.use(express.json());
app.use(cookieParser());
app.use("/api", router);

const PORT=8080 || process.env.PORT

let gfs;

connectDB().then(()=>{
  const conn = mongoose.connection;
  gfs = Grid(conn.db, mongoose.mongo);
  gfs.collection('pdfs');
  app.listen(PORT, ()=>{
    console.log("Server is running");
  });
});
```

Figură 3.7 - Cod sursă pentru gestionarea unui server

Sursă: Cod sursă propriu

Următorul pas a fost structurarea elementelor necesare pentru frontend, astfel am format diverse categorii:

- /components - acest folder conține elemente reutilizabile precum footer, header, carduri pentru afișarea de informații etc.;



- /pages - acest folder conține paginile principale ale website-ului: AboutUs.jsx, BookDetails.jsx, Home.jsx, login.jsx etc.;
- /commons - conține o listă cu toate rutele backend-ului și metodele aferente, constituind un fel de configurație pentru apelurile către backend.
- /context - acest folder se ocupă de gestionarea stării globale și este folosit în procese de autentificare, modificarea numărului de produse din coșul de cumpărături etc.

### 3.2.1 Implementarea funcționalității de autentificare/înregistrare utilizatori

Acest proces începe în backend, prin definirea unui model de utilizator care conține câmpuri pentru informații precum nume, email, parolă criptată și rol. Pentru înregistrare, serverul primește aceste date prin intermediul unei metode de tip POST, le validează, criptează parola și salvează utilizatorul în baza de date, răspunzând apoi cu un mesaj de succes sau de eroare. Pentru autentificare, serverul primește email-ul și parola, va verifica dacă utilizatorul există și dacă parola este corectă folosind metoda de comparare bcrypt, iar dacă autentificarea este în regulă, se generează un token JWT care este trimis înapoi frontend-ului. În ceea ce privește interfața, sunt create formulare pentru ambele procese care conțin câte un buton de confirmare, iar apăsarea acestuia generează trimiterea de cereri POST către backend, folosindu-se URL-urile definite în fișierul index.js din /commons. În cazul în care acțiunea este efectuată cu succes, token-ul JWT se salvează în memoria locală, fiind șters atunci când vizitatorul dorește să se deconecteze.

### 3.2.2 Implementarea panoului administrativ

Panoul administrativ este format din două componente principale: secțiunea de gestionare a utilizatorilor și secțiunea de gestionare a cărților. Pentru prima componentă, a fost creat fișierul AllUsers.jsx, care încarcă o listă cu toate conturile create pe platformă folosind funcția fetchAllUsers(). Aceasta trimite un request sau o cerere către summaryAPI.allUsers, după care datele vor fi afișate într-un tabel (fig. 3.10). Administratorul are posibilitatea de a modifica informații ale unui utilizator folosind un buton de editare, care va deschide o fereastră tip formular. Pentru gestionarea cărților a fost creată componenta AllProducts.jsx, care se ocupă de afișarea tuturor cărților disponibile ale bibliotecii. Asemănător cu AllUsers.jsx, această acțiune are loc prin apelarea funcției fetchAllProducts() (fig. 3.8), generând o cerere către endpoint-ul definit în summaryAPI.allBooks (fig. 3.9). Cărțile vor fi apoi afișate (fig. 3.11), iar în pagina respectivă există posibilitatea de a introduce volume noi în baza de date prin intermediul unui formular, sau de a edita informații deja existente.

```

src > pages > AllProducts.jsx > ...
1 import React, { useEffect, useState } from 'react'
2 import UploadProduct from '../components/UploadProduct'
3 import summaryAPI from '../commons';
4 import AdminBookCard from '../components/AdminBookCard';
5
6 const AllProducts = () => {
7   const [openUploadProduct, setOpenUploadProduct]=useState(false);
8   const [allBooks, setAllBooks]=useState([]);
9
10  const fetchAllProducts=async()=>{
11    const response=await fetch(summaryAPI.allBooks.url);
12    const dataRes=await response.json();
13    console.log(dataRes);
14
15    setAllBooks(dataRes?.data || []);
16  }
17
18  useEffect(()=>{
19    fetchAllProducts();
20  })

```

Figură 3.8 - Fragment de cod pentru funcția fetchAllProducts()

Sursă: Cod sursă propriu

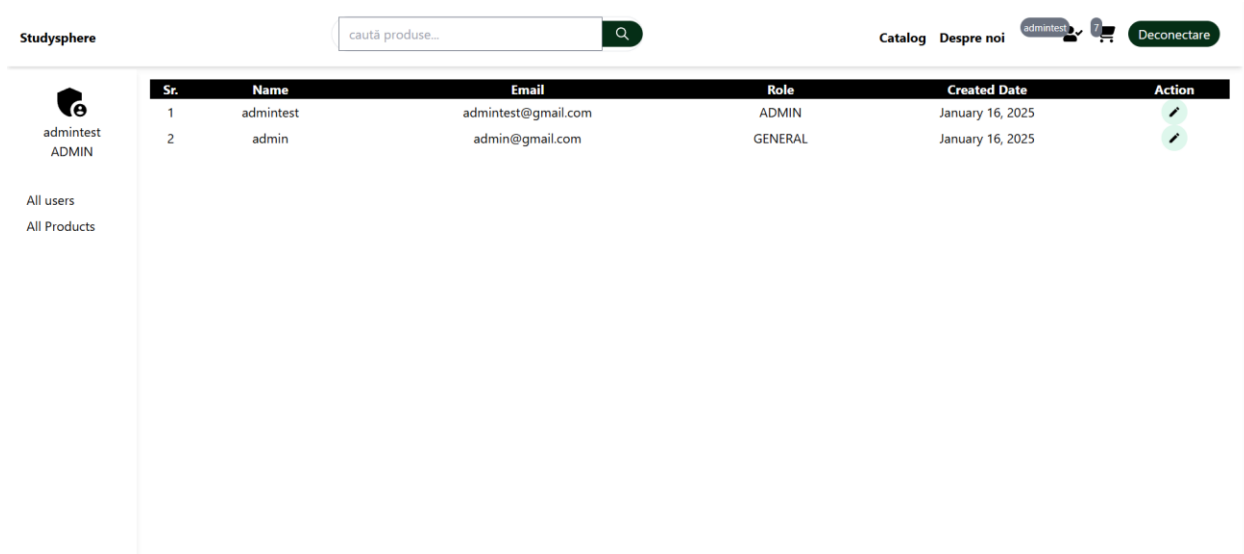
```

2 const domainBackend="http://localhost:8080";
3
4 const summaryAPI={
5   >   signUp:{ ...
8     },
9   >   signIn:{ ...
12    },
13  >   currentUser:{ ...
16    },
17  >   logoutUser:{ ...
20    },
21  >   allUsers:{ ...
24    },
25  >   updateUser:{ ...
28    },
29  >   uploadProduct:{ ...
32    },
33  >   allBooks:{ ...

```

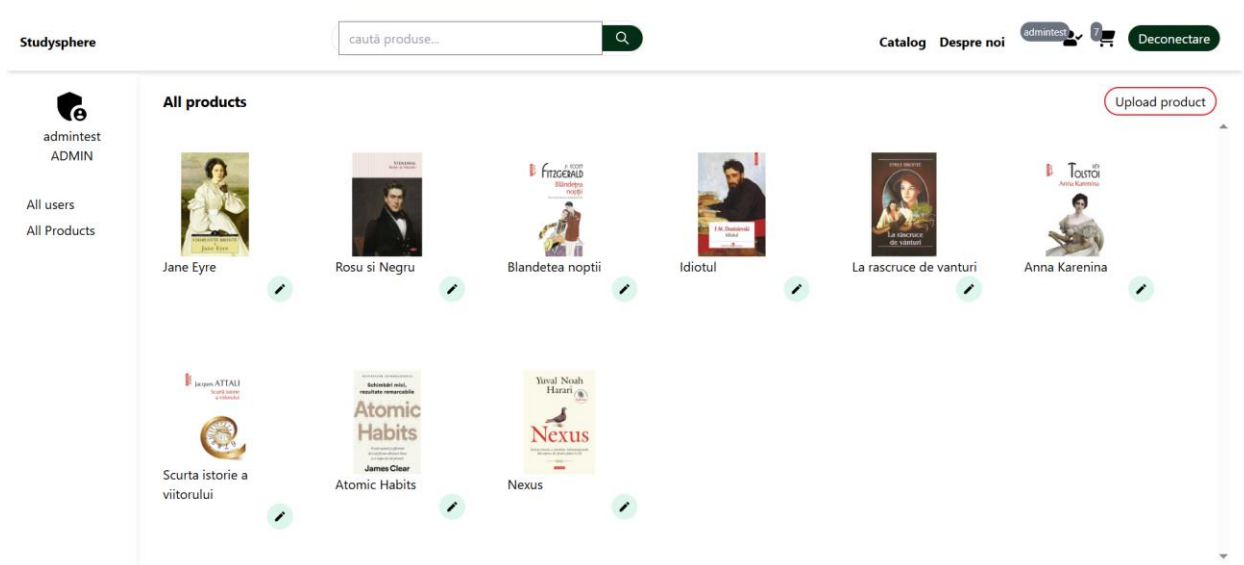
Figură 3.9 - Fragment de cod pentru summaryApi

Sursă: Cod sursă propriu



Figură 3.10 - Interfața panoului administrativ unde sunt afișați utilizatorii

Sursă: Captură de ecran din aplicația proprie



Figură 3.11 - Interfața panoului administrativ unde sunt afișate cărțile

Sursă: Captură de ecran din aplicația proprie

### 3.2.3 Implementarea catalogului digital

Implementarea catalogului de cărți a fost efectuată în două etape, integrând elemente atât din frontend, cât și din backend. Interfața acestuia permite afișarea de cărți, filtrarea în funcție de categorie, dar și adăugarea în coș a elementelor selectate. Extragerea tuturor obiectelor din baza de date s-a făcut cu ajutorul unui controller numit `getBookController` (fig. 3.12). Acesta folosește

modelul unui produs, definit într-un fișier separat, numit `productModel.js`, și metoda `find()` pentru returnarea tuturor înregistrărilor. În final, trimite un răspuns tip JSON care conține următoarele informații: `success: true`, `error: false`, `message: „All Books”`, și `data: allBooks`, care este de fapt lista propriu-zisă de cărți. În folderul de frontend am creat componenta `AllBooks.jsx`, care gestionează:

- Obținerea listei de cărți folosind funcția `fetchAllProducts` (fig. 3.13), prin intermediul căreia trimite o cerere de tip GET către ruta API, numită `allBooks`, și definită în `summaryAPI`. Răspunsul va fi stocat în două variabile, `allBooks` și `filteredBooks` (fig. 3.14), sub formă de JSON.
- Afișarea conținutului în pagină. Cărțile vor fi afișate folosind componenta `Card`, preluată din biblioteca `flowbite-react`. Fiecare element va conține o imagine cu coperta principală, titlul, autorul, și un buton „Adaugă în coș” care apelează funcția `handleAddToCart()`.
- Filtrarea după categorie. Genurile literare sunt afișate în sidebar, în partea din stânga a paginii într-un număr limitat. Utilizatorul poate alege să afișeze totalitatea acestora prin apăsarea butonului „Mai multe categorii”, sau poate tasta unul specific în bara de căutare. Atunci când un element este bifat sau debifat, variabila `selectedCategories` se actualizează în timp real, iar filtrarea se face cu ajutorul unui `useEffect`.

```
const productModel = require("../models/productModel")

const getBookController=async (req,res)=>{
  try {
    const allBooks=await productModel.find().sort({createdAt:-1});

    res.json({
      message:"All books",
      success:true,
      error:false,
      data:allBooks
    })
  } catch (err) {
    res.status(400).json({
      message:err.message || err,
      error:true,
      success:false
    })
  }
}
```

Figură 3.12 - Controller pentru extragerea cărților

Sursă: Cod sursă propriu

```

const handleAddToCart = async(e,id)=>{
  await addToCart(e,id)
  fetchUserAddToCart()
}

const fetchAllProducts=async()=>{
  const response=await fetch(summaryAPI.allBooks.url);
  const dataRes=await response.json();
  console.log("verificare date",dataRes);

  setAllBooks(dataRes?.data || []);

  setFilteredBooks(dataRes?.data || []);
}

useEffect(()=>{
  fetchAllProducts();
},[])

```

Figură 3.13 - Funcția `fetchAllProducts()` și `handleAddToCart()`

Sursă: Cod sursă propriu

```

useEffect(() => {
  const selected = Object.keys(selectedCategories).filter((key) => selectedCategories[key]);
  console.log(" selected filters", selected);
  if (selected.length === 0) {
    setFilteredBooks(allBooks);
  } else {
    const filtered = allBooks.filter((book) => {
      console.log("book.category:", book.category);
      return book.category.some(cat => selected.includes(cat));
    });

    setFilteredBooks(filtered);
  }
}, [selectedCategories, allBooks]);

const handleCategoryChange = (e) => {
  const { value, checked } = e.target;
  setSelectedCategories((prev) => ({
    ...prev,
    [value]: checked,
  }));
};

const filteredCategoryList = bookCategory.filter(cat =>
  cat.label.toLowerCase().includes(categorySearch.toLowerCase())
);
const displayedCategories = showAllCategories ? filteredCategoryList : filteredCategoryList.slice(0, 10);

```

Figură 3.14 - Filtrarea cărților în funcție de categorie

Sursă: Cod sursă propriu

### 3.2.4 Gestionarea coșului de cumpărături și efectuarea comenzilor

Afișarea produselor din coșul de cumpărături se realizează prin intermediul componentei ViewCart.jsx. Aceasta se ocupă de afișarea tuturor produselor pe care utilizatorul le-a adăugat în coș, oferind și posibilitatea ștergerii elementelor. Această acțiune actualizează interfața, cât și informațiile de pe server, printr-o cerere DELETE către API. De asemenea, există două butoane în partea din dreapta a paginii care afișează „Ridicare personală” și „Livrare la domiciliu”. Odată ce utilizatorul le apasă, va fi redirecționat către o pagină nouă, unde va finaliza procesul de rezervare al cărților selectate.

În cazul livrărilor la domiciliu, plățile se efectuează cu ajutorul Stripe Checkout, în funcția handlePayment() (fig. 3.15) din componenta frontend. În cadrul acesteia, se trimite o solicitare către backend, urmând să se creeze o sesiune de plată. În backend, am creat fișierul paymentController.js (fig. 3.16), unde sunt definite detalii precum metoda de plată, informații despre client, și lista cu produsele comandate. Fiind o bibliotecă publică, nu există prețuri pentru cărți, astfel parametrul unit\_amount este inițiat cu 0. Odată ce plata a fost finalizată, Stripe apelează ruta API configurată în fișierul webhook.js. Cărțile rezervate sunt extrase, precum și detaliile despre comandă cu privire la plată și metoda de livrare, și ulterior, sunt salvate în baza de date conform schemei orderModel. În final, coșul de cumpărături este golit, iar comanda este înregistrată pe numele utilizatorului conectat, acesta putând să o vizualizeze la secțiunea Comenzi din pagina principală.

```
const handlePayment = async (orderType) => {
  const stripePromise = await loadStripe('pk_test_51R10ewIPak9Bh8sysSVzAcEMU0jvcM1gmt9wnrBqrdzQ2RZwY9Im5C9gK7NshK1IKcQsrHIMiD908penB0o1DEvf00nvIhoafC')
  const response = await fetch(summaryAPI.payment.url, {
    method: summaryAPI.payment.method,
    credentials: 'include',
    headers: {
      "content-type": 'application/json'
    },
    body: JSON.stringify({
      cartItems: data,
      orderType: orderType
    })
  })

  const responseData = await response.json()

  if (responseData?.id) {
    stripePromise.redirectToCheckout({ sessionId: responseData.id })
  }
}
```

Figură 3.15 - Segment de cod din funcția handlePayment()

Sursă: Cod sursă propriu

```

const paymentController=async(req,res)=>{
  try {
    const {cartItems,orderType}=req.body;
    const user = await userModel.findOne({ _id : req.userId });
    const shippingOptions = orderType === "ridicare personala" ? [] : [
      {
        shipping_rate: 'shr_1R1PIWIPaK9Bh8syiHksvI19' // Transport pentru livrare
      }
    ];
  };
  const params = {
    submit_type : 'pay',
    mode : "payment",
    payment_method_types : ['card'],
    billing_address_collection : 'auto',
    shipping_options: shippingOptions,
    customer_email : user.email,
    metadata : {
      userId : req.userId,
      orderType: orderType
    },
    line_items : cartItems.map((item,index)=>{
      return{
        price_data : {
          currency : 'RON',
          product_data : {
            name : item.bookId.bookTitle,
            images : item.bookId.bookImage,
            metadata : {
              productId : item.bookId._id
            }
          },
          unit_amount : 0
        },
        quantity : 1
      }
    }),
    success_url : `${process.env.FRONTEND_URL}/success`,
    cancel_url : `${process.env.FRONTEND_URL}/cancel`,
  }
}

```

Figură 3.16 - Controller pentru efectuarea de plăți

Sursă: Cod sursă propriu

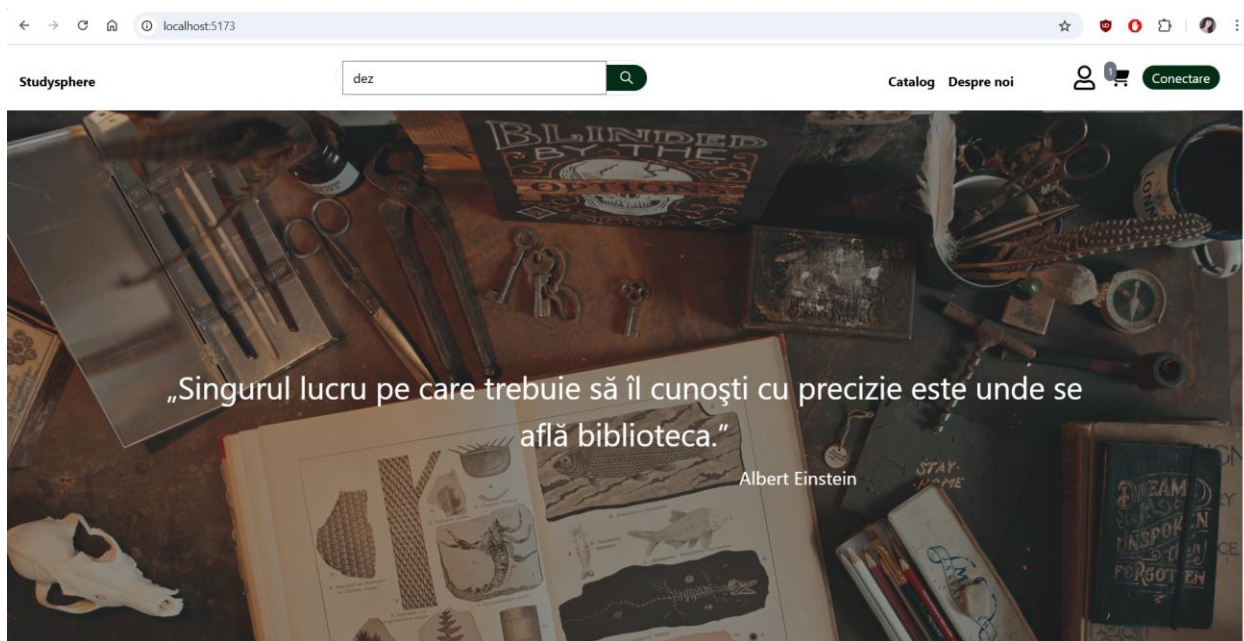
### 3.3. Prezentarea aplicației

Aplicația web realizată are scopul de a oferi publicului o platformă modernă și ușor de utilizat, cu scopul de a gestiona împrumuturile de cărți în cadrul bibliotecii Studysphere. Din acest punct de vedere, există două componente principale care au stat la baza construirii acestui website, și anume: funcționalitățile acestuia precum posibilitatea de înregistrare sau de efectuare comenzi, cât și design-ul propriu-zis, care constituie un instrument prin care utilizatorul poate avea parte de o experiență satisfăcătoare.

În ceea ce privește rezultatul digitalizării unei biblioteci publice, de cele mai multe ori acesta nu răspunde nevoilor unui cetățean de rând. Fie că este vorba de funcționalități mult prea complexe, probleme de conectare la server sau un aspect deloc intuitiv, consecința rămâne aceeași: utilizatori dezinteresați, care riscă să părăsească site-ul în câteva secunde, și pierderea credibilității instituției. Așadar, misiunea principală a acestui proiect este de a oferi un portal cu aspect modern

dar ușor de înțeles și de navigat, care se concentrează pe a satisface cerințele unui vizitator într-un mod cât mai eficient.

În primă instanță, când o persoană accesează platforma Studysphere, va fi întâmpinată de pagina principală (fig. 3.17).



Figură 3.17 - Interfața paginii principale a aplicației

Sursă: Captură de ecran din aplicația proprie

În cadrul acesteia, conținutul este structurat astfel încât să reprezinte un punct de interes pentru vizitatori. Partea de sus a paginii conține un header, sau un antet, format din următoarele secțiuni:

- Logo-ul bibliotecii, afișat în stânga-sus (fig. 3.18) . Poziționarea acestuia este strategică, deoarece va fi primul element pe care îl vor vedea utilizatorii, asigurând recunoașterea instituției. Conform unui studiu (Nielsen Norman Group, Whitenton K., 2016), o persoană este cu 89% mai predispusă să își amintească logo-ul unei companii atunci când sunt plasate în mod tradițional, anume stânga-sus.

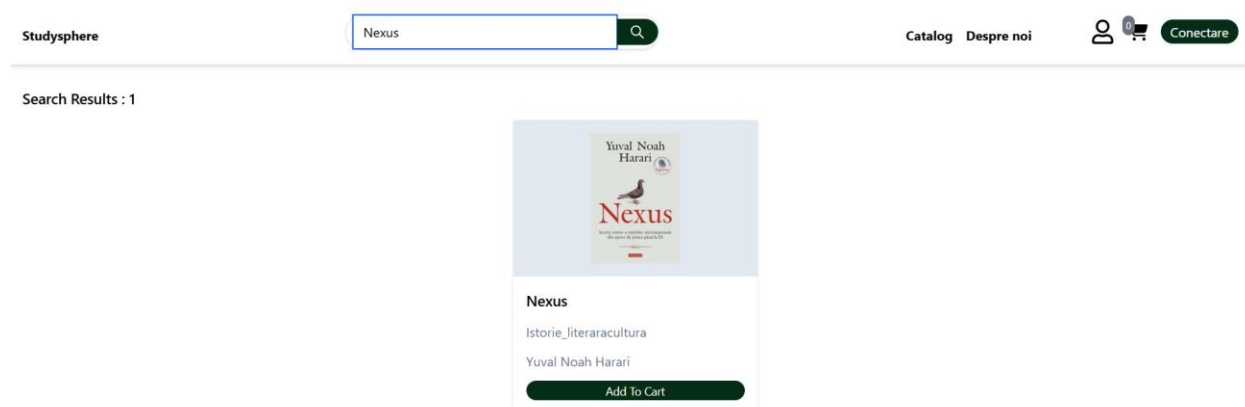


Figură 3.18 - Antetul/Header-ul aplicației

Sursă: Captură de ecran din aplicația proprie



- O bară de căutare, poziționată la mijloc, prin intermediul căreia vizitatorii pot căuta anumite cărți după titlu, autor sau gen literar (fig. 3.19). Existența ei îmbunătățește experiența utilizatorului, oferind o metodă rapidă și ușoară de căutare. În acest fel, timpul petrecut pe platformă crește considerabil, iar rata de finalizare a unei comenzi este mai mare.



Figură 3.19 - Bara de căutare din antetul aplicației

Sursă: Captură de ecran din aplicația proprie

- Un meniu, care conține principalele puncte de interes ale platformei: catalogul online, unde sunt afișate cărțile din colecția bibliotecii, secțiunea informativă a instituției, unde sunt descrise serviciile efectuate, și secțiunea referitoare la contul utilizatorului.

Sub acest antet se află o secțiune dinamică de conținut, menită să atragă atenția vizitatorilor prin aspectul plăcut și interesant. Astfel, containerul paginii principale afișează un videoclip sugestiv pentru misiunea platformei, peste care este plasat unul din citatele inspiraționale ale lui Albert Einstein: „Singurul lucru pe care trebuie să îl cunoști cu precizie este unde se află biblioteca.”.

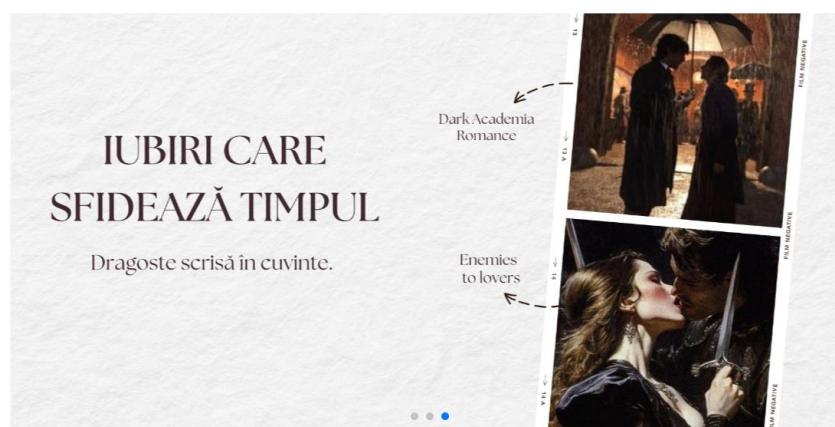
După o scurtă derulare în jos, este afișat un set de carduri animate care afișează câteva statistici precum numărul de cărți gestionate de bibliotecă, de resurse digitale, de autori, și de membri înregistrați (fig. 3.20). Componentele sunt modularizate prin intermediul unei componente separate CardItem.jsx, existând imagini personalizate pentru fiecare card, iar animația fluidă este realizată cu ajutorul bibliotecii React Spring.



Figură 3.20 - Set de cărți animate pentru a îmbunătăți designul interfeței

Sursă: Captură de ecran din aplicația proprie

În continuare, website-ul conține o componentă tip banner (fig. 3.21), implementată prin intermediul unei biblioteci moderne și eficiente, Swiper. Aceasta creează un carusel dinamic cu 3 imagini importate local, iar trecerea de la un cadru la altul se face prin glisare. Scopul este de a promova diverse genuri literare populare la momentul actual, cât și de a crea o primă impresie puternică.

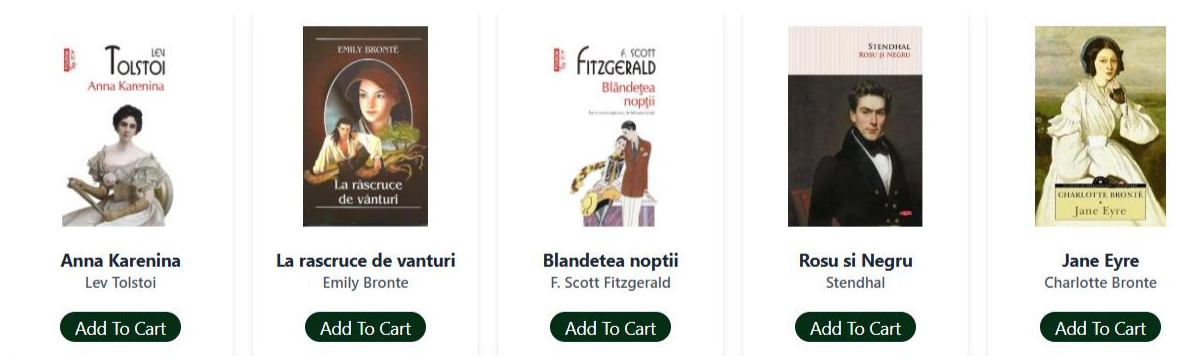


Figură 3.21 - Banner prezent pe pagina principală

Sursă: Captură de ecran din aplicația proprie

În partea de jos a platformei există o secțiune dedicată unei categorii de cărți alese de administrator (fig. 3.22). Aceasta poate fi schimbată săptămânal sau lunar, și afișează câteva volume reprezentative.

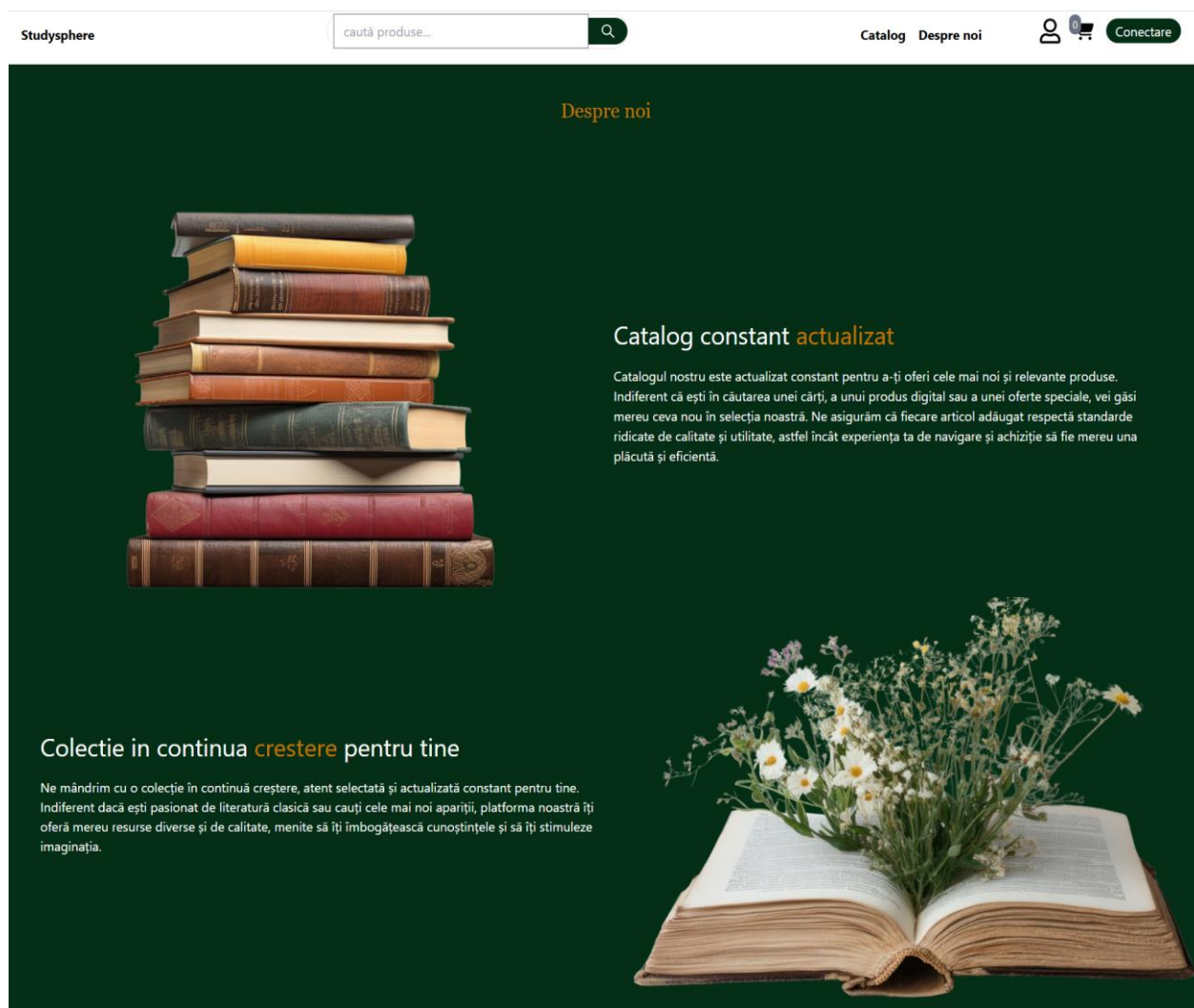
## Top books : literatura clasica



Figură 3.22 - Secțiune de cărți populare

Sursă: Captură de ecran din aplicația proprie

Pentru a sublinia valoarea pe care o aduce existența acestui website este utilizată pagina „Despre noi”, care poate fi accesată prin intermediul meniului din antet. Aceasta servește drept punct de legătură între companie și vizitatori, descriind obiectivele companiei și serviciile oferite. De asemenea, creează un sentiment de încredere pentru utilizator printr-o împărtășire transparentă de valori și misiuni. Această secțiune este stilizată cu un fundal verde-închis pentru a transmite senzația de calm, eleganță și profesionalism, iar pentru un contrast puternic a fost adăugat text alb și galben-închis, fiind ușor de citit. Pentru a adăuga un element dinamic, am ales folosirea librăriei Framer Motion cu scopul de a realiza animații bazate pe tranziții tip fade-in și slide. Conținutul se împarte în două blocuri: cel semnificativ pentru informații despre organizație (fig. 3.23 și fig. 3.24), și cel reprezentativ pentru serviciile oferite (fig. 3.25).



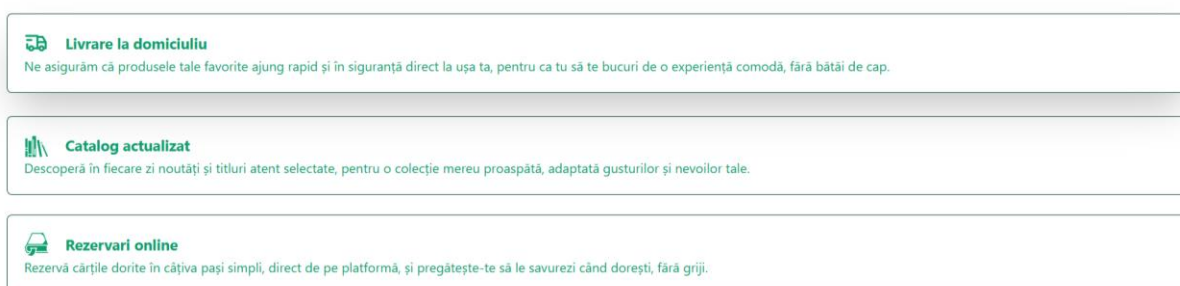
Figură 3.23 - Segmente din secțiunea „Despre noi”

Sursă: Captură de ecran din aplicația proprie



Figură 3.24 - Segmente din secțiunea „Despre noi”

Sursă: Captură de ecran din aplicația proprie



Figură 3.25 - Secțiunea de servicii oferite

Sursă: Captură de ecran din aplicația proprie

O altă componentă fundamentală pentru orice platformă online este formularul de înregistrare (fig. 3.27), însoțit de cel de conectare (fig. 3.26). Acestea oferă o experiență personalizată utilizatorului și constituie o metodă prin care acesta poate interacționa cu site-ul dincolo de simpla vizualizare a conținutului. Fie că este vorba de accesul la un dashboard, finalizarea unei comenzi, sau pur și simplu pentru salvarea anumitor preferințe, aceste formulare sunt esențiale pentru funcționalitate, cât și pentru satisfacția vizitatorilor.

În cadrul acestui proiect, o persoană poate accesa secțiunea de înregistrare/conectare prin intermediul butonului din dreapta-sus a antetului. Inițial va fi redirecționat către formularul de conectare, unde se va putea autentifica trimițând către backend un email și o parolă. Această componentă poate fi îmbunătățită prin implementarea unei protecții suplimentare la încercări greșite multiple, prin oferirea posibilității de conectare atunci când parola a fost uitată sau prin adăugarea unei confirmări prin trimiterea unui cod pe email.

Studysphere

caută produse...

Catalog Despre noi

Conectare

# Login

Email:

Introduceți adresa de email

Parolă:

Introduceți parola

Parolă uitată?

Login

Nu aveți un cont? [Sign-up](#)

Figură 3.26 - Formular de conectare

Sursă: Captură de ecran din aplicația proprie

În cazul în care vizitatorul nu are un cont, există posibilitatea de a crea unul prin completarea unui formular de înregistrare. Un astfel de formular poate seta tonul experienței unui individ cu platforma. În acest sens, dacă există prea multe câmpuri care trebuie completate, utilizatorii tind să abandoneze procesul. Astfel, pentru a crea o identitate digitală în baza de date a instituției Studysphere, sunt necesare următoarele informații: nume, email, parolă și confirmarea parolei. Există, însă, loc pentru posibile îmbunătățiri: validarea poate fi mai detaliată, impunându-se reguli referitoare la lungimea și structura parolei, deblocarea câmpurilor după validarea celor anterioare sau adăugarea unui spinner de încărcare la trimiterea formularului.

Studysphere

caută produse...

Catalog Despre noi

Conectare

# Sign up

Nume:

Introduceți numele dvs.

Email:

Introduceți o adresă de email validă

Parolă:

Introduceți parola dorită

Confirmați parola:

Reintroduceți parola de mai sus

Sign-up

Aveți deja un cont? [Login](#)

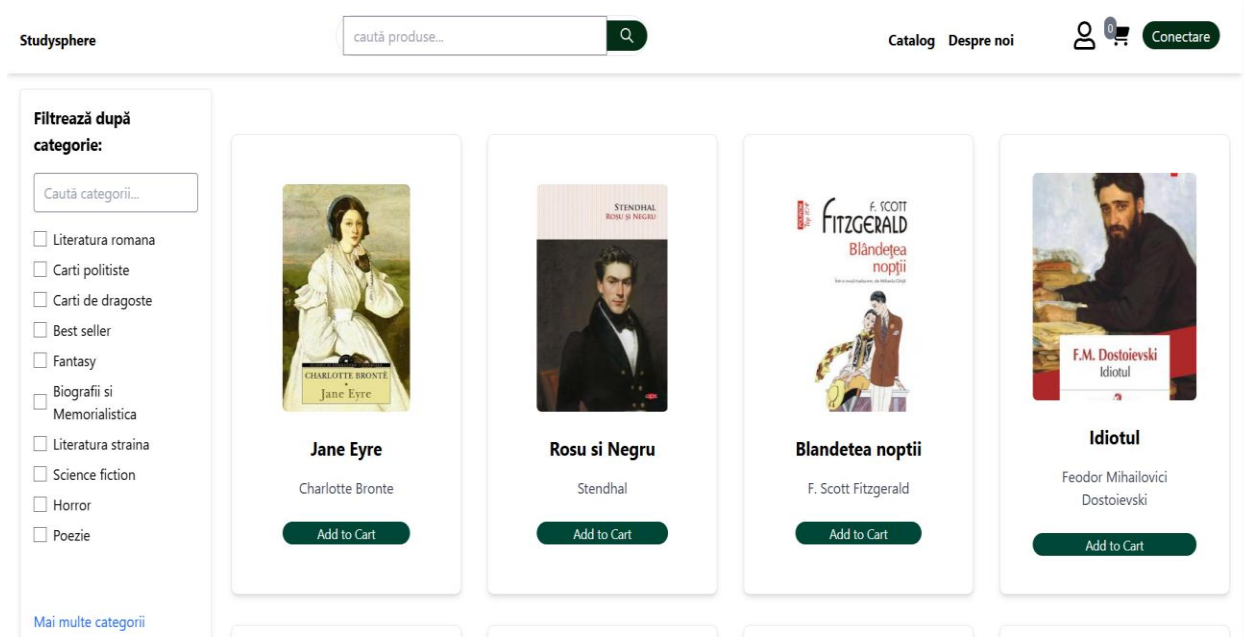
Figură 3.27 - Formular de înregistrare/creare cont digital

Sursă: Captură de ecran din aplicația proprie

Din perspectiva companiei, aceste formulare asigură colectarea de date. Odată ce un utilizator creează un cont, website-ul poate oferi conținut personalizat și poate monitoriza activitatea acestuia. În esență, cele două procese nu sunt doar elemente tehnice, ci și puncte cheie în experiența unui utilizator. Atunci când sunt implementate adecvat, acestea pot spori implicarea și pavează calea către o interacțiune de succes între public și site.

O platformă care oferă o gamă largă de resurse, precum o bibliotecă online, are un element crucial, și anume catalogul digital. Acesta servește drept punct de centralizare pentru produsele gestionate de instituție, și oferă utilizatorilor posibilitatea de a căuta și de a descoperi noi articole. În prezent, orice persoană dorește acces rapid și eficient la informație, astfel în cadrul platformei Studysphere, vizitatorul poate accesa catalogul printr-un simplu click în cadrul antetului de pe pagina principală. Redirectionarea este instantanee, iar afișarea listei de cărți deținute de bibliotecă se face în cadrul unei grile responsivă (fig. 3.28), fiecare produs fiind prezentat sub forma unui card care conține coperta, titlul, autorul, și un buton de „Adaugă în coș”.



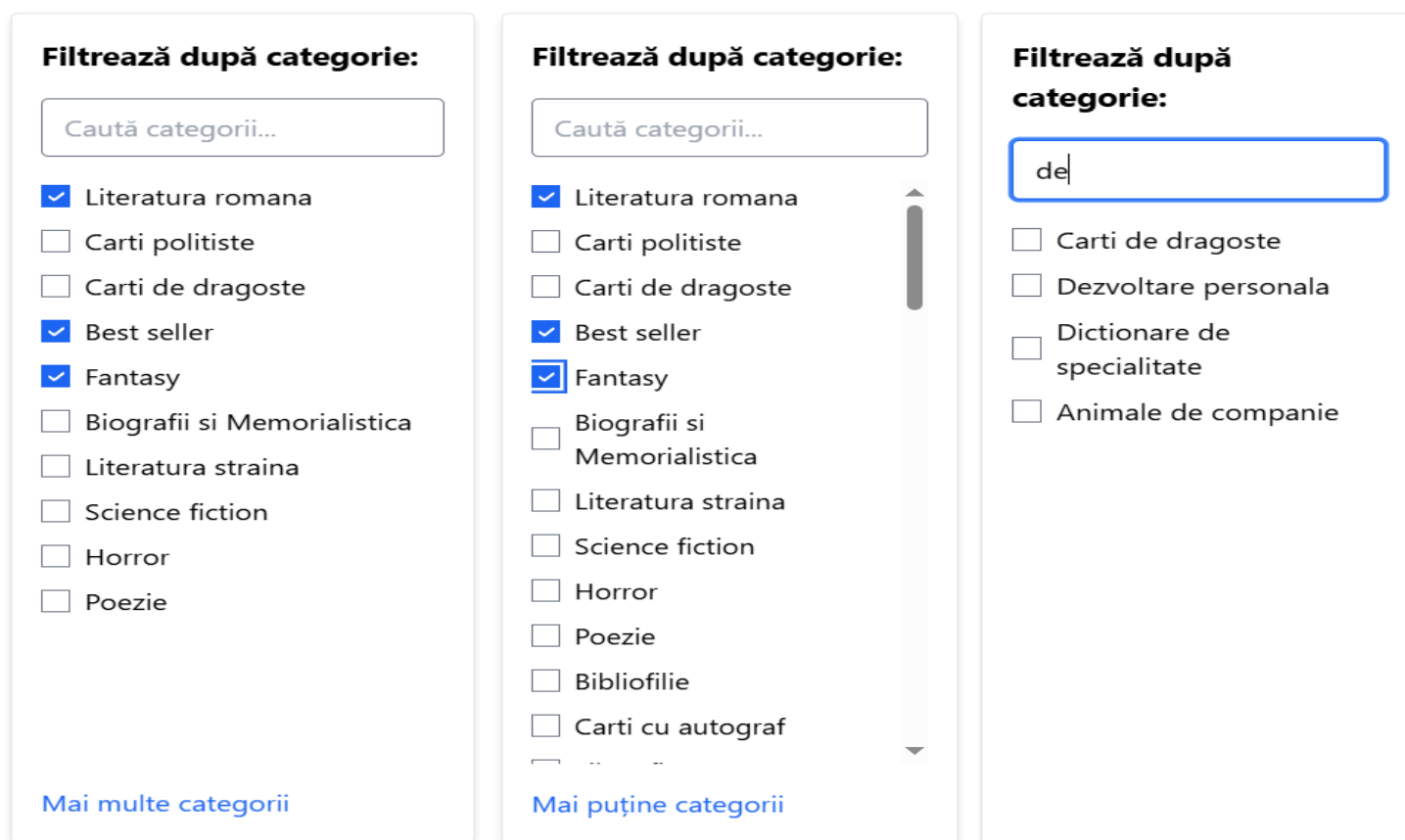


Figură 3.28 - Catalogul digital pentru biblioteca Studysphere

Sursă: Captură de ecran din aplicația proprie

Această secțiune a website-ului este concepută pentru a oferi o experiență intuitivă de căutare, având o interfață vizuală atractivă și funcțională. Una din funcționalitățile cheie este filtrarea avansată după gen literar (fig. 3.29), astfel utilizatorii pot selecta una sau mai multe categorii, iar rezultatul se actualizează în timp real. Pentru a evita o aglomerare vizuală, sunt expuse doar 10 categorii, însă există posibilitatea expandării listei. De asemenea, în containerul dedicat filtrării există și o bară de căutare, care permite găsirea rapidă a unei categorii printr-un simplu input de la tastatură. Câteva aspecte care pot fi îmbunătățite sunt: implementarea unui filtru persistent prin păstrarea informațiilor în localStorage, adăugarea mai multor filtre pentru o căutare cât mai eficientă.





Figură 3.29 - Sistemul de filtrare în funcție de gen literar

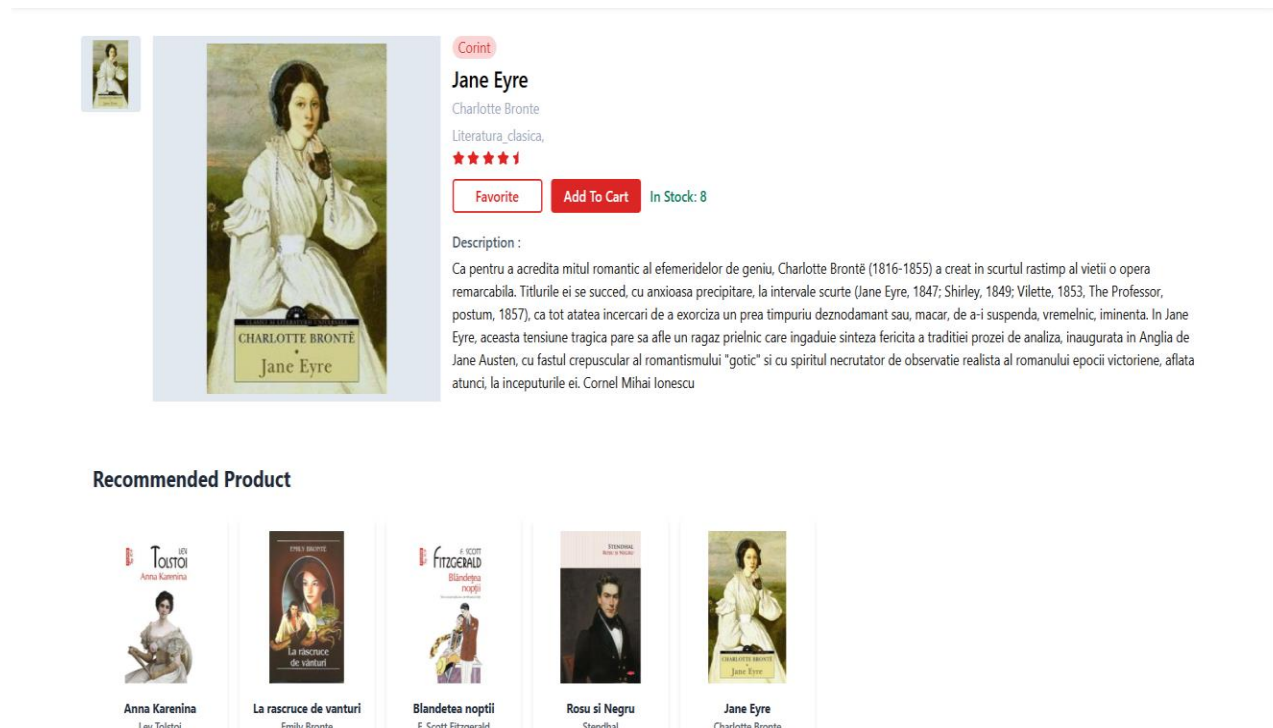
Sursă: Captură de ecran din aplicația proprie

Crearea unui catalog digital în cadrul aplicației a avut atât scopul de a crea o experiență plăcută pentru utilizator, cât și de a crea un suport pentru managementul inventarului în secțiunea de backend, optimizând accesibilitatea și organizarea resurselor. Componenta AllBooks.jsx reflectă direct această viziune, oferind o interfață intuitivă și funcțională, unde fiecare carte este afișată într-un card cu imagine, titlu, autor și buton de adăugare în coș, iar printr-un simplu clic pe imaginea sau titlul cărții, utilizatorul este redirecționat către o pagină dedicată acelei cărți.

Paginile individuale pentru fiecare carte sunt elemente esențiale în cadrul unei biblioteci digitale, oferind o descriere mai detaliată și ajutând la luarea unei decizii informate. Spre deosebire de un catalog, o astfel de pagină asigură o explorare mult mai profundă a volumului respectiv, evidențiind aspecte principale, cât și secundare.

Platforma Studysphere oferă informații despre fiecare carte (fig. 3.30), precum titlul, autorul, editura, genurile literare și descrierea. De asemenea, făcând legătura între backend și

frontend, există și o secțiune care arată câte exemplare mai sunt disponibile pentru împrumut. Utilizatorul poate să efectueze două acțiuni rapide, și anume: salvarea cărții în secțiunea de favorite și adăugarea acesteia în coșul de cumpărături, doar dacă este în stoc.



Figură 3.30 - Pagină descriptivă pentru fiecare carte

Sursă: Captură de ecran din aplicația proprie

Pentru creșterea accesibilității, cititorii pot beneficia de un exemplar PDF al cărții (fig. 3.31), în cazul în care acesta este disponibil. Dacă în baza de date cartea are asociat un fișier PDF, acesta este redat direct în pagină, printr-un iframe HTML.

Avantajele integrării PDF-ului sunt numeroase: utilizatorii pot să citească conținutul cărții direct din platformă, fără să fie nevoiți să descarce vreun document, afișarea directă în aplicație menține cititorul implicat, și riscul de abandon sau comutare între site-uri scade, iar cărțile într-o stare fragilă pot fi conservate.

Această funcționalitate adaugă un plus de valoare catalogului digital, aliniindu-se obiectivelor moderne ale bibliotecii de a facilita accesul rapid, liber și universal la cunoaștere.



Figură 3.31 - Secțiunea unde poate fi găsită varianta PDF a unei cărți

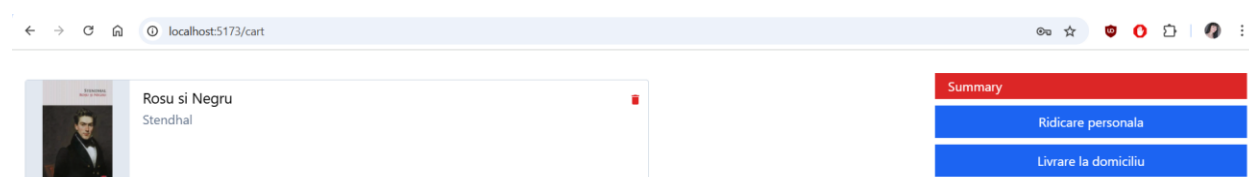
Sursă: Captură de ecran din aplicația proprie

În dezvoltarea unui website cu funcționalitate de comerț electronic sau gestionare a produselor (precum o librărie online sau o bibliotecă digitală), componentele precum „Coșul de cumpărături” și „Lista de favorite” joacă un rol esențial în îmbunătățirea experienței utilizatorului și în creșterea eficienței interacțiunii cu platforma.

Coșul de cumpărături permite utilizatorului să salveze temporar cărțile sau produsele dorite pentru o viitoare rezervare sau achiziție. Este un element fundamental pentru orice sistem de tranzacționare, permițând o vizualizare rapidă a articolelor selectate, modificarea conținutului prin ștergerea elementelor și inițierea plății.

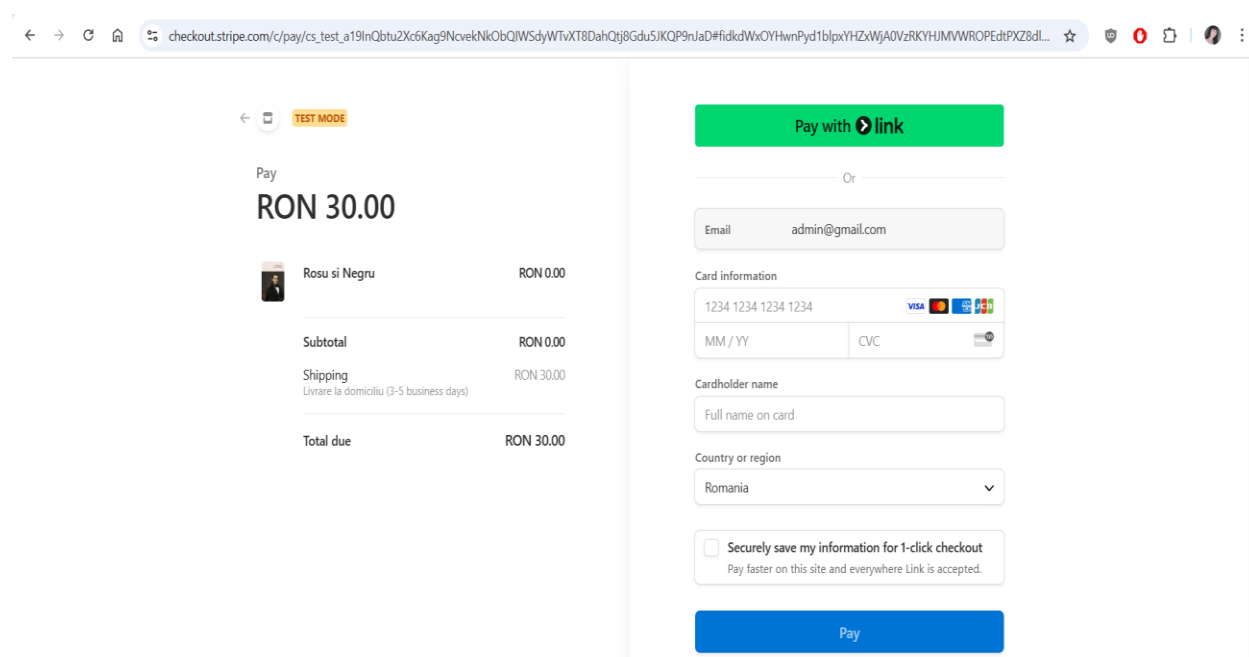
Componenta ViewCart.jsx din cadrul acestei aplicații oferă o funcționalitate esențială pentru gestionarea coșului de cumpărături, permițând utilizatorului să vizualizeze cărțile adăugate, să le elimine și să finalizeze comanda printr-un sistem de plată. Aceasta utilizează contextul

aplicației și funcții asincrone pentru a comunica cu backend-ul, de unde preia sau actualizează informațiile legate de coș. Fiecare produs este afișat într-un container individual, cu imaginea cărții, titlul, autorul și categoria, iar în partea dreaptă sus a fiecărui produs apare un buton de ștergere, care permite eliminarea acestuia. În partea dreaptă a interfeței (pe ecrane mari), se afișează un sumar al comenzii, dacă există produse în coș. Acest sumar include două butoane care inițiază procesul de plată: „Ridicare personală” și „Livrare la domiciliu” (fig. 3.32). Ambele apelează funcția `handlePayment`, care comunică cu backend-ul pentru a crea o sesiune Stripe corespunzătoare comenzii, iar utilizatorul este redirecționat către pagina Stripe pentru finalizarea plății (fig. 3.33 și fig. 3.34).



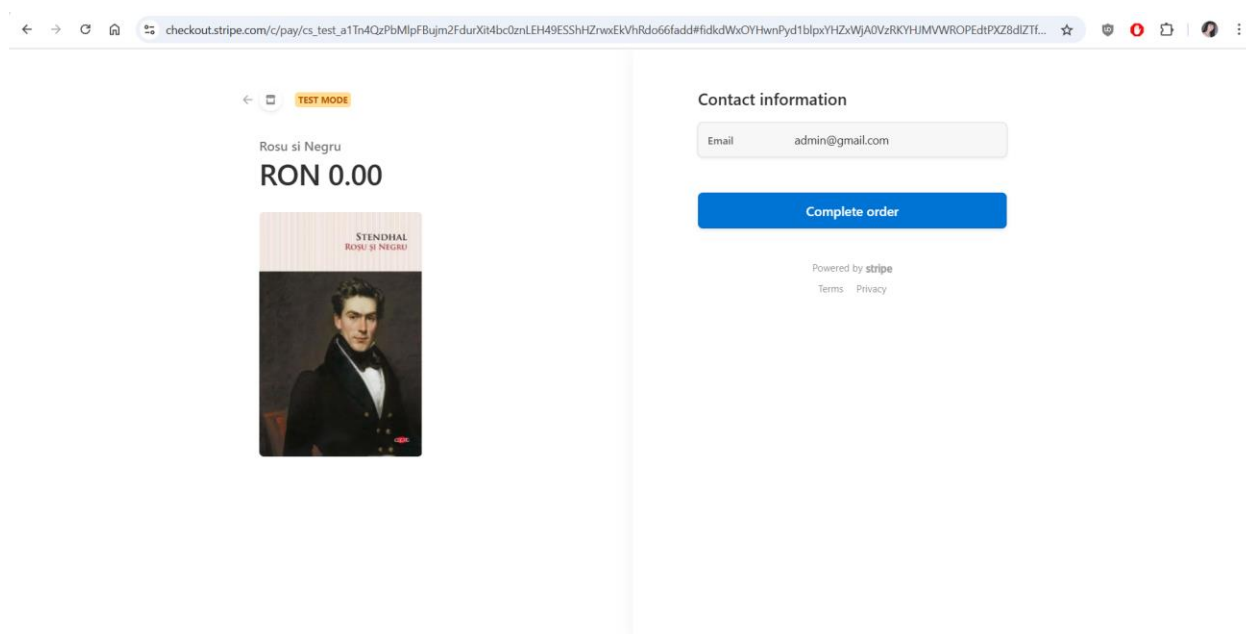
Figură 3.32 - Pagina unde pot fi vizualizate produsele adăugate în coșul de cumpărături

Sursă: Captură de ecran din aplicația proprie



Figură 3.33 - Pagină pentru efectuarea plății la comenzile cu livrare la domiciliu

Sursă: Captură de ecran din aplicația proprie

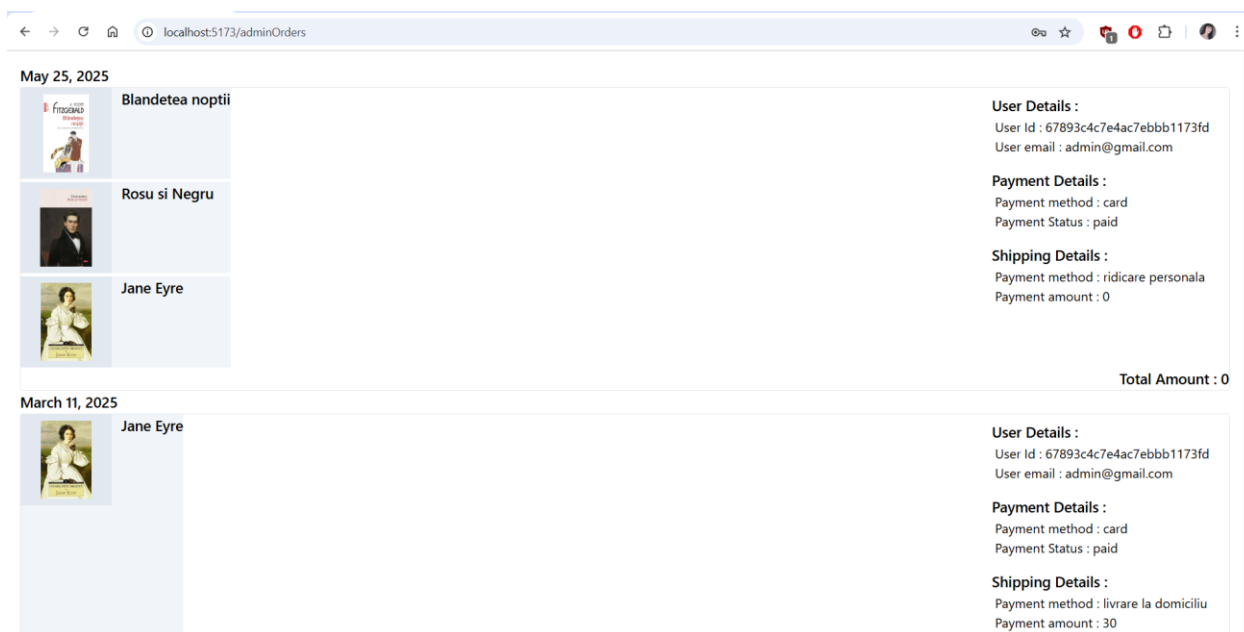


Figură 3.34 - Pagină pentru efectuarea tranzacției la comenzile cu ridicare personală

Sursă: Captură de ecran din aplicația proprie

Funcționalitatea de „favorite” permite utilizatorilor să salveze cărți sau produse de interes fără intenția imediată de cumpărare sau rezervare. Pentru platforma Studysphere, afișarea listei de cărți salvate ca favorite se face cu ajutorul componentei ViewFavs.jsx. Aceasta funcționează într-un mod similar cu componenta ViewCart, dar se concentrează exclusiv pe afișarea și gestionarea produselor marcate ca favorite, fără funcționalitate de plată.

Într-un website modern care implică tranzacții, pagina de comenzi reprezintă un element fundamental pentru transparență, control și încredere în relația dintre utilizator și platformă. Această funcționalitate este implementată în cadrul aplicației prin componenta OrdersPage.jsx, care permite vizualizarea comenzilor efectuate anterior de către un cititor, inclusiv detalii precum metoda de plată, lista cu produsele rezervate, tipul de livrare ales sau suma totală (fig. 3.35).



Figură 3.35 - Vizualizarea comenzilor efectuate

Sursă: Captură de ecran din aplicația proprie

Existența acestei secțiuni contribuie la personalizarea experienței unui utilizator, permite gestionarea comenzilor și ajută la creșterea încrederii. Deși este utilă, funcționalitatea poate fi semnificativ îmbunătățită: comenzile pot fi grupate sau filtrate după dată/status, se poate adăuga un buton prin care comanda poate fi printată sau exportată într-un fișier PDF.

## Concluzii

Proiectul realizat a avut ca scop principal dezvoltarea unei aplicații web dedicate digitalizării proceselor din cadrul unei biblioteci, în special prin facilitarea accesului utilizatorilor la catalogul de cărți, gestionarea listelor personale de favorite și procesarea comenzilor printr-un sistem modern de plată online. În acest sens, obiectivele principale au fost concepute astfel încât să reflecte necesitățile actuale ale utilizatorilor, dar și să sprijine eforturile bibliotecii de a-și eficientiza activitatea.

Pentru a realiza obiectivele definite la începutul proiectului, am dezvoltat componente funcționale în React care permit afișarea cărților adăugate în lista favoritelor, actualizarea lor în timp real, ștergerea produselor nedorite, dar și vizualizarea detaliată a comenzilor plasate, inclusiv informații despre metodele de plată și detaliile livrării.

Importanța digitalizării în domeniul bibliotecilor este una esențială, mai ales în contextul evoluției tehnologice rapide și al schimbărilor cu privire la modul în care oamenii consumă informația. Prin implementarea acestei aplicații, se facilitează accesul rapid la resursele bibliotecii fără a fi necesară prezența fizică, se optimizează procesele interne și se oferă o transparență mai mare asupra disponibilității produselor. Acest lucru contribuie nu doar la creșterea satisfacției utilizatorilor, ci și la o administrare mai eficientă a resurselor.

Pe parcursul dezvoltării, am întâmpinat mai multe provocări, însă, în final, acestea au adus valoare procesului de învățare. Gestionarea stării aplicației în mod asincron, sincronizarea datelor între backend și frontend, și asigurarea unui feedback corespunzător în timpul încărcării datelor au fost doar câteva dintre acestea. Totodată, integrarea serviciului Stripe a presupus înțelegerea tranzacțiilor monetare, asigurarea securității datelor și gestionarea eventualelor erori în procesul de checkout, aspecte critice pentru orice platformă ce prelucrează tranzacții financiare.

Contribuțiile mele personale se reflectă în crearea unor componente scalabile și ușor de întreținut, în structurarea logică a fluxului aplicației și în adoptarea celor mai bune practici legate de experiența utilizatorului. Am realizat un design responsiv, și am pus accent pe claritatea și consistența interfeței grafice. Mai mult, am dezvoltat funcționalități care permit o interacțiune intuitivă cu lista de favorite și de comenzi, oferind utilizatorilor un control complet asupra experienței lor.

Cu toate acestea, aplicația poate fi îmbunătățită prin implementarea unor funcționalități suplimentare. Printre acestea se numără introducerea unui sistem inteligent de recomandări, bazat pe comportamentul și preferințele utilizatorilor, care să stimuleze descoperirea de noi cărți. De asemenea, se pot dezvolta filtre avansate pentru căutare, opțiuni de feedback și rating pentru fiecare carte, precum și module dedicate administrării mai eficiente a utilizatorilor, stocurilor și comenzilor. O altă direcție valoroasă este

integrarea funcționalității de împrumut digital direct, ceea ce ar extinde utilitatea platformei dincolo de simpla comandă și achiziție.

În concluzie, proiectul reprezintă o contribuție relevantă în procesul de digitalizare a unei biblioteci, adaptându-se tendințelor actuale și aducând un plus de valoare atât instituției, cât și utilizatorilor acesteia. Provocările tehnice întâmpinate au fost oportunități importante de dezvoltare profesională, iar soluțiile implementate asigură o bază solidă pentru viitoare extinderi și adaptări la cerințele în continuă schimbare ale pieței și tehnologiei.



## Bibliografie

- Andreescu, A. și Rodica, M. (2009). *Dezvoltarea sistemelor software pentru managementul afacerilor*. București: Editura ASE.
- CERN (fără dată). *A short history of the web*. [online] Disponibil la: <https://home.cern/science/computing/birth-web/short-history-web>
- Eurostat. (2024). *Persons reading books in the last 12 months by sex, age, educational attainment and number of books*. [online] Disponibil la: [https://ec.europa.eu/eurostat/databrowser/product/view/ilc\\_scp27?lang=en](https://ec.europa.eu/eurostat/databrowser/product/view/ilc_scp27?lang=en)
- Express.js (fără dată). *Express.js Documentation*. [online] Disponibil la: <https://expressjs.com/>
- GeeksforGeeks (fără dată). *What is Single Page Application?* [online] Disponibil la: <https://www.geeksforgeeks.org/what-is-single-page-application/>
- HubSpot (2021). *Chartbeat's Website Engagement Data Reveals How Visitors Really Behave*. [online] Disponibil la: <https://blog.hubspot.com/marketing/chartbeat-website-engagement-data-nj>
- IBM (2021). *What is data modeling?* [online] Disponibil la: <https://www.ibm.com/think/topics/data-modeling>
- Kumar, K. and Kumar, M. (2020) *Digital Library Challenges and Opportunities: An Overview*. Disponibil la: <https://digitalcommons.unl.edu/libphilprac/5020/>
- Maeda, J. (fără dată). *Laws of Simplicity*. [online] Disponibil la: <https://lawsofsimplicity.com/>
- MindInventory / Rajput, M. (2021). *A Comprehensive Guide on Web Application Architecture*. [online] Disponibil la: <https://www.mindinventory.com/blog/web-application-architecture/>
- MongoDB (fără dată). *MongoDB Documentation*. [online] Disponibil la: <https://www.mongodb.com/docs/>
- Nielsen Norman Group, Whitenton, K. (2016). *Website Logo Placement for Maximum Brand Recall*. [online] Disponibil la: <https://www.nngroup.com/articles/logo-placement-brand-recall/>
- Norman, D.A. (2013). *The Design of Everyday Things*. [online] Disponibil la: <https://dl.icdst.org/pdfs/files4/4bb8d08a9b309df7d86e62ec4056ceef.pdf>
- Node.js (fără dată). *Node.js API Documentation*. [online] Disponibil la: <https://nodejs.org/docs/latest/api/>
- npm (fără dată). *bcrypt – npm*. [online] Disponibil la: <https://www.npmjs.com/package/bcrypt>
- npm (fără dată). *jsonwebtoken – npm*. [online] Disponibil la: <https://www.npmjs.com/package/jsonwebtoken>
- npm (fără dată). *npm Documentation*. [online] Disponibil la: <https://docs.npmjs.com/>

React Icons (fără dată). *React Icons Documentation*. [online] Disponibil la: <https://react-icons.github.io/react-icons/>

React Router (fără dată). *React Router Documentation*. [online] Disponibil la: <https://reactrouter.com/>

React Toastify, Fahim, F. (2024). *React Toastify Introduction*. [online] Disponibil la: <https://fkhadra.github.io/react-toastify/introduction/>

RESTful API (fără dată). *What is a RESTful API?* [online] Disponibil la: <https://restfulapi.net/>

Richardson, L. și Amundsen, M. (2013). *RESTful Web APIs*. [online] Disponibil la: [https://tdonker.nl/restapibooks/RESTful\\_Web\\_APIs.pdf](https://tdonker.nl/restapibooks/RESTful_Web_APIs.pdf)

Schwartz, B. (2004). *The Paradox of Choice: Why More Is Less*. [online] Disponibil la: [https://github.com/pankajb64/bookesis/blob/master/data/paradox\\_of\\_choice/The-Paradox-of-Choice-Barry-Schwartz.pdf](https://github.com/pankajb64/bookesis/blob/master/data/paradox_of_choice/The-Paradox-of-Choice-Barry-Schwartz.pdf)

Stripe (fără dată). *Get started with Stripe*. [online] Disponibil la: <https://docs.stripe.com/get-started>

Tailwind CSS (fără dată). *Tailwind CSS Documentation*. [online] Disponibil la: <https://v2.tailwindcss.com/docs>

Vite (fără dată). *Vite - Next Generation Frontend Tooling*. [online] Disponibil la: <https://vite.dev/guide/>

Visual Paradigm (fără dată). *Diagram Examples*. [online] Disponibil la: <https://circle.visual-paradigm.com/diagram-examples/>

Wikipedia (2025). *Biblioteca Batthyaneum*. [online] Disponibil la: [https://ro.wikipedia.org/wiki/Biblioteca\\_Batthyaneum](https://ro.wikipedia.org/wiki/Biblioteca_Batthyaneum)

Yale Law School (2007). *Exchanging Books in Western Europe*. [online] Disponibil la: [https://openyls.law.yale.edu/bitstream/handle/20.500.13051/17695/2007\\_Exchanging\\_Books\\_in\\_Western\\_Europe.pdf?sequence=2](https://openyls.law.yale.edu/bitstream/handle/20.500.13051/17695/2007_Exchanging_Books_in_Western_Europe.pdf?sequence=2)

## Anexe - Figuri

Figură 1.1 Evoluția numărului de biblioteci în România (1995-2023)	8
Figură 1.2 Digital Library Challenges and Opportunities: An Overview	10
Figură 1.3 Interfața catalogului digital pentru Biblioteca Națională a României	11
Figură 1.4 Interfața principală a site-ului pentru Biblioteca Metropolitană din București	12
Figură 2.1 - Diagrama cazurilor de utilizare a unei platforme online	17
Figură 2.2 - Diagrama de clase a unei platforme online de gestionare a unei biblioteci	18
Figură 2.3 - Diagrama de activitatea unei platforme online de gestionare a unei biblioteci	19
Figură 2.4 - Ierarhia unei platforme online de gestionare a unei biblioteci	21
Figură 2.5 - Schema bazei de date relaționale unei platforme de gestionare a unei biblioteci	23
Figură 2.6 - Diagrama de interacțiune pentru o platformă de gestionare a unei biblioteci	24
Figură 3.1 - Arhitectura unei aplicații web	26
Figură 3.2 - What is Single Page Application?	27
Figură 3.3 - Fragment de cod pentru gestionarea rutelor dinamice	27
Figură 3.4 - Fragment de cod pentru gestionarea rutelor imbricate	28
Figură 3.5 - Fragment de cod pentru stilizare, folosind Tailwind CSS	28
Figură 3.6 - Notificări Toastify afișate la o conectare/deconectare cu succes	28
Figură 3.7 - Cod sursă pentru gestionarea unui server	31
Figură 3.8 - Fragment de cod pentru funcția fetchAllProducts()	33
Figură 3.9 - Fragment de cod pentru summaryApi	33
Figură 3.10 - Interfața panoului administrativ unde sunt afișați utilizatorii	34
Figură 3.11 - Interfața panoului administrativ unde sunt afișate cărțile	34
Figură 3.12 - Controller pentru extragerea cărților	35
Figură 3.13 - Funcția fetchAllProducts() și handleAddToCart()	36
Figură 3.14 - Filtrarea cărților în funcție de categorie	36
Figură 3.15 - Segment de cod din funcția handlePayment()	37
Figură 3.16 - Controller pentru efectuarea de plăți	38
Figură 3.17 - Interfața paginii principale a aplicației	39
Figură 3.18 - Antetul/Header-ul aplicației	39

Figură 3.19 - Bara de căutare din antetul aplicației	40
Figură 3.20 - Set de cărți animate pentru a îmbunătăți designul interfeței	41
Figură 3.21 - Banner prezent pe pagina principală	41
Figură 3.22 - Secțiune de cărți populare	42
Figură 3.23 - Segmente din secțiunea „Despre noi”	43
Figură 3.24 - Segmente din secțiunea „Despre noi”	44
Figură 3.25 - Secțiunea de servicii oferite	44
Figură 3.26 - Formular de conectare	45
Figură 3.27 - Formular de înregistrare/creare cont digital	46
Figură 3.28 - Catalogul digital pentru biblioteca Studysphere	47
Figură 3.29 - Sistemul de filtrare în funcție de gen literar	48
Figură 3.30 - Pagină descriptivă pentru fiecare carte	49
Figură 3.31 - Secțiunea unde poate fi găsită varianta PDF a unei cărți	50
Figură 3.32 - Pagina unde pot fi vizualizate produsele adăugate în coșul de cumpărături	51
Figură 3.33 - Pagină pentru efectuarea plății la comenzile cu livrare la domiciliu	51
Figură 3.34 - Pagină pentru efectuarea tranzacției la comenzile cu ridicare personală	52
Figură 3.35 - Vizualizarea comenzilor efectuate	53

## Anexe – Tabele

Tabel 1: Cititori activi la biblioteci, grupați pe județe și localități

Rezultatele cautarii - Cititori activi la biblioteci pe judete si localitati			
Judete	Localitati	Perioade	
		Anul 2013	Anul 2023
		UM: Numar persoane	
		Numar persoane	Numar persoane
<b>TOTAL</b>	<b>TOTAL</b>	4004366	2419179

..... 8