O imagine care conține text, Font, captură de ecran, Grafică

Conținutul generat de inteligența artificială poate fi incorect.

# Simularea gestionării unei biblioteci în C

Student: Țăruș Adelina-Elena

Profesor coordonator: Asist. Ing. Olteanu Gabriela

București 2025

# Cuprins

[Motivare Alegere Temă 2](#_Toc198772292)

[Obiective 2](#_Toc198772293)

[Tehnologii Utilizate 2](#_Toc198772294)

[Studiu de caz – Tema aleasă 3](#_Toc198772295)

[Implementare pas cu pas 3](#_Toc198772296)

[Concluzie 8](#_Toc198772297)

[Bibliografie 8](#_Toc198772298)

## Motivare Alegere Temă

Am ales să dezvolt un sistem de gestiune a unei biblioteci într-o aplicație C, deoarece administrarea eficientă a cărților și a împrumuturilor este o necesitate practică în multe instituții de învățământ și biblioteci. Proiectul oferă o oportunitate bună de a aplica noțiuni fundamentale de programare, cum ar fi manipularea fișierelor, structuri, și gestionarea datelor în memorie.

## Obiective

În cadrul acestui proiect se propune implementarea unui sistem de gestiune a cărților și a împrumuturilor într-un mod interactiv, care permite utilizatorului să administreze eficient o colecție de cărți și evidența împrumuturilor. Interacțiunea cu utilizatorul se va realiza prin intermediul unui meniu intuitiv în linia de comandă, unde acesta poate adăuga cărți, căuta titluri, împrumuta și returna volume.

Pentru organizarea datelor într-un mod clar și eficient, se vor folosi structuri în limbajul C. Acestea vor reprezenta atât cărțile (cu informații precum ID, titlu și disponibilitate), cât și împrumuturile (cu detalii despre persoana care a împrumutat cartea și ID-ul acesteia).

Funcționalitățile esențiale ale aplicației includ: adăugarea unei cărți noi, căutarea după titlu, înregistrarea unui împrumut, precum și procesul de returnare a unei cărți. Fiecare acțiune actualizează starea internă a colecției și, acolo unde este cazul, și fișierul de evidență.

Datele relevante privind împrumuturile vor fi salvate într-un fișier text denumit imprumuturi.txt, pentru a păstra o evidență persistentă care poate fi consultată sau actualizată ulterior. Astfel, aplicația oferă o soluție simplă, dar funcțională, pentru gestionarea manuală a unei biblioteci de dimensiuni mici sau medii.

# Tehnologii Utilizate

-**Limbaj de programare: C**

C este un [limbaj de programare](https://ro.wikipedia.org/wiki/Limbaj_de_programare) [standardizat](https://ro.wikipedia.org/wiki/Standard). Este implementat pe majoritatea platformelor de calcul existente azi, și este cel mai popular limbaj de programare pentru scrierea de [software de sistem](https://ro.wikipedia.org/w/index.php?title=Software_de_sistem&action=edit&redlink=1).

C este un [limbaj de programare](https://ro.wikipedia.org/wiki/Limbaj_de_programare) relativ minimalist ce operează în strânsă legătură cu [hardware](https://ro.wikipedia.org/wiki/Hardware)-ul, fiind cel mai apropiat de [limbajul de asamblare](https://ro.wikipedia.org/wiki/Limbaj_de_asamblare) față de majoritatea celorlalte limbaje de programare.

Sintaxa limbajului C a stat la baza multor limbaje create ulterior și încă populare azi: [C++](https://ro.wikipedia.org/wiki/C%2B%2B), [Java](https://ro.wikipedia.org/wiki/Limbajul_Java), [JavaScript](https://ro.wikipedia.org/wiki/JavaScript), [C#](https://ro.wikipedia.org/wiki/C_Sharp), [D](https://ro.wikipedia.org/wiki/D_(limbaj_de_programare)).

Studiu de caz – Tema aleasă

## Implementare pas cu pas

1. Structuri definite:  
 - carte: ID, nume, autor, an, stoc

O imagine care conține text, Font, captură de ecran

Conținutul generat de inteligența artificială poate fi incorect.

- împrumut: ID carte, nume împrumutător, dată

O imagine care conține text, Font, captură de ecran

Conținutul generat de inteligența artificială poate fi incorect.

**Typedef struct** este o construcție în limbajul C care permite definirea unei structuri (o colecție de variabile de diferite tipuri) și atribuirea unui nume alias pentru acea structură. Acest alias simplifică utilizarea ulterioară a structurii, eliminând necesitatea de a scrie struct de fiecare dată.

2. Funcționalități principale:  
 - Adăugare carte – permite introducerea de noi cărți în sistem. O imagine care conține text, captură de ecran, Font, software

Conținutul generat de inteligența artificială poate fi incorect.

-**Strstr** este o funcție din **<string.h>** care caută prima apariție a unui șir de caractere (subșir) într-un alt șir de caractere.

-**fgets** este o funcție care citește un șir de caractere de la tastatură sau dintr-un fișier, inclusiv spațiile, până la un caracter **\n** sau până se atinge dimensiunea maximă specificată.

- Căutare carte – se poate căuta după nume, autor sau an.

O imagine care conține text, captură de ecran, Font

Conținutul generat de inteligența artificială poate fi incorect.

-**getchar** este o funcție din biblioteca standard C **(<stdio.h>)** care citește un singur caracter de la tastatură (din fluxul de intrare stdin).

- Împrumut carte – verifică stocul și salvează datele într-un fișier.

O imagine care conține text, captură de ecran, software, Font

Conținutul generat de inteligența artificială poate fi incorect.

O imagine care conține text, captură de ecran, Font, linie

Conținutul generat de inteligența artificială poate fi incorect.

- Returnare carte – crește stocul și elimină împrumutul din listă.

O imagine care conține text, captură de ecran, software, afișaj

Conținutul generat de inteligența artificială poate fi incorect.

- Afișare cărți și împrumuturi – listează toate înregistrările din sistem.

O imagine care conține text, captură de ecran, Font

Conținutul generat de inteligența artificială poate fi incorect.

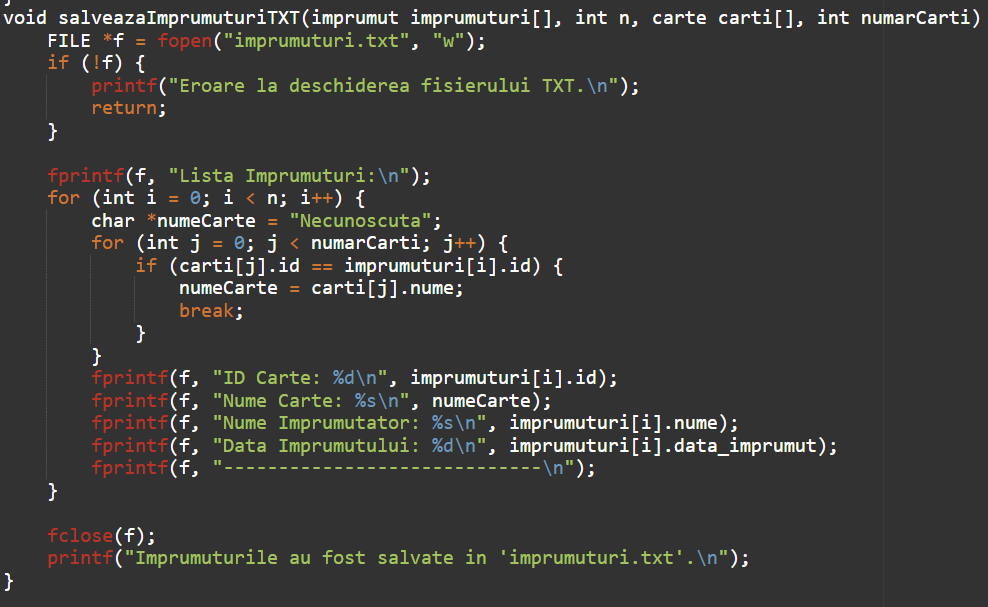
* Căutarea cărților – afișează cărțile disponibile

O imagine care conține text, captură de ecran, afișaj, software

Conținutul generat de inteligența artificială poate fi incorect.

- **strcspn** este o funcție din biblioteca **<string.h>** care returnează numărul de caractere de la începutul unui șir până la prima apariție a oricărui caracter dintr-un alt șir.

3. Fișier text (imprumuturi.txt):  
 - La fiecare împrumut, sistemul salvează automat informația într-un fișier pentru evidență ulterioară.



Folosirea **do-while** și **switch:**

**-do-while** este o structură de control care execută un bloc de cod cel puțin o dată, apoi repetă executarea atâta timp cât o condiție este adevărată.

**- switch** este o structură de control utilizată pentru a executa diferite blocuri de cod în funcție de valoarea unei expresii.

**O imagine care conține text, captură de ecran, Font

Conținutul generat de inteligența artificială poate fi incorect.**

# Concluzie

Proiectul a demonstrat eficiența limbajului C în gestionarea datelor prin structuri, dar și importanța modularității în dezvoltarea unui sistem funcțional. Interfața prin consolă este intuitivă, iar funcțiile acoperă un flux complet pentru o bibliotecă simplă. Se poate extinde cu funcționalități suplimentare precum autentificare sau interfață grafică.

# Bibliografie

[1]<https://ro.wikipedia.org/wiki/C_(limbaj_de_programare)#:~:text=C%20are%20urmatoarele%20caracteristici%20importante%3A%201%20Este%20un,date%20ce%20%C3%AEmpiedic%C4%83%20multe%20opera%C8%9Bii%20neinten%C8%9Bionate%20More%20items>

[2] <https://www.geeksforgeeks.org/c-program-for-hospital-management-system/>

[3]<https://github.com/Masjidul-60/Employee_payroll_system/blob/main/employee_Payroll_system.c>

[4] [CSA0405-Operating-Systems-Of-File-Systems-Implementation/16.for implementing random access file for processing the employee details..cpp at main · Jayaram192210314/CSA0405-Operating-Systems-Of-File-Systems-Implementation](https://github.com/Jayaram192210314/CSA0405-Operating-Systems-Of-File-Systems-Implementation/blob/main/16.for%20implementing%20random%20access%20file%20for%20processing%20the%20employee%20details..cpp)