

Universitatea Tehnică "Gheorghe Asachi" din Iași
Facultatea de Automatică și Calculatoare
Domeniul Calculatoare și Tehnologia Informației
Specializarea Tehnologia Informației

ECBD
Sistem de Gestiune a unei Biblioteci

Student:
Martin Maria, 1410A

Cadru didactic coordonator:
ș.l. dr.ing. Mironeanu Cătălin

An universitar 2023-2024

Cuprins

1. Descrierea proiectului.....	3
2. Structura colecțiilor.....	3
2.1 Colecția carte.....	3
2.2 Colecția membru.....	4
2.3 Colecția bibliotecar.....	5
2.4 Colecția imprumuturi.....	5
3. Descrierea funcționalităților și a operațiilor corespunzătoare din fișierele de script.....	6
3.1 Operații CRUD.....	6
3.2 Căutări simple.....	7
3.3 Căutări complexe.....	8

1. Descrierea proiectului

Baza de date “*proiectBiblioteca*” are ca scop oferirea unui suport de stocare a datelor și furnizarea unor funcții de operare și manevrare a acestor date (de tip CRUD, agregare, căutări complexe) pentru gestionarea unei biblioteci. Aceasta permite bibliotecarilor să gestioneze colecția de cărți, membrii bibliotecii, precum și împrumuturile de cărți.

Funcționalitățile principale implementate sunt:

- **Gestionarea cărților:** Adăugarea de noi cărți în colecție, modificarea informațiilor despre cărți existente, ștergerea cărților și vizualizarea detaliilor despre cărți.
- **Gestionarea membrilor:** Înregistrarea de noi membri, actualizarea informațiilor despre membri, ștergerea membrilor și vizualizarea profilurilor membrilor.
- **Gestionarea împrumuturilor:** Înregistrarea împrumuturilor de cărți, actualizarea detaliilor împrumuturilor și gestionarea returnării cărților.
- **Gestionarea bibliotecarilor:** Înregistrarea de noi bibliotecari, actualizarea informațiilor despre bibliotecari, ștergerea bibliotecarilor și vizualizarea profilurilor membrilor.

2. Structura colecțiilor

2.1 Colecția carte

Această colecție este responsabilă cu stocarea datelor despre cărțile din bibliotecă. Fiecare document din această colecție reprezintă o carte și conține informațiile necesare pentru identificarea și gestionarea acesteia.

Structura unui document din colecția “*carte*” este următoarea:

- **_id:** ObjectId()
- **titlu:** String
- **autor:** String
- **gen:** [String]
- **an_publicare:** Integer
- **isbn:** String
- **disponibil:** Boolean
- **detalii_carte:** { limba: String, numar_pagini: Integer, descriere: String }

Proprietăți:

- **_id:** Identificator unic generat automat pentru fiecare carte.
- **titlu:** Titlul cărții.
- **autor:** Autorul cărți.
- **gen:** Un vector de genuri literare.
- **an_publicare:** Anul în care a fost publicată cartea.
- **isbn:** Codul ISBN al cărții, un identificator unic pentru fiecare carte.

- **disponibil**: Disponibilitatea cărții (true dacă este disponibilă pentru împrumut, false dacă este împrumutată).
- **detalii_carte**: Un subdocument care conține detalii adiționale despre carte:
 - **limba**: Limba în care este scrisă cartea.
 - **numar_pagini**: Numărul de pagini ale cărții.
 - **descriere**: O mică descriere a conținutului cărții.

2.2 Colecția membru

Această colecție este responsabilă cu stocarea datelor despre membrii bibliotecii. Fiecare document din această colecție reprezintă un membru al bibliotecii și conține informațiile necesare pentru identificarea acestuia.

Structura unui document din colecția “*membru*” este următoarea:

- **_id**: ObjectId()
- **nume_membru**: { nume: String, prenume: String }
- **varsta**: Integer
- **email**: String
- **telefon**: String
- **adresa**: { strada: String, numar: Integer, oras: String }
- **data_inscriere**: Date
- **carti_imprumutate**: [{ carte_id: ObjectId, titlu: String, data_imprumut: Date, data_returnare: Date }]

Proprietăți:

- **_id**: Identificator unic generat automat pentru fiecare membru.
- **nume_membru**: Un subdocument care conține numele și prenumele membrului.
 - **nume**: Numele de familie al membrului.
 - **prenume**: Prenumele membrului.
- **varsta**: Vârsta membrului.
- **email**: Adresa de email a membrului.
- **telefon**: Numărul de telefon al membrului.
- **adresa**: Un subdocument care conține adresa membrului.
 - **strada**: Numele străzii.
 - **numar**: Numărul străzii.
 - **oras**: Orașul.
- **data_inscriere**: Data înregistrării membrului în bibliotecă.
- **carti_imprumutate**: Un vector de subdocumente care conține detalii despre cărțile împrumutate de către membru:
 - **carte_id**: Identificatorul cărții împrumutate.
 - **titlu**: Titlul cărții împrumutate.
 - **data_imprumut**: Data împrumutului.
 - **data_returnare**: Data returnării.

2.3 Colecția bibliotecar

Această colecție este responsabilă cu stocarea datelor despre bibliotecari. Fiecare document din această colecție reprezintă un bibliotecar și conține informațiile necesare pentru identificarea acestuia.

Structura unui document din colecția “*bibliotecar*” este următoarea:

- `_id`: ObjectId()
- `nume_bibliotecar`: { `nume`: String, `prenume`: String }
- `varsta`: Integer
- `email`: String
- `telefon`: String
- `adresa`: { `strada`: String, `numar`: Integer, `oras`: String }
- `data_angajare`: Date
- `responsabilitati`: [String]

Proprietăți:

- **`_id`**: Identificator unic generat automat pentru fiecare bibliotecar.
- **`nume_bibliotecar`**: Un subdocument care conține numele și prenumele bibliotecarului.
 - **`nume`**: Numele de familie al bibliotecarului.
 - **`prenume`**: Prenumele bibliotecarului.
- **`varsta`**: Vârsta bibliotecarului.
- **`email`**: Adresa de email a bibliotecarului.
- **`telefon`**: Numărul de telefon al bibliotecarului.
- **`adresa`**: Un subdocument care conține adresa bibliotecarului.
 - **`strada`**: Numele străzii.
 - **`numar`**: Numărul străzii.
 - **`oras`**: Orașul.
- **`data_angajare`**: Data angajării bibliotecarului.
- **`responsabilitati`**: Un vector de responsabilități ale bibliotecarului, cum ar fi "împrumuturi", "returnari", "gestionare".

2.4 Colecția împrumuturi

Această colecție este responsabilă cu stocarea datelor despre împrumuturile de cărți. Fiecare document din această colecție reprezintă un împrumut și conține informațiile necesare pentru identificarea și gestionarea acestuia.

Structura unui document din colecția “*împrumuturi*” este următoarea:

- `_id`: Integer
- `membru_id`: ObjectId()
- `bibliotecar_id`: ObjectId()
- `carti`: [{ `carte_id`: ObjectId(), `data_imprumut`: Date, `data_returnare`: Date }]

Proprietăți:

- **_id**: Identificator unic generat automat pentru fiecare împrumut.
- **membru_id**: Referință la membrul care a împrumutat cartea, utilizat pentru a lega împrumutul de membru.
- **bibliotecar_id**: Referință la bibliotecarul care a gestionat împrumutul, utilizat pentru a lega împrumutul de bibliotecar.
- **carti**: Un vector de subdocumente care conține detalii despre cărțile împrumutate:
 - **carte_id**: Identificatorul cărții împrumutate.
 - **data_imprumut**: Data împrumutului.
 - **data_returnare**: Data returnării.

3. Descrierea funcționalităților și a operațiilor corespunzătoare din fișierele de script

3.1 Operații CRUD

Operațiile CRUD (Create, Read, Update, Delete) sunt esențiale pentru manipularea datelor din colecțiile utilizate în aplicație. Acestea permit adăugarea, citirea, actualizarea și ștergerea documentelor din colecțiile “*carte*”, “*membru*”, “*bibliotecar*” și “*imprumuturi*”.

În scriptul *script_a.js* s-a realizat inserarea documentelor în colecțiile “*carte*”, “*membru*”, “*bibliotecar*” și “*imprumuturi*”. Inserarea s-a efectuat printr-o singură comandă `insertMany`, pentru fiecare colecție, permițând adăugarea mai multor documente în cadrul unei singure operațiuni.

În scriptul *script_b.js* s-au realizat operațiile de actualizare și ștergere a documentelor din colecțiile “*carte*”, “*membru*”, “*bibliotecar*” și “*imprumuturi*”. Scriptul utilizează funcții precum `updateOne`, `updateMany`, `deleteOne` și `deleteMany` pentru a efectua aceste operațiuni.

Operații pentru colecția “*carte*”:

- **Update ISBN pentru carte**: modifică codul ISBN pentru cartea cu titlul "Crime și pedeapsa".
- **Delete carte după ISBN**: șterge cartea care are codul ISBN specificat.

Operații pentru colecția “*membru*”:

- **Update adresă membru după nume**: modifică adresa membrului cu numele specificat.
- **Delete membri după vârstă**: șterge membrii care au vârsta între 25 și 40 de ani.

Operații pentru colecția “*bibliotecar*”:

- **Update număr de telefon pentru bibliotecar**: modifică numărul de telefon al bibliotecarului cu ID-ul specificat.
- **Update responsabilități pentru bibliotecari cu vârsta > 40**: adaugă responsabilitatea "creare fise membri" pentru toți bibliotecarii care au vârsta mai mare de 40 de ani.

- **Update pentru adăugarea unei responsabilități noi unui bibliotecar:** adaugă responsabilitatea "împrumuturi" unui bibliotecar specific.
- **Delete bibliotecar după email:** șterge bibliotecarul care are adresa de email specificată.

Operații pentru colecția “*împrumuturi*”:

- **Update pentru schimbarea bibliotecarului într-un împrumut:** modifică bibliotecarul responsabil pentru împrumutul cu ID-ul 8.
- **Delete împrumuturi pentru un anumit membru:** șterge toate împrumuturile care aparțin unui membru specific.

3.2 Căutări simple

Scriptul *script_c.js* conține diverse căutări simple ce demonstrează utilizarea funcției *find()*.

Operații pentru colecția “*carte*”:

- **Căutare cărți după gen și sortare după titlu:** returnează toate cărțile din genul “*Fictiune*” și le sortează crescător după titlu.
- **Căutare cărți după disponibilitate și ordonare descrescătoare după anul publicării:** returnează cărțile disponibile, sortate descrescător după anul publicării.
- **Căutare cărți care au fost publicate înainte de 1900 și ordonare după anul publicării:** returnează cărțile publicate înainte de anul 1900, sortate crescător după anul publicării.
- **Paginare: afișare primele 3 cărți:** returnează primele trei cărți din colecție.
- **Căutare cărți care sunt publicate înainte de 1900 sau după 1950:** returnează cărțile publicate înainte de 1900 sau după 1950.

Operații pentru colecția “*membru*”:

- **Căutare membri care au număr de telefon:** returnează membrii care au numărul de telefon specificat.
- **Căutare membri cu vârsta ≥ 20 și ≤ 30 și sortare după vârstă:** returnează membrii cu vârsta între 20 și 30 de ani, sortate crescător după vârstă.
- **Căutare membri din Constanța cu vârsta > 20 :** returnează membrii din orașul Constanța cu vârsta mai mare de 20 de ani.
- **Căutare membri care au împrumutat cel puțin o carte și au număr de telefon specificat:** returnează membrii care au împrumutat cel puțin o carte și au număr de telefon specificat.

Operații pentru colecția “*bibliotecar*”:

- **Căutare bibliotecar după oraș:** returnează bibliotecarii din orașul Cluj-Napoca.
- **Căutare bibliotecar după responsabilitate:** returnează bibliotecarii care au responsabilitatea de împrumuturi.
- **Căutare bibliotecar fără email:** returnează bibliotecarii care nu au un email specificat.
- **Sortare după vârstă:** returnează toți bibliotecarii, sortați crescător după vârstă.

Operații pentru colecția “*împrumuturi*”:

- **Căutare împrumut după carte:** returnează împrumuturile care conțin o anumită carte.
- **Căutare împrumuturi fără cărți:** returnează împrumuturile care nu conțin nicio carte.
- **Paginare: afișare ultimele 2 împrumuturi:** returnează ultimele două împrumuturi din colecție.
- **Căutare împrumuturi cu data de returnare \geq decât cea specificată:** returnează împrumuturile care au cărți cu data de returnare mai mare sau egală cu o dată specificată.

3.3 Căutări complexe

Scriptul *script_d.js* conține operații complexe realizate asupra colecțiilor. Aceste operații includ utilizarea cursorilor pentru procesări suplimentare și utilizarea framework-ului aggregate pentru a obține rezultate detaliate și structurate din mai multe colecții.

Operații pentru colecția “*carte*”:

- **Update pentru cărțile care nu sunt disponibile (data retur > data curentă):** folosește un cursor pentru a găsi toate împrumuturile care au cărți cu data returnării mai mare decât data curentă și setează disponibilitatea acestor cărți la *false*.

Operații pentru colecția “*membru*”:

- **Update la membru, adăugare titlu cărților împrumutate în vectorul de cărți împrumutate:** parcurge toate împrumuturile și adaugă titlul cărților împrumutate în vectorul de cărți împrumutate din documentele membrilor.

Operații pentru colecția “*bibliotecar*”:

- **Ștergere responsabilitate "returnari" pentru bibliotecarii care fac și returnări și gestionare:** folosește un cursor pentru a găsi bibliotecarii care au atât responsabilitatea "returnari" cât și "gestionare", și elimină responsabilitatea "returnari" din lista lor.

Operații pentru colecția “*împrumuturi*”:

- **Extragerea numărului de împrumuturi pentru fiecare membru:** folosește framework-ul aggregate pentru a grupa împrumuturile pe baza membru_id, calculând numărul de împrumuturi pentru fiecare membru. Rezultatele sunt sortate descrescător după numărul de împrumuturi, iar detaliile membrilor sunt obținute folosind operatorul \$lookup pentru a face legătura cu colecția membru. Rezultatele sunt proiectate pentru a afișa numele membrului și numărul de împrumuturi.