

## Lucrarea 10

### Proiect web dinamic *Spring Boot*

Să se implementeze un proiect web dinamic *Spring Boot* care permite realizarea operațiilor *CRUD* asupra unei tabele *carti* din baza de date *bd\_lab10*. Pentru afișarea datelor în front-end se va utiliza artefactul *Thymeleaf*. Operațiile asupra bazei de date se vor realiza utilizând *Spring Data JPA*. Pentru fiecare carte se va reține *isbn*-ul (cheia primară), *titlul* și *autorul*. Aplicația va avea un aspect asemănător cu cel din figura 1:

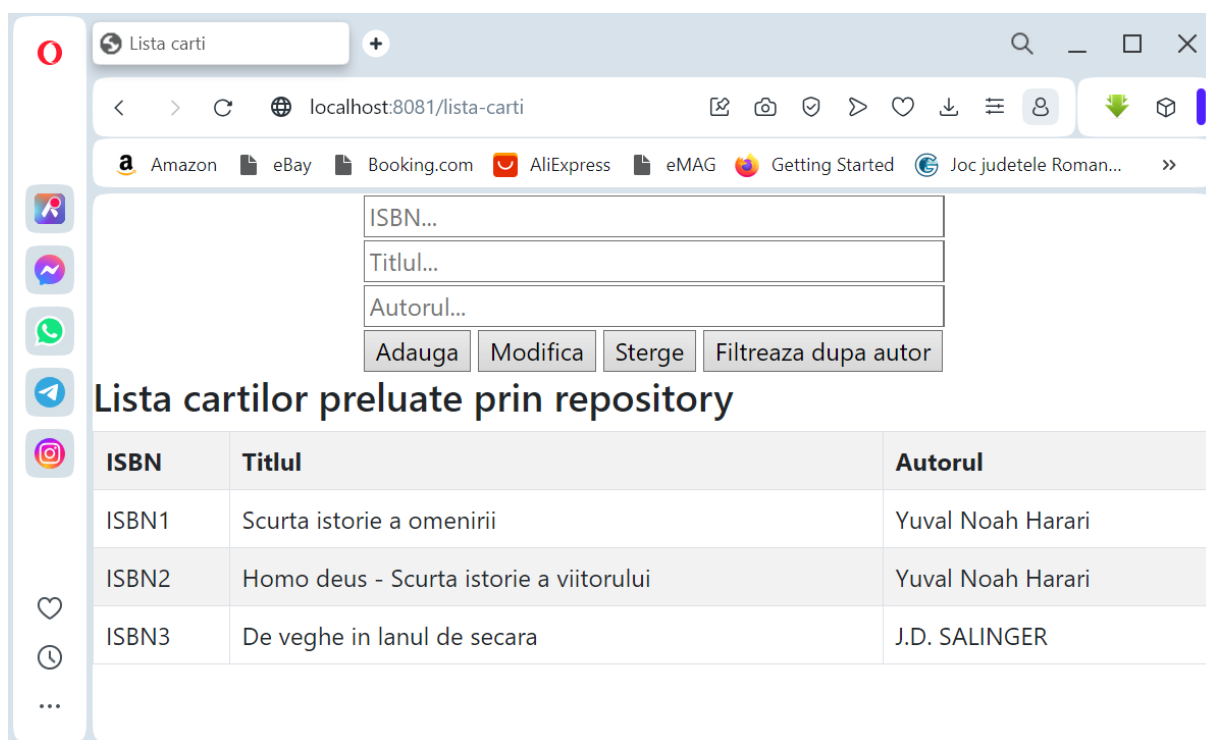


Fig. 1- front-end-ul aplicației

Proiectul va fi configurat încât baza de date *bd\_lab10* să se creeze prin program, la fel și tabela *carti*.

Se va crea fișierul *src/main/resources/data.sql* care asigură precompletarea tabelii *carti*, imediat după ce aceasta a fost creată, cu datele cărților din figura 1.

Acționarea butoanelor va fi însoțită de mesaje care vor fi afișate în locul mesajului *Lista cartilor preluate prin repository*.

Adăugarea se va realiza doar dacă au fost introduse toate caracteristicile cărții și va fi însoțită de unul din mesajele: *Adaugare realizata cu succes!* sau *Adaugarea nu se realizează dacă nu completați toate caracteristicile!*

Operația de modificare presupune căutarea după ISBN a cărții care se dorește să fie modificată. Dacă aceasta nu se găsește va fi afișat mesaj-ul: *Nu se gasete nici o carte cu isbn-ul introdus*. Dacă se găsește cartea, aceasta se modifică în acord cu valorile introduse în

interfață și se va afișa un mesaj care va indica reușita operației: *Cartea cu ISBN-ul ISBN1 a fost modificata!*

Ștergerea se realizează utilizând doar informația completată în câmpul *ISBN*, și va fi însoțită de mesaje corespunzătoare care indică succesul sau eșecul operației.

Filtrarea după autor va afișa toate cărțile autorului al cărui nume a fost precizat în caseta de text și va fi însoțită de mesaje de genul: *Cărțile următoare aparțin autorului Yuval Noah Harari*. Actionarea butonului de filtrare fără a completa numele vreunui autor va determina afișarea tuturor cărților, însoțită de un mesaj corespunzător.

## ***Indicații de rezolvare***

Suportul teoretic pentru rezolvarea lucrării de laborator, însoțit de exemple ajutătoare se găsește în curs, capitolul 6. Spring Boot, subcapitolul 6.14 Dezvoltarea unui proiect web dinamic cu front-end HTML + Thymeleaf.

Se va genera un proiect **Spring Boot** utilizând **Spring Initializr** <https://start.spring.io>.

Proiectul va utiliza limbajul de programare Java, versiunea 17 și gestiunea proiectului va fi asigurată de Maven.

Proiectul va utiliza dependențele de mai jos:

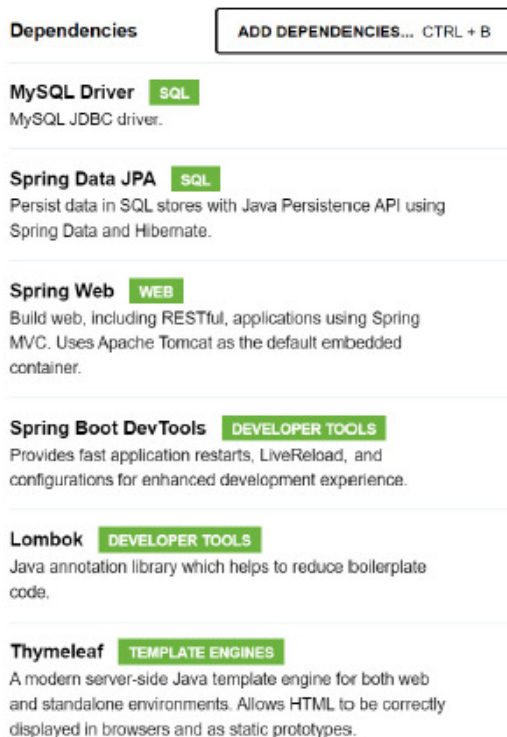


Fig. 2 – dependențele proiectului

Se va dezvolta clasa persistentă (entitatea) *Carte* care se mapează la tabela *MySQL* *carti*. Se va dezvolta repository-ul *CarteRepository* care este un repository *Spring Data JPA*. Acesta oferă acces la date și operează asupra entității *Carte* oferind operațiile CRUD.

Se va dezvolta controllerul *CarteWebController* va gestiona cererile HTTP lansate de client (browser).

Controller-ul va răspunde la cerile de tip GET lansate de client (browser) către URL-ul <http://localhost:8081/lista-carti> trimițând către browser pagina *carti.html*.

Documentul *carti.html* se creează în directorul *src/main/resources/templates* și utilizează sablonul *Thymeleaf* care îi asigură dinamică și permite afișarea datelor din Java în *html*.

Documentul *carti.html* are în partea de început un formular *HTML*, care utilizează metoda *POST* pentru a trimite date către o metodă a controllerului *CarteWebController*.

```
<form method="post" action="operatii">

<input type="text" name="isbn" placeholder="ISBN..." size="46"/>
<!--...-->

<input type="submit" name="adauga" value="Adauga"/>
<!--...-->

</form>
```

Datele din formular pot să fie accesate în metoda apelată din controller, prin parametrii de intrare ai acestei metode. Parametrii de intrare ai metodei trebuie să aibă aceleași denumiri ca și componentele *html*.

În metoda apelată din controller, se poate determina dacă un anumit buton a fost apăsat analizând valoarea pe care o are parametrul de intrare al metodei, care corespunde butonului. Acest parametru de intrare este un *String* care este *null* când butonul nu este apăsat și are valoarea egală cu textul de pe buton când acesta este apăsat.

Controller-ul va răspunde la cererea *POST* lansată către metodă, trimițând către browser pagina *carti.html* actualizată cu datele și mesajele corespunzătoare, rezultate în urma efectuării operației solicitate.

Mesajele se vor afișa sub formular, iar datele cărților vor fi afișate într-un tabel care va fi stilizat utilizând *Bootstrap*: <https://getbootstrap.com/docs/5.1/content/tables/>.