**Descrierea Proiectului**

**Proiect:** *Catalog de Cărți*

**Descriere:**  
Acest proiect este o aplicație web single-page (SPA) care permite utilizatorilor să gestioneze un catalog de cărți. Utilizatorii pot adăuga cărți printr-un formular, pot vizualiza cărțile într-o listă sub formă de carduri, pot vizualiza detaliile complete ale unei cărți, iar, de asemenea, pot șterge o carte din catalog. Aplicația este dezvoltată folosind **React**, o bibliotecă JavaScript pentru construirea interfețelor de utilizator.

**Tehnologii utilizate:**

* **React** - pentru construirea aplicației frontend.
* **HTML** și **CSS** - pentru structura și stilizarea interfeței utilizatorului.
* **JavaScript** (ES6+) - pentru manipularea stării aplicației și interactivitatea acesteia.

**Descrierea Componentelor Aplicației**

Aplicația este structurată în mai multe componente care interacționează între ele pentru a furniza funcționalitatea completă. Iată o descriere a componentelor și a modului în care interacționează:

1. **BookForm**:
   * **Rol:** Permite utilizatorilor să adauge o carte în catalog prin completarea unui formular.
   * **Proprietăți:**
     + onAddBook: o funcție care este apelată atunci când utilizatorul trimite formularul, pentru a adăuga cartea în lista de cărți.
   * **Interacțiune:** După completarea formularului și trimiterea acestuia, datele sunt trimise către componenta părinte (App) printr-o funcție de callback (onAddBook), care adaugă cartea în starea aplicației.
2. **BookCard**:
   * **Rol:** Afișează informațiile unei cărți (titlu, autor, gen, descriere) sub formă de card.
   * **Proprietăți:**
     + book: un obiect care conține informațiile cărții.
     + onDelete: o funcție care permite ștergerea cărții.
   * **Interacțiune:** Componenta permite ștergerea cărții printr-un buton de ștergere, apelând funcția onDelete din componenta părinte.
3. **BookList**:
   * **Rol:** Afișează o listă de cărți sub formă de carduri.
   * **Proprietăți:**
     + books: o listă cu obiectele cărților.
     + onDelete: o funcție care permite ștergerea cărților.
   * **Interacțiune:** Iterează prin lista de cărți și pentru fiecare carte apelează componenta BookCard pentru a o vizualiza.
4. **BookDetails**:
   * **Rol:** Afișează informații complete despre o carte selectată.
   * **Proprietăți:**
     + book: un obiect care conține informațiile complete ale cărții selectate.
   * **Interacțiune:** Componenta afișează detalii suplimentare atunci când utilizatorul face clic pe un card din BookList. Se actualizează cu informațiile cărții selectate.
5. **App**:
   * **Rol:** Componenta principală a aplicației care gestionează starea și interacțiunile între componente.
   * **Stare:**
     + books: o listă de cărți adăugate.
     + selectedBook: cartea selectată pentru a vizualiza detalii.
   * **Interacțiune:** App gestionează adăugarea și ștergerea cărților, selectarea cărții pentru vizualizarea detaliilor, și transmite aceste funcționalități către componentele corespunzătoare.

**Gestionarea Validării și Trimiterii Datelor**

1. **Validarea datelor**:
   * **Validarea la nivel de formular:**
     + Atunci când utilizatorul completează formularul pentru a adăuga o carte, câmpurile title și author sunt obligatorii. Validarea este efectuată la nivelul formularului, folosind atributul HTML required pentru câmpurile respective.
     + Dacă aceste câmpuri sunt goale, formularul nu va permite trimiterea datelor.
   * **Validarea pe frontend:** Deși în acest proiect validarea este minimală, în aplicațiile mai complexe, validarea suplimentară ar putea fi implementată pentru a verifica, de exemplu, lungimea textului sau formatul acestuia (pentru descriere sau gen).
2. **Trimiterea datelor:**
   * **Adăugarea cărții:** După completarea formularului, datele sunt trimise printr-o funcție de callback (onAddBook) către componenta părinte (App). Aici, datele sunt adăugate în lista de cărți (starea books).
   * **Ștergerea unei cărți:** Atunci când utilizatorul apasă pe butonul de ștergere, funcția onDelete este apelată și cartea respectivă este eliminată din lista de cărți.
3. **Gestionarea stării:**
   * În React, starea este gestionată folosind hook-ul useState. Starea aplicației (lista de cărți și cartea selectată) este centralizată în componenta App, care este responsabilă de propagarea acesteia către celelalte componente.

Astfel, validarea și trimiterea datelor sunt gestionate prin interacțiunea dintre componentele de formular și componenta principală (App), iar modificările de stare sunt realizate folosind React hooks.