

Universitatea Tehnică din Cluj-Napoca

Catedra de Calculatoare și Tehnologia Informației

Sistem de management a cărților

Student: Trifu Diana Maria

Coechipier: Ilieș Gabriela

Grupa: 30233

Îndrumător de proiect: Eneia Nicolae Todoran

Data: 4 ianuarie 2021

MINISTERUL EDUCAȚIEI NAȚIONALE



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

CUPRINS

Rezumat	3
Miniproiect	4
Diagrama de Usecase-uri.....	5
Randarea. Relația „părinte-copil” între componente.....	6
Diagrama de implementare.....	7
Implementare	8
Rezultate experimentale	9
Bibliografie	14
Anexe	15

Rezumat

Acest proiect a avut ca scop dezvoltarea unei aplicații web de recomandare a cărților pentru utilizatori, pe baza rating-ului acordat cărților citite. Această aplicație îi permite utilizatorului să caute cărți după titlul acesteia, să aduge cărți în WishList, sau într-o listă care conține cărțile pe care acesta le citește în prezent. Totodată, acesta va avea posibilitatea de a adăuga cărțile pe care le-a terminat de citit într-o listă separată(ReadBooks). Cărțile pe care acesta le are în WishList le poate adăuga în lista de cărți active (ActiveList). Utilizatorului îi este permis să acorde și rating, reprezentat printr-o notă de la 1 la 5, dar doar cărților pe care acesta le-a citit și le-a adăugat în ReadList. Pentru a face posibile toate aceste operațiuni acesta trebuie să facă log in prin intermediul unui username și a unei parole.

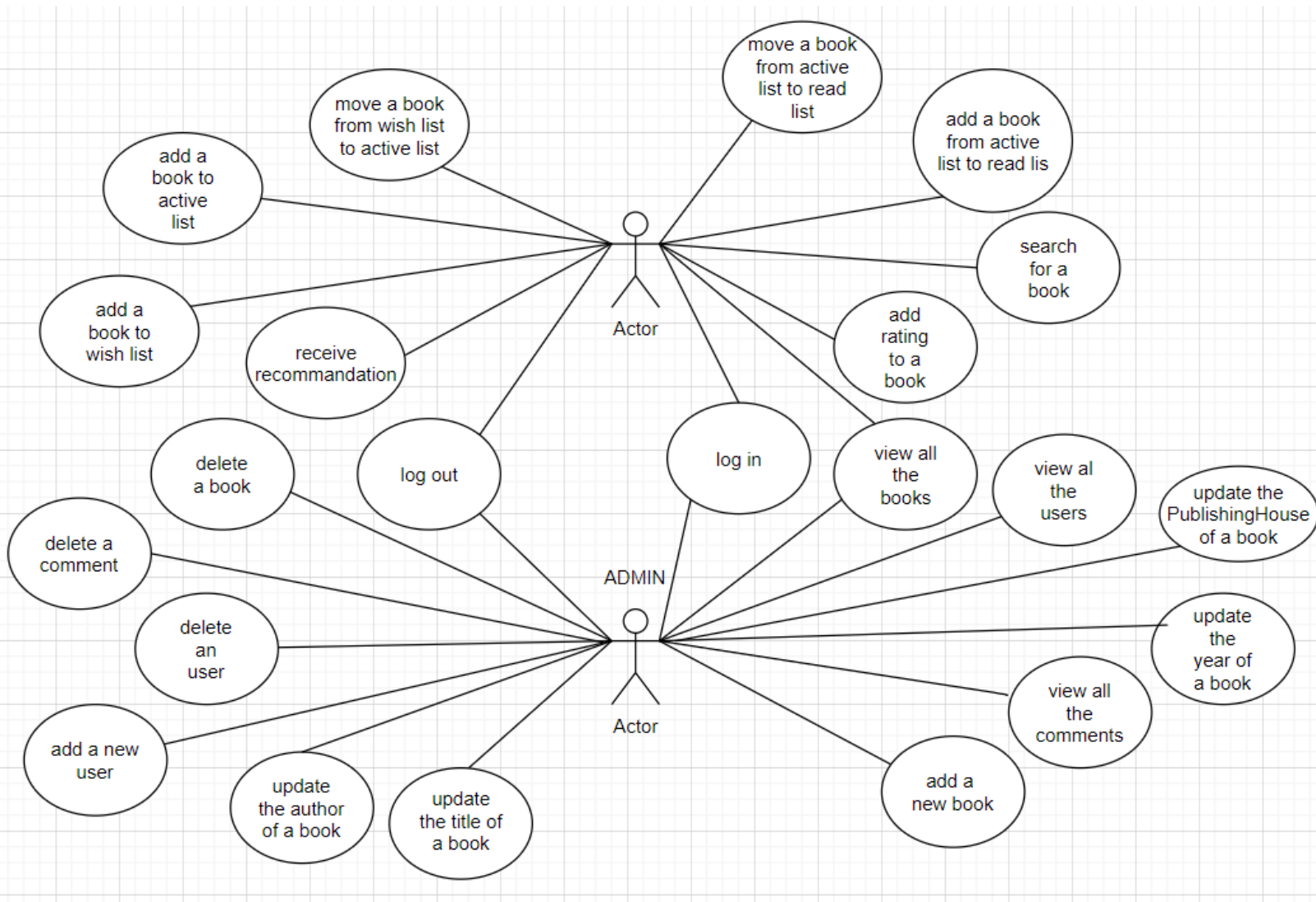
Accesul la această aplicație are și adminul aplicației care poate vedea toți utilizatorii, toate cărțile și toate comentariile. De asemenea, poate adăuga cărți noi în sistem, noi utilizatori, dar poate să îi și șteargă. Totodată, acesta poate să modifice informațiile cărților deja existente. În cazul în care acestuia i se par anumite comentarii nepotrivite în cadrul aplicației, acesta le poate șterge. Ca și utilizatorul, și acesta trebuie să se înregistreze cu un username și o parolă pentru a i se permite accesul spre administrarea aplicației.

Această aplicație a fost realizată cu ajutorul următoarelor limbaje de programare și tehnologii: React JS, C#, ASP.net.

Miniproiect

Inițial, miniproiectul, a reprezentat un mic demo care demonstrează legătura dintre cele 3 componente principale ale proiectului: Model, View, Controller. Pentru acest demo s-a folosit biblioteca HTTP Axios pentru a face posibile request-urile către server, pentru preluarea datelor. Pentru a demonstra legătura dintre partea de View(React.JS, HTML, CSS, JavaScript) și cele de Model și Controller (C#, ASP.net) am construit un buton care, odată apăsător afișează toate cărțile care se află în fișierul în care sunt serializate toate informațiile: cărțile(id, titlu, autor, editura, anul apariției, comentariile și rating-ul), utilizatorii(id, , parolă, variabila de determinare a acceselor(admin sau user) și listele sale de cărți: WishList, ReadList și ActiveList. Pentru a face posibilă afișarea tuturor titlurilor de cărți s-a făcut request către server pentru metoda corenspozătoare, preluându-se datele într-un șir. Mai apoi din acest șir de obiecte de tip carte se preiau titlurile acestea și se afișează prin intermediul unei liste neordonate. Toate aceste operații sunt declanșate de evenimentul "onClick", la apăsarea butonului respectiv.

Diagrama de Usecase-uri



Randarea. Relația „părinte-copil” între componente

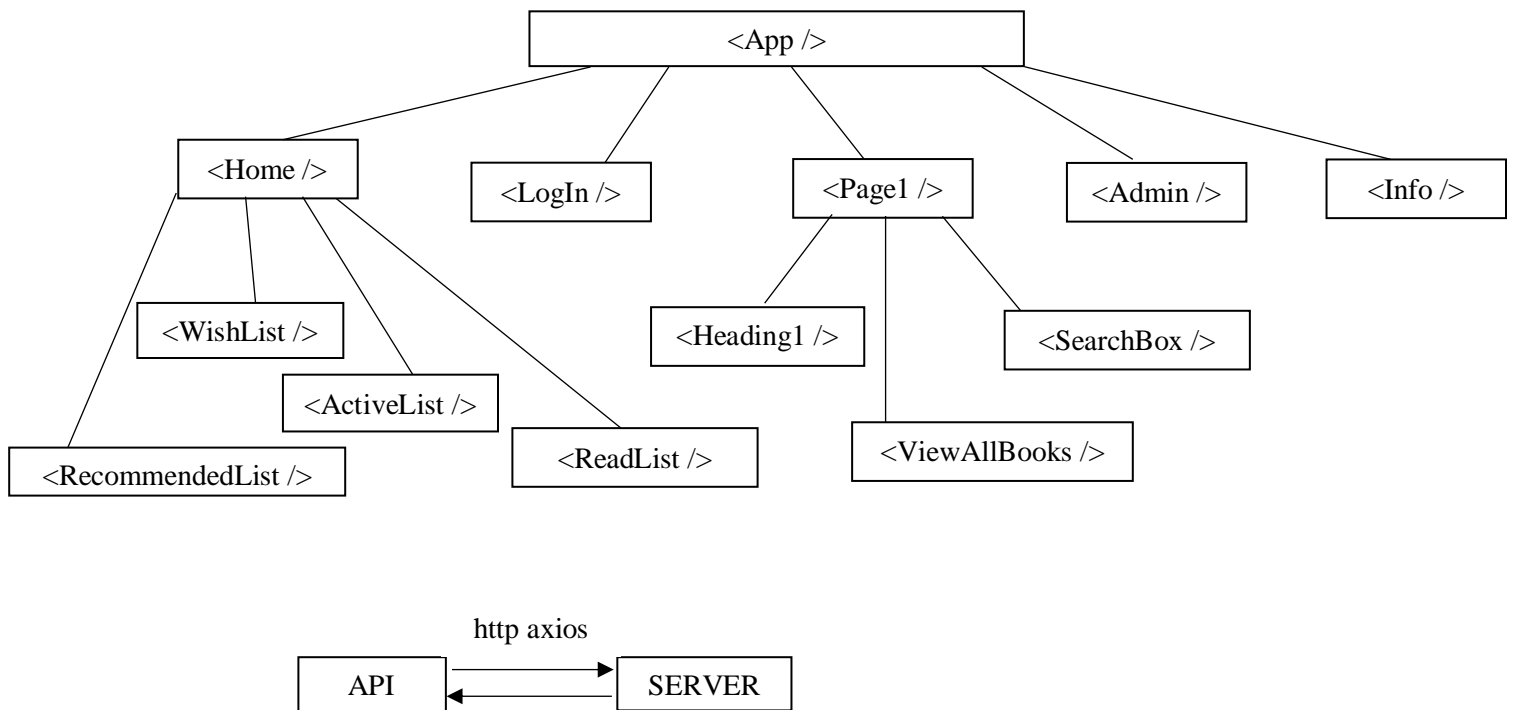
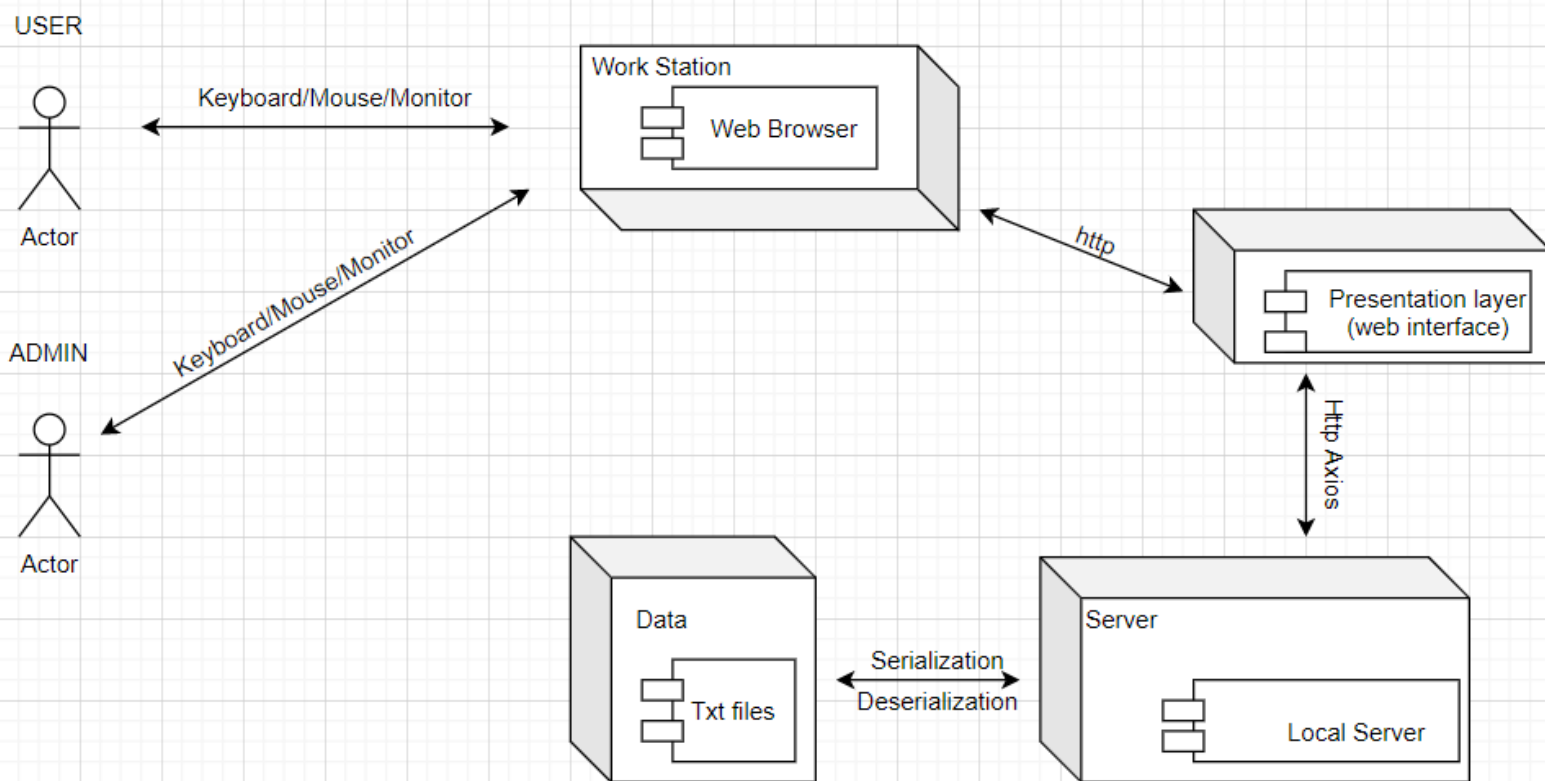


Diagrama de implementare



Implementare

Componente:


1. **LogIn.js**: când această componentă este randată în componenta principală „App.js”, în browser se va randa pagin de autentificare pentru oricare dintre cele două tipuri de actori: utilizator și administrator.
2. **Page1.js**: această componentă randează la rândul lui alte 3 componente(Header1.js, SearchBox.js, ViewAllBooks.js) pentru a forma împreună pagina de „Home” a utilizatorului.
3. **Header1.js**: această componentă alcătuiește antetul paginii de „Home”: bara cu butonul de „Log out” și cel care îl redirecționează pe utilizator spre pagina contului său.
4. **SearchBox.js**: componenta aceasta reprezintă textfield-ul și butonul de căutare, prin intermediul cărora utilizatorul poate căuta cărți în funcție de cuvinte care fac parte din titlul acestora.
5. **ViewAllBooks.js**: această componentă contribuie cu afișarea tuturor cărților aflate în sistem(titlul acestora), însoțite de două butoane care oferă posibilitatea autorului de a adăuga cartea respectivă în una dintre următoarele două liste: ActiveList(cărțile pe care le citește în prezent), sau WishList(cărțile care și-ar dori să le citească).
6. **Home.js**: această componentă este asemănătoare cu „Page1” care ajută la împreunarea altor componente pentru a forma contul utilizatorului. Cele 4 componente sunt: Wishlist.js, ActiveList.js, ReadList.js și RecommendedBooks.js.
7. **RecommendedBooks.js**: această componentă preia cărțile recomandate utilizatorului respectiv și le afișează într-o listă pentru a fi vizibile acestuia în contul său.
8. **ActiveList.js**: această componentă preia cărțile pe care utilizatorul le-a marcat ca fiind în curs de citire și le afișează într-o listă pentru a fi vizibile acestuia în contul său. Fiecare titlu de carte din această listă este însoțit cu un buton care oferă posibilitatea adăugării cărții respective în lista cărților citite: ReadList.
9. **ReadList.js**: această componentă preia cărțile pe care utilizatorul respectiv le-a marcat ca fiind citite și le afișează într-o listă pentru a fi vizibile acestuia în contul său. Odată adăugate în această listă, utilizatorul le poate acorda un rating, reprezentat printr-un număr de la 0 la 5.
10. **WishList.js**: această componentă conține cărțile pe care utilizatorul respectiv le-a marcat ca fiind cărți pe care acesta și-ar dori să le citească în viitor și le afișează într-o listă pentru a fi vizibile acestuia în contul său. Fiecare titlu de carte din această listă este însoțit cu un buton care oferă posibilitatea adăugării cărții respective în lista cărților aflate în curs de citire: ActiveList.
11. **Admin.js**: componenta aceasta este pagina de „Home” randată după o autentificare reușită a unui utilizator de tip administrator. Această pagină conține toate câmpurile, însoțite de butoane corespunzătoare, pentru ca administratorul să poată realiza acțiuni pentru administrarea acestei aplicații. În antetul paginii se mai găsesc și două butoane: „Log out” și „Info”.
12. **Info.js**: în această componentă sunt preluate informații din server precum: utilizatorii, comentariile cărților și titlurile tuturor cărților, pentru ca administratorul să aibă o buna vedere de ansamblu asupra acțiunilor realizate de administratori în cadrul aplicației. Aceasta pagină este randată la apăăsarea butonului „Info” din componenta precedentă.
13. **API.js**: aceasta componentă folosește biblioteca HTTP AXIOS pentru a face request-uri către metodele dezvoltate în cadrul server-ului.

Rezultate experimentale

Pagina de autentificare pentru utilizator



Please sign in


user
user 

Sign in

Pagina de „Home” a utilizatorului

[My page](#) [Sign out](#)

Welcome, user!



All the Books:

Try to get away

Add to WishList

Add to ActiveList

Try to make it possible

Add to WishList

Add to ActiveList

Christmas Morning

Add to WishList

Add to ActiveList

We need to reconnect

Add to WishList

Add to ActiveList

Top 10 restaurants in London

Add to WishList

Add to ActiveList

How to win more money

Add to WishList

Add to ActiveList

How to cook

Add to WishList

Add to ActiveList


Activați Windows

Accesați Setări pentru a activa Windows.

Opțiunea de „căutare” a unei cărți după un cuvânt conținut de titlu

[My page](#) [Sign out](#)

Welcome, user!



How to win more money

Add to WishList


Add to ActiveList

How to cook

Add to WishList

Add to ActiveList

MINISTERUL EDUCAȚIEI NAȚIONALE



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Contul utilizatorului accesată după apăsarea butonului „My Page” din colțul din stânga sus a paginii de „Home”. Momentan utilizatorul nu are nicio carte în listacărților aflate în cur de citire



My Home Page

My WishList:

Try to make it possible

Add to ActiveList

The books I have already read:

Try to get away!

1 2 3 4 5

How to cook

1 2 3 4 5

Top 10 restaurants in London

1 2 3 4 5

We need to reconnect

1 2 3 4 5

Christmas Morning

1 2 3 4 5

Now I'm reading:

Activați Windows

Accesați Setări pentru a activa Windows.

Lista de cărți recomandate utilizatorului

Books that we recommend for you:

Christmas Morning

Try to make it possible

Try to get away

How to win more money

How to cook

Top 10 restaurants in London

We need to reconnect



Universitatea Tehnică din Cluj-Napoca

Catedra de Calculatoare și Tehnologia Informației

Pagina de autentificare a administratorului



Please sign in

admin
admin 

Sign in

Pagina de „Home” a administratorului

Sign out Info

Welcome, Admin!

Title
Year
Author
Publishing House
Short Description

Add Book

Id

Delete Book

IsAdmin
Username
Password

Add User

Id

Delete User

Id

Delete Comment

Id
New Author

Update Author

Id
New Year

Update Year

Id
New PublishingHouse

Update PublishingHouse

Id
New Title

Update Title

MINISTERUL EDUCAȚIEI NAȚIONALE



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Pagina unde administratorului poate vedea informații legate de aplicație, care poate fi accesată după apăsarea butonului de „Info” din colțul din stânga sus a paginii



All books:

0	Try to get away
1	Try to make it possible
2	Christmas Morning
3	We need to reconnect
4	Top 10 restaurants in London
5	How to win more money
6	How to cook

All users:

0	diana
1	gabi
2	user
3	user1
4	admin

All comments:

0	Good Book
1	A good book for rainy days spent inside :)
2	Awesome book! Can't stop reading it!!
3	I can't wait for the next volume!
4	Nice book, but I think it is more appropriate for women
5	An old but good book for rainy days
6	My children want me to read it every Christmas
7	I just bought this book and my child wants me to read it every night before bed
8	I bought this book for my 17y girl and now she is spending less time on Instagram and Facebook
9	You can find all this kind of information on the Internet, you don't have to read this book. Just wasted my time
10	I received this book as a present for my birthday and I find it quite interesting
11	An interesting book, but I'm not sure that I can use those ideas in real life
12	Nothing written there is real!!
13	I'm trying a new recipe every day and I started to enjoy cooking since I bought it
14	It's a good book, delicious recipes, but I can't tell that it is appropriate for beginners

Activați Windows
Accesați Setări pentru a activa Windows.



Bibliografie

- [1] <https://reactjs.org/tutorial/tutorial.html>
- [2] <https://www.toptal.com/react/managing-view-state-with-react>
- [3] <https://medium.com/createdd-notes/understanding-mvc-architecture-with-react-6cd38e91fef9>
- [4] „Modern web-development using reactjs - S Aggarwal - International Journal of Recent Research Aspects, 2018” :
<http://ijrra.net/Vol5issue1/IJRA-05-01-27.pdf>
- [5] „React js essentials – A Fedosejev, 2015”
https://books.google.ro/books?hl=ro&lr=&id=Rhl1CgAAQBAJ&oi=fnd&pg=PP1&dq=mvc+react&ots=JkryrGAQNE&sig=vKppZRZFFJjT4wKiiM4fshEU-7w&redir_esc=y#v=onepage&q=mvc%20react&f=false

Anexe

Anexa A

Cod sursă miniproiect

```
//conexiunea cu server-ul aplicatiei
import axios from 'axios';

//////////////////////////////////USER//////////////////////////////////
//search for book by title
export const search = (title) =>{
    const response = `http://localhost:59427/api/book/search/title/${title}`
    return axios.get(response)
}

//get read books of an use
export const getReadBooks = (idUser) =>{
    const response = `http://localhost:59427/api/user/readBooks/${idUser}`
    return axios.get(response)
}

//get active books of an user
export const getActiveBooks = (idUser) =>{
    const response = `http://localhost:59427/api/user/activeBooks/${idUser}`
    return axios.get(response)
}

//get wishlist of an user
export const getWishList = (idUser) =>{
    const response = `http://localhost:59427/api/user/wishList/${idUser}`
    return axios.get(response)
}

//add rating to a book
export const addRating = (idUser, idBook, rating) =>{
    const response = `http://localhost:59427/api/user/rating/${idUser}/${idBook}/${rating}`
    return axios.put(response)
}
```

```
//add a book to active books
export const addBookToActiveBooks = (idUser, idBook) =>{
    const response = `http://localhost:59427/api/user/activeBook/${idUser}/${idBook}`
    console.log(response)
    return axios.put(response)
}

//add a book to read books
export const addBookToReadBooks = (idUser, idBook) =>{
    const response = `http://localhost:59427/api/user/readBook/${idUser}/${idBook}`
    return axios.put(response)
}

//add a book to wishlist
export const addBookToWishList = (idUser, idBook) =>{
    const response = `http://localhost:59427/api/user/wishList/${idUser}/${idBook}`
    return axios.put(response)
}

//delete book from active books
export const deleteBookFromActiveBooks = (idUser, idBook) =>{
    const response = `http://localhost:59427/api/user/deleteActiveBook/${idUser}/${idBook}`
    return axios.delete(response)
}

//delete book from wishlist
export const deleteBookFromWishList = (idUser, idBook) =>{
    const response = `http://localhost:59427/api/user/deleteWishListBook/${idUser}/${idBook}`
    return axios.delete(response)
}

//get book by id
export const getBook = (id) =>{
    const response = `http://localhost:59427/api/book/id/${id}`
    return axios.get(response)
}
```



```
//////////////////////////////////USER+ADMIN//////////////////////////////////
//get all books
export const getAllBooks = () =>{
    const response = 'http://localhost:59427/api/book'
    return axios.get(response)
}

//////////////////////////////////LOGIN//////////////////////////////////
//get id of an user by username
export const getUserId = (username) =>{
    const response = `http://localhost:59427/api/user/${username}`
    return axios.get(response)
}

//verify the user
export const login = (username, password) =>{
    const response = `http://localhost:59427/api/user/verify/${username}/${password}`
    return axios.get(response)
}

//////////////////////////////////ADMIN//////////////////////////////////
//add book
export const addBook = (title, author, year, pubHouse, description) =>{
    const response = `http://localhost:59427/api/book/add/${title}/${author}/${year}/${pubHouse}/${description}`
    return axios.post(response)
}

//add user
export const addUser = (isAdmin, username, password) =>{
    const response = `http://localhost:59427/api/user/add/${isAdmin}/${username}/${password}`
    return axios.post(response)
}

//delete book
export const deleteBook = (idBook) =>{
    const response = `http://localhost:59427/api/book/delete/${idBook}`
    return axios.delete(response)
}
```

```
}

//delete user
export const deleteUser = (idUser) =>{
    const response = `http://localhost:59427/api/user/delete/${idUser}`
    return axios.delete(response)
}

//delete comment
export const deleteComment = (idComment) =>{
    const response = `http://localhost:59427/api/comment/delete/${idComment}`
    return axios.delete(response)
}

//update the title of a book
export const updateTitle = (id, title) =>{
    const response = `http://localhost:59427/api/book/updateTitle/${id}/${title}`
    return axios.put(response)
}

//update the author of a book
export const updateAuthor = (id, author) =>{
    const response = `http://localhost:59427/api/book/updateAuthor/${id}/${author}`
    return axios.put(response)
}

//update the year of a book.
export const updateYear = (id, year) =>{
    const response = `http://localhost:59427/api/book/updateYear/${id}/${year}`
    return axios.put(response)
}

//update the publishing house of a book
export const updatePubHouse = (id, pubHouse) =>{
    const response = `http://localhost:59427/api/book/updatePH/${id}/${pubHouse}`
    return axios.put(response)
}
```

```
//get all comments
export const getAllComments = () =>{
    const response = 'http://localhost:59427/api/comment'
    return axios.get(response)
}

//get all users
export const getAllUsers = () =>{
    const response = 'http://localhost:59427/api/user'
    return axios.get(response)
}

import React, {useState} from 'react';
import './API.js'
import {getAllBooks} from './API'

const ListBooks = (props) => {
    const call = props.com.map(({Title}) => Title);
    const call1 = props.com.map(({Id}) => Id);
    return(
        <div>
            <div>
                {call.map(titlu =>
                    <div>
                        <p>{titlu}</p>
                    </div>
                )}
            </div>

            <div STYLE = "position:absolute; LEFT:600px ;TOP:92px">
                {call1.map(id =>
                    <div>
                        <p>{id}</p>
                    </div>
                )}
            </div>
        </div>
    );
}
```

```
const Info = (props) => {
  const [books, setBooks] = useState([])
  const [users, setUsers] = useState([])
  const [comments, setComments] = useState([])
  return(
    <div align = "center">
      <h4>All books:</h4>
      {(() => {getAllBooks().then((res) => setBooks(res.data)))}())}
      <ListBooks com={books}/>
    </div>
  );
}

export default Info;
```

Anexa B

Cod sursă proiect

```
import React, {useState} from 'react';
import './API.js'
import {getActiveBooks} from './API'
import {addBookToReadBooks} from './API'
import {deleteBookFromActiveBooks} from './API'

const ActiveList = (props) => {
  const titlesArr = []
  const idsArr = []
  const buttonsArr = []
  const call = props.com.map(({Title}) => Title);
  const call1 = props.com.map(({Id}) => Id);
  const finalArr = []

  function buttonsGenerate(){
    for (let i = 0; i < idsArr.length; i++) {
      var button = <button onClick = {(()) => handleClick(i)}>
        Add to ReadList
      </button>
      buttonsArr.push(button);
    }
  }

  function makeArr(){
    call.map(titlu => titlesArr.push(titlu))
    call1.map(id => idsArr.push(id))
  }

  function handleClick(index){
    for(let i = 0; i < idsArr.length; i++){
      if(i === index){
        addBookToReadBooks(props.pr, idsArr[i])
        deleteBookFromActiveBooks(props.pr, idsArr[i])
      }
    }
  }
}
```

```
function final(){
  for (let i = 0; i < titlesArr.length; i++) {
    finalArr.push(titlesArr[i]);
    finalArr.push(buttonsArr[i]);
  }
}

return(
  <div>
    {makeArr()}
    {buttonsGenerate()}
    {final()}
    {finalArr.map(item => <div>{item}</div>)}
  </div>
);
}

const ActiveBooks = (props) => {
  const [books, setBooks] = useState([])
  return(
    <div align = "center">
      <br />
      <br />
      <h4>Now I'm reading:</h4>
      {(() => {getActiveBooks(props.id).then((res) => setBooks(res.data))})}
    </div>
    <ActiveList com={books} pr = {props.id}/>
  );
}

export default ActiveBooks;
```

```
import React, {useState} from 'react';
import './API.js'
import {addBook} from './API'
import {deleteBook} from './API'
import {addUser} from './API'
import {deleteUser} from './API'
import {deleteComment} from './API'
import {updateTitle} from './API'
import {updateAuthor} from './API'
import {updateYear} from './API'
import {updatePubHouse} from './API'

const Admin = (props) => {
  //datele pe care adminul trebuie sa le introduca pentru a adauga o noua carte
  in sistem
  const [title, setTitle] = useState('')
  const [id, setId] = useState()
  const [author, setAuthor] = useState('')
  const [year, setYear] = useState()
  const [pubHouse, setPubHouse] = useState('')
  const [description, setDescription] = useState('')
  //datele pe care adminul trebuie sa le introduca pentru a sterge o carte din
  sistem
  const [idd, setIdd] = useState()
  //datele pe care adminul trebuie sa le introduca pentru a adauga un nou user
  const [idUser, setIdUser] = useState()
  const [isAdmin, setIsAdmin] = useState()
  const [username, setUsername] = useState('')
  const [password, setPassword] = useState('')
  //datele pe care adminul trebuie sa le introduca pentru a sterge un user din
  sistem
  const [idUserd, setIdUserd] = useState()
  //datele pe care adminul trebuie sa le introduca pentru a sterge un comentari
  u existent
  const [idComment, setidComment] = useState()
  //datele pe care adminul trebuie sa le introduca pentru a modifica titlul unei
  i carti
  const [idTitle, setIdTitle] = useState()
  const [newTitle, setNewTitle] = useState('')
  //datele pe care adminul trebuie sa le introduca pentru a modifica autorul unei
  ei carti
  const [idAuthor, setIdAuthor] = useState()
  const [newAuthor, setNewAuthor] = useState('')
```

```
//datele pe care adminul trebuie sa le introduca pentru a modifica anul apari
tiei unei carti
const [idYear, setIdYear] = useState()
const [newYear, setNewYear] = useState('')
//datele pe care adminul trebuie sa le introduca pentru a modifica editura un
ei carti
const [idPh, setIdPh] = useState()
const [newPh, setNewPh] = useState('')
return(
  <span>
    <nav class="navbar navbar-light bg-light">
      <form class="form-inline">
        <button class="btn btn-sm btn-outline-
secondary" type="button" onClick = {props.action}>Sign out</button>
        <button class="btn btn-sm btn-outline-
secondary" type="button" onClick = {props.action1}>Info</button>
      </form>
    </nav>

    <div align = "center">
      <h3>Welcome, Admin!</h3>
    </div>

    <input
      STYLE="position:absolute; TOP:158px; LEFT:430px"
      type = "text"
      placeholder="Id"
      value={idd}
      onChange={ (e) => {
        setIdd(e.target.value)
      }}/>

    <button
      type="button" class="btn btn-outline-danger"
      STYLE="position:absolute; TOP:200px; LEFT:430px"
      onClick = { () => {deleteBook(idd); console.log("Clicked")}}
    >
      Delete Book
    </button>

    <input
      STYLE="position:absolute; TOP:430px; LEFT:1030px"
      type = "text"
      placeholder="Id"
```



```
        value={idTitle}
        onChange={(e) => {
            setIdTitle(e.target.value)
        }}/>

<input
    STYLE="position:absolute; TOP:460px; LEFT:1030px"
    type = "text"
    placeholder="New Title"
    value={newTitle}
    onChange={(e) => {
        setNewTitle(e.target.value)
    }}/>

<button
    type="button" class="btn btn-outline-secondary"
    STYLE="position:absolute; TOP:500px; LEFT:1030px"
    onClick = {() => {updateTitle(idTitle, newTitle); console.log("Clicked")}}
    >
        Update Title
    </button>

<input
    STYLE="position:absolute; TOP:158px; LEFT:1150px"
    type = "text"
    placeholder="Id"
    value={idComment}
    onChange={(e) => {
        setidComment(e.target.value)
    }}/>

<button
    type="button" class="btn btn-outline-danger"
    STYLE="position:absolute; TOP:200px; LEFT:1150px"
    onClick = {() => {deleteComment(idComment); console.log("Clicked")}}
    >
        Delete Comment
    </button>

<input
    STYLE="position:absolute; TOP:158px; LEFT:910px"
    type = "text"
```

```
        placeholder="Id"
        value={idUserd}
        onChange={(e) => {
            setIdUserd(e.target.value)
        }}/>

<button
    type="button" class="btn btn-outline-danger"
    STYLE="position:absolute; TOP:200px; LEFT:910px"
    onClick = {() => {deleteUser(idUserd); console.log("Clicked")}}
>
    Delete User
</button>

<input
    STYLE="position:absolute; TOP:188px; LEFT:670px"
    type = "text"
    placeholder="IsAdmin"
    value={isAdmin}
    onChange={(e) => {
        setIsAdmin(e.target.value)
    }}/>
<input
    STYLE="position:absolute; TOP:218px; LEFT:670px"
    type = "text"
    placeholder="Username"
    value={username}
    onChange={(e) => {
        setUsername(e.target.value)
    }}/>
<input
    STYLE="position:absolute; TOP:248px; LEFT:670px"
    type = "text"
    placeholder="Passweord"
    value={password}
    onChange={(e) => {
        setPassword(e.target.value)
    }}/>
<button
    STYLE="position:absolute; TOP:290px; LEFT:670px"
    type="button" class="btn btn-outline-secondary"
    onClick = {() => {addUser(isAdmin, username,password); console
.log("Clicked")}}
```

```
>
    Add User
</button>

<input
    STYLE="position:absolute; TOP:188px; LEFT:190px"
    type = "text"
    placeholder="Title"
    value={title}
    onChange={(e) => {
        setTitle(e.target.value)
    }}/>
<input
    STYLE="position:absolute; TOP:218px; LEFT:190px"
    type = "text"
    placeholder="Year"
    value={year}
    onChange={(e) => {
        setYear(e.target.value)
    }}/>
<input
    STYLE="position:absolute; TOP:248px; LEFT:190px"
    type = "text"
    placeholder="Author"
    value={author}
    onChange={(e) => {
        setAuthor(e.target.value)
    }}/>
<input
    STYLE="position:absolute; TOP:278px; LEFT:190px"
    type = "text"
    placeholder="Publishing House"
    value={pubHouse}
    onChange={(e) => {
        setPubHouse(e.target.value)
    }}/>
<input
    STYLE="position:absolute; TOP:308px; LEFT:190px"
    type = "text"
    placeholder="Short Description"
    value={description}
    onChange={(e) => {
        setDescription(e.target.value)
    }}/>
```

```
<button
  STYLE="position:absolute; TOP:350px; LEFT:190px"
  type="button" class="btn btn-outline-secondary"
  onClick = {(e) => {addBook(title, author, year, pubHouse, descriptio
n); console.log("Clicked")}}>
  Add Book
</button>

<input
  STYLE="position:absolute; TOP:430px; LEFT:300px"
  type = "text"
  placeholder="Id"
  value={idAuthor}
  onChange={(e) => {
    setIdAuthor(e.target.value)
  }}/>
<input
  STYLE="position:absolute; TOP:460px; LEFT:300px"
  type = "text"
  placeholder="New Author"
  value={newAuthor}
  onChange={(e) => {
    setNewAuthor(e.target.value)
  }}/>
<button
  STYLE="position:absolute; TOP:500px; LEFT:300px"
  type="button" class="btn btn-outline-secondary"
  onClick = {(e) => {updateAuthor(idAuthor, newAuthor); console.log("C
licked")}}>
  Update Author
</button>

<input
  STYLE="position:absolute; TOP:430px; LEFT:550px"
  type = "text"
  placeholder="Id"
  value={idYear}
  onChange={(e) => {
    setIdYear(e.target.value)
  }}/>
<input
  STYLE="position:absolute; TOP:460px; LEFT:550px"
  type = "text"
  placeholder="New Year"
```

```
        value={newYear}
        onChange={(e) => {
            setNewYear(e.target.value)
        }}/>
    <button
        type="button" class="btn btn-outline-secondary"
        STYLE="position:absolute; TOP:500px; LEFT:550px"
        onClick = {() => {updateYear(idYear, newYear); console.log("Click
ked")}}}
    >
        Update Year
    </button>

    <input
        STYLE="position:absolute; TOP:430px; LEFT:790px"
        type = "text"
        placeholder="Id"
        value={idPh}
        onChange={(e) => {
            setIdPh(e.target.value)
        }}/>
    <input
        STYLE="position:absolute; TOP:460px; LEFT:790px"
        type = "text"
        placeholder="New PublishingHouse"
        value={newPh}
        onChange={(e) => {
            setNewPh(e.target.value)
        }}/>
    <button
        type="button" class="btn btn-outline-secondary"
        STYLE="position:absolute; TOP:500px; LEFT:790px"
        onClick = {() => {updatePubHouse(idPh, newPh); console.log("Click
ked")}}}
    >
        Update PublishingHouse
    </button>

</span>
);
}

export default Admin;
```

```
//API.js
//get recommendation names
//done
export const getRecommendationNames = (idUser) =>{
    const response = `http://localhost:59427/api/book/rec/names/${idUser}`
    return axios.get(response)
}

//the header of the page where user can search for books and see the books the sy
stem recommend
import React, {Component} from "react"
import Img from './user.png'

class Header1 extends Component{
    render(){
        return(
            <span>
                <nav class="navbar navbar-light bg-light">
                    <form class="form-inline">
                        <button class="btn btn-sm btn-outline-
secondary" type="button" onClick = {this.props.action1}>My page</button>
                        <button class="btn btn-sm btn-outline-
secondary" type="button" onClick = {this.props.action}>Sign out</button>
                    </form>
                </nav>
                <div align = "center">
                    <h3>Welcome, user!</h3>
                    <img src={Img} alt="User's Icon" width="120" height="120"/>
                </div>
            </span>
        )
    }
}

export default Header1;
```

```
//PAGINA "ACASA"
//pagina este randata dupa ce user-ul da click pe butonul "Home"
//contul user-ului
import React from "react"
import WishList from "./WishList"
import ReadBooks from "./ReadBooks"
import ActiveBooks from "./ActiveBooks"
import ViewAllBooks from "./ViewAllBooks"
import Img from './goback.png'

const Home = (props) => {
  return(
    <span>
      <nav class="navbar navbar-light bg-light">
        <form class="form-inline">
          <input type="Image" src={Img} width="40" height="40" onC
lick = {props.action}/>
        </form>
      </nav>

      <div align = "center">
        <h3>My Home Page</h3>
        <WishList id = {props.id} />
        <ReadBooks id = {props.id} />
        <ActiveBooks id = {props.id} />
        <ViewAllBooks id = {props.id} stop = {props.stop} stopv = {pr
ops.stopv} titles={props.titles} change={props.change}/>
      </div>
    </span>
  );
}

export default Home;
```

```
import React, { useState } from 'react';
import './API.js'
import {getAllBooks} from './API'
import {getAllUsers} from './API'
import {getAllComments} from './API'
import Img from './goback.png'

const ListBooks = (props) => {
  const call = props.com.map(({Title}) => Title);
  const call1 = props.com.map(({Id}) => Id);
  return(
    <div>
      <div>
        {call.map(titlu =>
          <div>
            <p>{titlu}</p>
          </div>
        )}
      </div>

      <div STYLE = "position:absolute; LEFT:600px ;TOP:92px">
        {call1.map(id =>
          <div>
            <p>{id}</p>
          </div>
        )}
      </div>
    </div>
  );
}

const ListUsers = (props) => {
  const call = props.com.map(({Username}) => Username);
  const call1 = props.com.map(({Id}) => Id);
  return(
    <div>
      <div>
        {call.map(username =>
          <div>
            <p>{username}</p>
          </div>
        )}
      </div>
    </div>
  );
}
```



```
        <div STYLE = "position:absolute; LEFT:660px ;TOP:457px">
            {call1.map(id =>
                <div>
                    <p>{id}</p>
                </div>
            )}
        </div>

    </div>
);
}

const ListComments = (props) => {
    const call = props.com.map(({Text}) => Text);
    const call1 = props.com.map(({Id}) => Id);
    return(
        <div>
            <div>
                {call.map(text =>
                    <div>
                        <p>{text}</p>
                    </div>
                )}
            </div>

            <div STYLE = "position:absolute; LEFT:310px ;TOP:742px">
                {call1.map(id =>
                    <div>
                        <p>{id}</p>
                    </div>
                )}
            </div>
        </div>
    );
}

const Info = (props) => {
    const [books, setBooks] = useState([])
    const [users, setUsers] = useState([])
    const [comments, setComments] = useState([])
    return(
        <div align = "center">
            <nav class="navbar navbar-light bg-light">
```

```
        <form class="form-inline">
            <input type="Image" src={Img} width="40" height="40" onClick = {props.actionAdmin}/>
        </form>
    </nav>
    <h4>All books:</h4>
    {(() => {getAllBooks().then((res) => setBooks(res.data)))}())}

    <ListBooks com={books}/>
    <br />
    <br />
    <h4>All users:</h4>
    {(() => {getAllUsers().then((res) => setUsers(res.data)))}())}

    <ListUsers com={users}/>
    <br />
    <br />
    <h4>All comments:</h4>
    {(() => {getAllComments().then((res) => setComments(res.data)))}())}

    <ListComments com={comments}/>
</div>
);
}

export default Info;
```

```
import React, {useState} from "react"
import './API.js'
import {login} from './API'

const Login = (props) =>{
  const [username, setUsername] = useState('')
  const [password, setPassword] = useState('')

  function logging(res){
    if(res === 0){
      props.husername(username);
      props.actionUser();
    }else if(res === 1){
      props.actionAdmin();
    }else{
      props.action();
    }
  }

  return(
    <div className = "text-center">
      <form className = "form-signin">
        

        <br/>
        <br/>
        <br/>

        <h1 className="h3 mb-3 font-weight-
normal">Please sign in</h1>

        <label for="inputUsername" className="sr-only">
          Username
        </label>
        <input type="username" id="inputUsername" value = {username}
placeholder="Username" required autoFocus onChange={({ target }) => setUsername(
target.value)}></input>

        <br/>

        <label for="inputPassword" className="sr-only">
```

```
        Password
        </label>
        <input type="password" id="inputPassword" value = {password}
placeholder="Password" required onChange={({ target }) => setPassword(target.valu
e)}}></input>

        <br/>
        <br/>

        <button class="btn btn-outline-
primary" type="button" onClick={() => login(username, password).then((res) => log
ging(res.data))}>
            Sign in
        </button>
    </form>
</div>
)
}

export default Login;

import React from "react"
import Header1 from "./Header1"
import SearchBox from "./SearchBox"
import ViewAllBooks from "./ViewAllBooks"

class Page1 extends React.Component {

    render(){
        return(
            <div>
                <Header1 action = {this.props.action} action1 = {this.props.actio
n1}/>

                <SearchBox id = {this.props.id}/>
            </div>
        );
    }
}

export default Page1;
```

```
import React, { useState } from 'react';
import './API.js'
import {getAllComments} from './API'

const CommentsList = (props) => {
  const call = props.com.map(({Text}) => Text);
  return(
    <div>
      {call.map(coment =>
        <li>
          {coment}
        </li>
      )}
    </div>
  );
}

const MainContent = () => {
  const [comments, setComments] = useState([])
  return(
    <div>
      <br />
      <br />
      <button onClick={() => getAllComments().then((res) => setComments(res
        .data))}>
        Afiseaza comentariile!
      </button>
      <CommentsList com={comments}/>
    </div>
  );
}

export default MainContent;
```

```
import React, { useState } from 'react';
import './API.js'
import {getReadBooks} from './API'
import {addRating} from './API'

const List = (props) => {
  const titlesArr = []
  const idsArr =[]
  const finalArr = []
  const call = props.com.map(({Title}) => Title);
  const call1 = props.com.map(({Id}) => Id);
  const buttonsArr = []

  function buttonsGenerate(){
    //j for index, i for rating value
    //fac cate 5 butoane cu acelasi id, dar cu valori deiferite
    for(let j = 0; j < titlesArr.length; j++){
      for (let i = 1; i < 6; i++) {
        var button = <button onClick = {(()) => handleClick(j, i)}>
          {i}
        </button>
        buttonsArr.push(button);
      }
    }
  }

  function handleClick(index, rating){
    for(let i = 0; i < idsArr.length; i++){
      if(i === index){
        console.log("user: " + props.pr);
        console.log("book_id: " + idsArr[i]);
        console.log("rating: " + rating);
        addRating(props.pr, idsArr[i], rating);
      }
    }
  }

  function makeArr(){
    call.map(titlu => titlesArr.push(titlu));
    call1.map(id => idsArr.push(id));
  }

  function final(){
```

```
        for(let i = 0; i<titlesArr.length; i++){
            finalArr.push(titlesArr[i]);
            finalArr.push(buttonsArr[i*5]);
            finalArr.push(buttonsArr[i*5+1]);
            finalArr.push(buttonsArr[i*5+2]);
            finalArr.push(buttonsArr[i*5+3]);
            finalArr.push(buttonsArr[i*5+4]);
        }
    }

    function handleItem(item){
        if(typeof item === "string"){
            return <div>{item}</div>
        }else{
            return <div STYLE = "display: inline-block">{item}</div>
        }
    }

    return(
        <div>
            {makeArr()}
            {buttonsGenerate()}
            {final()}
            {finalArr.map(item => handleItem(item))}
        </div>
    );
}

const ReadBooks= (props) => {
    const [books, setBooks] = useState([])
    return(
        <div align = "center">
            <br />
            <br />
            <h4>The books I have already read:</h4>
            {(() => {getReadBooks(props.id).then((res) => setBooks(res.data))})()}
        </div>
        <List com = {books} pr = {props.id}/>
    );
}

export default ReadBooks;
```

```
import React, {useState} from 'react';
import './API.js'
import {search} from './API'
import {addBookToWishList} from './API'
import {addBookToActiveBooks} from './API'

const BooksList = (props) => {
  const wishButttons = []
  const activeButtons = []
  const idsArr = []
  const titlesArr =[]
  const call = props.books.map(({Title}) => Title);
  const call1 = props.books.map(({Id}) => Id);
  const finalArr = []

  function buttonWishGenerate(){
    for (let i = 0; i < idsArr.length; i++) {
      var button = <button onClick = {() => handleClick1(i)}>
        Add to WishList
      </button>
      wishButttons.push(button);
    }
  }

  function buttonActiveGenerate(){
    for (let i = 0; i < idsArr.length; i++) {
      var button = <button onClick = {() => handleClick2(i)}>
        Add to ActiveList
      </button>
      activeButtons.push(button);
    }
  }

  function handleClick2(index){
    for(let i = 0; i < idsArr.length; i++){
      if(i === index){
        addBookToActiveBooks(props.pr, idsArr[i])
      }
    }
  }

  function makeArr(){
    call.map(titlu => titlesArr.push(titlu))
  }
}
```



```
    call1.map(id => idsArr.push(id))
  }

  function handleClick1(index){
    for(let i = 0; i < idsArr.length; i++){
      if(i === index){
        addBookToWishList(props.pr, idsArr[i])
      }
    }
  }

  function final(){
    for (let i = 0; i < titlesArr.length; i++) {
      finalArr.push(titlesArr[i]);
      finalArr.push(wishButttons[i]);
      finalArr.push(activeButtons[i]);
    }
  }

  return(
    <div>
      {makeArr()}
      {buttonActiveGenerate()}
      {buttonWishGenerate()}
      {final()}
      {finalArr.map(item => <div>{item}</div>)}
    </div>
  );
}

const SearchBox = (props) => {
  const [term, setTerm] = useState('')
  const [books, setBooks] = useState([])
  return (
    <div align = "center">
      <br />
      <br />
      <input
        type = "text"
        placeholder="Search..."
        value={term}
        onChange={(e) => {
          setTerm(e.target.value)
        }}/>
    </div>
  )
}
```

```
      <button onClick={() => search(term).then((res) => setBooks(res.data))}>
        Search
      </button>
      <br />
      <BooksList books={books} pr={props.id}/>
    </div>
  )
}

export default SearchBox;
```

```
import React, { useState } from 'react';
import './API.js'
import {getRecommandationNames} from './API'

const RecommendedBooksList = (props) => {
  function condition(){
    console.log(props.stopv)
    if(props.com1.length > 0 && props.stopv === 0){
      props.stop();
      props.change(props.com1);
    }

    return(
      <div>
        {condition()}
        {props.titles.map((titlu) => <div>{titlu}</div>)}
        {console.log(props.titles)}
      </div>
    );
  }

  Const RecommendedBooks = (props) => {
    const [books, setBooks] = useState([])
    return(
      <div align = "center">
        <br />
        <br />
        <h4>Books that we recommend for you:</h4>
        {(() => {getRecommandationNames(props.id).then((res) => setBooks(res.
data))))}()
        <RecommendedBooksList com1 = {books} pr = {props.id} stop = {props.s
top} stopv = {props.stopv} titles={props.titles} change = {props.change}/>
      </div>
    );
  }

  export default RecommendedBooks;
```

```
import React, { useState } from 'react';
import './API.js'
import {getWishList} from './API'
import {addBookToActiveBooks} from './API'
import {deleteBookFromWishList} from './API'

const List = (props) => {
  const titlesArr = []
  const idsArr = []
  const buttonsArr = []
  const call = props.com.map(({Title}) => Title);
  const call1 = props.com.map(({Id}) => Id);
  const finalArr = []

  function buttonsGenerate(){
    for (let i = 0; i < idsArr.length; i++) {
      var button = <button onClick = {() => handleClick(i)}>
        Add to ActiveList
      </button>
      buttonsArr.push(button);
    }
  }

  function final(){
    for (let i = 0; i < titlesArr.length; i++) {
      finalArr.push(titlesArr[i]);
      finalArr.push(buttonsArr[i]);
    }
  }

  function makeArr(){
    call.map(titlu => titlesArr.push(titlu))
    call1.map(id => idsArr.push(id))
  }

  function handleClick(index){
    for(let i = 0; i < idsArr.length; i++){
      if(i === index){
        addBookToActiveBooks(props.pr, idsArr[i])
        deleteBookFromWishList(props.pr, idsArr[i])
      }
    }
  }
}
```

```
    return(  
      <div>  
        {makeArr()}  
        {buttonsGenerate()}  
        {final()}  
        {finalArr.map(item => <div>{item}</div>)}  
      </div>  
    );  
}  
  
const WishList= (props) => {  
  const [books, setBooks] = useState([])  
  return(  
    <div align = "center">  
      <br />  
      <br />  
      <h4>My WishList:</h4>  
      {(() => {getWishList(props.id).then((res) => setBooks(res.data))})()}  
  
      <List com={books} pr = {props.id}/>  
    </div>  
  );  
}  
  
export default WishList;
```

```
//AFISAREA TUTUROR CARTILOR DIN INTREGUL SISTEM
import React, { useState } from 'react';
import './API.js'
import {getAllBooks, addBookToActiveBooks, addBookToWishList} from './API'

const Books = (props) => {
  const titlesArr = []
  const idsArr = []
  const buttonsWish = []
  const buttonsActive = []
  const call = props.com.map(({Title}) => Title);
  const call1 = props.com.map(({Id}) => Id);
  const finalArr = []

  function handleClickActive(index){
    for(let i = 0; i < idsArr.length; i++){
      if(i === index){
        addBookToActiveBooks(props.pr, idsArr[i])
      }
    }
  }

  function handleClickWish(index){
    for(let i = 0; i < idsArr.length; i++){
      if(i === index){
        addBookToWishList(props.pr, idsArr[i])
      }
    }
  }

  function wishGenerate(){
    for (let i = 0; i < idsArr.length; i++) {
      var button = <button onClick = {() => handleClickWish(i)}>
        Add to WishList
      </button>
      buttonsWish.push(button);
    }
  }
}
```

```
function activeGenerate(){
  for (let i = 0; i < idsArr.length; i++) {
    var button = <button onClick = {() => handleClickActive(i)}>
      Add to ActiveList
    </button>
    buttonsActive.push(button);
  }
}

function makeArr(){
  call.map(titlu => titlesArr.push(titlu))
  call1.map(id => idsArr.push(id))
}

function final(){
  for (let i = 0; i < idsArr.length; i++) {
    finalArr.push(titlesArr[i]);
    finalArr.push(buttonsWish[i]);
    finalArr.push(buttonsActive[i]);
  }
}

function handleItem(item){
  if(typeof item === "string"){
    return <div>{item}</div>
  }else{
    return <div STYLE = "display: inline-block">{item}</div>
  }
}

return(
  <div>
    {makeArr()}
    {wishGenerate()}
    {activeGenerate()}
    {final()}
    {finalArr.map(item => handleItem(item))}
  </div>
);
}

const ViewAllBooks = (props) => {
```

```
const [books, setBooks] = useState([])
return(
  <div align = "center">
    <br />
    <br />
    <h4>All the Books:</h4>
    {(() => {getAllBooks().then((res) => setBooks(res.data)))}())}
    <Books com = {books} pr = {props.id}/>
  </div>
);
}
```

export default ViewAllBooks;