

Aplikacja do zarządzania książkami

Zarządzanie książkami

Lista książek

hobbit - piotrek, Przygodowa

Merida waleczna - Pan maciek, Horror

App.js

```
import React, { useState } from 'react';
```

```
import 'bootstrap/dist/css/bootstrap.min.css';
```

```
/*  
*****  
*/
```

```
* klasa: App
```

```
*
```

```
* opis: Aplikacja do zarządzania książkami umożliwiająca dodawanie,  
*       usuwanie, edytowanie i przeglądanie książek. Każda książka  
*       posiada tytuł, autora, gatunek i grafikę.
```

```
*
```

```
* autor: Bartek Pyka
```

```
*
```

```
*****/  
*/
```

```
function App() {
```

```
  const [books, setBooks] = useState([]);
```

```
  const [newBook, setNewBook] = useState({ title: '', author: '', genre: '', image: '' });
```

```

const [editIndex, setEditIndex] = useState(-1);

// Zmiana wartości w formularzu
const handleInputChange = (e) => {
  setNewBook({ ...newBook, [e.target.name]: e.target.value });
};

// Dodawanie książki do tablicy
const handleAddBook = (e) => {
  e.preventDefault();
  if (editIndex >= 0) {
    const updatedBooks = books.map((item, index) => index === editIndex ? newBook :
item);
    setBooks(updatedBooks);
    setEditIndex(-1);
  } else {
    setBooks([...books, newBook]);
  }
  setNewBook({ title: '', author: '', genre: '' });
};

// Odpalenie edycji książki (podmienia wartości w formularzu na wartości z książki)
const handleEditBook = (index) => {
  setNewBook(books[index]);
  setEditIndex(index);
};

// Usuwanie książki z tablicy (po prostu ją usuwa)
const handleDeleteBook = (index) => {
  const updatedBooks = books.filter((_, i) => i !== index);
  setBooks(updatedBooks);
};

return (
  <div className="container mt-4">
    <h1>Zarządzanie książkami</h1>

    { /* Formularz dodawania książki */ }
    <form onSubmit={handleAddBook} className="mb-4">
      <input type="text" name="title" value={newBook.title} onChange={handleInputChange}
placeholder="Tytuł książki" className="form-control my-2" required />
      <input type="text" name="author" value={newBook.author} onChange={handleInputChange}
placeholder="Autor książki" className="form-control my-2" required />
      <input type="text" name="genre" value={newBook.genre} onChange={handleInputChange}
placeholder="Gatunek książki" className="form-control my-2" required />
      <button type="submit" className="btn btn-primary">{editIndex >= 0 ? 'Zapisz zmiany' :
'Dodaj książkę'}</button>
    </form>

    <h2>Lista książek</h2>
  </div>
);

```

```

    { /* Wyświetlanie wszystkich książek w tablicy */
    {books.length > 0 ? (
      <ul className="list-group">
        {books.map((book, index) => (
          <li key={index} className="list-group-item">
            <strong>{book.title}</strong> - {book.author}, {book.genre}
            <div className="float-right">
              <button onClick={() => handleEditBook(index)} className="btn btn-info btn-sm
mx-1">Edytuj</button>
              <button onClick={() => handleDeleteBook(index)} className="btn btn-danger
btn-sm">Usuń</button>
            </div>
          </li>
        ))}
      </ul>
    ) : (
      <p>Brak książek na liście.</p>
    )}
  </div>
);
}

export default App;

```