

Book Allocation System – Full Stack Project

Tech Stack:

- **Frontend:** React.js, Tailwind CSS, React Router, Axios, React-Toastify
 - **Backend:** Node.js, Express.js, MongoDB, Mongoose, JWT, bcrypt
 - **UX Enhancements:** React-Toastify (for notifications), React-Spinners (for loaders)
-

Folder Structure

Backend (book-backend/)

```
pgsql
CopyEdit
.
├── controllers/
│   ├── authController.js
│   └── bookController.js
├── middleware/
│   └── authMiddleware.js
├── models/
│   ├── User.js
│   └── Book.js
├── routes/
│   ├── auth.js
│   └── book.js
├── .env
└── server.js
```

Frontend (book-frontend/)

```
css
CopyEdit
.
├── src/
│   ├── components/
│   │   ├── BookCard.jsx
│   │   ├── EditModal.jsx
│   │   └── Header.jsx
│   ├── pages/
│   │   ├── Home.jsx
│   │   ├── Login.jsx
│   │   └── Register.jsx
│   └── services/
│       └── api.js
```

```
├── App.jsx
├── main.jsx
└── index.css
```

□ Backend Setup

✦ Environment Variables (.env)

```
ini
CopyEdit
PORT=5000
MONGO_URI=your_mongodb_connection
JWT_SECRET=your_jwt_secret_key
```

✦ Install & Run

```
bash
CopyEdit
cd book-backend
npm install
npm run dev
```

🔒 Auth Routes (/api/auth)

- POST /register — Register new user
- POST /login — Login and receive token in HTTP-only cookie

📖 Book Routes (/api/books)

- All routes are protected by JWT middleware
 - GET / — Get all books
 - POST / — Add a book
 - PUT /:id — Update a book (only by uploader)
 - DELETE /:id — Delete a book (only by uploader)
-

🌐 Frontend Setup

✦ Install & Run

```
bash
CopyEdit
cd book-frontend
npm install
npm run dev
```

☞ Axios Setup (src/services/api.js)

```
js
CopyEdit
import axios from "axios";

const API = axios.create({
  baseURL: "http://localhost:5000/api",
  withCredentials: true,
});

export default API;
```

🔄 Application Workflow

🔒 Auth Flow

1. **Register** → Validates input → Sends data to /api/auth/register
 2. **Login** → Validates input → Receives token cookie and username → Saves username to localStorage
 3. **Auth Middleware** checks JWT from cookie in every protected route
-

📖 Book Management Flow

1. **Home Page Loads**
 - Auth-protected route (/home)
 - Fetches books via GET /books
 - Shows loader using react-spinners
 2. **Uploading a Book**
 - Form with title & author
 - Calls POST /books
 - Success → toast + refresh list
 3. **Editing a Book**
 - Opens modal
 - Edits fields
 - Calls PUT /books/:id
 - Success → toast + refresh list
 4. **Deleting a Book**
 - Calls DELETE /books/:id
 - Success → toast + refresh list
-

🔗 UI Features

- 🌈 Gradient background and modern card design
 - 🍞 Toasts for every action using react-toastify
 - ⌛ Loading spinner for data fetching
 - 📦 Reusable components (BookCard, EditModal, Header)
 - 🔒 Protected route using token presence (/home)
 - 🚪 Logout clears cookie and local storage
-

🔗 Deployment Tips

🔒 Backend

- Host on **Render**, **Railway**, or **VPS**
- Set JWT_SECRET, MONGO_URI in deployed env

🌐 Frontend

- Host on **Vercel** or **Netlify**
- Set API_BASE_URL in .env and update api.js:

```
js
CopyEdit
const API = axios.create({
  baseURL: import.meta.env.VITE_API_BASE_URL,
  withCredentials: true,
});
```

🚀 Future Enhancements

- Add file/image upload with Multer (e.g. book cover)
- Pagination or search in book list
- Theme toggle (light/dark)
- Admin panel for managing users/books
- Profile page for each user