Library Management System (Angular + Node.js)

This project is a **Library Management System** built using **Angular** (for the frontend) and **Node.js with Express** (for the backend). The goal is to allow users to **view books**, apply **filters**, and **add books** (only if logged in).

Project Overview

- **Frontend:** Angular (to create an interactive UI).
- **Backend:** Node.js + Express (to handle API requests and authentication)
- **Database: MongoDB or MySQL**

Features

✅ View books on the dashboard (open for all users)
✅ Apply filters (search books by Genre, Year, or Author)
✅ Login system (only authorized users can add books)
✅ Add new books (only after logging in)
✅ Protect certain pages (using **Auth Guards**)

1. Folder Structure

A good folder structure makes your project **organized** and **maintainable.**

📌 Library Management System: Page Descriptions

This project consists of **four main pages**:

1. **Login Page** – Users/Admins log in.

2. **Dashboard** (Home Page) – Shows books (**only for logged-in users**).
3. **Add Book Page** – Only **admins** can add books.
4. **Book Details Page** – Shows full book info.

Each page has specific **UI elements** and **access control**.

1️⃣ Login Page

📍 **Route:** `/login`
👥 **Accessible by:** Everyone
📌 **Purpose:** Allows **users & admins** to log in.

🔹 Features to Include

✔ **Username & Password Fields** – Users enter credentials.
✔ **Login Button** – If correct:

- Normal **users** go to the **Dashboard**.
- **Admins** go to the **Dashboard** (with an "Add Book" option).

✔ **Error Messages** – Show an error if credentials are incorrect.
✔ **Store Role in Local Storage** – Save `"role": "user"` or `"role": "admin"`.

2️⃣ Dashboard (Home Page)

📍 **Route:** `/dashboard`
👥 **Accessible by: Logged-in users and admins**
📌 **Purpose:** Shows the **book list**.

🔹 Features to Include

✔ **Book List** – Displays books with title, author, and genre.
✔ **Filters** – Users can filter books by:

- **Genre**
- **Author**

- **Publication Year**
  ✔ **Search Bar** – Search books by title.
  ✔ **Navigation Bar** – Includes **Dashboard, Logout, and Add Book** (only for admins).
  ✔ **Access Control** – If **not logged in**, redirect to **Login Page**.
  ✔ **Click on Book** – Navigates to **Book Details Page**.

3️⃣ Add Book Page

📍 **Route:** `/add-book`
👥 **Accessible by: Only logged-in admin**
📌 **Purpose:** Allows **admin** to **add a new book** to the library.

🔹 Features to Include

✔ **Title Input Field** – Enter the book's title.
✔ **Author Input Field** – Enter the author's name.
✔ **Genre Dropdown** – Select a genre.
✔ **Publication Year Field** – Choose the book's year of publication.
✔ **Submit Button** – Saves the book and redirects to the **Dashboard**.
✔ **Form Validation** – Ensure all fields are filled before submission.
✔ **Navigation Restriction** – If a **non-logged-in admin** tries to access, **redirect them to Login**.

4️⃣ Book Details Page

📍 **Route:** `/book/:id`
👥 **Accessible by:** Everyone
📌 **Purpose:** Displays **detailed information** about a book.

🔹 Features to Include

✔ **Book Title, Author, Genre, and Year** – Show detailed book info.
✔ **Description** – Add a **short description** about the book.
✔ **Back Button** – Navigate back to the **Dashboard**.

🚀 Final Summary

Page	Route	Features	Accessible By

Dashboard	/dashboard	Book list, filters, search	Everyone
Login	/login	Login form, authentication	Everyone
Add Book	/add-book	Form for adding books (title, author, etc.)	admin Only
Book Details	/book/:id	Shows full details of a selected book	Everyone

💡 To **enhance** the project, students can:

1. **Add a Book Cover Image** – Allow users to upload an image for each book.
2. **Implement User Registration** – Add a signup page for new users.
3. **Enable Editing & Deleting Books** – Let logged-in users modify or remove books.
4. **Improve UI with Bootstrap/Tailwind** – Make the design more attractive.
5. **Store Data in a Database (MongoDB or MySQL)** – Instead of a JSON file.

Summary

- Build a simple **Library Management System** with Angular & Node.js.
- Users can **view books**, **apply filters**, and **add books (if logged in)**.
- Use **AuthGuard** to protect the "Add Book" page.
- You can use ChatGPT for reference. However, whatever code you write must be understood before submission.
- Please note that plagiarism will not be tolerated.