Angular Take-Home Assignment

Overview:

Build a simple “Book Library” app in Angular that supports listing, adding, editing, and deleting books.

Deliverables

- A GitHub repo (or zip) containing your Angular project.
- A README with setup/run instructions and any notes.

Requirements

1. **Project Structure**
   - Lazy-load a LibraryModule at route /library.
   - Have a shared CoreModule for singleton services.
2. **Data Layer & DI**
   - Define an IBook interface (id, title, author, year, genre).
   - Create an IBookService interface/abstract class with CRUD methods.
   - Implementations suggestions:
     - HttpBookService (uses real HTTP to a mock REST API).
     - InMemoryBookService (in-app array).
   - Wire up dependency injection so you can swap services in CoreModule.
3. **Components & Routing**
   - **BookListComponent**: displays all books (optional pagination).
   - **BookDetailComponent**: add/edit form using Reactive Forms.
   - Use route parameters and a resolver to fetch book data before loading BookDetailComponent.

4. **Delete & Notifications**
    - Implement book deletion with a confirmation.
    - It can be a plus: Show success/error toasts via a NotificationService

5. **Error Handling & Logging**
    - Global HTTP error handling with an interceptor.
    - Log errors through a LoggerService (wrapper over console).
6. **Unit Tests (Optional)**
    - Write Jasmine/Karma tests for at least one service and one component, using mocks/stubs.

Key Focus:

- **SOLID**: interfaces, DI, single-responsibility services, open/closed components.
- **Angular Best Practices**: modules, lazy loading, Reactive Forms, HTTP interceptors.
- **Code Quality & UX**: readable code, graceful error handling, clear notifications.
- **Testing**: meaningful unit tests.

Good luck! 🚀📚📚