E-Library System – Project Specification

1. Core Modules

a) User (Reader) Dashboard

- **Browse Books** – list of available books with search, filters (by author, genre, price, etc.).
- **Book Details** – title, description, author, price (if paid), preview sample (e.g. first few pages).
- **Cart & Checkout** – add books to cart, make payment via **Paystack**.
- **Library** – see purchased books, read them inside the platform (PDF/EPUB viewer).
- **Transaction History** – list of all purchases, amount, date, and status.
- **Profile Management** – update email, password, payment details.

b) Admin Dashboard

- **User Management** – create, block, or delete users.
- **Book Management** – upload books (with metadata: title, author, price, description, cover image, file).
- **Transaction Monitoring** – view all transactions made by users (success, failed, pending).
- **Revenue Reports** – daily/weekly/monthly earnings summary.
- **Notifications** – send announcements (email/SMS/in-app) to users.
- **Roles & Permissions** – admin vs staff (staff can upload books but not manage finances).

2. Book Pricing & Payments

- Each book can be:
  - **Free** → instantly added to user's library.
  - **Paid** → requires Paystack payment.
- **Paystack Integration**:
  - Initialize transaction on backend.
  - Redirect/inline popup for payment.
  - Verify transaction with Paystack API.
  - Save transaction in DB (status, amount, reference, userId, bookId).
- **Payment Status**:
  - `pending`, `successful`, `failed`, `refunded`.

3. Transaction History

a) For Users:

- Filter by date or status.

- Show:
    - Transaction Reference
    - Book Title
    - Amount
    - Date
    - Status (Success/Failed)

b) For Admins:

- See **all user transactions**.
- Export to CSV/Excel.
- Insights like "Top-selling books" and "Highest spending users".

4. Other Features

- **Search & Recommendations** – recommend books based on purchases.
- **Ratings & Reviews** – users can review books (optional).
- **Security**:
    - JWT-based authentication.
    - Role-based access (Admin/User).
    - reCAPTCHA
    - 2FA via OTP
    - Email validation
- **Scalability**:
    - Store book files on cloud storage (e.g. AWS S3, GCP Bucket, DigitalOcean Spaces). optional
    - Cache frequently accessed books/transactions. optional

5. Tech Stack (Suggested)

- **Frontend**: React (for fast UI).
- **Backend**: Node.js (Express) or Laravel.
- **Database**: PostgreSQL or MySQL.
- **Storage**: AWS S3 / Cloudinary for book files & covers. For cloud, multer for server
- **Payments**: Paystack API.
- **Authentication**: JWT .

6. Database Schema (Simplified)

Users Table

- id, name, email, password, role (`admin`/`user`)

Books Table

- id, title, author, description, price, file_url, cover_url

Transactions Table

- id, user_id, book_id, amount, status, reference, created_at

UserLibrary Table

- id, user_id, book_id (represents purchased/downloaded books)