**Backend API Technical Requirement Document**

**BOOKS MANAGEMENT SYSTEM – LIST TRANSACTION API**

**1. Overview**

This document outlines the requirements for creating backend API endpoints to list book transactions for the Books Management System.

**2. Objective**

* Develop API endpoints to retrieve a list of book transactions.
* Implement pagination and filtering options for transaction listing.
* Integrate the backend with the database to fetch transaction data.

**3. Success Criteria**

* API returns a success response with transaction data upon a successful request.
* API returns an error response if there are issues fetching transaction data.
* Transaction data is securely retrieved from the database.

**4. Endpoints**

* **GET /api/transactions**
  + **Purpose:** Retrieve a list of book transactions.
  + **Query Parameters:**
    - **page**: Page number for pagination (default: 1)
    - **limit**: Number of transactions per page (default: 10)
    - **userId**: Filter transactions by user ID (optional)
    - **bookId**: Filter transactions by book ID (optional)
    - **status**: Filter transactions by status (optional)
  + **Response:**
    - Success (HTTP 200 OK):

Json { "success": true, "message": "Transactions retrieved successfully", "data": [ { “id”: “number”,"borrowingId": "string", "userId": "string", "bookId": "string", "transDate": "string", // Format: YYYY-MM-DD "status": "string" // e.g., "borrowed", "returned" }, ... ], "pagination": { "currentPage": "number", "totalPages": "number", "pageSize": "number", "totalItems": "number" } }

* + - Error (HTTP 500 Internal Server Error):

Json { "success": false, "message": "Internal server error" }

**5. Pagination**

* Implement pagination to limit the number of transactions returned per request.
* Allow users to specify the page number and limit through query parameters.

**6. Filtering**

* Implement filtering options to search for specific transactions based on criteria such as user ID, book ID, and status.
* Allow users to specify filtering parameters through query parameters.

**7. Error Handling**

* Handle server errors and database query errors appropriately, returning an HTTP 500 Internal Server Error status code with an error message.