**Backend API Technical Requirement Document**

**BOOKS MANAGEMENT SYSTEM – LIST BOOK API**

**1. Overview**

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

**2. Objective**

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

**3. Success Criteria**

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

**4. Endpoints**

* **GET /api/books**
  + **Purpose:** Retrieve a list of books.
  + **Query Parameters:**
    - **page**: Page number for pagination (default: 1)
    - **limit**: Number of books per page (default: 10)
    - **search**: Search term to filter books (optional)
  + **Response:**
    - Success (HTTP 200 OK):

Json { "success": true, "message": "Books retrieved successfully", "data": [ { "id": "string", "title": "string", "genre": "string", "publisher": "string", "author": "string", "bookNumber": "string", "publicationDate": "string" }, ... ], "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 books returned per request.
* Allow users to specify the page number and limit through query parameters.

**6. Filtering**

* Implement filtering options to search for specific books based on criteria such as title, author, genre, etc.
* Allow users to specify search 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.