**Backend API Technical Requirement Document**

**BOOKS MANAGEMENT SYSTEM – COMPLETE TRANSACTION API**

**1. Overview**

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

**2. Objective**

* Develop API endpoints to allow users to mark book transactions as complete (e.g., book returned).
* Implement validation for completing transaction data.
* Integrate the backend with the database to update transaction information securely.

**3. Success Criteria**

* API returns a success response upon successful completion of a transaction.
* API returns an error response upon failure to complete the transaction.
* Transaction status is securely updated in the database.

**4. Endpoints**

* **PUT /api/transactions/{transactionId}/complete**
  + **Purpose:** Mark a transaction as complete.
  + **Request Parameters:**
    - **transactionId**: The ID of the transaction to be completed.
  + **Request Body:**

Json { "completedDate": "string" // Format: YYYY-MM-DD }

* + **Response:**
    - Success (HTTP 200 OK):

Json { "success": true, "message": "Transaction completed successfully", "data": { "transactionId": "string", "userId": "string", "bookId": "string", "transDate": "string", // Format: YYYY-MM-DD "completedDate": "string",// Format: YYYY-MM-DD "status": "completed" } }

* + - Error (HTTP 400 Bad Request):

Json { "success": false, "message": "Validation error: Invalid data provided" }

* + - Error (HTTP 404 Not Found):

Json { "success": false, "message": "Transaction not found" }

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

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

**5. Validation**

* Validate that the **completedDate** is provided in the request body.
* Ensure the transaction ID exists in the database before attempting to update it.

**6. Error Handling**

* Invalid data provided: Return HTTP 400 Bad Request status code with an appropriate error message.
* Transaction not found: Return HTTP 404 Not Found status code with an appropriate error message.
* Handle server errors and database query errors appropriately, returning an HTTP 500 Internal Server Error status code with an error message.

**7. Data Storage**

* Update the transaction status and completed date securely in the database.
* Ensure data integrity by validating and sanitizing user input before updating it in the database.