**Project Documentation**

**Library Management System using**

**Overview**

This documentation provides instructions on how to run the application, interact with the API endpoints, and use the implemented basic authentication. The application is a Library Management System API built with Laravel, featuring endpoints for managing books, patrons, and borrowing records, with added security through basic authentication and logging middleware.

# Library Management System API This is a Library Management System API built with Laravel 11.

The system allows librarians to manage books, patrons, and borrowing records.

It features basic authentication and logging middleware.

## Features:

Manage books: Add, update, delete, and retrieve book details.

Manage patrons: Add, update, delete, and retrieve patron details.

Manage borrowing records: Allow patrons to borrow and return books.

Basic authentication to secure the API endpoints.

Logging middleware to log method calls, exceptions, and performance metrics.

**1. Running the Application**

**Prerequisites**

* **PHP**: 8.0 or higher
* **Composer**: Dependency management tool for PHP
* **Laravel**: The framework used for building the application
* **Database**: Set up (e.g., MySQL, SQLite)

**Steps**

1. **Clone the Repository**

git clone <repository-url>

cd <repository-directory>

1. **Install Dependencies**

Run Composer to install the required packages:

composer install

1. **Set Up Environment File**

Copy the example environment file and configure the database settings:

cp .env.example .env

Edit the .env file to configure your database and other environment settings.

1. **Generate Application Key**

php artisan key:generate

1. **Run Migrations**

Apply database migrations to set up the database schema:

php artisan migrate

1. **Start the Laravel Development Server**

php artisan serve

The application will be available at http://127.0.0.1:8000.

**2. API Endpoints**

**Authentication**

The API uses Basic Authentication. You need to include the Authorization header with the Basic scheme.

**Default Credentials:**

* **Username**: admin
* **Password**: admin

**Endpoints**

**Books**

* **List All Books**

GET /api/books

http://127.0.0.1:8001/api/books

Retrieves a list of all books.

* **Get Book Details**

GET /api/books/{id}

http://127.0.0.1:8001/api/books/10

Retrieves details of a specific book by its ID.

* **Add New Book**

POST /api/books

http://127.0.0.1:8001/api/books

Creates a new book entry. Requires a JSON payload with book details.

* **Update Book**

PUT /api/books/{id}

http://127.0.0.1:8001/api/books/10

Updates an existing book's information. Requires a JSON payload with updated details.

* **Delete Book**

DELETE /api/books/{id}

http://127.0.0.1:8001/api/books/10

Removes a book from the library by its ID.

**Patrons**

* **List All Patrons**

GET /api/patrons

http://127.0.0.1:8001/api/patrons

Retrieves a list of all patrons.

* **Get Patron Details**

GET /api/patrons/{id}

http://127.0.0.1:8001/api/patrons/20

Retrieves details of a specific patron by ID.

* **Add New Patron**

POST /api/patrons

http://127.0.0.1:8001/api/patrons

Creates a new patron entry. Requires a JSON payload with patron details.

* **Update Patron**

PUT /api/patrons/{id}

http://127.0.0.1:8001/api/patrons/20

Updates an existing patron's information. Requires a JSON payload with updated details.

* **Delete Patron**

DELETE /api/patrons/{id}

http://127.0.0.1:8001/api/patrons/20

Removes a patron from the system by their ID.

**Borrowing Records**

* **Borrow Book**

POST /api/borrow/{bookId}/patron/{patronId}

http://127.0.0.1:8001/api/borrow/9/patron/9

Allows a patron to borrow a book. Replace {bookId} and {patronId} with actual IDs.

* **Return Book**

PUT /api/return/{bookId}/patron/{patronId}

http://127.0.0.1:8001/api/return/3/patron/3

Records the return of a borrowed book by a patron. Replace {bookId} and {patronId} with actual IDs.

**3. Middleware**

**Basic Authentication Middleware**

Located at app/Http/Middleware/BasicAuthMiddleware.php. It checks the request for valid credentials and denies access if authentication fails.

**Logging Middleware**

Located at app/Http/Middleware/LogMiddleware.php. It logs request details, performance metrics, and any exceptions that occur.

**4. Testing API Endpoints**

You can use tools like Postman to interact with the API endpoints.

**Postman Setup:**

1. **Create a New Request**
   * Set the request method (GET, POST, PUT, DELETE).
   * Enter the API URL.
2. **Add Authentication**
   * Go to the Authorization tab.
   * Select Basic Auth.
   * Enter the username (admin) and password (admin).
3. **Send the Request**

Click the Send button to make the request and view the response.

**Summary**

This documentation outlines how to set up and interact with the Library Management System API built with Laravel. Ensure you configure your environment correctly, use basic authentication for secure access, and leverage the API endpoints to manage books, patrons, and borrowing records effectively.