

Library Management System - API Endpoints

Base URL

`http://localhost:5000/api`

Authentication Endpoints

1. User Registration

- POST /auth/register
- Body:

```
{  
  "name": "John Doe",  
  "email": "john@example.com",  
  "password": "password123",  
  "role": "user" // or "admin"  
}  
  
• Response:  
  
{  
  "success": true,  
  "message": "User registered successfully",  
  "user": {  
    "id": 1,  
    "name": "John Doe",  
    "email": "john@example.com",  
    "role": "user"  
  },  
  "token": "jwt_token_here"  
}
```

2. User Login

- POST /auth/login
- Body:

```
{  
  "email": "john@example.com",  
  "password": "password123"  
}  
  
• Response:  
  
{  
  "success": true,  
  "message": "User logged in successfully",  
  "user": {  
    "id": 1,  
    "name": "John Doe",  
    "email": "john@example.com",  
    "role": "user"  
  },  
  "token": "jwt_token_here"  
}
```

```

    "message": "Login successful",
    "user": {
        "id": 1,
        "name": "John Doe",
        "email": "john@example.com",
        "role": "user"
    },
    "token": "jwt_token_here"
}

```

3. Get Current User

- GET /auth/me
- Headers: Authorization: Bearer {token}
- Response:

```

{
    "success": true,
    "user": {
        "id": 1,
        "name": "John Doe",
        "email": "john@example.com",
        "role": "user"
    }
}

```

4. Logout

- POST /auth/logout
- Headers: Authorization: Bearer {token}
- Response:

```

{
    "success": true,
    "message": "Logged out successfully"
}

```

Books Endpoints

5. Get All Books (Browse/Search)

- GET /books?search=query&genre=Fiction&available=true&page=1&limit=12
- Query Params:
 - search (optional): Search by title, author, or ISBN
 - genre (optional): Filter by genre
 - available (optional): Filter by availability (true/false)

- `page` (optional): Page number for pagination
- `limit` (optional): Number of items per page

- **Response:**

```
{
  "success": true,
  "books": [
    {
      "id": 1,
      "title": "The Midnight Library",
      "author": "Matt Haig",
      "isbn": "978-0525559474",
      "genre": "Fiction",
      "publisher": "Viking",
      "year": 2020,
      "quantity": 5,
      "available": 3,
      "coverImage": "https://example.com/image.jpg",
      "description": "Book description here"
    }
  ],
  "pagination": {
    "currentPage": 1,
    "totalPages": 5,
    "totalBooks": 50,
    "limit": 12
  }
}
```

6. Get Book by ID

- `GET /books/:id`
- **Response:**

```
{
  "success": true,
  "book": {
    "id": 1,
    "title": "The Midnight Library",
    "author": "Matt Haig",
    "isbn": "978-0525559474",
    "genre": "Fiction",
    "publisher": "Viking",
    "year": 2020,
    "quantity": 5,
    "available": 3,
    "coverImage": "https://example.com/image.jpg",
```

```
        "description": "Book description here"
    }
}
```

7. Add New Book (Admin Only)

- **POST /books**
- **Headers:** Authorization: Bearer {admin_token}
- **Body:**

```
{
    "title": "The Midnight Library",
    "author": "Matt Haig",
    "isbn": "978-0525559474",
    "genre": "Fiction",
    "publisher": "Viking",
    "year": 2020,
    "quantity": 5,
    "available": 5,
    "coverImage": "https://example.com/image.jpg",
    "description": "Book description here"
}
```

- **Response:**

```
{
    "success": true,
    "message": "Book added successfully",
    "book": { /* book object */ }
}
```

8. Update Book (Admin Only)

- **PUT /books/:id**
- **Headers:** Authorization: Bearer {admin_token}
- **Body:** (same as Add New Book)
- **Response:**

```
{
    "success": true,
    "message": "Book updated successfully",
    "book": { /* updated book object */ }
}
```

9. Delete Book (Admin Only)

- **DELETE /books/:id**
- **Headers:** Authorization: Bearer {admin_token}

- Response:

```
{
  "success": true,
  "message": "Book deleted successfully"
}
```

Borrowings Endpoints

10. Borrow a Book

- POST /borrowings
- Headers: Authorization: Bearer {token}
- Body:

```
{
  "bookId": 1
}
```

- Response:

```
{
  "success": true,
  "message": "Book borrowed successfully",
  "borrowing": {
    "id": 1,
    "userId": 1,
    "bookId": 1,
    "borrowDate": "2025-11-28T10:00:00Z",
    "dueDate": "2025-12-12T10:00:00Z",
    "status": "active"
  }
}
```

11. Get User's Active Borrowings

- GET /borrowings/my-borrowings
- Headers: Authorization: Bearer {token}
- Response:

```
{
  "success": true,
  "borrowings": [
    {
      "id": 1,
      "book": {
        "id": 1,
        "title": "The Great Gatsby",
        "author": "F. Scott Fitzgerald",
        "genre": "Fiction"
      }
    }
  ]
}
```

```

        "title": "The Midnight Library",
        "author": "Matt Haig",
        "coverImage": "url"
    },
    "borrowDate": "2025-11-28",
    "dueDate": "2025-12-12",
    "status": "active",
    "daysRemaining": 14
}
]
}

```

12. Return a Book

- PUT /borrowings/:id/return
- Headers: Authorization: Bearer {token}
- Response:

```
{
  "success": true,
  "message": "Book returned successfully",
  "borrowing": {
    "id": 1,
    "returnDate": "2025-12-10",
    "status": "returned"
  }
}
```

13. Renew a Book

- PUT /borrowings/:id/renew
- Headers: Authorization: Bearer {token}
- Response:

```
{
  "success": true,
  "message": "Book renewed successfully",
  "borrowing": {
    "id": 1,
    "newDueDate": "2025-12-26"
  }
}
```

14. Get All Borrowings (Admin Only)

- GET /borrowings?status=active&page=1&limit=10
- Headers: Authorization: Bearer {admin_token}
- Query Params:

- **status** (optional): Filter by status (active, returned, overdue)
- **page, limit**: Pagination
- **Response:**

```
{
  "success": true,
  "borrowings": [
    {
      "id": 1,
      "user": {
        "id": 1,
        "name": "John Doe"
      },
      "book": {
        "id": 1,
        "title": "The Midnight Library",
        "author": "Matt Haig"
      },
      "borrowDate": "2025-11-28",
      "dueDate": "2025-12-12",
      "returnDate": null,
      "status": "active"
    }
  ],
  "pagination": { /* pagination info */ }
}
```

Reservations Endpoints

15. Reserve a Book

- **POST /reservations**
- **Headers:** Authorization: Bearer {token}
- **Body:**

```
{
  "bookId": 1
}
```

- **Response:**

```
{
  "success": true,
  "message": "Book reserved successfully",
  "reservation": {
    "id": 1,
    "userId": 1,
    "bookId": 1,
    "borrowDate": "2025-11-28",
    "dueDate": "2025-12-12",
    "status": "reserved"
  }
}
```

```

    "bookId": 1,
    "reservationDate": "2025-11-28",
    "expiryDate": "2025-12-05",
    "status": "active"
  }
}

```

16. Get User's Reservations

- **GET /reservations/my-reservations**
- **Headers:** Authorization: Bearer {token}
- **Response:**

```

{
  "success": true,
  "reservations": [
    {
      "id": 1,
      "book": {
        "id": 1,
        "title": "The Midnight Library",
        "author": "Matt Haig",
        "coverImage": "url"
      },
      "reservationDate": "2025-11-28",
      "expiryDate": "2025-12-05",
      "status": "active"
    }
  ]
}

```

17. Cancel Reservation

- **DELETE /reservations/:id**
- **Headers:** Authorization: Bearer {token}
- **Response:**

```

{
  "success": true,
  "message": "Reservation cancelled successfully"
}

```

18. Pickup Reserved Book

- **PUT /reservations/:id/pickup**
- **Headers:** Authorization: Bearer {token}
- **Response:**

```
{
  "success": true,
  "message": "Book picked up successfully",
  "reservation": {
    "id": 1,
    "status": "picked-up",
    "pickupDate": "2025-11-30"
  }
}
```

19. Get All Reservations (Admin Only)

- GET /reservations?status=active&page=1&limit=10
- Headers: Authorization: Bearer {admin_token}
- Response:

```
{
  "success": true,
  "reservations": [
    {
      "id": 1,
      "user": {
        "id": 1,
        "name": "John Doe"
      },
      "book": {
        "id": 1,
        "title": "The Midnight Library"
      },
      "reservationDate": "2025-11-28",
      "expiryDate": "2025-12-05",
      "status": "active"
    }
  ],
  "pagination": { /* pagination info */ }
}
```

History Endpoints

20. Get User's Borrowing History

- GET /history/borrowings
- Headers: Authorization: Bearer {token}
- Response:

```
{
```

```

    "success": true,
    "history": [
        {
            "id": 1,
            "book": {
                "id": 1,
                "title": "The Midnight Library",
                "author": "Matt Haig",
                "coverImage": "url"
            },
            "borrowDate": "2025-10-28",
            "dueDate": "2025-11-11",
            "returnDate": "2025-11-10",
            "status": "returned",
            "daysOverdue": 0
        }
    ]
}

```

21. Get User's Reservation History

- GET /history/reservations
- Headers: Authorization: Bearer {token}
- Response:

```

{
    "success": true,
    "history": [
        {
            "id": 1,
            "book": {
                "id": 1,
                "title": "The Midnight Library",
                "author": "Matt Haig",
                "coverImage": "url"
            },
            "reservationDate": "2025-10-20",
            "expiryDate": "2025-10-27",
            "pickupDate": "2025-10-25",
            "status": "picked-up"
        }
    ]
}

```

Members Endpoints (Admin Only)

22. Get All Members

- GET /members?search=query&page=1&limit=10
- Headers: Authorization: Bearer {admin_token}
- Query Params:
 - search (optional): Search by name or email
 - page, limit: Pagination
- Response:

```
{  
  "success": true,  
  "members": [  
    {  
      "id": 1,  
      "name": "John Doe",  
      "email": "john@example.com",  
      "role": "user",  
      "joinDate": "2025-01-15",  
      "activeBorrowings": 2,  
      "totalBorrowings": 15  
    }  
  ],  
  "pagination": { /* pagination info */ }  
}
```

23. Get Member by ID

- GET /members/:id
- Headers: Authorization: Bearer {admin_token}
- Response:

```
{  
  "success": true,  
  "member": {  
    "id": 1,  
    "name": "John Doe",  
    "email": "john@example.com",  
    "role": "user",  
    "joinDate": "2025-01-15",  
    "activeBorrowings": 2,  
    "totalBorrowings": 15,  
    "overdueBooks": 0  
  }  
}
```

24. Update Member

- PUT /members/:id
- Headers: Authorization: Bearer {admin_token}
- Body:

```
{  
  "name": "John Doe",  
  "email": "john@example.com",  
  "role": "user"  
}
```

- Response:

```
{  
  "success": true,  
  "message": "Member updated successfully",  
  "member": { /* updated member object */ }  
}
```

25. Delete Member

- DELETE /members/:id
- Headers: Authorization: Bearer {admin_token}
- Response:

```
{  
  "success": true,  
  "message": "Member deleted successfully"  
}
```

Reports Endpoints (Admin Only)

26. Get Popular Books Report

- GET /reports/popular-books?from=2025-10-01&to=2025-10-31
- Headers: Authorization: Bearer {admin_token}
- Response:

```
{  
  "success": true,  
  "data": [  
    {  
      "id": 1,  
      "title": "The Midnight Library",  
      "author": "Matt Haig",  
      "genre": "Fiction",  
      "totalBorrows": 152  
    }  
  ]  
}
```

```
        }
    ]
}
```

27. Get All Borrowings Report

- GET /reports/borrowings?from=2025-10-01&to=2025-10-31&status=all
- Headers: Authorization: Bearer {admin_token}
- Response:

```
{
  "success": true,
  "data": [
    {
      "id": 1,
      "memberName": "John Smith",
      "bookTitle": "The Midnight Library",
      "author": "Matt Haig",
      "borrowDate": "2025-10-15",
      "dueDate": "2025-11-15",
      "status": "active"
    }
  ]
}
```

28. Get All Reservations Report

- GET /reports/reservations?from=2025-10-01&to=2025-10-31
- Headers: Authorization: Bearer {admin_token}
- Response:

```
{
  "success": true,
  "data": [
    {
      "id": 1,
      "memberName": "Alice Cooper",
      "bookTitle": "The Midnight Library",
      "author": "Matt Haig",
      "reservationDate": "2025-10-20",
      "expiryDate": "2025-10-27",
      "status": "active"
    }
  ]
}
```

29. Get Overdue Books Report

- GET /reports/overdue-books
- Headers: Authorization: Bearer {admin_token}
- Response:

```
{  
  "success": true,  
  "data": [  
    {  
      "id": 1,  
      "memberName": "Mike Brown",  
      "bookTitle": "Dune",  
      "author": "Frank Herbert",  
      "dueDate": "2025-11-05",  
      "daysOverdue": 20,  
      "fineAmount": 10.00  
    }  
  ]  
}
```

30. Get Member Activity Report

- GET /reports/member-activity?from=2025-10-01&to=2025-10-31
- Headers: Authorization: Bearer {admin_token}
- Response:

```
{  
  "success": true,  
  "data": [  
    {  
      "id": 1,  
      "name": "John Smith",  
      "email": "john.smith@email.com",  
      "totalBorrowings": 25,  
      "activeBorrowings": 3,  
      "totalReservations": 8,  
      "overdueBooks": 0  
    }  
  ]  
}
```

Notifications Endpoints

31. Get User Notifications

- **GET /notifications**
- **Headers:** Authorization: Bearer {token}
- **Response:**

```
{  
  "success": true,  
  "notifications": [  
    {  
      "id": 1,  
      "type": "due-soon",  
      "title": "Book Due Soon",  
      "message": "Your book 'The Midnight Library' is due in 2 days",  
      "isRead": false,  
      "createdAt": "2025-11-26T10:00:00Z"  
    }  
  ],  
  "unreadCount": 5  
}
```

32. Mark Notification as Read

- **PUT /notifications/:id/read**
- **Headers:** Authorization: Bearer {token}
- **Response:**

```
{  
  "success": true,  
  "message": "Notification marked as read"  
}
```

33. Mark All Notifications as Read

- **PUT /notifications/read-all**
- **Headers:** Authorization: Bearer {token}
- **Response:**

```
{  
  "success": true,  
  "message": "All notifications marked as read"  
}
```

34. Delete Notification

- **DELETE /notifications/:id**
- **Headers:** Authorization: Bearer {token}

- Response:

```
{  
  "success": true,  
  "message": "Notification deleted successfully"  
}
```

Dashboard Endpoints

35. Get User Dashboard Stats

- GET /dashboard/user
- Headers: Authorization: Bearer {token}
- Response:

```
{  
  "success": true,  
  "stats": {  
    "activeBorrowings": 3,  
    "activeReservations": 1,  
    "overdueBooks": 0,  
    "totalBorrowed": 25  
  }  
}
```

36. Get Admin Dashboard Stats

- GET /dashboard/admin
- Headers: Authorization: Bearer {admin_token}
- Response:

```
{  
  "success": true,  
  "stats": {  
    "totalBooks": 500,  
    "totalMembers": 150,  
    "activeBorrowings": 120,  
    "overdueBooks": 15,  
    "totalReservations": 25  
  }  
}
```

Notes

Authentication

- All protected routes require JWT token in Authorization header:
`Authorization: Bearer {token}`
- Tokens should be stored in localStorage/sessionStorage on frontend
- Token should be included in all authenticated requests

Error Responses

All error responses follow this format:

```
{  
  "success": false,  
  "message": "Error message here",  
  "errors": [] // Optional validation errors  
}
```

Status Codes

- 200 - Success
- 201 - Created
- 400 - Bad Request (validation errors)
- 401 - Unauthorized (no token or invalid token)
- 403 - Forbidden (insufficient permissions)
- 404 - Not Found
- 500 - Internal Server Error

Pagination

For endpoints with pagination: - Default `limit: 10` - Default `page: 1` - Max `limit: 100`