A. Authentication Endpoints

1. POST /api/auth/login

Request:

json

```
{
  "username": "string",
  "password": "string"
}
```

Response:

json

```
{
  "token": "jwt_token_string",
  "user": {
    "id": "number",
    "username": "string",
    "email": "string",
    "avatar": "url_string"
    “role”:”string”
  }
}
```

2. GET /api/auth/me

Response:

json

```
{
  "user": {
    "id": "number",
    "username": "string",
    "email": "string",
    "avatar": "url_string",
     }
}
```

B. Books Endpoints

3. GET /api/latest-news

Response:

json

```
{
  "news": [
{
      “id”:”number”,
      “name”:”string”,
      “details”:”string”,
      “coverImage”:”string”,
      “creationDate”:”date-string”
}
],
"pagination": {
    "page": "number",
    "limit": "number",
    "totalRecords": "number",
    "totalPages": "number"
  }


}
```

4. GET /api/books

GET /api/books?page=1&limit=10&genre=Fiction&search=harry

Query Parameters:

- page (optional): number
- limit (optional): number
- genre (optional): string
- search (optional): string

Response:

json

```
{
  "books": [
    {
      "id": "number",
      "title": "string",
      "author": "string",
      "coverImage": "url_string",
      "rating": "number",
      "genre": "string",
      “tag”:”string”,
      "publishedDate": "date_string",
      "available": "boolean"
    }
  ],

  "pagination": {
    "page": "number",
    "limit": "number",
    "totalRecords": "number",
    "totalPages": "number"
  }

}
```

5. GET /api/books/{id}

Response:

json

```
{
  "id": "number",
  "title": "string",
  "author": "string",
  "coverImage": "url_string",
  "description": "string",(description and summary are same)
  "rating": "number", ==(separate table)
  "genre": "string",
  “tag”:”string”,
  "publishedDate": "date_string",
  "publisher": "string",
  "isbn": "string",
  "available": "boolean",
  "audioClip": "url_string_or_null",
  "pdfUrl": "url_string",
  "totalRatings": "number",
  "totalReviews": "number",
  “status”:”status”,
}
```

6. GET /api/books/{id}/borrow

Request:

json

```
{
  "id": "number"
}
```

Response:

json

```
{
  "title": "string",
  "author": "string",
  "coverImage": "string",
  "availableFrom": "date_string",
  "mustReturnBy": "date_string",
}
```

7. POST /api/books/{id}/borrow

Request:

json

```
{
  "id": "number"
}
```

Response:

json

```
{
  "success": "boolean",
  “borrowingDate”:””string,
  "dueDate": "date_string",
  "message": "string"
}
```

C. Library Endpoints

8. GET /api/books/{id}/reviews

Response:

json

```
{
  "reviews": [
    {
      "id": "number",
      "userId": "number",
      "userName": "string",
      "userAvatar": "url_string",
      "rating": "number",
      "comment": "string",
      "date": "date_string",
      "helpful": "number",
      }
  ],
  "averageRating": "number",
  "totalReviews": "number",
  "ratingBreakdown": {
    "5": "number",
    "4": "number",
    "3": "number",
    "2": "number",
    "1": "number"
  }
}
```

9. POST /api/books/{id}/reviews

Request:

json

```
{
  "userId": "number",
  "rating": "number",
  "comment": "string"
}
```

Response:

json

```
{
  "success": "boolean",
  "reviewId": "number",
  "message": "string"
}
```

10. POST /api/reviews/{id}/helpful

Request:

json

```
{
  "userId": "number",
  "helpful": "boolean"
}
```

Response:

json

```
{
  "success": "boolean",
  "helpfulCount": "number"
}
```

D. User Library Endpoints

11. GET /api/users/{id}/library

Response:

json

```
{
  "borrowedBooks": [
    {
      "id": "number",
      "title": "string",
      "author": "string",
      "coverImage": "url_string",
      "borrowDate": "date_string",
      "dueDate": "date_string"
    }
  ],
}
```

E. Admin Endpoints

Dashboard

12. GET /api/admin/dashboard/stats

```
{
  "success": "boolean",
  "data": {
    "borrowedBooks": "number",
    "returnedBooks": "number",
    "overdueBooks": "number",
    "totalBooks": "number",
    "newMembers": "number",
    "borrowsPending": "number"
  }
}
```

13. GET /api/admin/dashboard/checkout-stats

```
{
  "success": true,
  "data": {
    "totalBorrowed": "number",
    "totalReturned": "number",
    "totalOverdue": "number",
    "borrowedToday": "number",
    "returnedToday": "number",
    "overdueToday": "number",
    "activeBorrowers": "number"
  }
}
```

14. GET /api/admin/dashboard/overdue-history

```
{
  "success": true,
  "data": [
    {
      "bookId": "number",
      "bookName": "string",
      "username": "string",
     “email”:”string”,
     “dueDate”:”date_string”

    }
      ],
    "pagination": {
    "page": "number",
    "limit": "number",
    "totalRecords": "number",
    "totalPages": "number"
  }

}
```

15. GET /api/admin/dashboard/borrowed-request

```
{
  "success": "boolean",
  "data": [
    {
     "requestId": "number",
     "bookId": "number",
      "bookName": "string",
      "userName": "string",
      "borrowedDate": "date",
      "returnedDate": "date",
      "status": "string"   // (pending | accepted | rejected)
    }
  ],
  "pagination": {
    "page": "number",
    "limit": "number",
    "totalRecords": "number",
    "totalPages": "number"
  }
}
```

15. PATCH /api/admin/dashboard/borrowed-request/{requestId}

Request

```
{
  "status": "string"   // accepted | rejected
}
```

Response

```
{
  "success": "boolean",
  "message": "string"
}
```

Manage Book

16. GET /api/manage-books

Response:

json

```
{
  "books": [
    {
      "id": "number",
      "title": "string",
      "author": "string",
      "coverImage": "url_string",
      "genre": "string",
      "numberofcopy": "number",
      “”g
    }
  ],

  "pagination": {
    "page": "number",
    "limit": "number",
    "totalRecords": "number",
    "totalPages": "number"
  }

}
```

17. POST /api/manage-books

Request:

json

```
{
  "title": "string",
  "author": "string",
  "coverImage": "url_string",
  "description": "string",
  "genre": "string",
  “tag”:”string”,
  "publishedDate": "date_string",
  "publisher": "string",
  "numberofcopy": "number",
  "isbn": "string",
  "audioClip": "url_string_or_null",
  "pdfUrl": "url_string",
  “status”:”status”,


}
```

Response

```
{
  “bookId”:”number”,
  "success": "boolean",
  "message": "string"
}
```

18. PUT /api/manage-books/{bookId}

Request:

json

```
{
  "title": "string",
  "author": "string",
  "coverImage": "url_string",
  "genre": "string",
  "numberofcopy": "number"

}
```

Response

```
{
  "success": "boolean",
  "message": "string"
}
```

19. DELETE /api/manage-books/{bookId}

Response:

```
json
{
  "success": "boolean",
  "message": "string"
}
```

Manage Category

20. GET /api/category

Response:

json

```
{
  "category": [
    {
      "id": "number",
      "name": "string",
      "slug": "string",
    }
  ],

  "pagination": {
    "page": "number",
    "limit": "number",
    "totalRecords": "number",
    "totalPages": "number"
  }

}
```

21. POST /api/category

Request

```
{
  “name”:”string”,
  “slug”:”string”
}
```

Response

```
{
  "success": "boolean",
  “categoryId”:”number”,
  "message": "string"
}
```

22. PUT /api/category/{categoryId}

Request

```
{
  “name”:”string”,
  “slug”:”string”
}
```

Response

```
{
  "success": "boolean",
  “categoryId”:”number”,
  "message": "string"
}
```

23. DELETE /api/category/{categoryId}

```
{
  "success": "boolean",
  "categoryId": "number",
  "message": "string"
}
```

HTTP Status Codes

- `200` - Success
- `201` - Created
- `400` - Bad Request
- `401` - Unauthorized
- `403` - Forbidden
- `404` - Not Found
- `409` - Conflict
- `422` - Validation Error
- `500` - Internal Server Error
- `503` - Service Unavailable

Authentication

Most endpoints require authentication via JWT token in the Authorization header:

Authorization: Bearer <jwt_token>