

# API Documentation for Book Nest

## 1. Authentication APIs

### a) Register User

**Endpoint:** /auth/register

**Method:** POST

Description: Registers a new user.

**Request Body:**

```
{  
  "name": "string",  
  "email": "string",  
  "role": "string",  
  "password": "string"  
}
```

**Response:**

Success: 201 Created

```
{  
  "message": "User registered successfully"  
}
```

Failure: 400 Bad Request, 500 Internal Server Error

## **b) Login User**

**Endpoint:** /auth/login

**Method:** POST

Description: Authenticates a user and returns a JWT token.

**Request Body:**

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

**Response:**

Success: 200 OK

```
{  
  "token": "string",  
  "role": "string"  
}
```

Failure: 404 Not Found, 400 Bad Request, 500 Internal Server Error

---

## **2. Book APIs**

### **a) Add a Book**

**Endpoint:** /book/add

**Method:** POST

Description: Adds a new book to the system (Admin only).

Request Headers: Authorization: Bearer token

**Request Body:**

```
{  
  "title": "string",  
  "author": "string",  
  "copies": "number",  
  "image": "file (optional)"  
}
```

**Response:**

Success: 201 Created

```
{  
  "message": "Book added successfully",  
  "book": {  
    "title": "string",  
    "author": "string",  
    "copies": "number",  
    "image": "string (URL)"  
  }  
}
```

Failure: 500 Internal Server Error

**b) Delete a Book**

**Endpoint:** /book/:id/delete

**Method:** DELETE

Description: Deletes a book by ID (Admin only).

Request Headers: Authorization: Bearer token

**Response:**

Success: 200 OK

```
{  
  "message": "Book deleted successfully"  
}
```

Failure: 500 Internal Server Error

**c) Get All Books**

**Endpoint:** /book/

**Method:** GET

Description: Fetches all books in the library (Can be accessed by anyone).

**Response:**

Success: 200 OK

```
[  
  {  
    "_id": "string",  
    "title": "string",  
    "author": "string",  
    "copies": "number",  
    "image": "string (URL)"  
  }  
]
```

Failure: 500 Internal Server Error

#### d) **Issue a Book**

**Endpoint:** /book/issue

**Method:** POST

Description: Issues a book to a user (Librarian only).

Request Headers: Authorization: Bearer token

**Request Body:**

```
{  
  "bookId": "string"  
}
```

**Response:**

Success: 200 OK

```
{  
  "message": "Book issued successfully",  
  "book": {  
    "copies": "number",  
    "issued": true,  
    "lastIssued": "string (ISO date)"  
  }  
}
```

Failure: 404 Not Found, 400 Bad Request, 500 Internal Server Error

#### e) **Get Issued Books**

**Endpoint:** /book/issued-books

**Method:** GET

Description: Retrieves all issued books (Librarian Only).

**Response:**

Success: 200 OK

```
[  
  {  
    "_id": "string",  
    "title": "string",  
    "author": "string",  
    "copies": "number",  
    "issued": true,  
    "lastIssued": "string (ISO date)"  
  }  
]
```

Failure: 500 Internal Server Error

---

### 3. User Profile APIs

#### a) Get Profile

**Endpoint:** /profile

**Method:** GET

Description: Retrieves the authenticated user's profile.

Request Headers: Authorization: Bearer token

**Response:**

Success: 200 OK

```
{  
  "_id": "string",
```

```
"name": "string",  
"email": "string",  
"role": "string"  
}
```

Failure: 500 Internal Server Error

**This API is implemented in:**

- a) Librarian: /librarian/profile
- b) Member: /member/profile
- c) Admin: /admin/profile

---