

# Book Platform

## API Documentation

---

### Base URL

http://localhost:3001/api

---

### Auth Endpoints (/auth)

#### Register a New User

- **POST** /auth/register

- **Body (JSON):**

```
{  
  "username": "yourUsername",  
  "email": "your@email.com",  
  "password": "yourPassword"  
}
```

- **Response:**

201 Created

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

---

### Login

- **POST** /auth/login

- **Body (JSON):**

```
{  
  "email": "your@email.com",  
  "password": "yourPassword"  
}
```

- **Response:**  
200 OK  
{ "message": "Login successful", "user": { "username": "...", "email": "..." } }
- 

### Get All Users (Admin)

- **GET** /auth/users
  - **Response:**  
200 OK  
Returns an array of users (without passwords).
- 

### Delete a User (Admin)

- **DELETE** /auth/users/:id
  - **Response:**  
200 OK  
{ "message": "User deleted" }
- 

### Books Endpoints (/books)

#### Add a New Book

- **POST** /books
  - **Body:** multipart/form-data
    - title (string)
    - author (string)
    - description (string)
    - Category (string)
    - [image](#) (file, optional)
  - **Response:**  
201 Created  
Returns the created book object.
-

## Get All Books

- **GET** /books
  - **Response:**  
200 OK  
Returns an array of books.
- 

## Get Book by ID

- **GET** /books/:id
  - **Response:**  
200 OK  
Returns the book object.
- 

## Delete Book by ID

- **DELETE** /books/:id
  - **Response:**  
200 OK  
{ "message": "Book deleted" } or error if not found.
- 

## Special Books Endpoints (/specialbooks)

### Add a New Special Book

- **POST** /specialbooks
- **Body:** multipart/form-data
  - title (string)
  - author (string)
  - description (string)
  - Category (string)
  - [image](#) (file, optional)
- **Response:**  
201 Created  
Returns the created special book object.

---

## Get All Special Books

- **GET** /specialbooks
- **Response:**  
200 OK  
Returns an array of special books.

---

## Get Special Book by ID

- **GET** /specialbooks/:id
- **Response:**  
200 OK  
Returns the special book object.

---

## Delete Special Book by ID

- **DELETE** /specialbooks/:id
- **Response:**  
200 OK  
{ "message": "Special book deleted successfully" } or error if not found.

---

## Set New Release for Special Book

- **PATCH** /specialbooks/:id/newrelease
- **Body (JSON):**  
{ "newReleasesId": ".." }
- **Response:**  
200 OK  
Returns the updated special book.

---

## Set ForUId for Special Book

- **PATCH** /specialbooks/:id/foru

- **Body (JSON):**

```
{ "forUId": "..."} 
```

- **Response:**

200 OK

Returns the updated special book.

---

### Set TrainingId for Special Book

- **PATCH** /specialbooks/:id/training

- **Body (JSON):**

```
{ "trainingId": "..."} 
```

- **Response:**

200 OK

Returns the updated special book.

---

### Comments Endpoints (/comments)

#### Get All Comments for a Book

- **GET** /comments/:bookModel/:bookId

- **Response:**

200 OK

Returns an array of comments.

---

#### Add a Comment

- **POST** /comments

- **Body (JSON):**

```
{  
  
  "bookId": "...",
```

```
"bookModel": "...",  
"user": "...",  
"comment": "...",  
"rating": 5  
}
```

- **Response:**  
201 Created  
Returns the created comment.
- 

## Favorites Endpoints (/favorites)

### Get All Favorites for a User

- **GET** /favorites/:user
  - **Response:**  
200 OK  
Returns an array of favorite books for the user.
- 

### Add a Book to Favorites

- **POST** /favorites
- **Body (JSON):**

```
{  
  "user": "...",  
  "bookId": "...",  
  "bookModel": "..."  
}
```

- **Response:**  
201 Created  
Returns the created favorite.

---

## Remove a Book from Favorites

- **DELETE** /favorites

- **Body (JSON):**

```
{  
  "user": "...",  
  "bookId": "...",  
  "bookModel": "..."  
}
```

- **Response:**

200 OK

```
{ "success": true }
```