LIBRARY MANAGEMENT SYSTEM

Submitted By: Vedavathi and Neeru Rani

ABSTRACT

Library management system is a project which aims in developing a computerized system to maintain all the daily work of a library .This project has many features which are generally not available in normal library management systems like facility of user login. It also has a facility of admin login through which the admin can monitor the whole system . It also has a facility where students after logging in their accounts can see list of books issued and its issue date and return date. Overall this project of ours is being developed to help the students as well as staff of the library to maintain the library in the best way possible and also reduce the human efforts.

ADMIN-REGISTER:-

This feature is used by the admin to register into the system. They are required to enter admin name,emailId,password,address and phone number before they are allowed to enter the system.

post API- **http://localhost:3000/admin/register**

body: {

```
        "name": "monu",
    "email":"monu1234@gmail.com",
    "password":"monu1234",
    "address":"haryana",
    "phoneno":"123456789"
}
```

success response : {


```
"jwt": null,
"_id": "5e8df0247a631c001787af42",
"adminName": "monu2",
"adminEmailId": "monu123456@gmail.com",
"adminPassword": "$2a$10$05I7X9uqfAtoDBwHzjFcNeB15iDPvbpF.S5pgT5nCnLTLOkdb2GQu",
"adminAddress": "haryana",
"adminPhoneNo": 123456789,
"date": "2020-04-08T15:39:16.696Z",
"__v": 0
```

}

ADMIN-LOGIN:-

This feature is used by the admin to login into the system. They are required to enter admin emailId and password number before they are allowed to enter the system.It will return an authentication token to add a user/book and to issue/return a book.

post API- **http://localhost:3000/admin/login**

Body : {
"email":"monu1234@gmail.com",
"password":"monu1234"
}
success response : {

"jwttoken":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjVlOGRjZTJiYjdkZjViMzdmNGRhNWUyNCIsImlhdCI6MTU4NjM1OTQ1MCwiZXhwIjoxNTg5OTU5NDUwfQ.UPVoKksQm_TChHunwZJLQbrOXpQ14YkDpphf9l56p0l",
"status": "successful login",
"code": "202"
}

USER-CREATE:-

This feature is used by the admin to create a user into the system.They are required to enter name,emailId,password,address and phone number.

post API- **http://localhost:3000/users**

Headers:-

Authorization: jwt token from login

Body:-


```
{
        "name": "monu",
    "email":"monu1234@gmail.com",
    "password":"monu1234",
    "address":"haryana",
    "phoneno":"123456789"
  }
```


success response : {


```
  "message": "create user",
  "user": {
    "_id": "5e8df1a07a631c001787af43",
    "name": "monu",
    "email": "monu1234@gmail.com",
    "password": "$2b$10$3DfVqHtXMLYeEQ.ChvHBauUlvcA4Zk4eS1s0cyMPCdKkNTx3xX8r.",
    "address": "haryana",
    "phoneno": 123456789,
```

```
    "createdAt": "2020-04-08T15:45:36.159Z",
    "updatedAt": "2020-04-08T15:45:36.159Z",
    "__v": 0
  }
}
```


USER-GET:-
In this feature we get the all users list and a particular user.

get API- **http://localhost:3000/users** **this is for get all users list.**

get API-**http://localhost:3000/users/id** **this is for a particular user.**

success response :-


```
[
  {
    "_id": "5e8df1a07a631c001787af43",
    "name": "monu",
    "email": "monu1234@gmail.com",
    "password": "$2b$10$3DfVqHtXMLYeEQ.ChvHBauUIvcA4Zk4eS1s0cyMPCdKkNTx3xX8r.",
    "address": "haryana",
    "phoneno": 123456789,
    "createdAt": "2020-04-08T15:45:36.159Z",
    "updatedAt": "2020-04-08T15:45:36.159Z",
    "__v": 0
  }
]
```


USER-UPDATE:-
In this feature we update all details for a user. They are required to enter _id of the user in body.

put API-**http://localhost:3000/users**

DELETE-USER:-
In this feature we DELETE users.

delete API-**http://localhost:3000/users/id**

BOOK-CREATE:-

This feature is used by the admin to create a book into the system.They are required to enter name,stock,author and title .

post API- -**http://localhost:3000/books**

body:- {

"name": "math",

"author":"R.s. CHAND",

"stock":"20"

}

success response:-{


```
  "message": "create book",
  "book": {
    "time": 1586359555296,
    "_id": "5e8df30e7a631c001787af44",
    "name": "math",
    "stock": 20,
    "createdAt": "2020-04-08T15:51:42.417Z",
    "updatedAt": "2020-04-08T15:51:42.417Z",
    "__v": 0
  }
}
```


BOOK-GET:-

In this feature we get the all books list and a particular book.

get API--**http://localhost:3000/books** **this is for get all books list.**

get API-**http://localhost:3000/users/id** **this is for a particular book.**

BOOK-UPDATE:-

In this feature we update stock for a book. They are required to enter _id of the book in body.

put API--**http://localhost:3000/books**

body:- {

"_id": "5e8df30e7a631c001787af44",

"name": "math",

"author":"R.s. CHAND",

"stock":"30"

```
  }
 success response:-{
  "message": "update book",
  "result": {
    "n": 1,
    "nModified": 1,
    "opTime": {
      "ts": "6813369774291025921",
      "t": 9
    },
    "electionId": "7fffffff0000000000000009",
    "ok": 1,
    "$clusterTime": {
      "clusterTime": "6813369774291025921",
      "signature": {
        "hash": "YTeHkdOlkNGa6ZqFp1J8ZV5SPMU=",
        "keyId": "6795219646225055747"
      }
    },
    "operationTime": "6813369774291025921"
  }
}
```


DELETE-USER:-
In this feature we DELETE books.

delete API-**http://localhost:3000/books/id**

Issue a book :-
In this feature we issue a book. They are required to enter userId,bookId and "returnStatus": false . for this user all update when we get this user we also see book details.

post API-**http://localhost:3000/logs/addToLog**
Body:-


```
{
        "bookId": "5e8df30e7a631c001787af44",
        "userId": "5e8df1a07a631c001787af43",
     "returnStatus":false
  }
 success response:-{
```

```
    "_id": "5e8df4787a631c001787af46",
    "userId": "5e8df1a07a631c001787af43",
    "bookId": "5e8df30e7a631c001787af44",
    "returnStatus": false,
    "date": "2020-04-08T15:57:44.627Z",
    "__v": 0
}
```


UPDATE-LOG:-
In this feature we update the book issue or return .

patch API-**http://localhost:3000/logs/logsid**

DELETE-LOG:-
In this feature we delete the book logs.

delete API-**http://localhost:3000/logs/logsid**

GET-LOG:-
In this feature we get all the book logs.

get API-**http://localhost:3000/logs/logsid**