iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** XII **Month of publication:** December 2022

DOI: https://doi.org/10.22214/ijraset.2022.48449

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue XII Dec 2022- Available at www.ijraset.com

Book Record Management API

Tushar Agarwal[1], Shruti Ahuja[2]
[1]*Student, B.Tech, CSE, Mahavir Swami Institute of Technology, Guru Gobind Singh Indraprastha University, Delhi, India*
[2]*Project Guide, Department of Computer Science Engineering, Mahavir Swami Institute of Technology, Guru Gobind Singh Indraprastha University, Delhi, India*

Abstract: The Book Record Management API aims to track all the books and user information and to subscription type, the user wishes to enroll. The API will be able to handle all the necessary operations for the display of complete book or user information.

I. INTRODUCTION

The number of services providing publicly available web Application Programming Interfaces (APIs) has been growing rapidly. Several studies point out that developers have moved from Simple Object Access Protocol (SOAP) or Remote Procedure Call (RPC) to deploying Representational State Transfer (REST) web services, as the means for consumers to use their services. This is corroborated by major websites like Google, Facebook, or Twitter, which are now deploying REST services to provide easy access to their valuable data resources, while promoting their businesses.
REST technology is generally preferred to the more robust Simple Object Access Protocol (SOAP) technology because REST uses less bandwidth, simple and flexible making it more suitable for internet usage. It's used to fetch or give some information from a web service. All communication done via REST API uses only HTTP request.

A. Problem Statement

Libraries are used to store books, but require a system to navigate to a specific book or specific content within a book. A Book record management system is an infrastructure that allows users to search books and book content, add/remove, and download selected books.

B. Working

A request is sent from client to server in the form of a web URL as HTTP GET or POST or PUT or DELETE request. After that, a response comes back from the server in the form of a resource which can be anything like HTML, XML, Image, or JSON. But now JSON is the most popular format being used in Web Services.

C. Objective

The objective of the "Book Record Management API" is to handle the entire activity of a library. The API keeps track of all the information about the books in the library, their cost, their complete details and total number of books available in the Library. The user will find it easy in this automated system rather than using the manual writing system. The system contains a database where all the information will be stored safely. The system is user-friendly and error free.

II. BACKGROUND AND RELATED WORK

The API will expose the endpoints as below:

1) Produce a list of all the books with their authors, genres, issued year and issued user.
2) Produce a list of all the users with their issued books and subscription type.
3) Provide the support for creating, updating and deleting a book.
4) Provide the support for creating, updating and deleting a user.

III. METHODOLOGY

A request (the URL you access) contains four components, which are:

1) The endpoint, which is the URL with the structure root-endpoint/?
2) The method with one of the five possible types (GET, POST, PUT, PATCH, DELETE)

©IJRASET: All Rights are Reserved

3) The headers, serving various functions, including authentication and providing information about the content of the body (you can use the -H or --header option to send HTTP headers)

4) Data (or body), that's what you send to the server through the -d or --data option with POST, PUT, PATCH, or DELETE requests.

The HTTP requests allow you to operate with the database, such as:

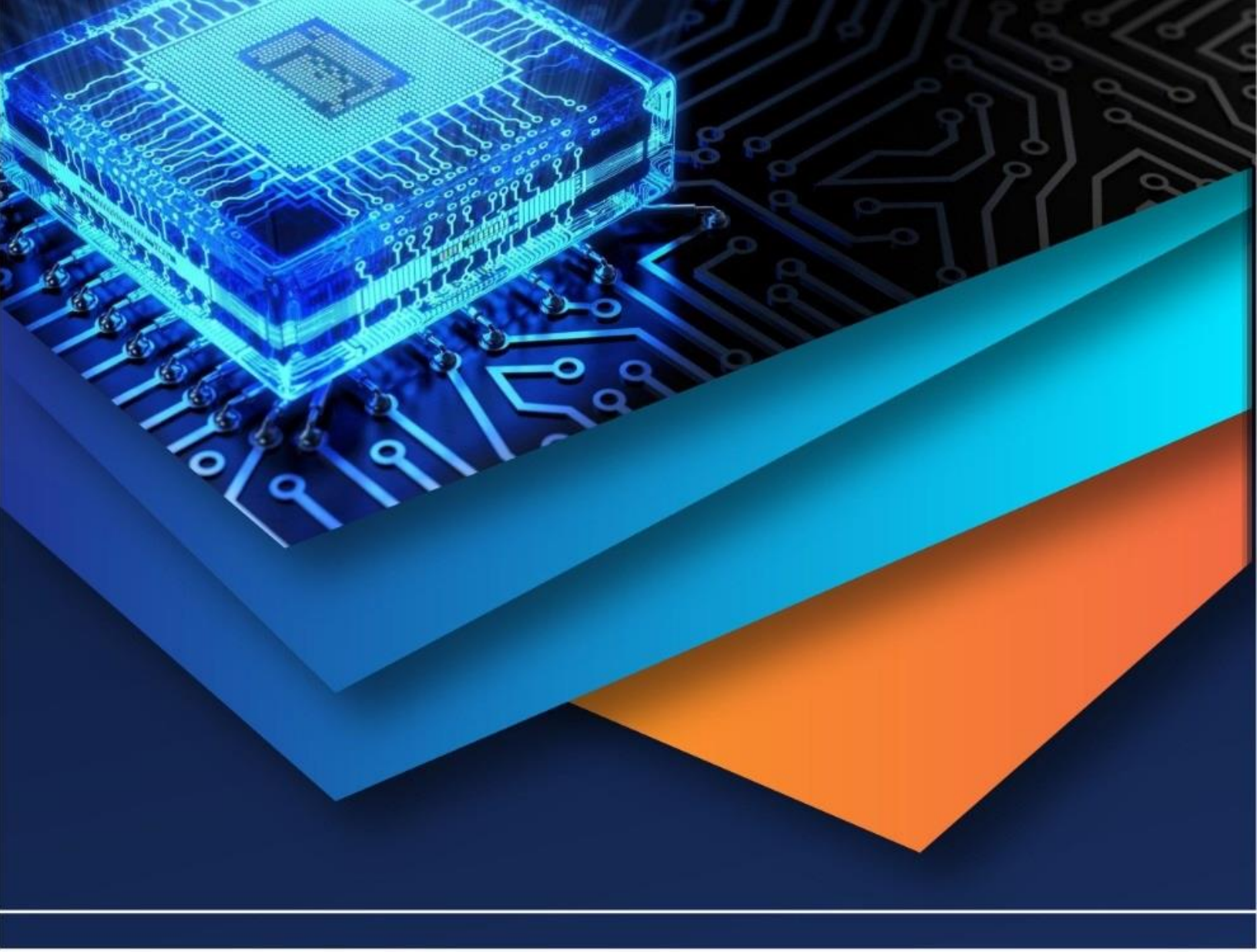
- POST request to create records
- GET request to read or get a resource (a document or image, a collection of other resources) from the server
- PUT and PATCH requests to update records
- DELETE request to delete a resource from a server

IV. CONCLUSION

This paper mainly focuses on how we can improve the traditional method of working of a library because the traditional method includes doing all the things in manual mode which is slow, less efficient, less secure, and difficult to manage. The solution to this is an online book record management API which take care of all the work by automating and digitizing the whole process. Our application is based on nodejs and is linked to a Non-SQL database MONGODB. The backend is supported and connected with database using mongoose package, its libraries and APIs. With the increase in the workload of the library, new features can be added to the existing application to make it relevant in the future as well.

REFERENCES

[1] PrasannaPillai, SonalSingn, ShreyaThakare, "Android Application for Library Automation".
[2] Akazue Maureen and Ojeme Blessing "Design of Automated Library Management System for States Universities in Nigeria".
[3] R.Dinesh, S.R.ArunPravin, M.Aravindhan, D.Rajeshwari. "Implementing BRICKS, a Digital Library Management System ".
[4] R.Dinesh, S.R.ArunPravin, M.Aravindhan, D.Rajeshwari. "Library Access System Smartphone Application Using Android ".
[5] https://auth0.com/blog/rest-architecture-part-1-building-api/
[6] https://restfulapi.net/rest-architectural-constraints/
[7] https://www.w3schools.com/js/js_api_intro.asp



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)