Project - 3 Revision

By - Ravi Kumar Sharma (fn-plu-85MKV)

Project Name: Book-Management

1.Project overview:

Role: Backend Developer, Tech Stack: Javascript, Nodejs, MongoDB.

Created a books management website backend with multiple api's for user , books and reviewers as well,
Also used jwt authentication for protected api's .

This is a backend projects where user can register and login himself where users can read reviews and give ratings on books. like Goodreads.com

2.Problem statement:

- We need to develop three schemas, to store the details of books, users and reviewers.
- We need to build an POST API to generate and store book data.
- We need to create another API to generate and save users details into the database
- To retrieve the data from the database we need to create a GET API.

3.Approach:

- Our server must first be set up using the express library (creating an index.js page).
- In order to store the data in a structured manner, a schema must be created using Mongoose.
- For the first POST API, we need to send the input data in the request body, and then the input must be validated to ensure that the input is valid.
- Following input from the request body, we must save the data in the database.
- For the second API, we must take the request body as input, and apply validations to ensure that the information is correct.
- After receiving data from the request body, we must store it in the database.
- For the get book API we need to pass the input in the request parameters, we must also perform validations on the input.

- After receiving the input and validating it we then need to return the details of the books by retrieving it from the database.

- For the update book API we need to pass the input in the request parameters, we must also perform validations on the input.
- After receiving the input and validating it we then need to return the details of the updated books by retrieving it from the database.

- For the delete book API we need to pass the input in the request parameters, we must also perform validations on the input.
- After receiving the input and validating it we then need to return the details of the deleted books by retrieving it from the database.

- For the review API we need to pass the input in the request parameters, we must also perform validations on the input and then store it in the database.
- After receiving the input and validating it we then need to return the details of the review by retrieving it from the database.

- For the update review API we need to pass the input in the request parameters, we must also perform validations on the input and then store it in the database.
- After receiving the input and validating it we then need to return the details of the updated review by retrieving it from the database.

- For the delete review API we need to pass the input in the request parameters, we must also perform validations on the input and then store it in the database.
- After receiving the input and validating it we then need to return the details of the deleted review by retrieving it from the database.

5.Final code solution (GitHub link):

https://github.com/RaviiSharma/Project-Books-Managament