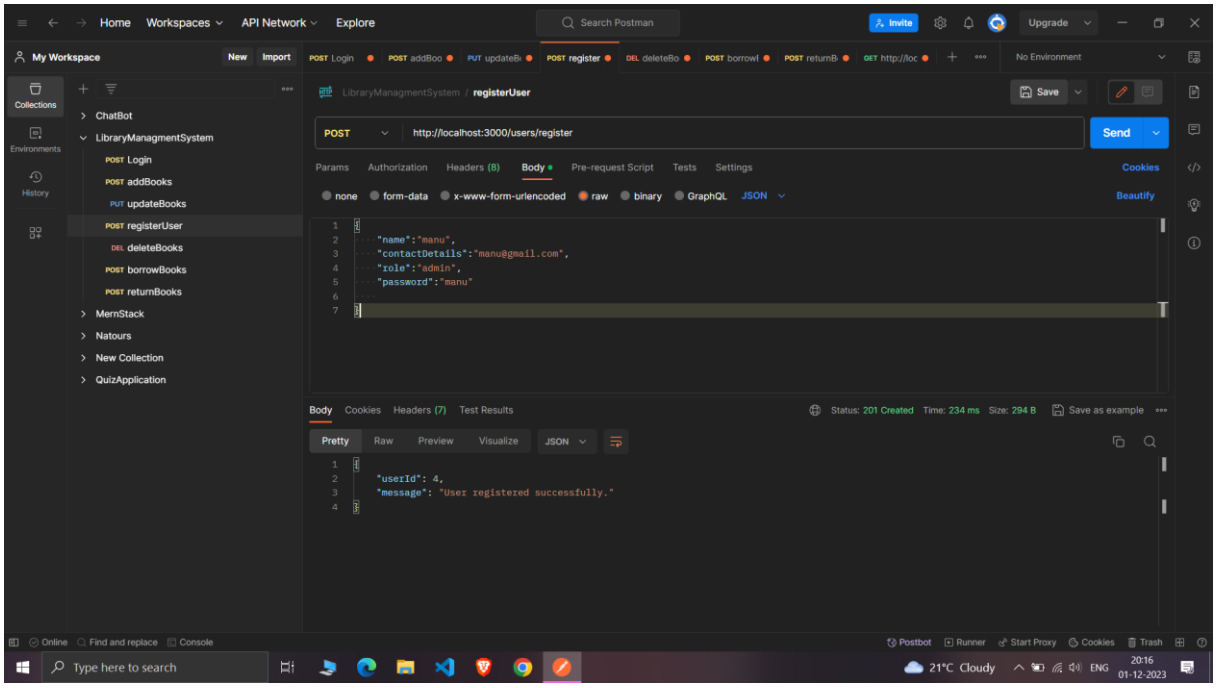
Instructions On How To Run API Endpoints

Registration Of User:-

Endpoint: POST /users/register

Body:

```
{
 "name": "John Doe",
"contactDetails": "john@example.com",
"role": "user",
"password": "your_password"
}
```

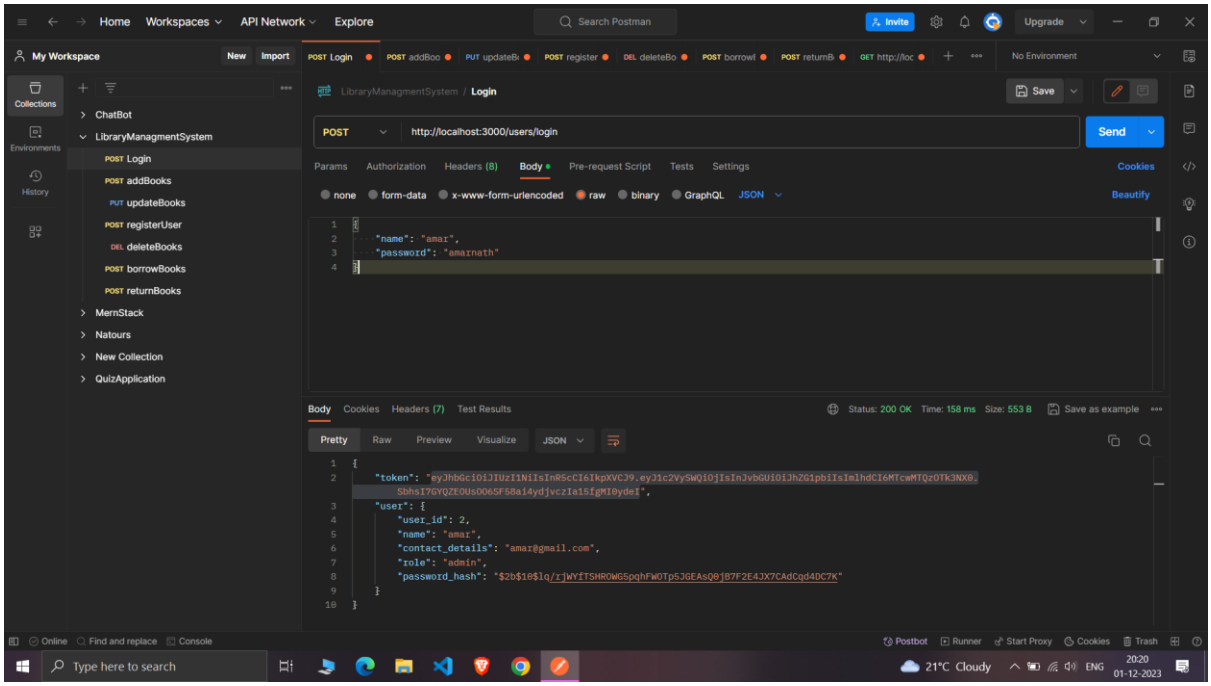


User Login

Endpoint: POST /users/login

Body:

```
{
"name": "amar",
"password": "amarnath"
}
```



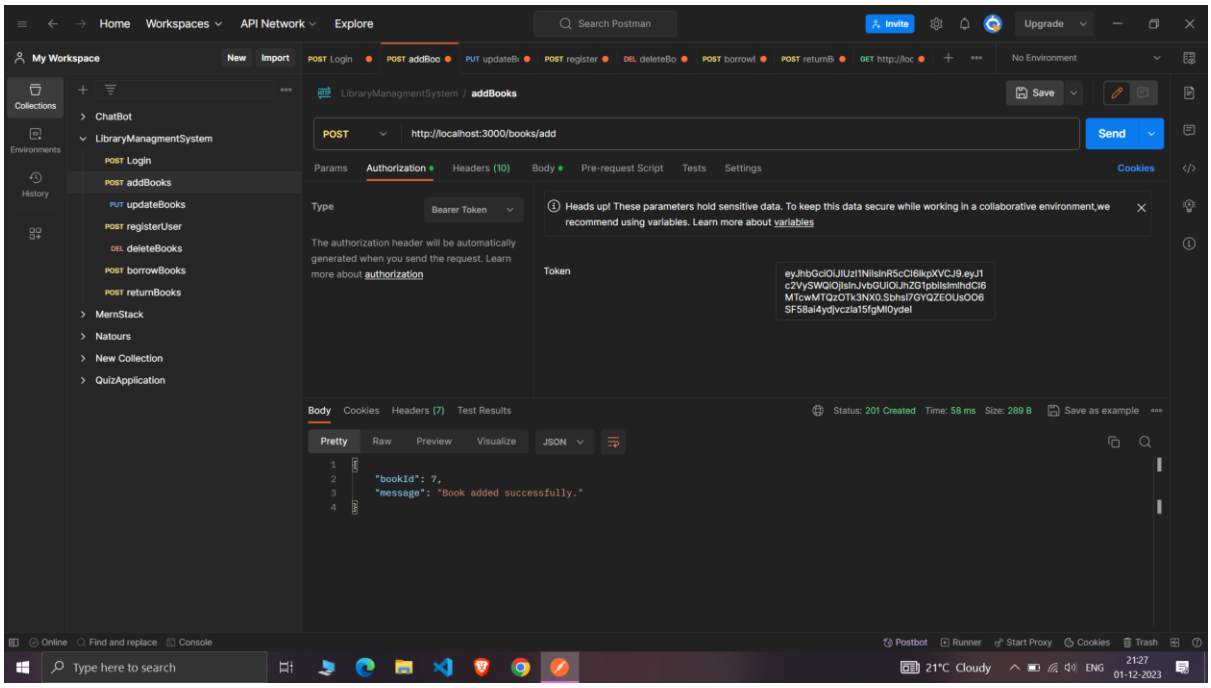
Next, I did Book Management:-

Adding Book:-

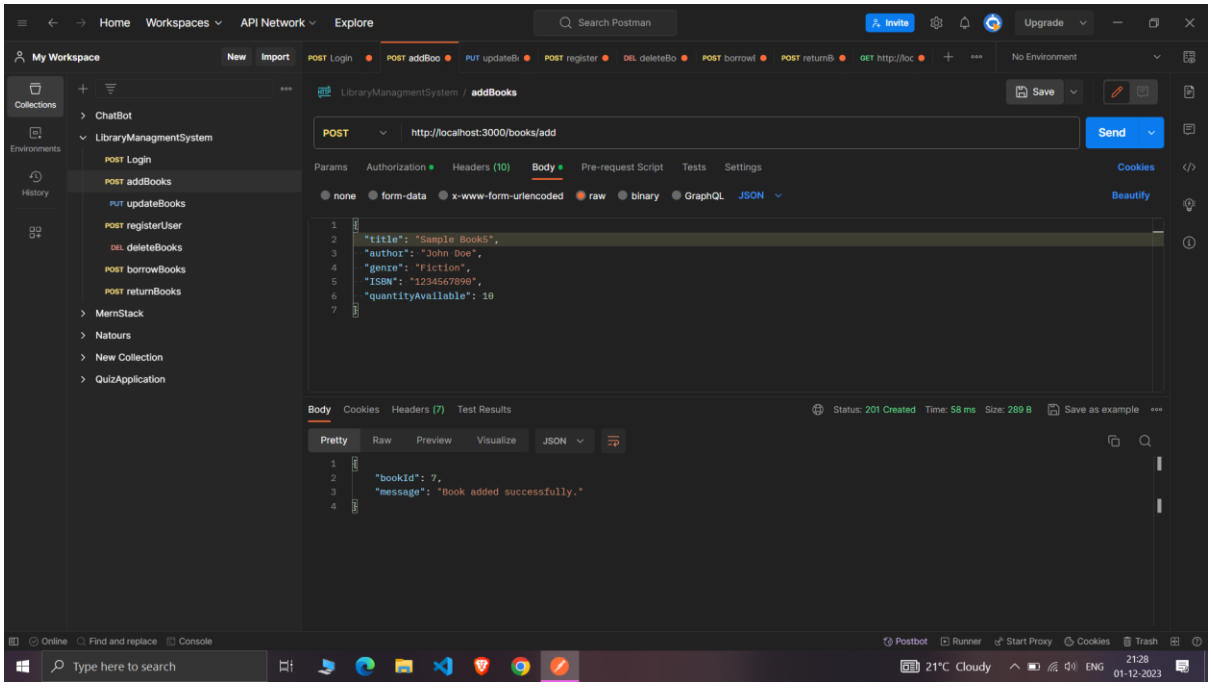
He should be authorized so we keep the below and include token there

And 2 roles are there as Admin And User ,Admin only can add books, update books, delete books.

Authorized User:-



Adding Book:-

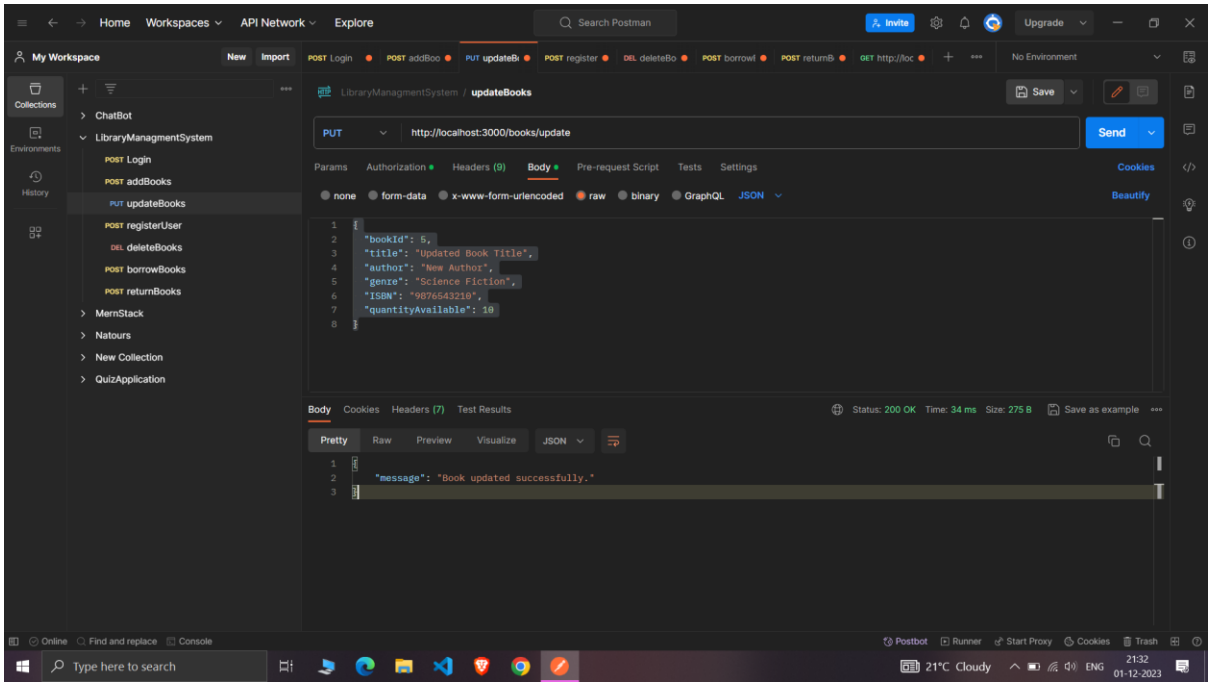


Update Book (Admin only):

Endpoint: PUT books/update

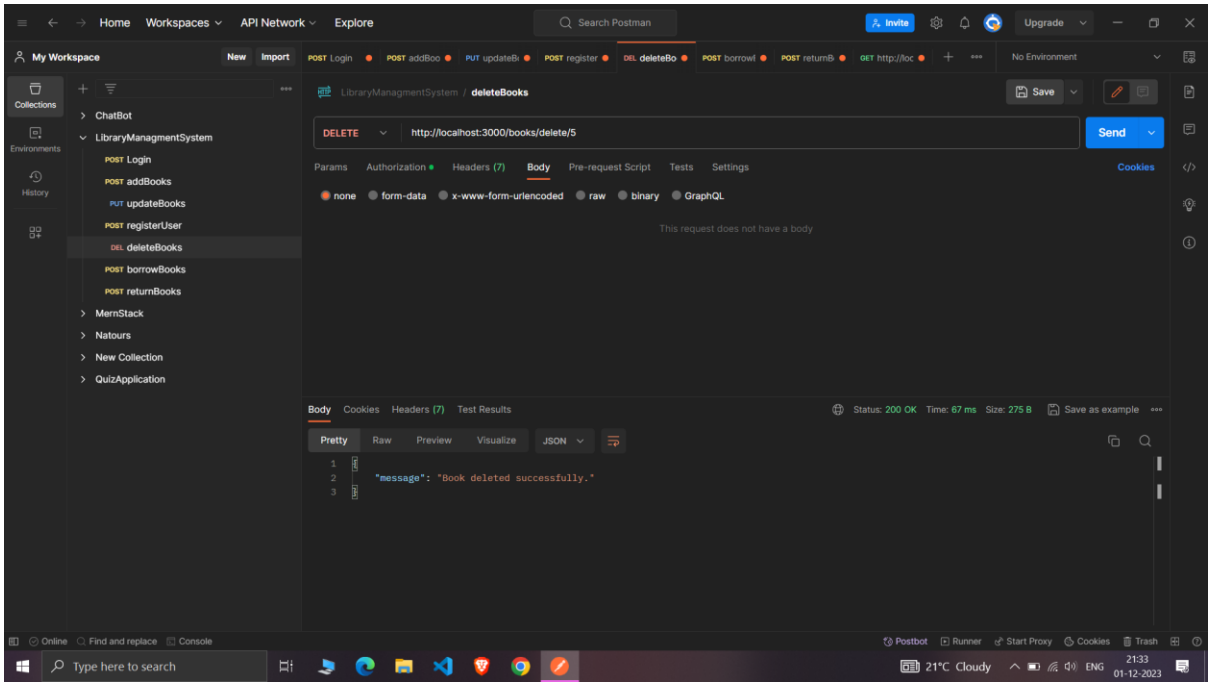
Body:-

```
{
 "bookId": 5,
"title": "Updated Book Title",
 "author": "New Author",
 "genre": "Science Fiction",
 "ISBN": "9876543210",
"quantityAvailable": 10
}
```



Delete Book (Admin only):

Endpoint: DELETE books/delete/bookId



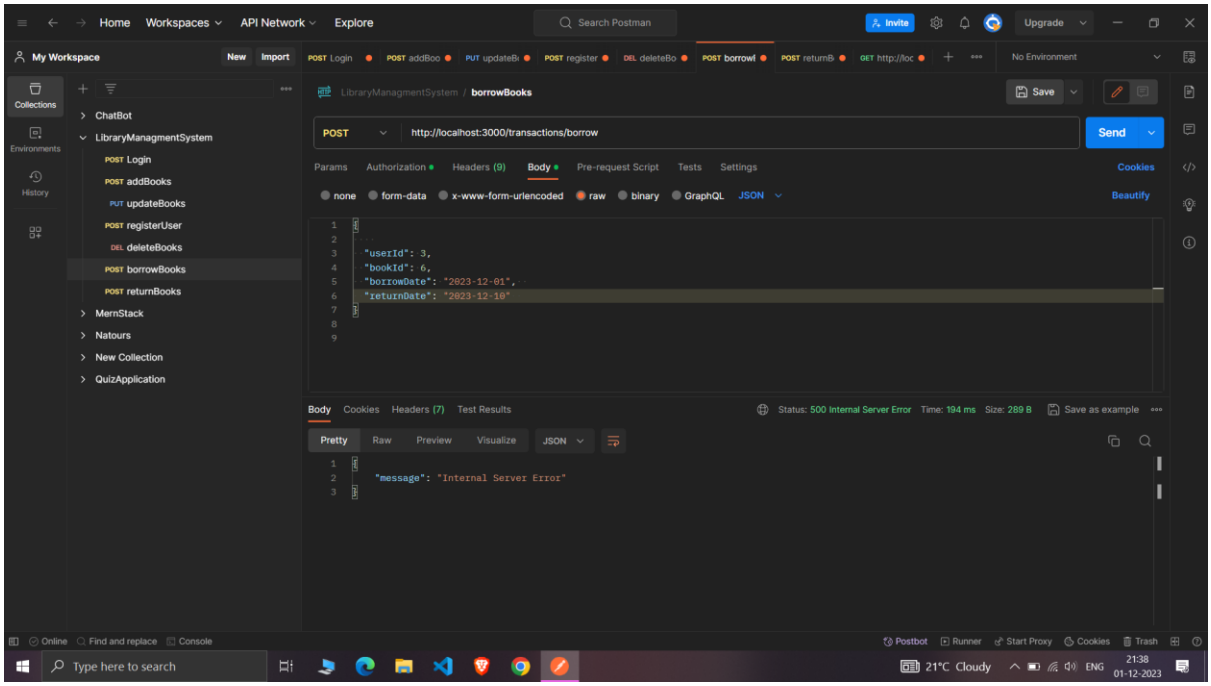
Transactions Management:-

Borrowing Books:-

EndPoint: POST transactions/borrow

Body:-

```
{
 "userId": 3,
 "bookId": 6,
"borrowDate": "2023-12-01
"returnDate": "2023-12-10"
}
```

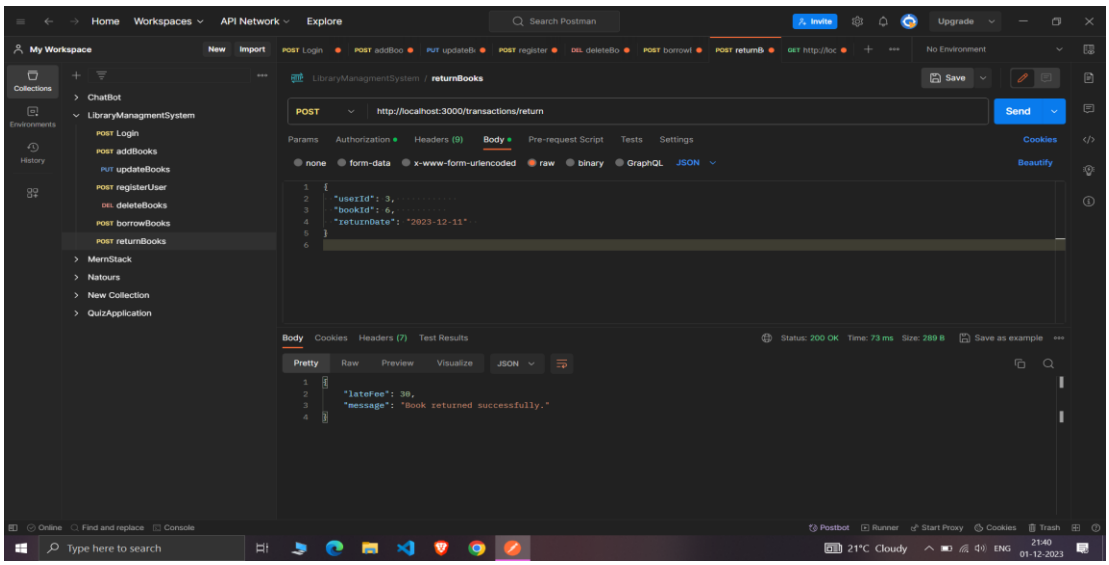


Return Book:

Endpoint: POST transactions/return

Body:-

```
{
"userId": 3,
 "bookId": 6,
 "returnDate": "2023-12-11"
}
```

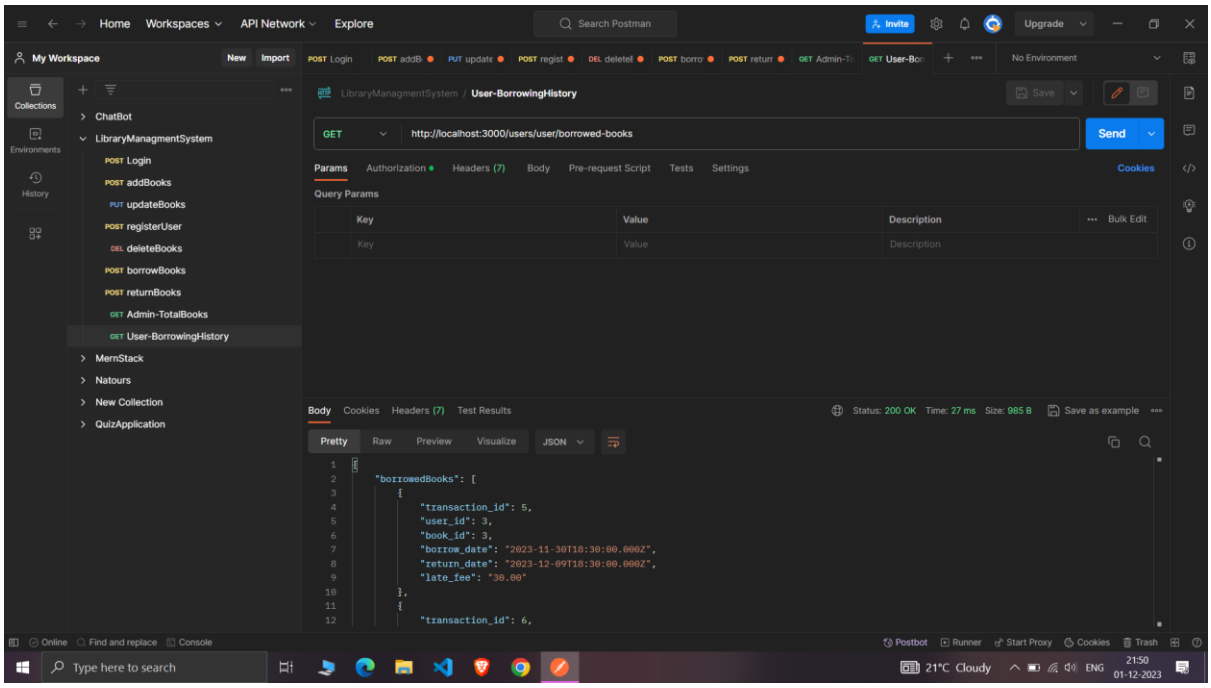


Note:-For the above both In Transactions we should have authorization headers i.e..,Tokens in the post man.

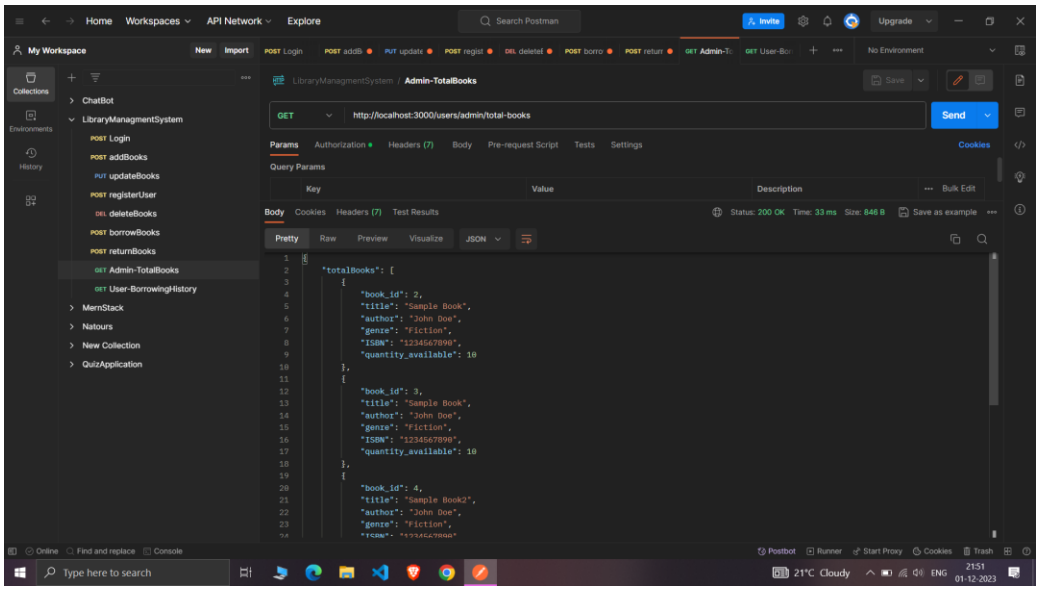
Finally, we can show the User borrowing history and Admin can see the total number of books present now in Library by following:-

User Borrowing History:-

Endpoint:-GET users/user/borrowed-books



Admin Endpoint:-GET users/admin/total-books



This is my sql workbench code for database creation:-

```
CREATE DATABASE library_management_system;
USE library_management_system;
CREATE TABLE Users (
  user_id INT PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(255) NOT NULL,
  contact_details TEXT, -- Using TEXT for flexibility
  role ENUM('user', 'admin') DEFAULT 'user' NOT NULL
);
CREATE TABLE Books (
  book_id INT PRIMARY KEY AUTO_INCREMENT,
  title VARCHAR(255) NOT NULL,
  author VARCHAR(255) NOT NULL,
  genre VARCHAR(255),
  ISBN VARCHAR(20) NOT NULL,
  quantity_available INT NOT NULL DEFAULT 0
);
CREATE TABLE Transactions (
  transaction_id INT PRIMARY KEY AUTO_INCREMENT,
  user_id INT,
  book_id INT,
  borrow_date DATE,
  return_date DATE,
  late_fee DECIMAL(10, 2) ,
  FOREIGN KEY (user_id) REFERENCES Users(user_id),
  FOREIGN KEY (book_id) REFERENCES Books(book_id)
);
ALTER TABLE Users
ADD COLUMN password_hash VARCHAR(255);
```