Fiction Passion Board

MID TERM REPORT

OF MINI PROJECT

BACHELOR OF TECHNOLOGY

Computer Science and Engineering Branch

SUBMITTED BY :

AYUSHI BATHAM (191500204)

VANI JAIN (191500893)

VASUNDHARA TIWARI (191500902)

SUPERVISED BY :

MR. AMIR KHAN

(TECHNICAL TRAINER)

GLA University, Mathura
(2021-2022)

Content

Abstract

1. Introduction
    a. General Introduction to the Project
    b. Area of computer science
2. Problem Definition
3. Objective
4. Technology being used
5. Methodology
6. Implementation
7. Progress
8. Screenshots

Abstract

We all cannot disagree with the fact that there are a lot of people in our society who are interested in reading new books and in our busy schedule it becomes very difficult to search for new books offline as it takes more time. Sometimes it creates a lot of haphazard to collect required reviews of a book to prefer for reading.

Introduction

1.1 General Introduction to the Project

Currently in our society, the people used to search for the books in the library as per their interest which becomes very frustrating to find. There is no central system within our reach which allow user to search for the books which they will find interested in reading. We are totally dependent on the views of our friends, colleagues, family, etc. to know some good books to read which results in confusing reviews of some books.

We are going to design a system for a library which will keep track of the user rating's and their reviews for the books they are reading. This will help user to know more about the content of the book before reading. As well as we will provide a system where user's can rate the books after reading which will be a source of information for other users. We will be designing a website which will facilitate the readers of the books as well as the library administrator.

1.2 Area of Computer Science

The computer has brought revolution in every sphere of human life, whether it is business, education field, governance, medical science etc. The computer has reduced the human work load, businesses are going global and everything is available at the click of mouse. Most of the users have to look for the reviews of others to find some good books, which is very time-consuming task. By this application they can access reviews easily.

Problem Definition

Most of the people nowadays prefer online shopping, this become more convenient for them as most of the things they want are available online and can be purchased with a single click. There are some areas of online shopping which needs to be covered like most of the time one has to visit shop to buy goods related to construction. So, we are trying to make an online platform which provide these types of construction material goods.

Objectives of the project

Fiction Passion Board is a platform where readers can provide and access the reviews and ratings of the books. The objective of this project is to collect reviews and ratings of different readers on a book as well as giving them a facility to provide ratings and reviews for the books and its content they have read.

Technology

Hardware Requirements :-

- Computer System with minimum 8GB of RAM

Software Requirements :-

- Windows/Linux OS
- Visual Code Studio
- Postman
- Robo3T

Programming Languages :-

- JavaScript
- MERN Stack

Methodology

We are using React based Webpage Portal as the frontend with the backend made using NodeJs and MongoDB database. To see what are the ratings and reviews of the books given by different readers, user have to scroll through the landing page of the books, where the different books are listed.

The modules used in the website are listed as follows :-

1. Home Page
2. Book Catalogue
3. Book Info display Page
4. Reviews and Rating Page

Implementation Details

Webpage based Portal :

We are working on this in two parts viz frontend and backend. Firstly, we are working on backend after that will move towards frontend. There might be some changes in the backend while working with frontend.

1. **Backend development :** The project commences with designing of fundamental book schema and rating schema. We will use APIs for creating different routes. Fiction Passion Board will display the books of a library. The data will be sent to backend using APIs.

   An application program interface (API) is a set of routines, protocols and tools for building software applications. Basically, an API specifies how software components should interact.

   The data will be sent in JSON format.

   JSON is a lightweight data-interchange format it is easy for humans to read and write. It is easy for machines to parse and generate.

   We are using MongoDB Atlas for cloud storage during development phase because of its flexibility and scalability of document database, available as a fully managed service.

   MongoDB Atlas is the global cloud database service for modern applications. Deploy fully managed MongoDB across AWS, Azure, or GCP. Best-in-class automation and proven practices guarantee availability, scalability, and compliance with the most demanding data security and privacy standards

2. **Frontend development :** We will be using ReactJs, HTML, Bootstrap for frontend development part. On the home page, there will be basic functionalities like various books with their basic details like name, author name, publication, etc. But for rating the contents of the book user have to navigate to the rating page by using the button provided.

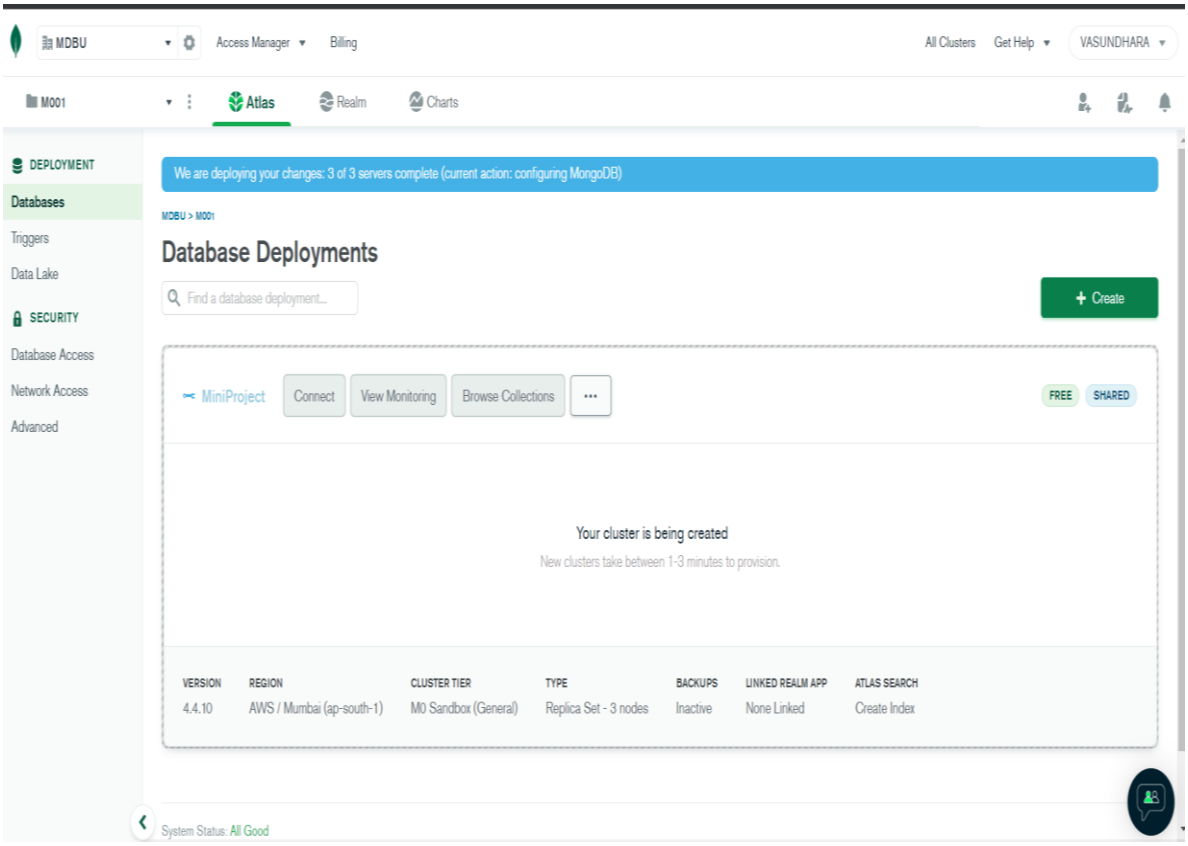
PROGRESS

Web page based Portal :

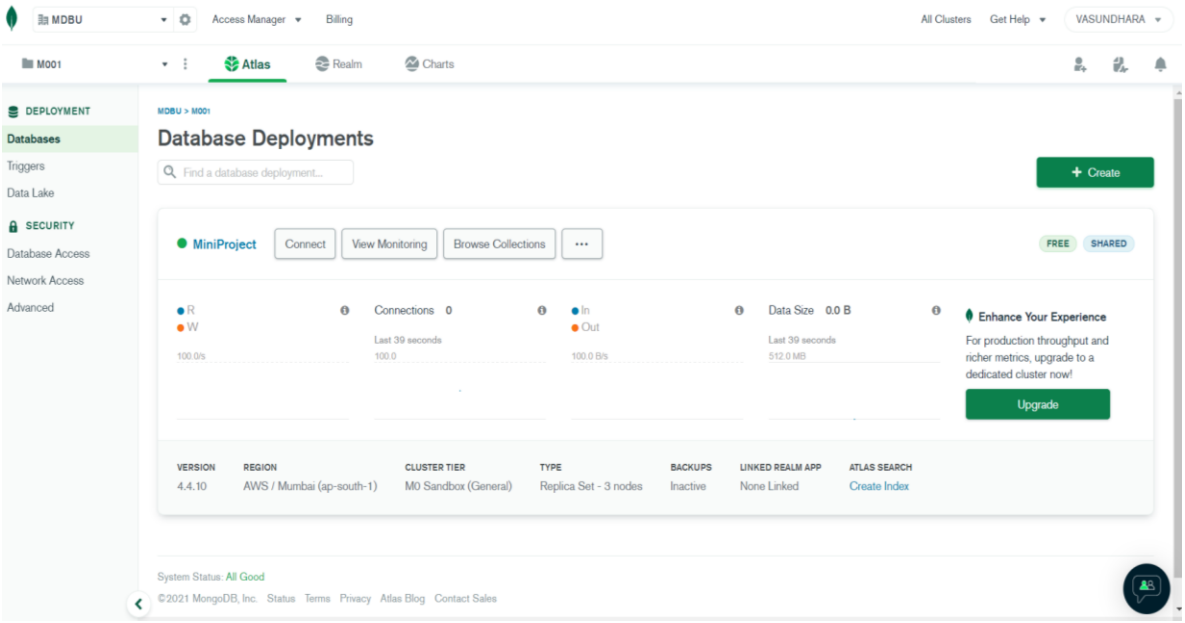
- Created online database on MongoDB Atlas, so that it can be used for Web Page based Portal.
- Successfully connected Backend of the Web Page based Portal with database.
- Created a basic schema for the Book.
- Created all the controllers required for adding a new book to the database.
- Created all the controllers required for retrieving a book from the database.
- Created all the required controllers for updating the details of the books in the database.
- Created all the required controllers for updating the ratings of the books in the database.
- Created the route for adding a new book to the database as http://localhost:XXXXX /api/addBook
- Created the route for retrieving book from the database as http://localhost:XXXXX /api/book/:id
- Created the route for retrieving book rating as http://localhost:XXXXX /api/bookRating/:id
- Created the dummy homepage for the frontend.

Screenshots

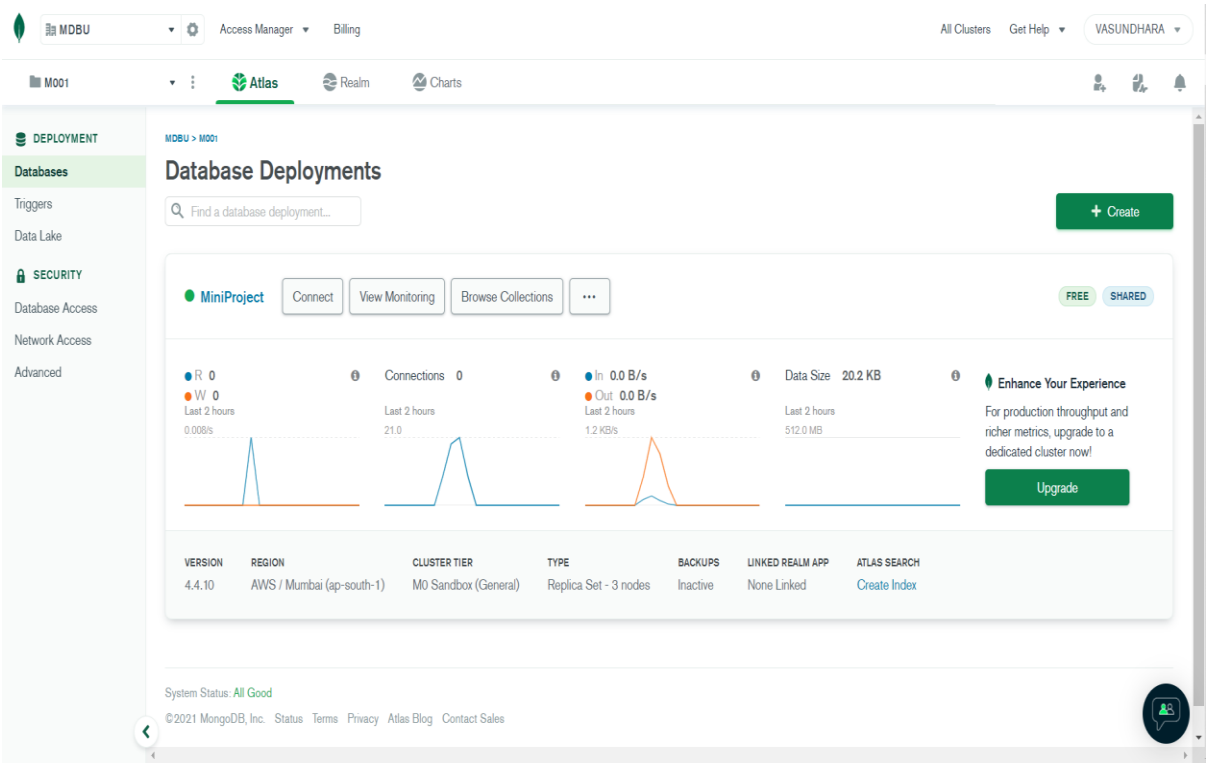
Creating MongoDB cluster

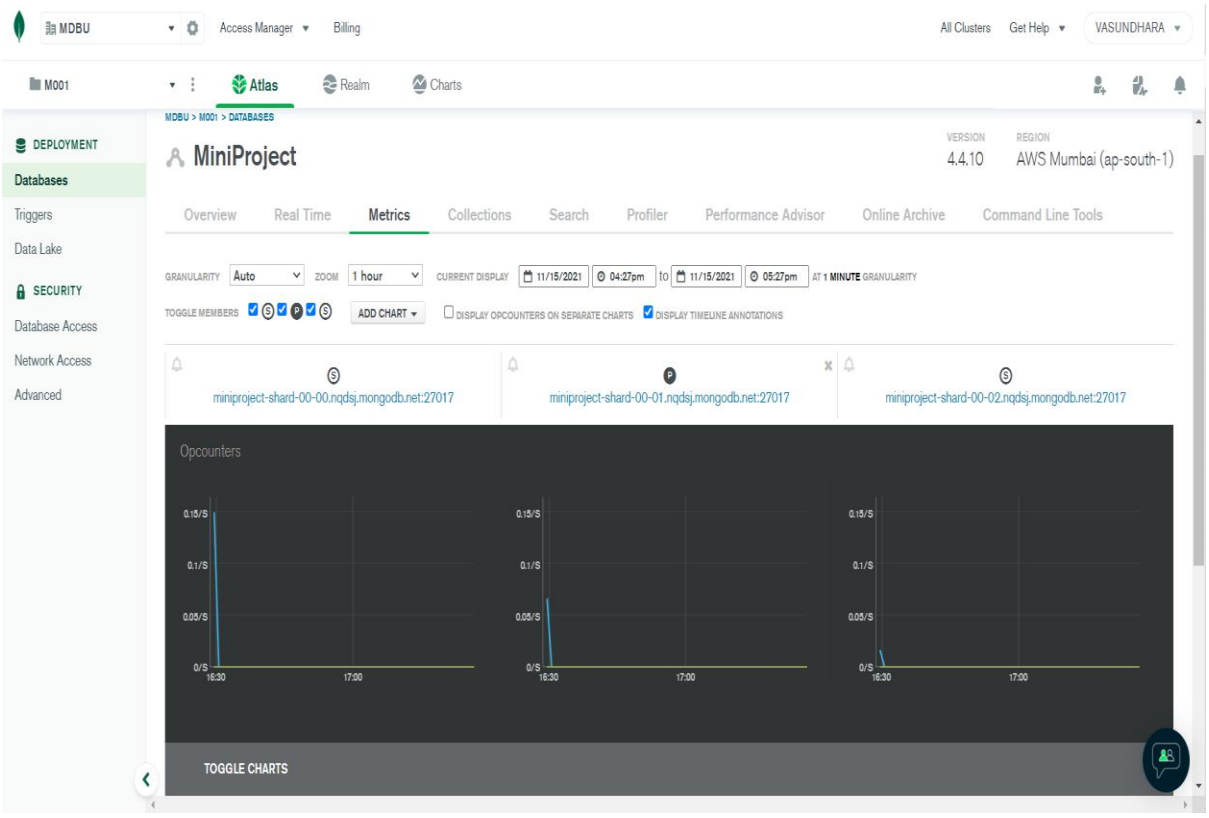


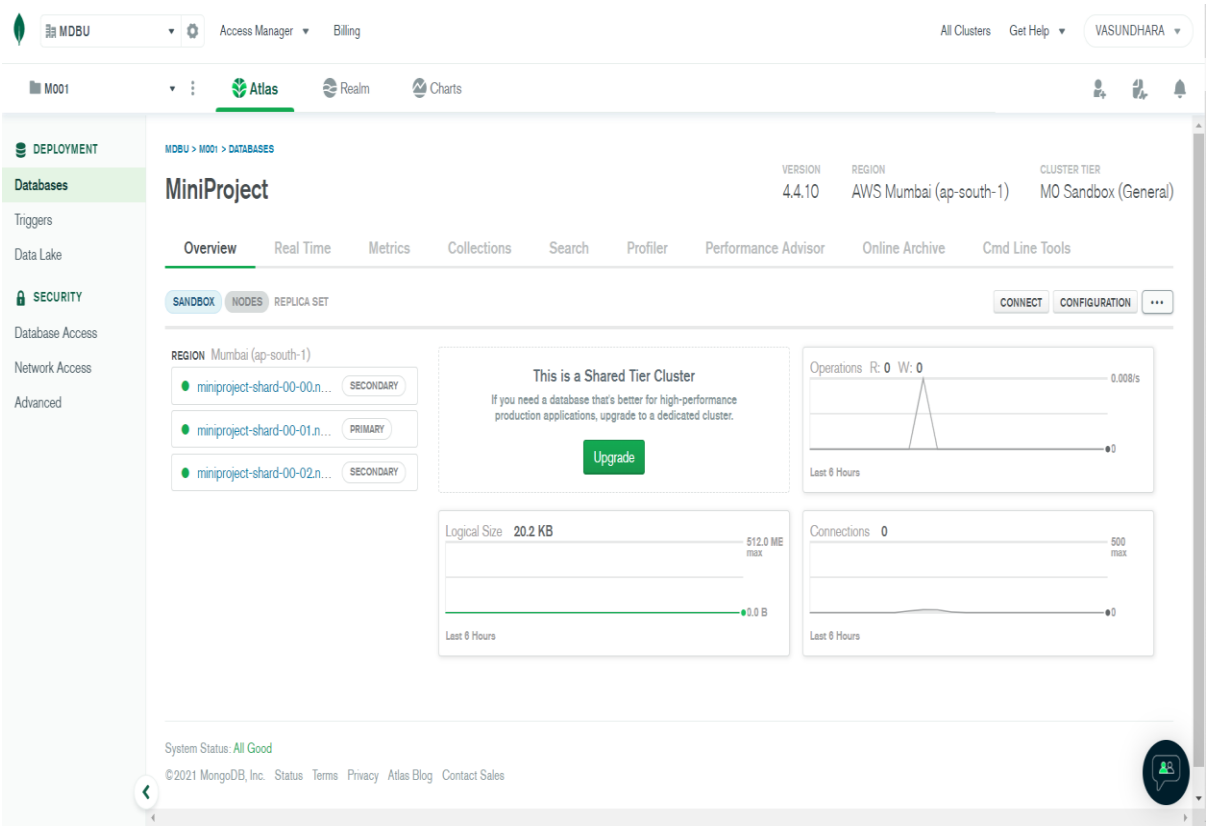
Successfully created the cluster



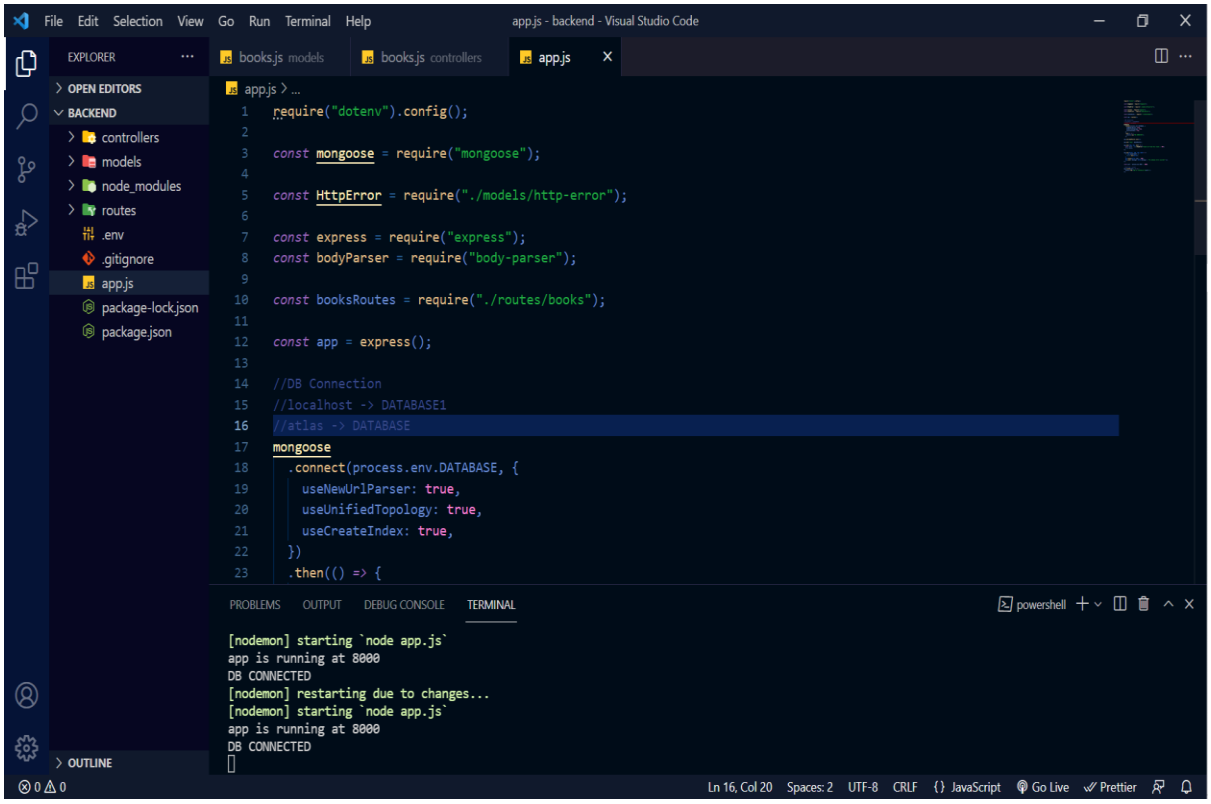
Displaying database metrices after creating the cluster

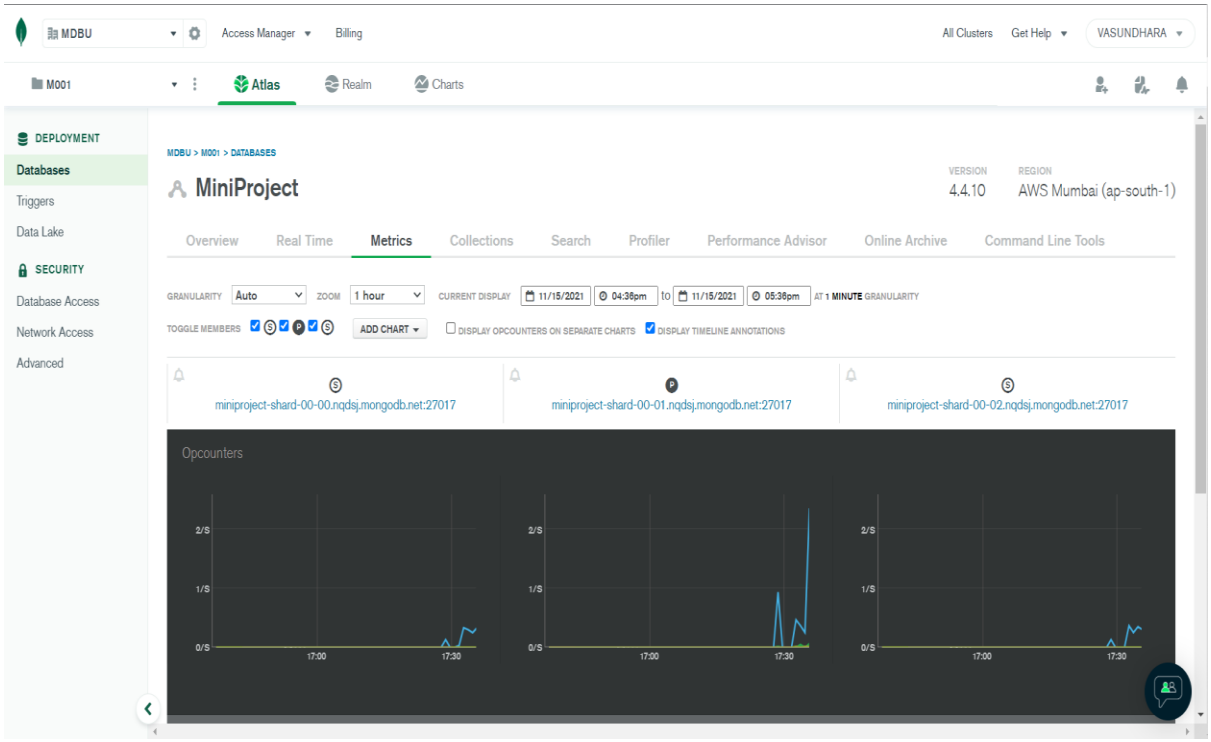






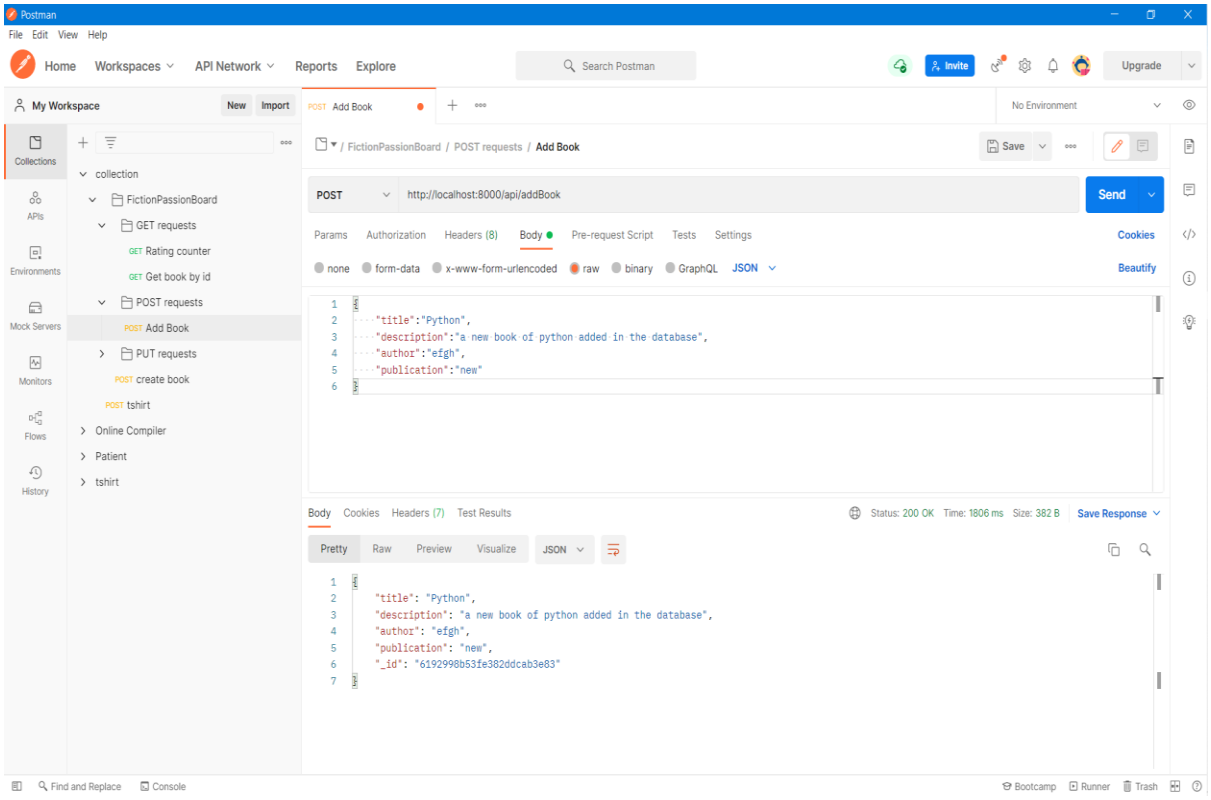
Successfully linked Project with MongoDB Cloud Storage (MongoDB Atlas): -



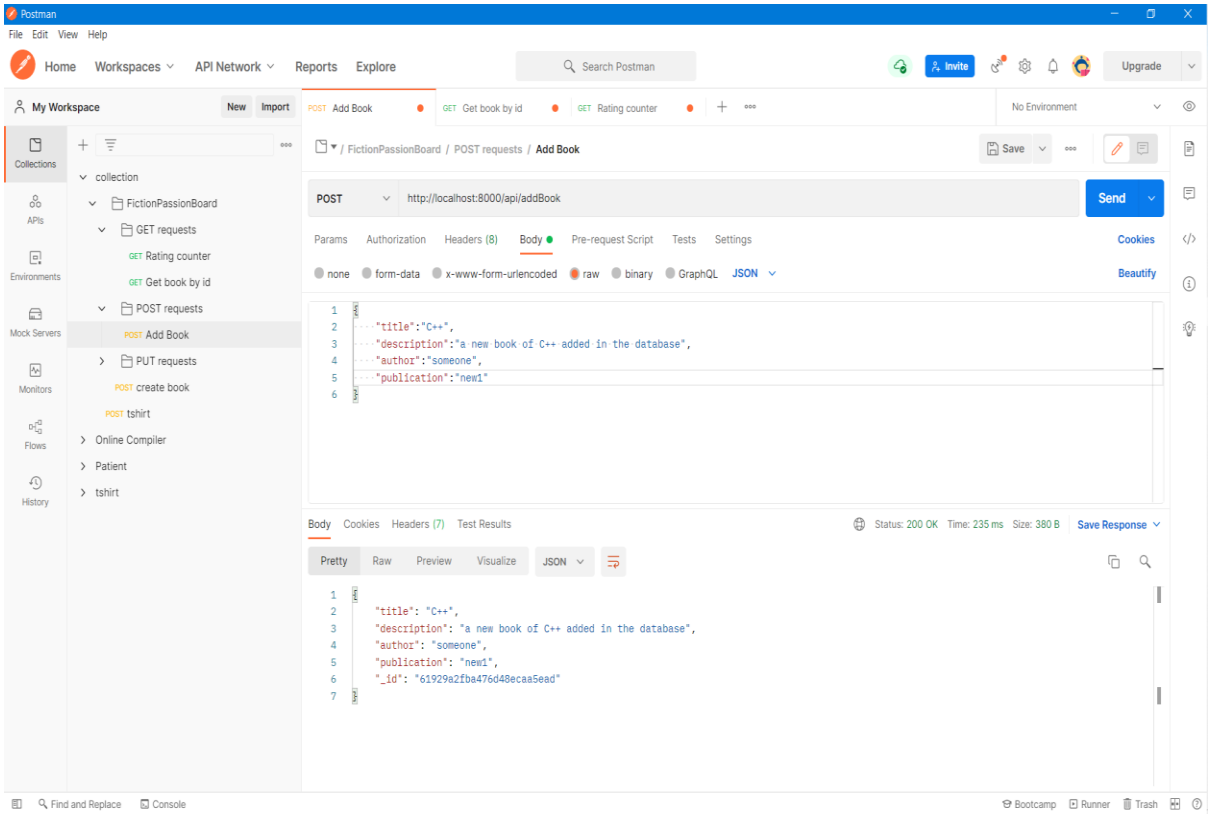


Adding books to the database

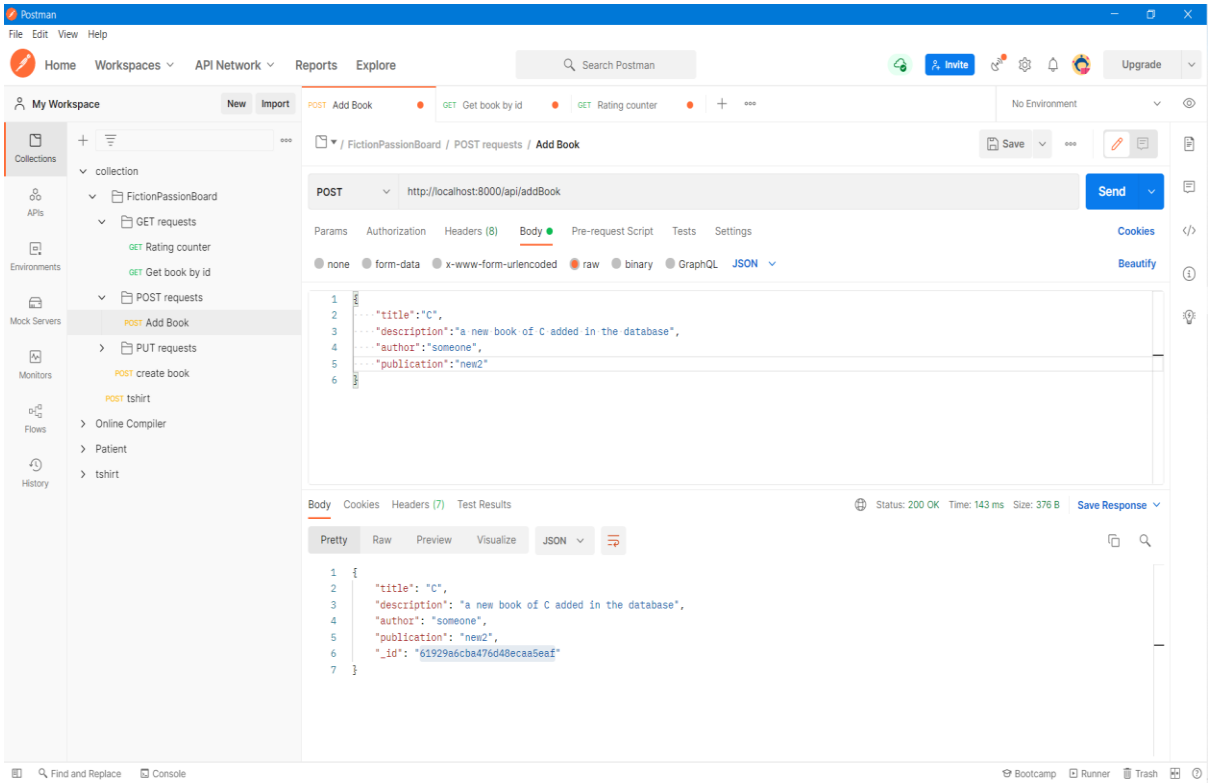
Book 1



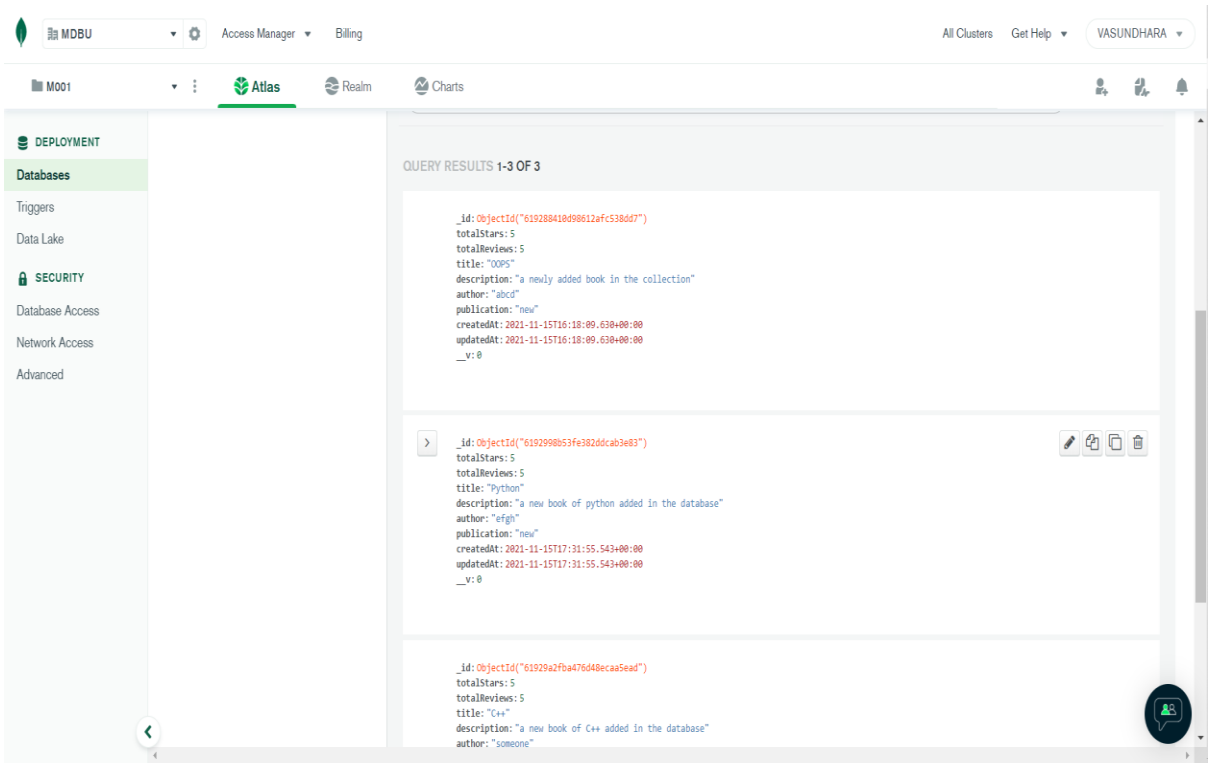
Book 2

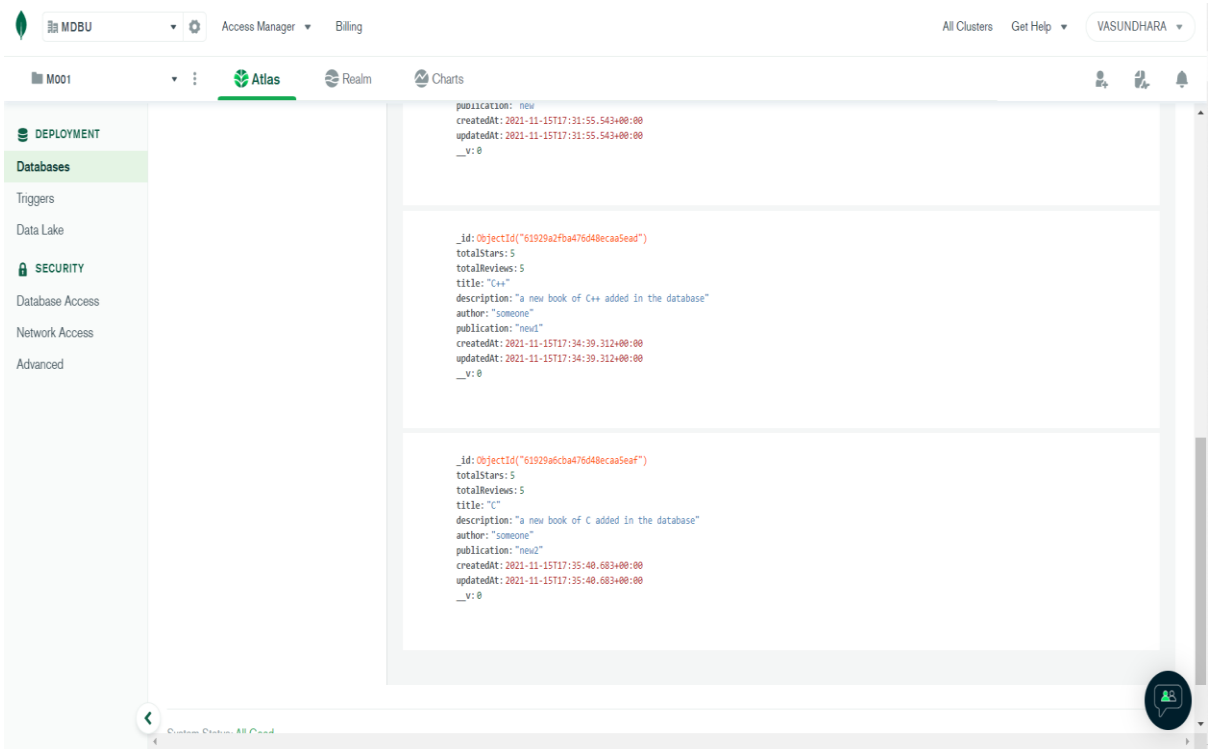

Postman
File Edit View Help
Home Workspaces API Network Reports Explore
Search Postman
Invite
Upgrade
My Workspace
New Import
POST Add Book
GET Get book by id
GET Rating counter
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
collection
FictionPassionBoard
GET requests
Rating counter
Get book by id
POST requests
Add Book
PUT requests
create book
tshirt
Online Compiler
Patient
tshirt
/ FictionPassionBoard / POST requests / Add Book
Save
POST
http://localhost:8000/api/addBook
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Beautify
{
"title":"C++",
"description":"a new book of C++ added in the database",
"author":"someone",
"publication":"new1"
}
Body Cookies Headers (7) Test Results
Status: 200 OK Time: 235 ms Size: 380 B
Save Response
Pretty Raw Preview Visualize JSON
{
"title": "C++",
"description": "a new book of C++ added in the database",
"author": "someone",
"publication": "new1",
"_id": "61929a2fba476d48ecaa5ead"
}
Find and Replace
Console
Bootcamp
Runner
Trash

Book 3

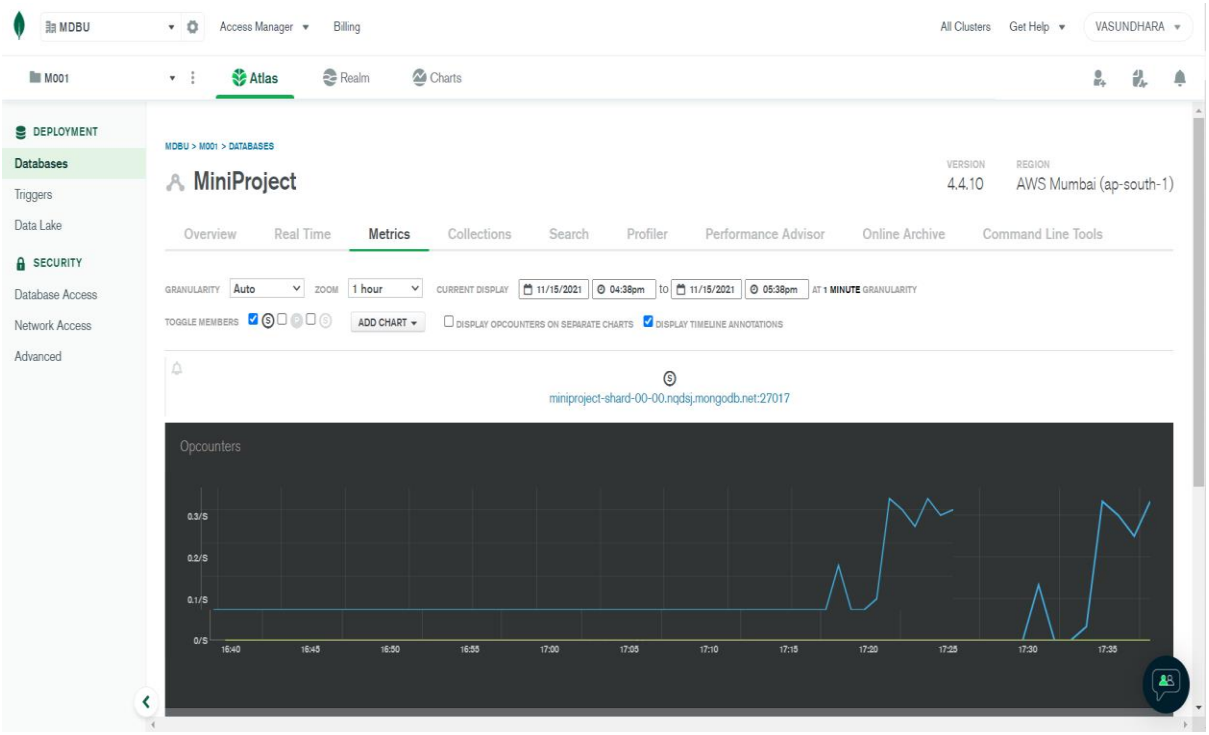

Postman
File Edit View Help
Home Workspaces API Network Reports Explore
Search Postman
Invite
Upgrade
My Workspace
New Import
POST Add Book
GET Get book by id
GET Rating counter
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
collection
FictionPassionBoard
GET requests
Rating counter
Get book by id
POST requests
Add Book
PUT requests
create book
tshirt
Online Compiler
Patient
tshirt
/ FictionPassionBoard / POST requests / Add Book
Save
POST
http://localhost:8000/api/addBook
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Beautify
{
"title":"C",
"description":"a new book of C added in the database",
"author":"someone",
"publication":"new2"
}
Body Cookies Headers (7) Test Results
Status: 200 OK Time: 143 ms Size: 376 B
Save Response
Pretty Raw Preview Visualize JSON
{
"title": "C",
"description": "a new book of C added in the database",
"author": "someone",
"publication": "new2",
"_id": "61929a6cba476d48ecaa5eaf"
}
Find and Replace
Console
Bootcamp
Runner
Trash

Collection after adding the books



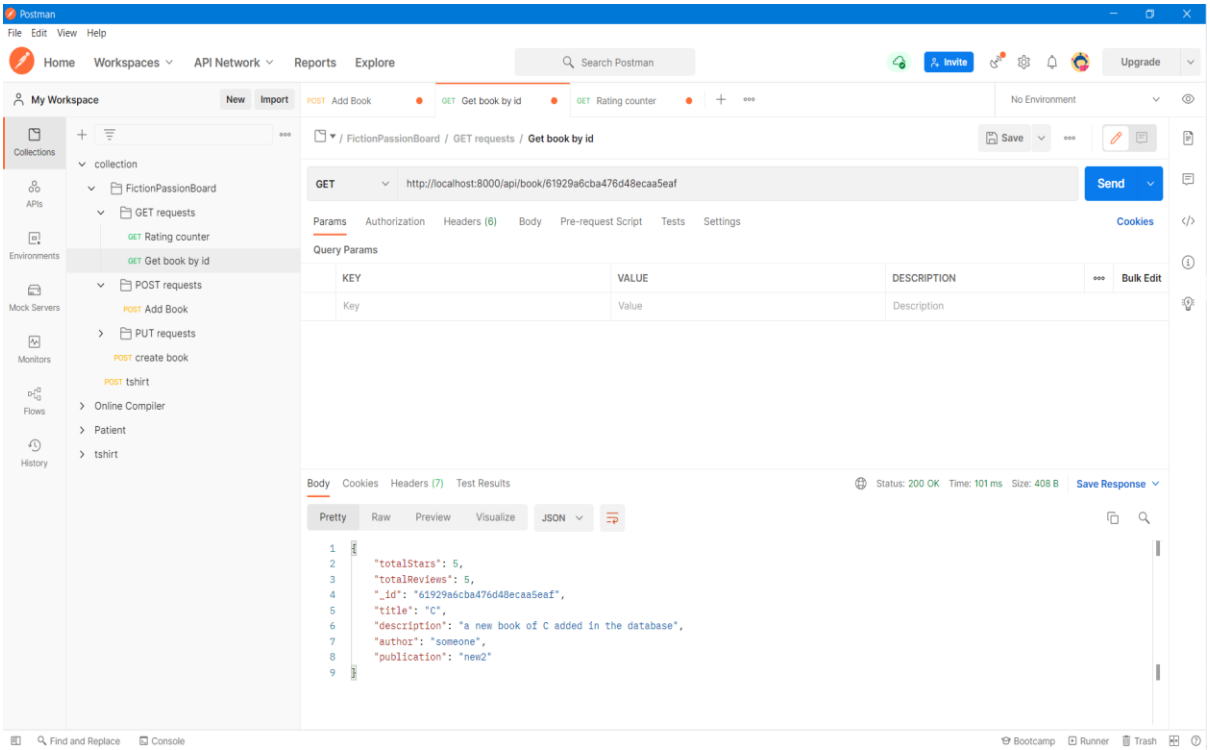


Database status after adding the books

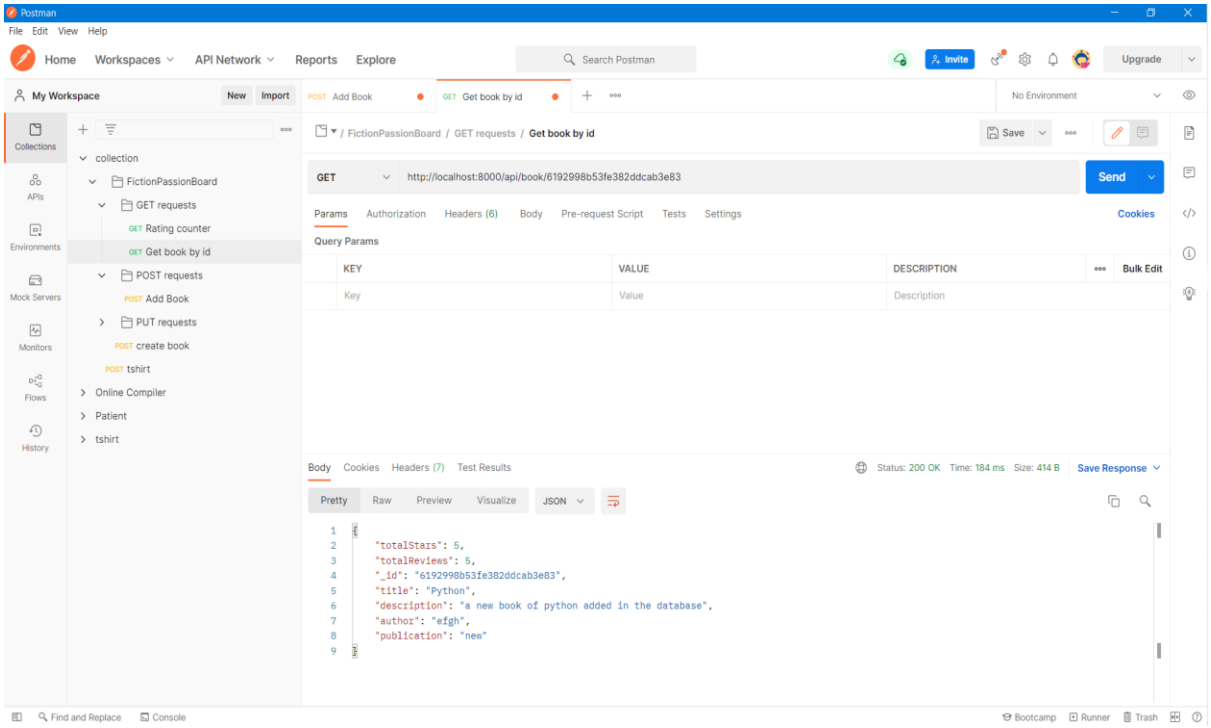


Fetching a book from the database

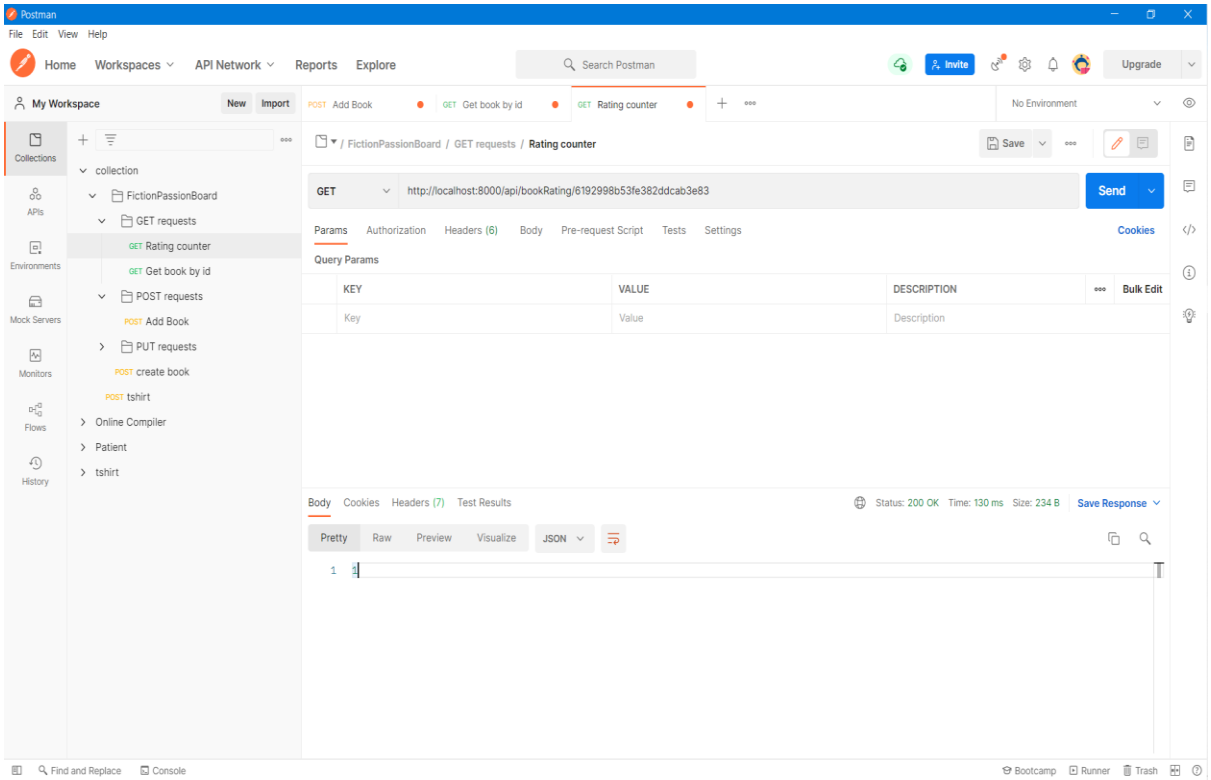
Book 1



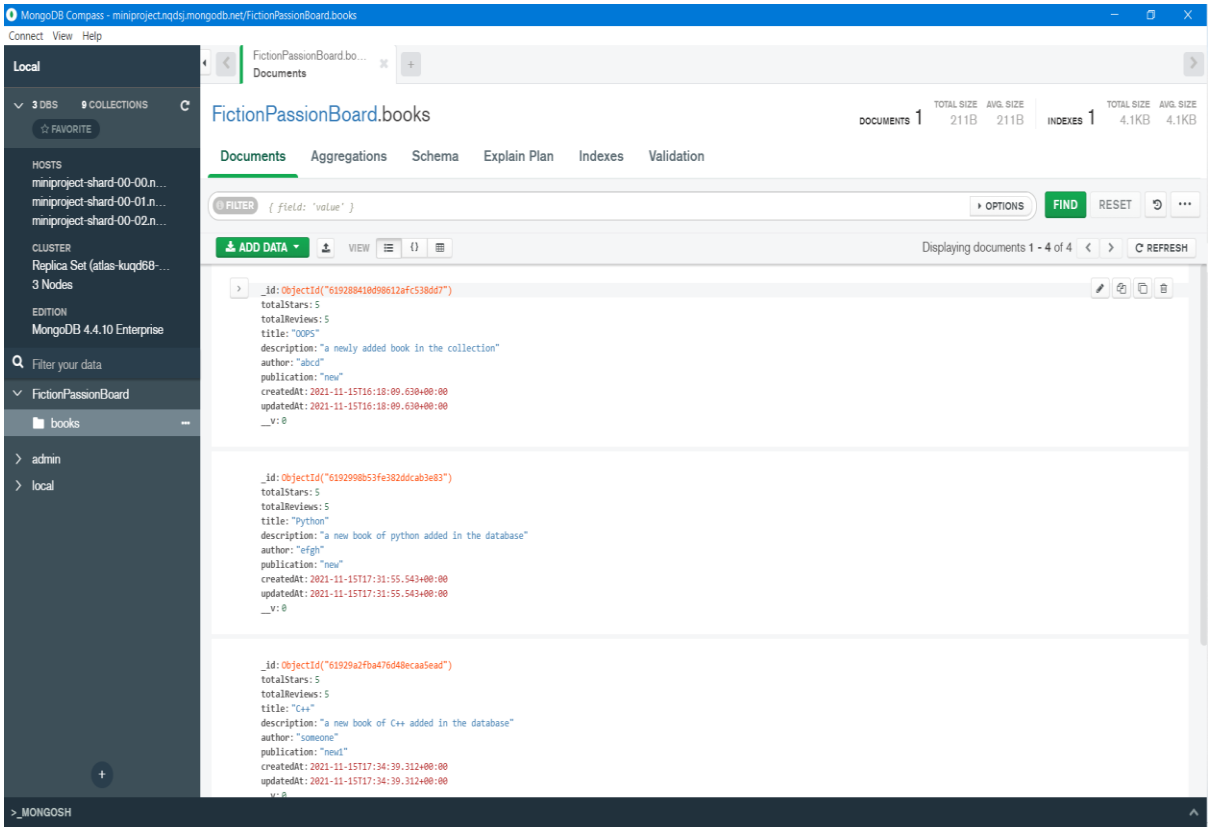
Book 2



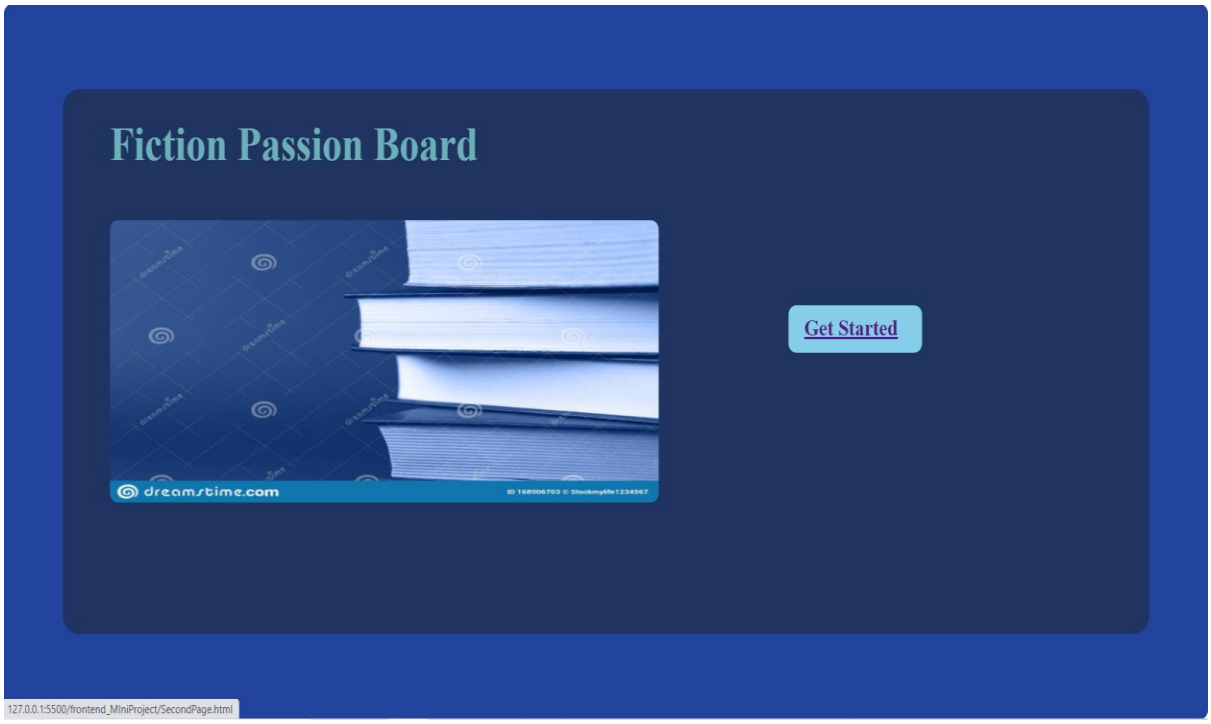
Fetching rating of a book from the database



The database after performing the above operations



Dummy Landing Page of the Website



Mobile View of the website