2. Write a spring boot application use Mongodb database implement BookStore database include the members I'd,bookname,book athorname,And write the following Operations :

-Post

-Get

-Delete

Step 1:   Open Spring Boot Suite app,Click on create new spring starter project, add project name.

Step 2: Add the following dependency

- Spring Web

-MongoDB

-Lombok

-DevTools

Step 3: Create 3 packages and create some classes and interfaces inside the Packages.

1. model

2. repository

3. controller

Step 4: Inside the model package Creating a simple class name as Book inside the Book.java file.

```
package com.ii; import org.springframework.data.annotation.Id;

import org.springframework.data.mongodb.core.mapping.Document;

import lombok.AllArgsConstructor; import lombok.Data; import

lombok.NoArgsConstructor;


//Annotations

@Data
```

```
}


@AllArgsConstructor

@NoArgsConstructor

@Document (collection="Book")


//Class public

class Book { //

Attributes

        @Id

        private Integer id;

private String bookname;

private String authorname;


        public Integer getId() {

                return id;

        }

        public void setId(Integer id) {

                this.id = id;

        }

        public String getBookname() {

                return bookname;

        }

        public void setBookame(String bookname) {
                this.bookname = bookname;

        }

        public double getAuthorname() {

                return authorname;

        }

        public void setAuthorname(String authorname) {
this.authorname = authorname;

        }
```

```
}
```

Step 5: Inside the repository package Create a simple interface and name the interface as BookRepo. This interface is going to extend the MongoRepository .

```
// Java Program to Illustrate BookRepo File


import com.globallogic.spring.mongodb.model.Book; import
org.springframework.data.mongodb.repository.MongoRepository; public
interface BookRepo extends MongoRepository <Book, Integer>
 {
}
```

Step 6: Inside the controller package Inside the package create one class named as BookController and perform the operations.

```
import com.globallogic.spring.mongodb.model.Book; import
com.globallogic.spring.mongodb.repository.BookRepo; import
org.springframework.beans.factory.annotation.Autowired; import
org.springframework.web.bind.annotation.*;
import java.util.List; //
Annotation
@RestController
// Class public class
BookController {
 @Autowired  private
BookRepo repo; //Post
Operation

 @PostMapping("/addBook")  public String
saveBook (@RequestBody Book book){
repo.save(book);  return "Added Successfully";
 }
//Get Operation
```

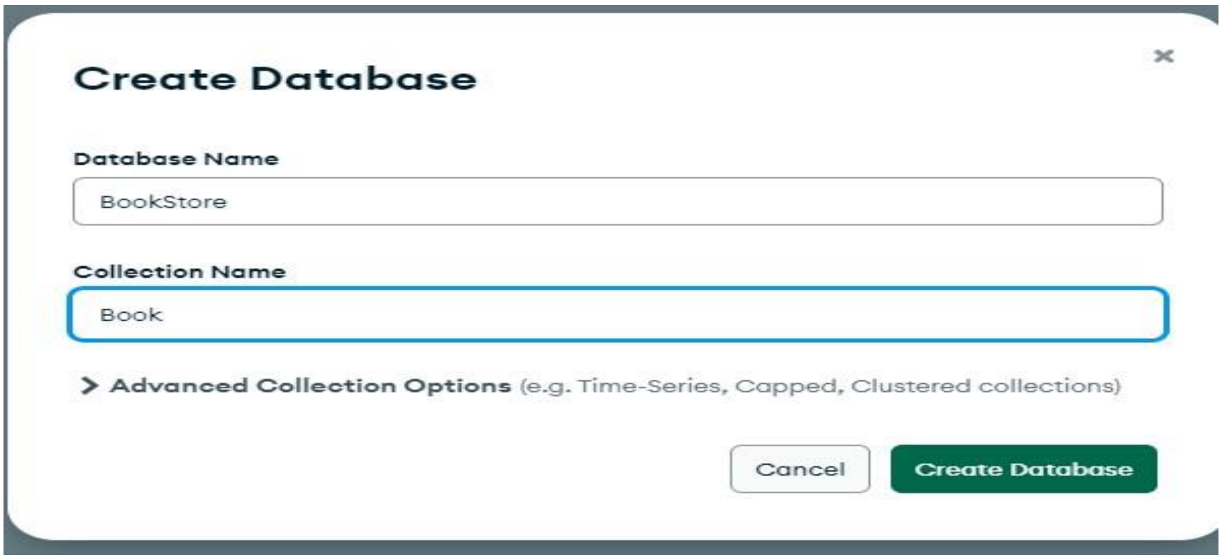
```
 @GetMapping("/findAllBooks")
public List<Book> getBooks() {  return
repo.findAll();
 }
//Delete Operation


 @DeleteMapping("/delete/{id}")  public String
deleteBook(@PathVariable int id){
repo.deleteById(id);  return "Deleted
Successfully";
 } }
```

Step 7: Below is the code to connect the Mongodb Server Write in the application.properties file .

```
server.port = 8989
# MongoDB Configuration server.port:8989
spring.data.mongodb.host=localhost
spring.data.mongodb.port=27017
spring.data.mongodb.database=BookStore
```

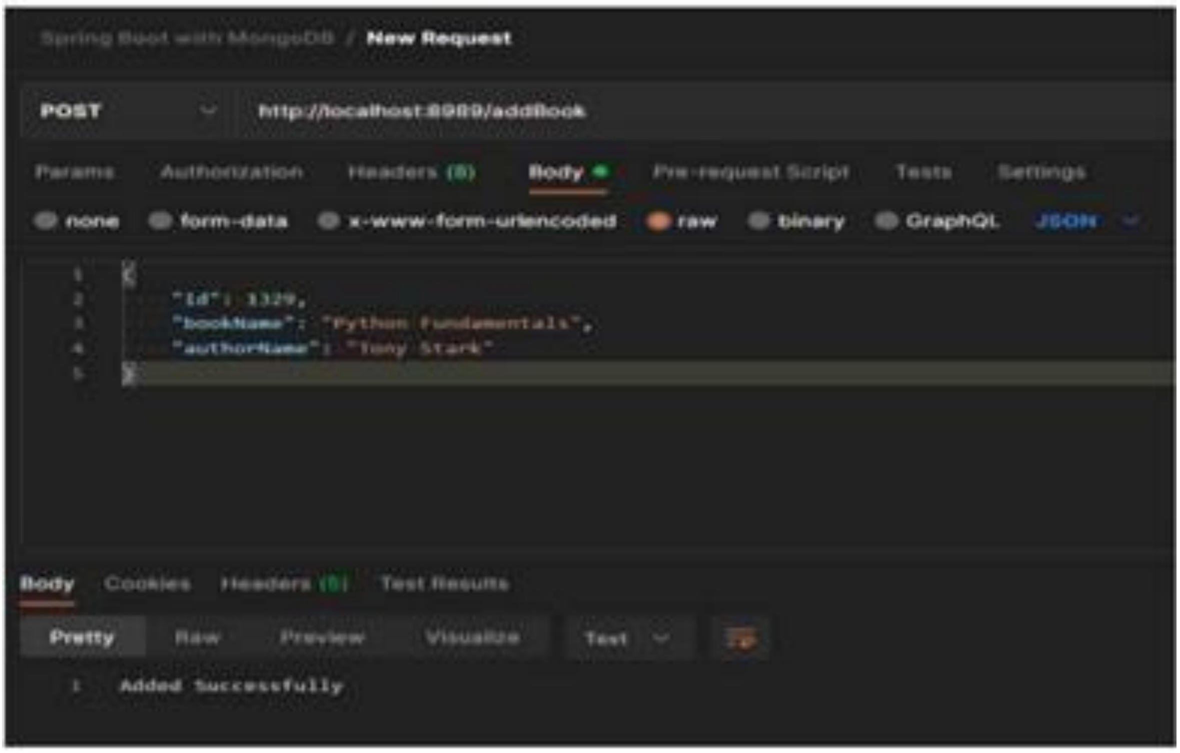
Step 8: Inside the MongoDB Compass Go to your MongoDB Compass and create a Database named BookStore and inside the database create a collection named Book as seen in the below image.


Create Database
Database Name
BookStore
Collection Name
Book
Advanced Collection Options (e.g. Time-Series, Capped, Clustered collections)
Cancel
Create Database

Now run your application and let's test the endpoints in Postman and also refer to our MongoDB Compass.

Testing the Endpoint in Postman

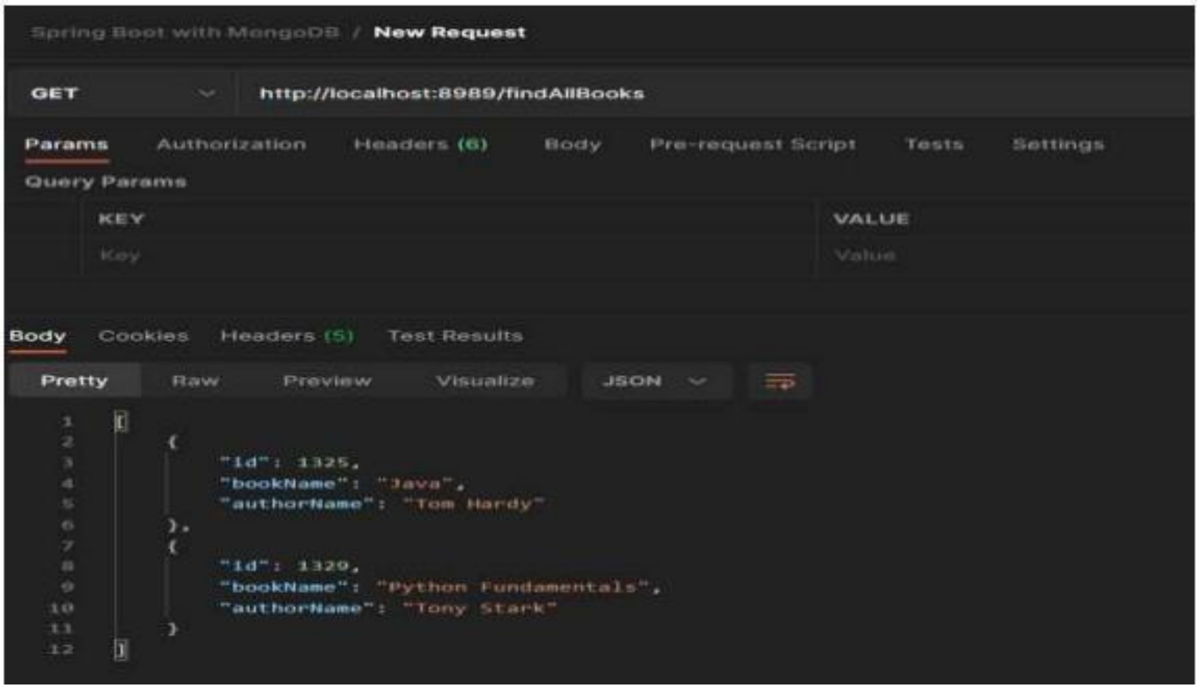
Endpoint 1: POST – http://localhost:8989/addBook



```
{
    "id": 1329,
    "bookName": "Python Fundamentals",
    "authorName": "Tony Stark"
}
```

```
Added Successfully
```

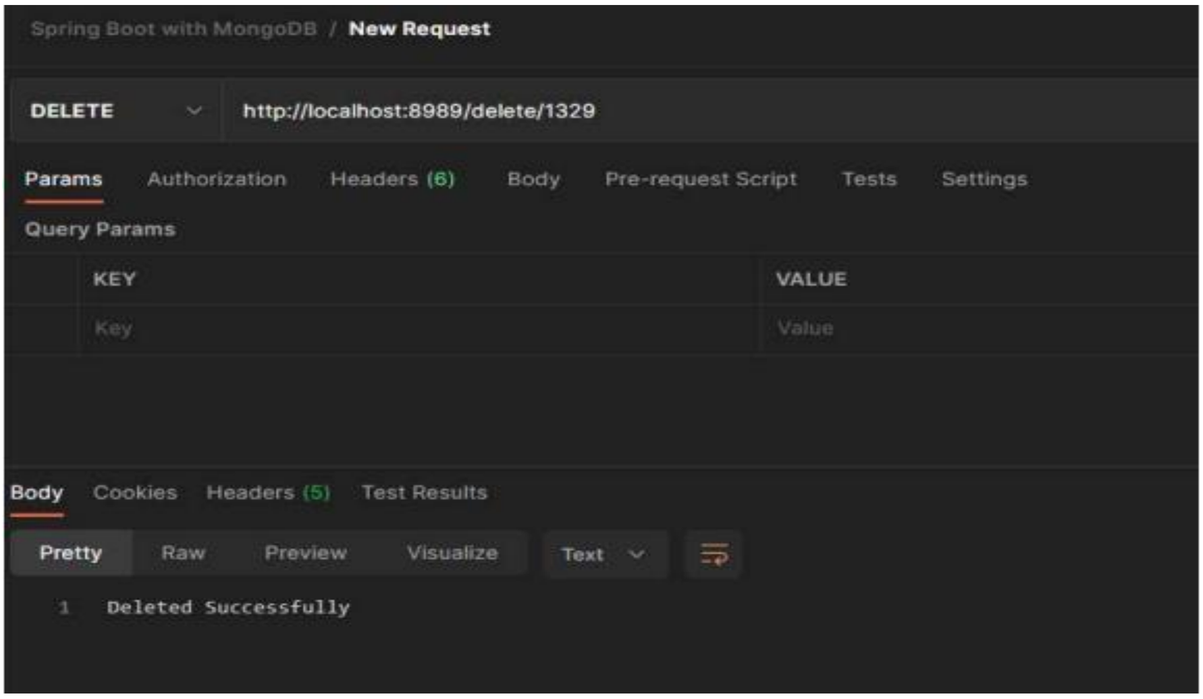
Endpoint 2: GET – http://localhost:8989/findAllBook


Spring Boot with MongoDB / **New Request**

GET http://localhost:8989/findAllBooks

```
[
    {
        "id": 1325,
        "bookName": "Java",
        "authorName": "Tom Hardy"
    },
    {
        "id": 1329,
        "bookName": "Python Fundamentals",
        "authorName": "Tony Stark"
    }
]
```

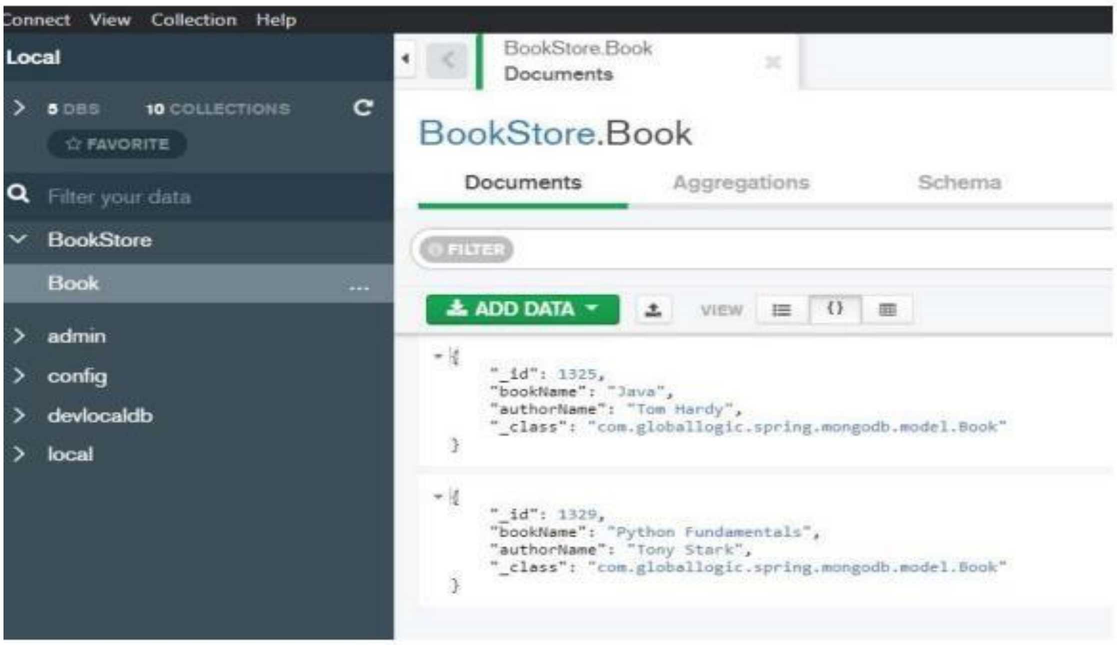
Endpoint 3: DELETE – http://localhost:8989/delete/1329


Spring Boot with MongoDB / **New Request**

DELETE http://localhost:8989/delete/1329

```
Deleted Successfully
```

Finally, MongoDB Compass is as depicted in the below image as shown below as follows:



```
{
    "_id": 1325,
    "bookName": "Java",
    "authorName": "Tom Hardy",
    "_class": "com.globallogic.spring.mongodb.model.Book"
}

{
    "_id": 1329,
    "bookName": "Python Fundamentals",
    "authorName": "Tony Stark",
    "_class": "com.globallogic.spring.mongodb.model.Book"
}
```