Lab Mid

Program: BS Software Engineering

Department of Computer Science

Submitted to:	**Ma’am Yasmeen**
Submitted by:	**Habiba Ashfaq**
Roll no:	**SP22-BSE-032**

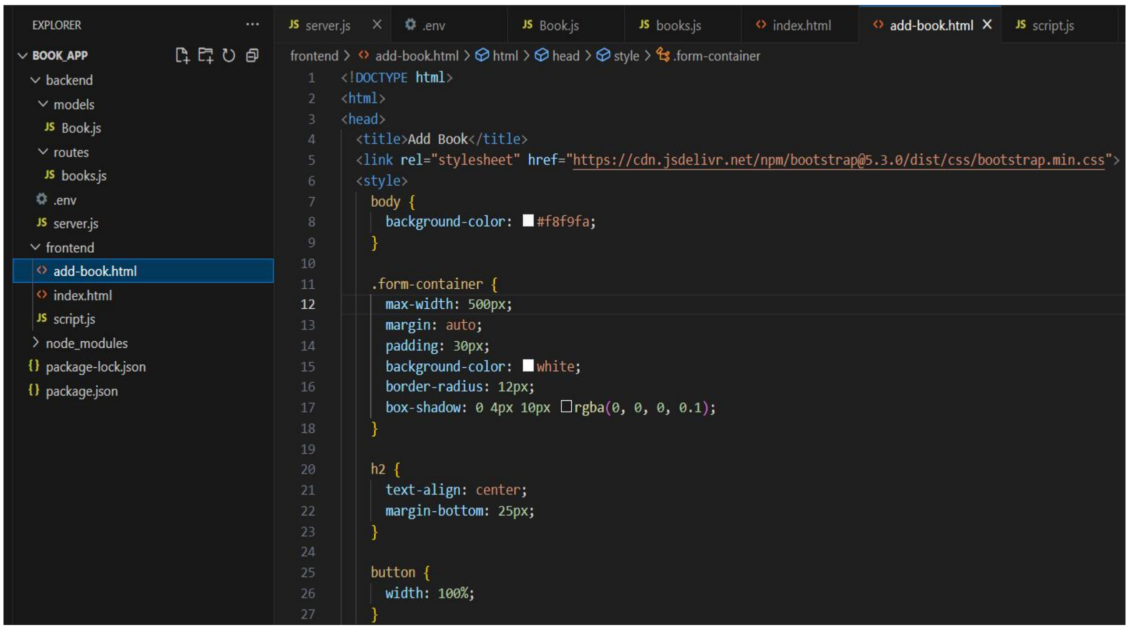
Course Title: Advanced Web Technologies

COMSATS University Islamabad, Vehari Campus.

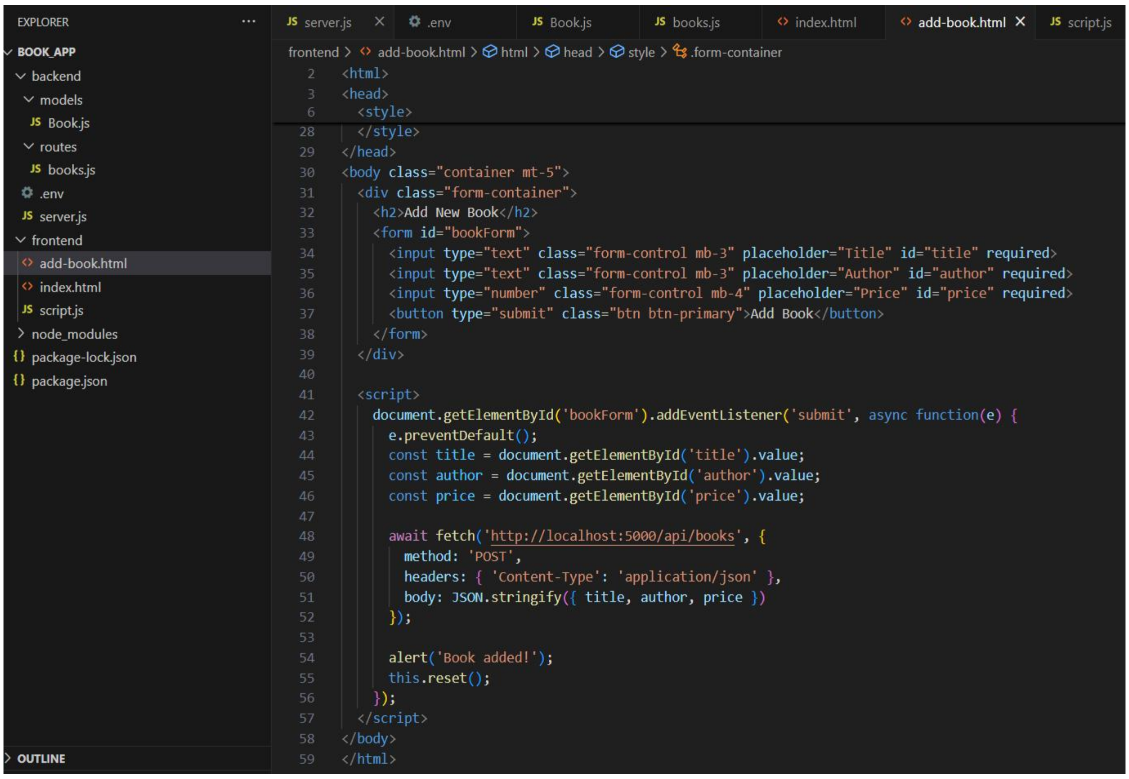
Project Name: Book Management System

Frontend:

Add Book:

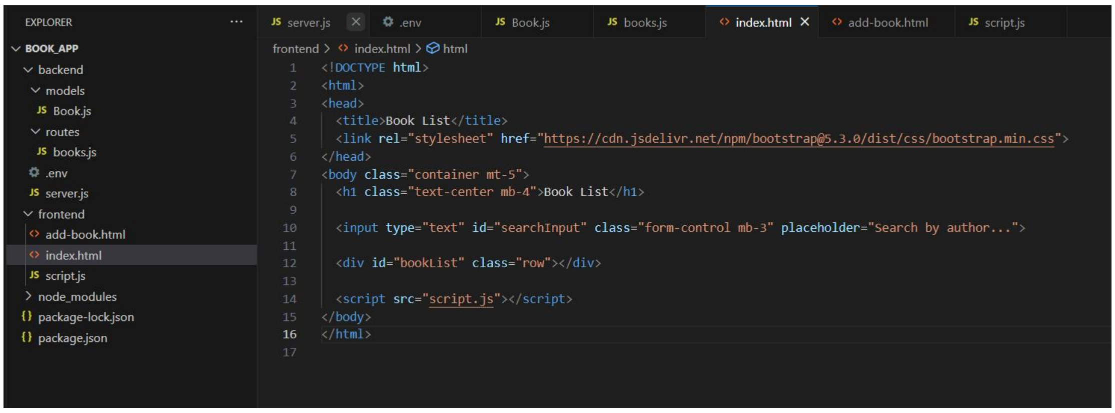


```html
<!DOCTYPE html>
<html>
<head>
  <title>Add Book</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
  <style>
    body {
      background-color: #f8f9fa;
    }

    .form-container {
      max-width: 500px;
      margin: auto;
      padding: 30px;
      background-color: white;
      border-radius: 12px;
      box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
    }

    h2 {
      text-align: center;
      margin-bottom: 25px;
    }

    button {
      width: 100%;
    }
```



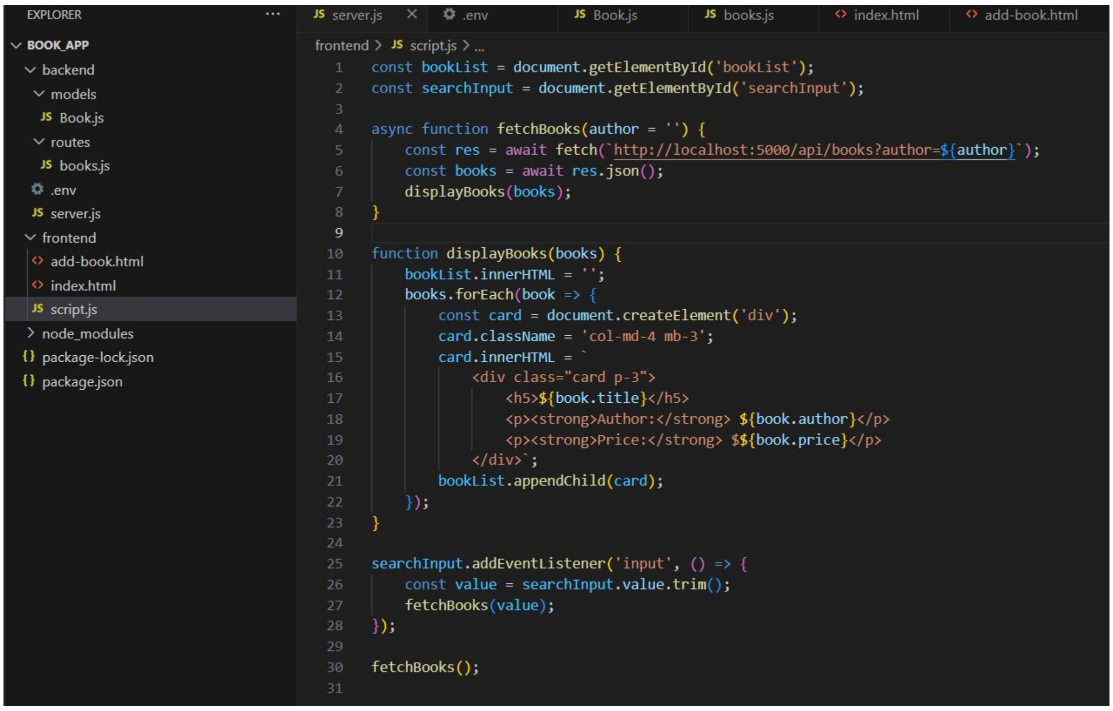
```html
<html>
<head>
  <style>
  </style>
</head>
<body class="container mt-5">
  <div class="form-container">
    <h2>Add New Book</h2>
    <form id="bookForm">
      <input type="text" class="form-control mb-3" placeholder="Title" id="title" required>
      <input type="text" class="form-control mb-3" placeholder="Author" id="author" required>
      <input type="number" class="form-control mb-4" placeholder="Price" id="price" required>
      <button type="submit" class="btn btn-primary">Add Book</button>
    </form>
  </div>

  <script>
    document.getElementById('bookForm').addEventListener('submit', async function(e) {
      e.preventDefault();
      const title = document.getElementById('title').value;
      const author = document.getElementById('author').value;
      const price = document.getElementById('price').value;

      await fetch('http://localhost:5000/api/books', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify({ title, author, price })
      });

      alert('Book added!');
      this.reset();
    });
  </script>
</body>
</html>
```

Index file for Book List:



```html
<!DOCTYPE html>
<html>
<head>
  <title>Book List</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
</head>
<body class="container mt-5">
  <h1 class="text-center mb-4">Book List</h1>

  <input type="text" id="searchInput" class="form-control mb-3" placeholder="Search by author...">

  <div id="bookList" class="row"></div>

  <script src="script.js"></script>
</body>
</html>
```

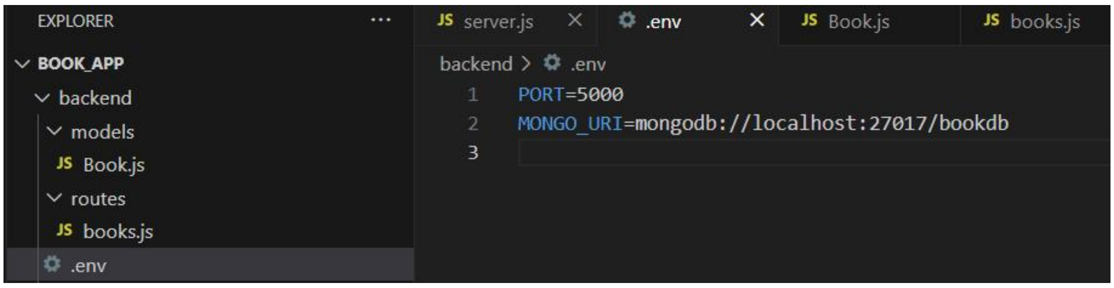
Script File:



```javascript
const bookList = document.getElementById('bookList');
const searchInput = document.getElementById('searchInput');

async function fetchBooks(author = '') {
    const res = await fetch(`http://localhost:5000/api/books?author=${author}`);
    const books = await res.json();
    displayBooks(books);
}

function displayBooks(books) {
    bookList.innerHTML = '';
    books.forEach(book => {
        const card = document.createElement('div');
        card.className = 'col-md-4 mb-3';
        card.innerHTML = `
            <div class="card p-3">
                <h5>${book.title}</h5>
                <p><strong>Author:</strong> ${book.author}</p>
                <p><strong>Price:</strong> $${book.price}</p>
            </div>`;
        bookList.appendChild(card);
    });
}

searchInput.addEventListener('input', () => {
    const value = searchInput.value.trim();
    fetchBooks(value);
});

fetchBooks();
```

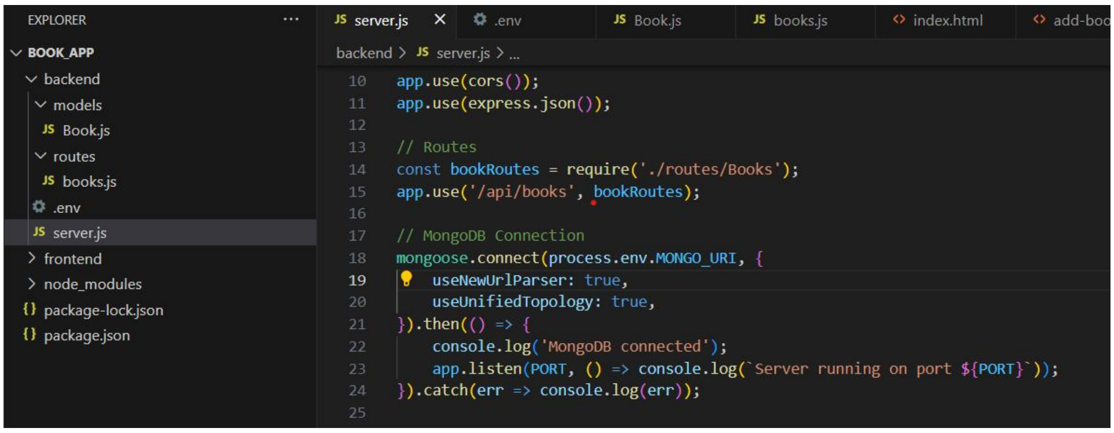
Backend:

.env file:



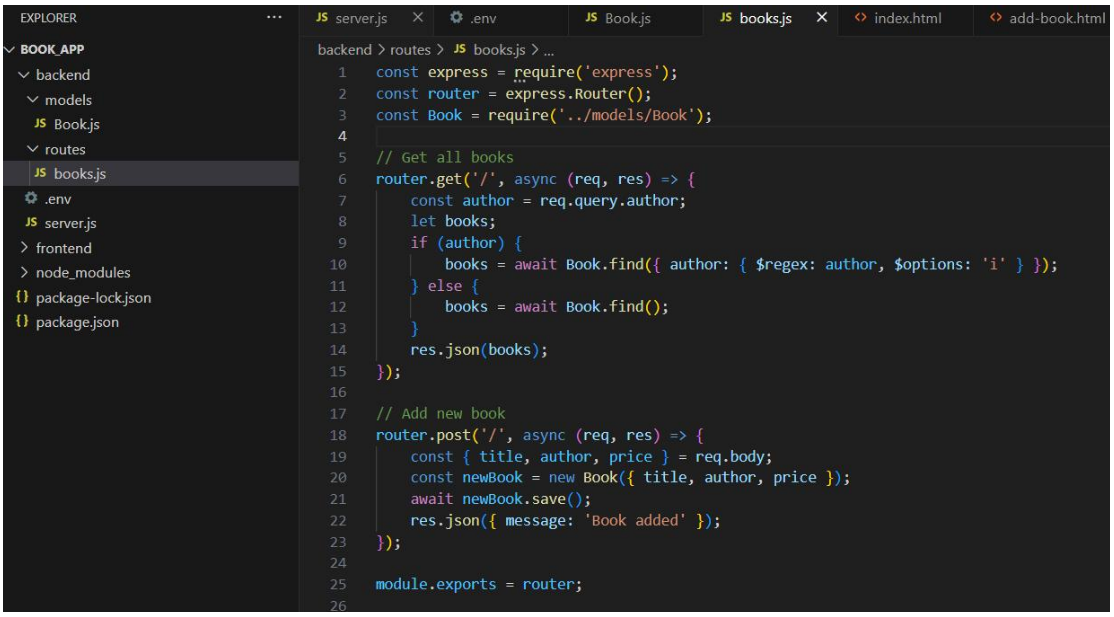
```
PORT=5000
MONGO_URI=mongodb://localhost:27017/bookdb
```

Server.js file:



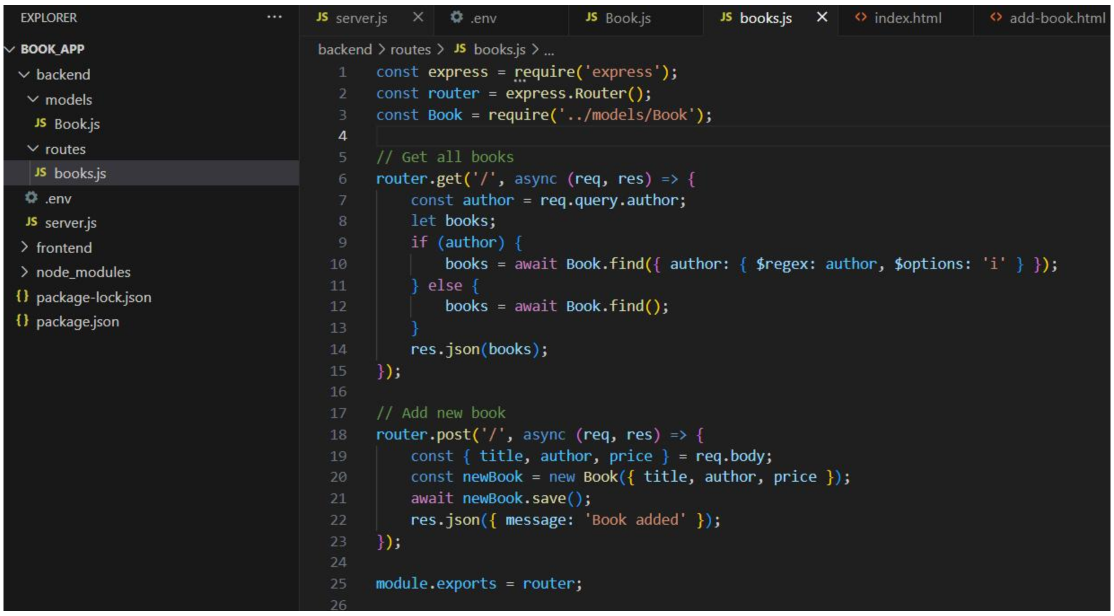
```javascript
app.use(cors());
app.use(express.json());

// Routes
const bookRoutes = require('./routes/Books');
app.use('/api/books', bookRoutes);

// MongoDB Connection
mongoose.connect(process.env.MONGO_URI, {
    useNewUrlParser: true,
    useUnifiedTopology: true,
}).then(() => {
    console.log('MongoDB connected');
    app.listen(PORT, () => console.log(`Server running on port ${PORT}`));
}).catch(err => console.log(err));
```

Routes file:



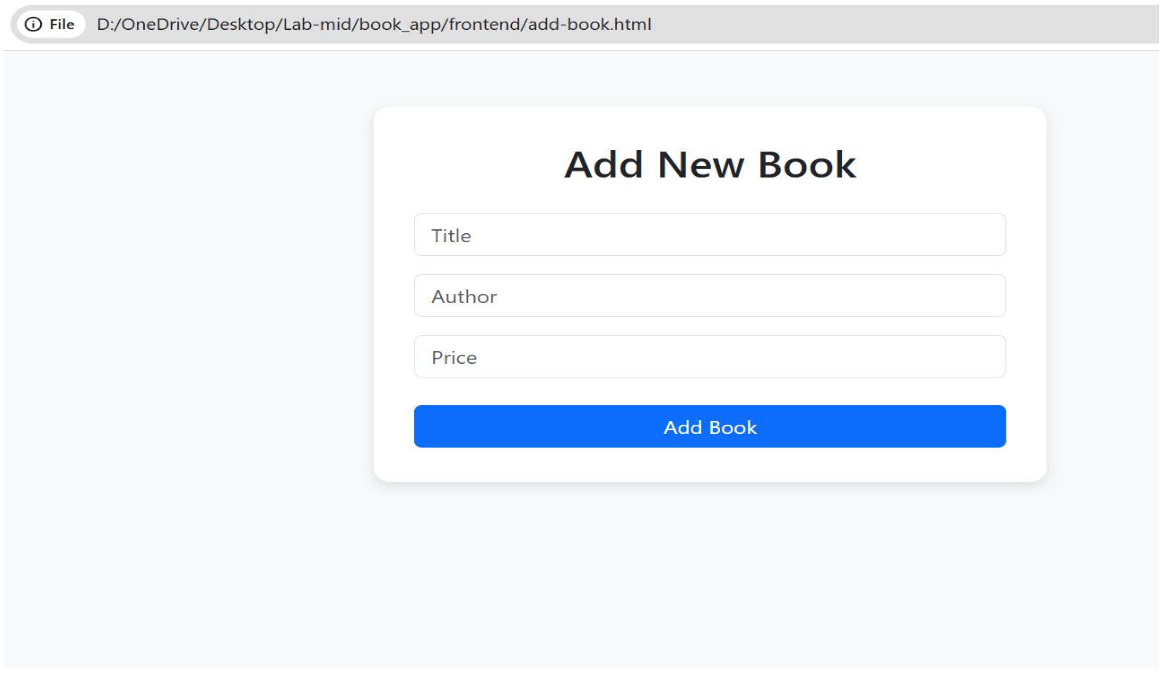
```javascript
const express = require('express');
const router = express.Router();
const Book = require('../models/Book');

// Get all books
router.get('/', async (req, res) => {
    const author = req.query.author;
    let books;
    if (author) {
        books = await Book.find({ author: { $regex: author, $options: 'i' } });
    } else {
        books = await Book.find();
    }
    res.json(books);
});

// Add new book
router.post('/', async (req, res) => {
    const { title, author, price } = req.body;
    const newBook = new Book({ title, author, price });
    await newBook.save();
    res.json({ message: 'Book added' });
});

module.exports = router;
```

Models file:

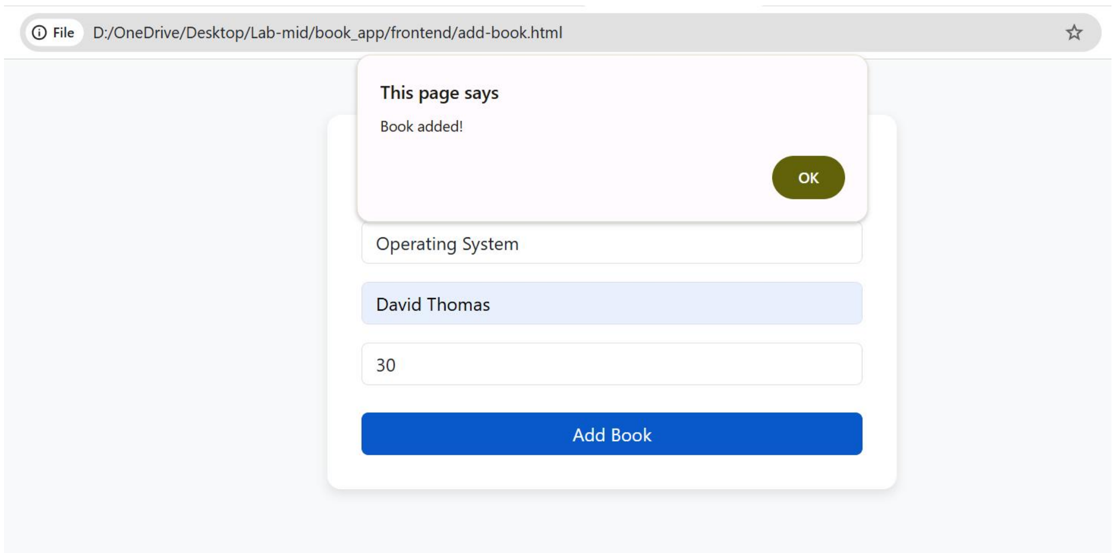


```
const express = require('express');
const router = express.Router();
const Book = require('../models/Book');

// Get all books
router.get('/', async (req, res) => {
    const author = req.query.author;
    let books;
    if (author) {
        books = await Book.find({ author: { $regex: author, $options: 'i' } });
    } else {
        books = await Book.find();
    }
    res.json(books);
});

// Add new book
router.post('/', async (req, res) => {
    const { title, author, price } = req.body;
    const newBook = new Book({ title, author, price });
    await newBook.save();
    res.json({ message: 'Book added' });
});

module.exports = router;
```

Output:



Book added successfully:



D:/OneDrive/Desktop/Lab-mid/book_app/frontend/add-book.html

This page says

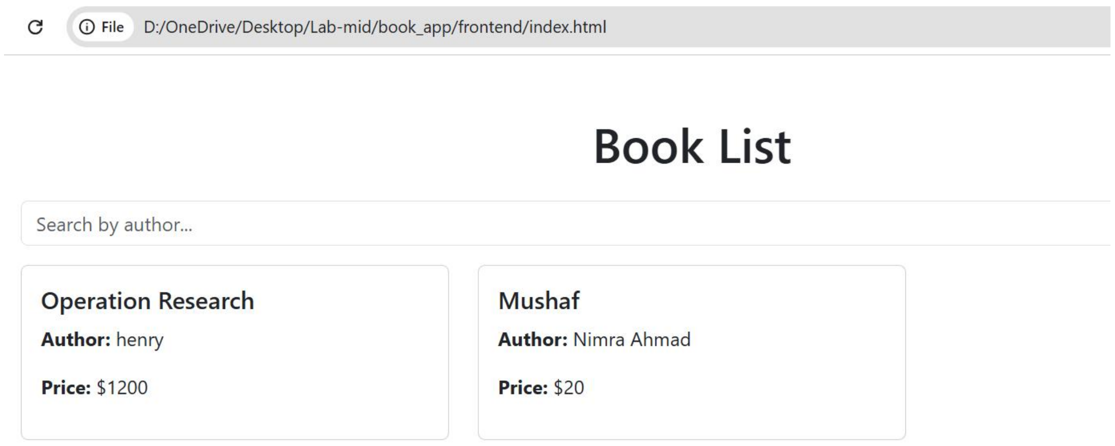
Book added!

Operating System

David Thomas

30

Book List:



D:/OneDrive/Desktop/Lab-mid/book_app/frontend/index.html

Book List

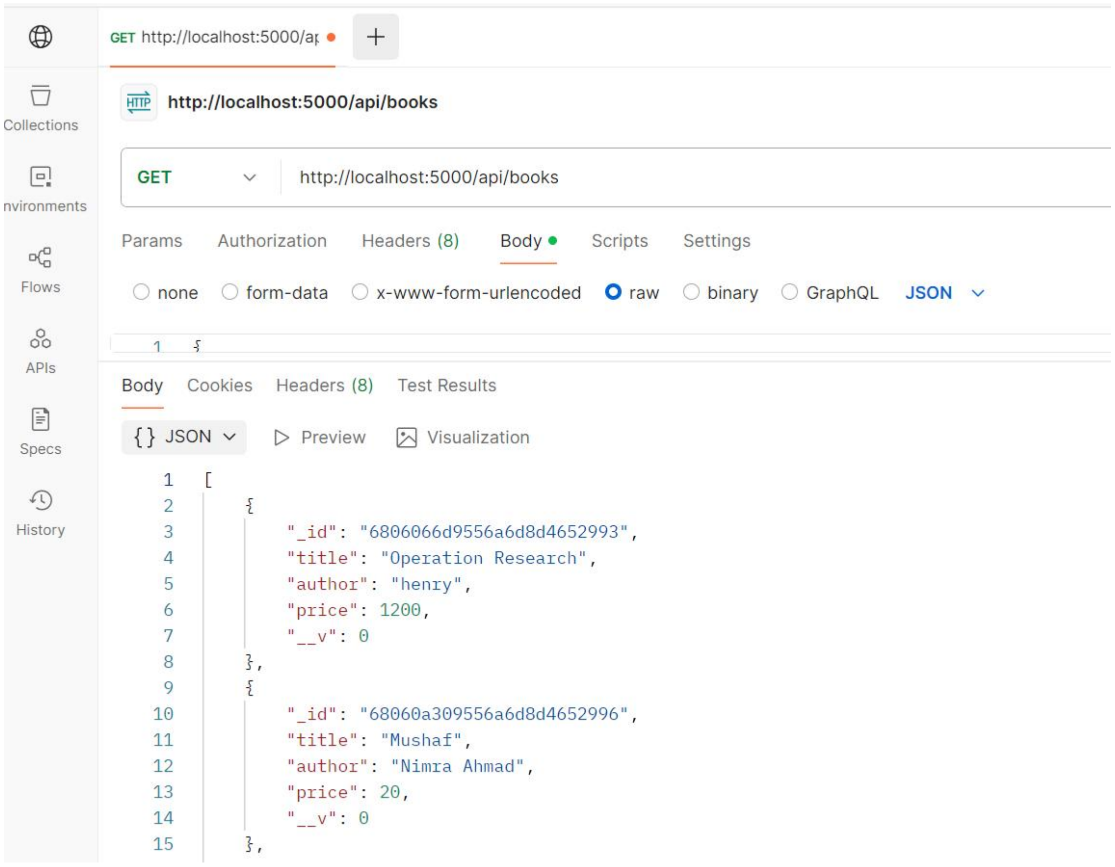
Search by author...

Operation Research
Author: henry
Price: $1200

Mushaf
Author: Nimra Ahmad
Price: $20

Methods Output:

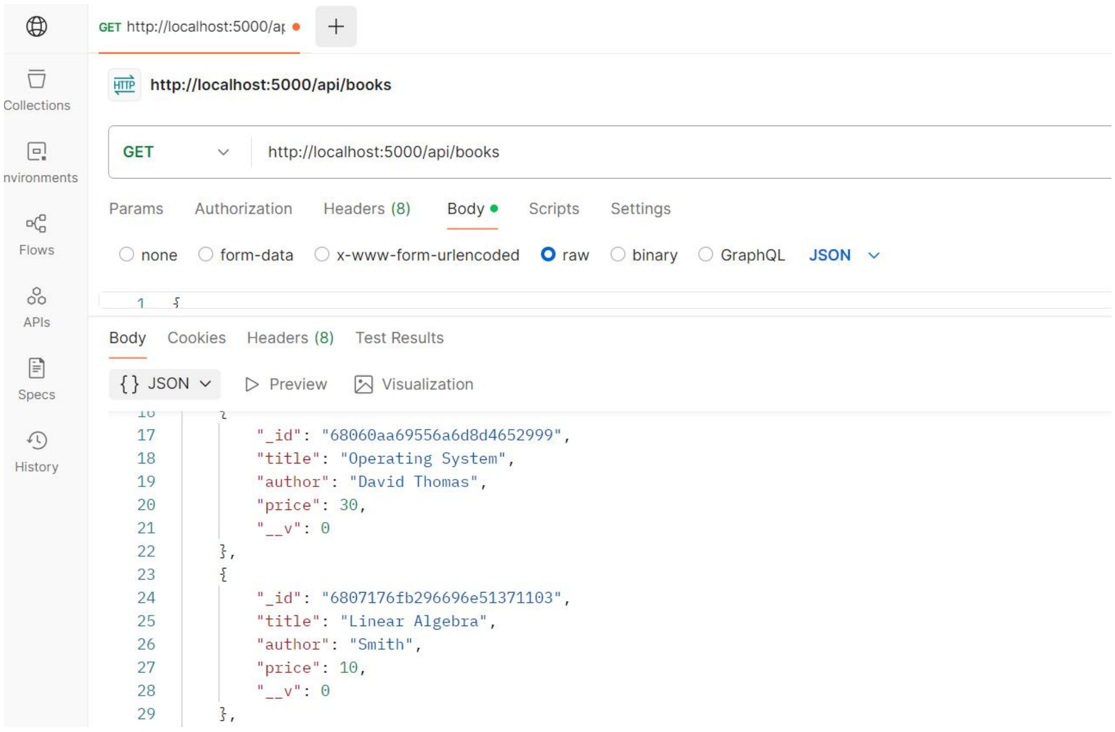
Get Method:



http://localhost:5000/api/books

GET http://localhost:5000/api/books

```
[
    {
        "_id": "6806066d9556a6d8d4652993",
        "title": "Operation Research",
        "author": "henry",
        "price": 1200,
        "__v": 0
    },
    {
        "_id": "68060a309556a6d8d4652996",
        "title": "Mushaf",
        "author": "Nimra Ahmad",
        "price": 20,
        "__v": 0
    },
```

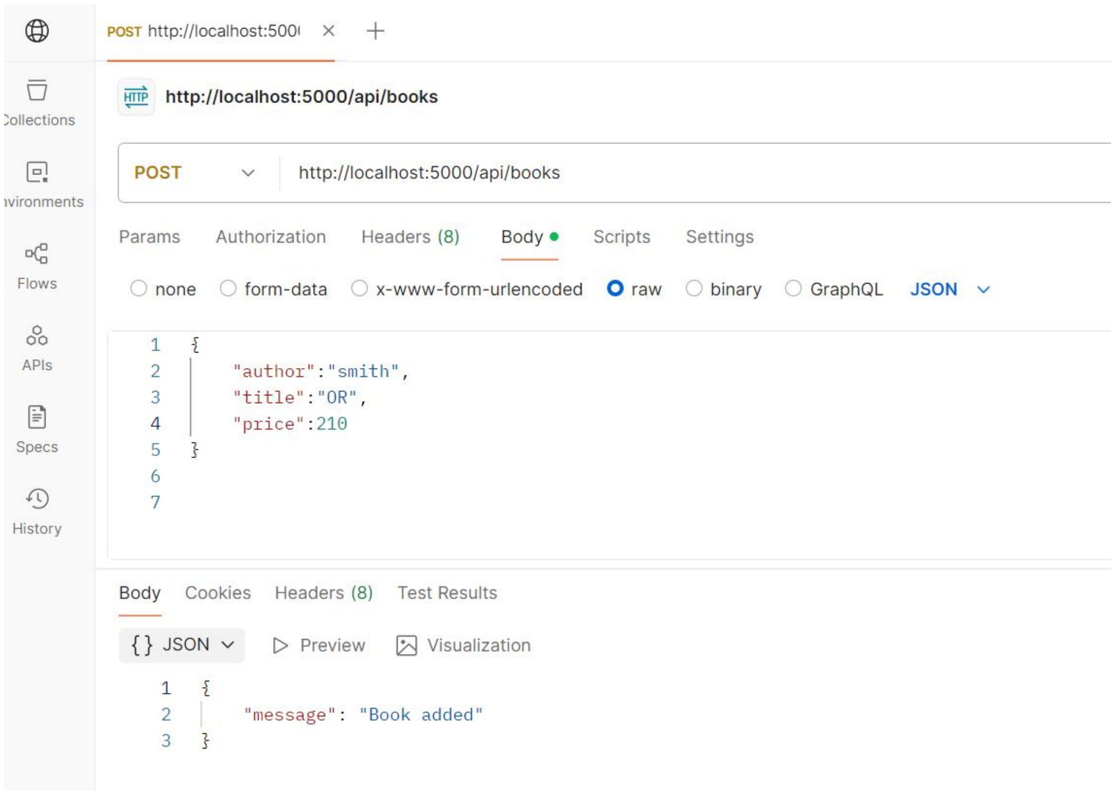


http://localhost:5000/api/books

GET http://localhost:5000/api/books

```
        "_id": "68060aa69556a6d8d4652999",
        "title": "Operating System",
        "author": "David Thomas",
        "price": 30,
        "__v": 0
    },
    {
        "_id": "6807176fb296696e51371103",
        "title": "Linear Algebra",
        "author": "Smith",
        "price": 10,
        "__v": 0
    },
```

Post Method:



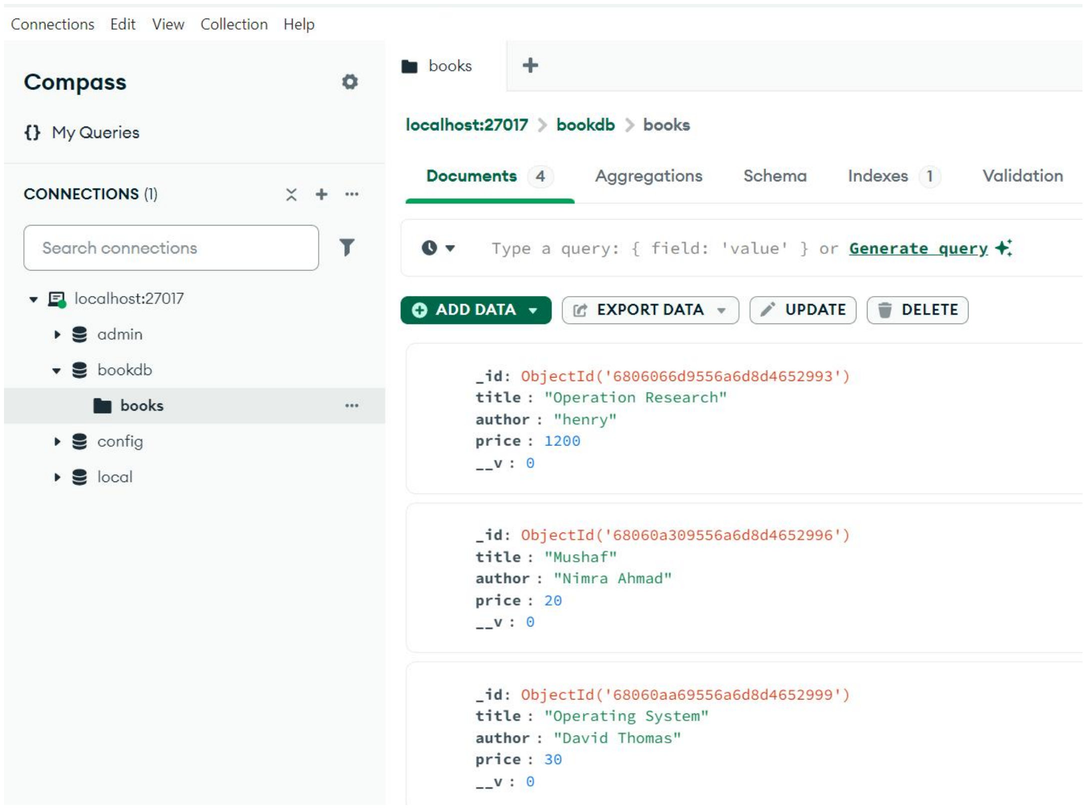
http://localhost:5000/api/books

POST http://localhost:5000/api/books

```
{
    "author":"smith",
    "title":"OR",
    "price":210
}
```

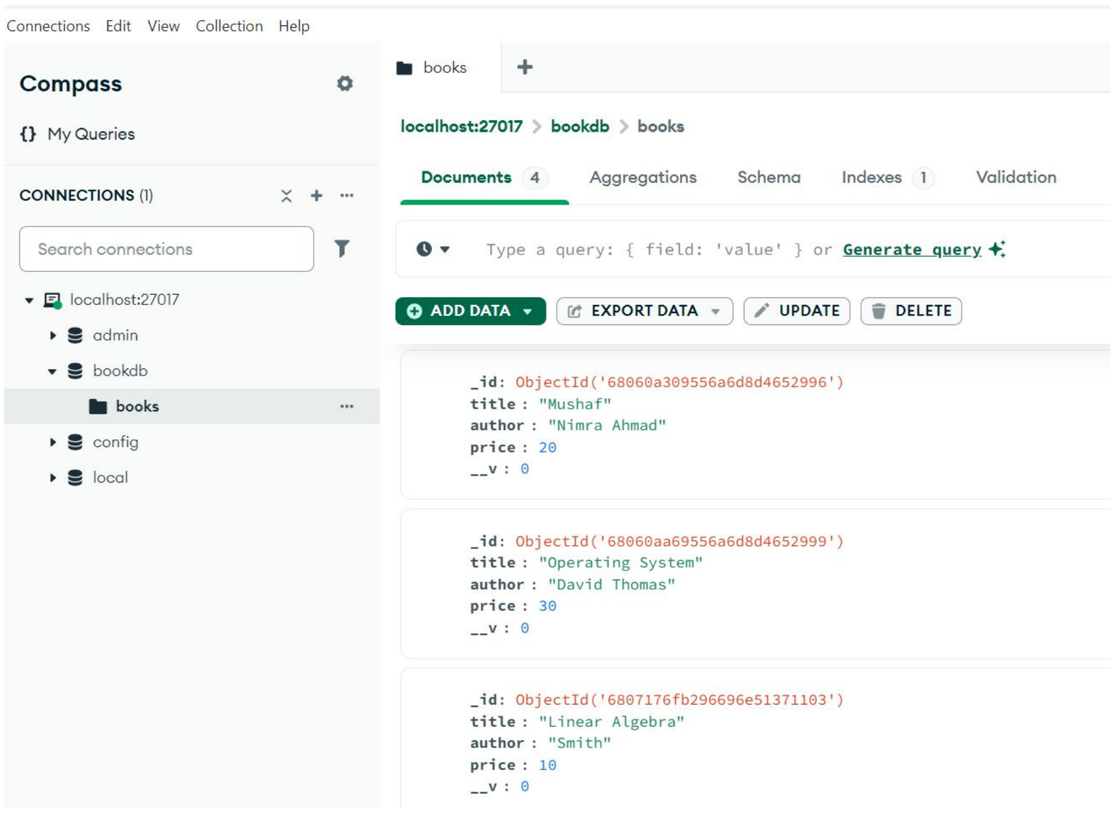
```
{
    "message": "Book added"
}
```

Mongodb:



localhost:27017 > bookdb > books

```
_id: ObjectId('6806066d9556a6d8d4652993')
title : "Operation Research"
author : "henry"
price : 1200
__v : 0

_id: ObjectId('68060a309556a6d8d4652996')
title : "Mushaf"
author : "Nimra Ahmad"
price : 20
__v : 0

_id: ObjectId('68060aa69556a6d8d4652999')
title : "Operating System"
author : "David Thomas"
price : 30
__v : 0
```



```
_id: ObjectId('68060a309556a6d8d4652996')
title : "Mushaf"
author : "Nimra Ahmad"
price : 20
__v : 0

_id: ObjectId('68060aa69556a6d8d4652999')
title : "Operating System"
author : "David Thomas"
price : 30
__v : 0

_id: ObjectId('6807176fb296696e51371103')
title : "Linear Algebra"
author : "Smith"
price : 10
__v : 0
```