E-BOOK CRUD OPERATIONS USING SPRINGBOOT

Applications.properties

```
spring.application.name=book
spring.datasource.url=jdbc:mysql://localhost:3306/database_name
spring.datasource.username=root
spring.datasource.password=root
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
```

Book Application.java

```
package com.example.book;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class BookApplication {
        public static void main(String[] args) {
                SpringApplication.run(BookApplication.class, args);
        }
}
```

Entity Class -Book.java

```
package com.example.book;
import jakarta.persistence.*;

@Entity
public class Book {
   @Id
   @GeneratedValue(strategy = GenerationType.IDENTITY)
   private int id;

   private String title;
   private String author;
   private double price;

   // Getters and Setters
   public int getId() { return id; }
   public void setId(int id) { this.id = id; }

   public String getTitle() { return title; }
   public void setTitle(String title) { this.title = title; }

   public String getAuthor() { return author; }
   public void setAuthor(String author) { this.author = author; }

   public double getPrice() { return price; }
   public void setPrice(double price) { this.price = price; }
}
```

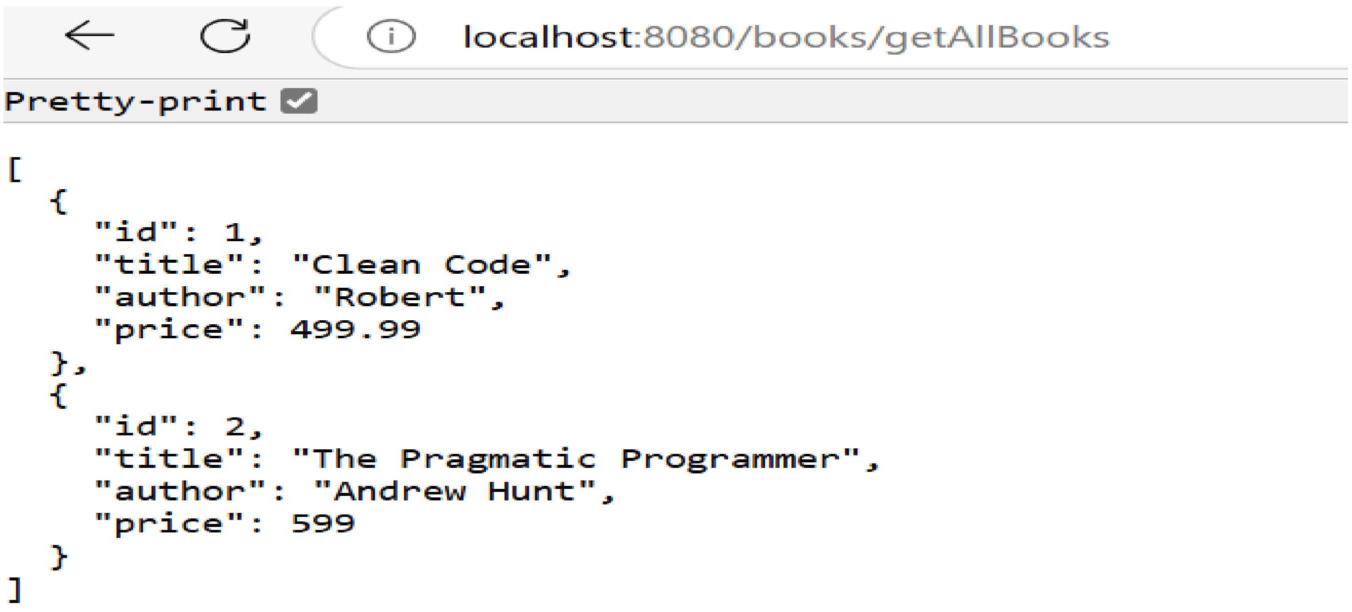
Controller class – BookController.java

```
package com.example.book;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import java.util.List;
import java.util.Optional;

@RestController
@RequestMapping("/books")
public class BookController {

   @Autowired
   private BookRepository bookRepo;

   // I) Get all books
   @GetMapping("/getAllBooks")
   public List<Book> getAllBooks() {
      return bookRepo.findAll();
   }

   // II) Get book by ID
   @GetMapping("/getBookById/{id}")
   public Optional<Book> getBookById(@PathVariable int id) {
      return bookRepo.findById(id);
   }

   // III-a) Add book
   @PostMapping("/addBook")
   public Book addBook(@RequestBody Book book) {
      return bookRepo.save(book);
   }

   // III-b) Update book
   @PutMapping("/updateBook/{id}")
   public Book updateBook(@PathVariable int id, @RequestBody Book updatedBook) {
      return bookRepo.findById(id).map(book -> {
         book.setTitle(updatedBook.getTitle());
         book.setAuthor(updatedBook.getAuthor());
         book.setPrice(updatedBook.getPrice());
         return bookRepo.save(book);
      }).orElseGet(() -> {
         updatedBook.setId(id);
         return bookRepo.save(updatedBook);
      });
   }

   // III-c) Delete book
   @DeleteMapping("/deleteBook/{id}")
   public void deleteBook(@PathVariable int id) {
      bookRepo.deleteById(id);
   }
}
```

Repository Interface – BookRepository.java

```
package com.example.book;
import org.springframework.data.jpa.repository.JpaRepository;

public interface BookRepository extends JpaRepository<Book, Integer> {
}
```

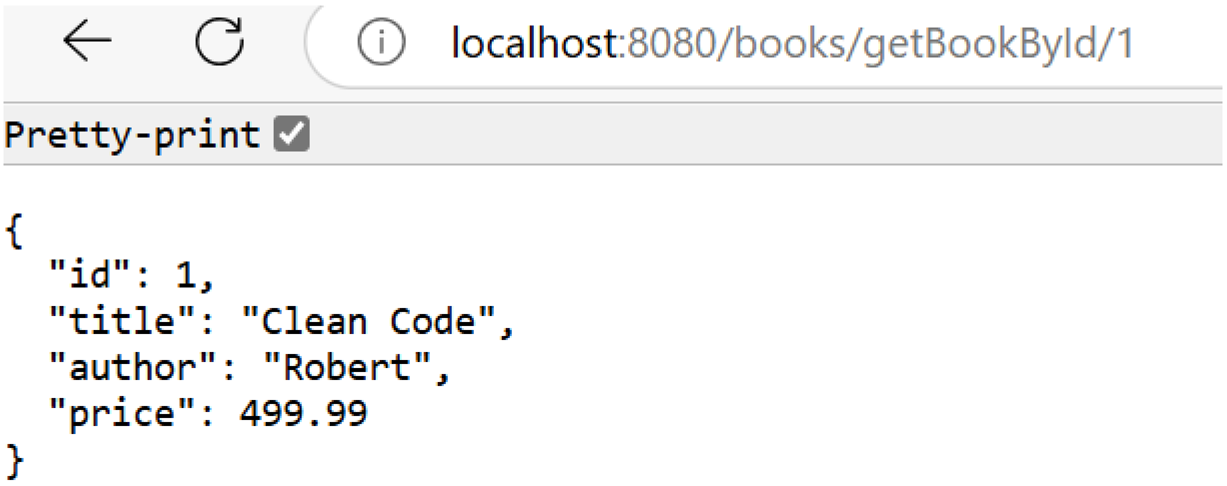
OUTPUTS



localhost:8080/books/getAllBooks

Pretty-print ✓

```
[
  {
    "id": 1,
    "title": "Clean Code",
    "author": "Robert",
    "price": 499.99
  },
  {
    "id": 2,
    "title": "The Pragmatic Programmer",
    "author": "Andrew Hunt",
    "price": 599
  }
]
```



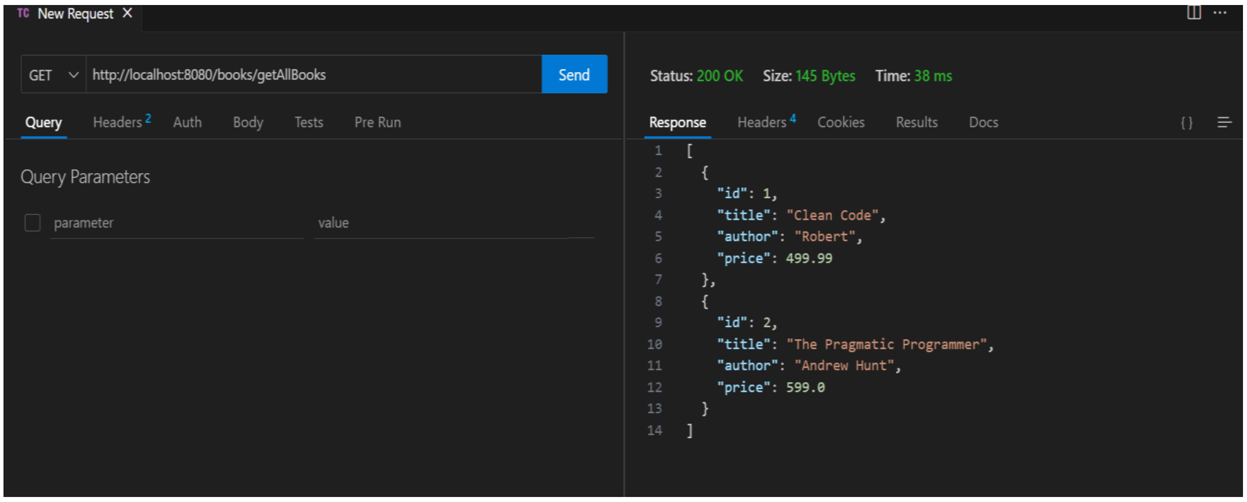
localhost:8080/books/getBookById/1

Pretty-print ✓

```
{
  "id": 1,
  "title": "Clean Code",
  "author": "Robert",
  "price": 499.99
}
```

TEST SPRINGBOOT API USING THUNDER CLIENT/POSTMAN

1) GET http://localhost:8080/books/getAllBooks

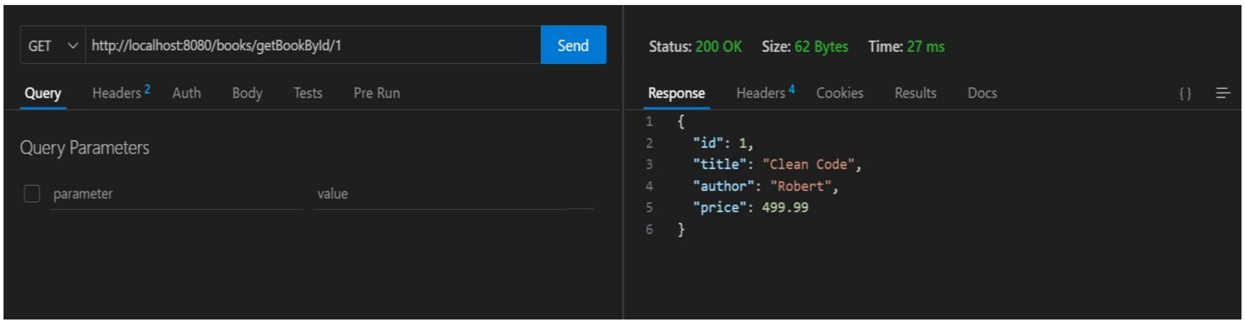


http://localhost:8080/books/getAllBooks

Status: 200 OK Size: 145 Bytes Time: 38 ms

```
[
  {
    "id": 1,
    "title": "Clean Code",
    "author": "Robert",
    "price": 499.99
  },
  {
    "id": 2,
    "title": "The Pragmatic Programmer",
    "author": "Andrew Hunt",
    "price": 599.0
  }
]
```

2) GET http://localhost:8080/books/getBookById/1

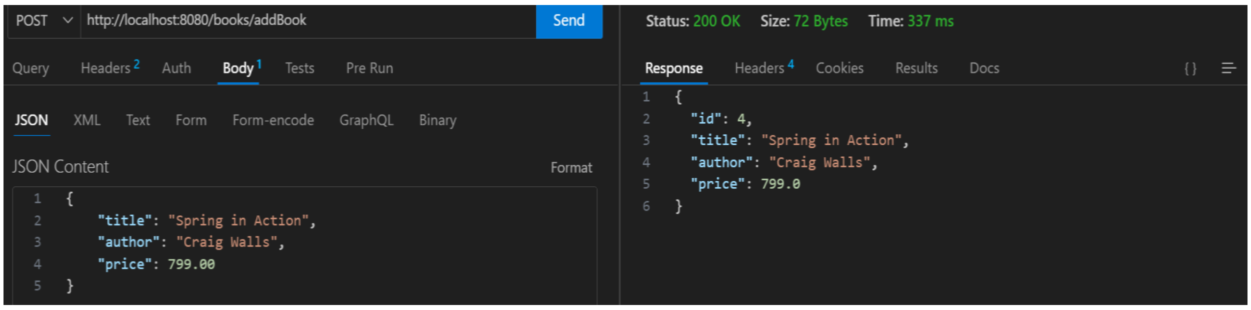


http://localhost:8080/books/getBookById/1

Status: 200 OK Size: 62 Bytes Time: 27 ms

```
{
  "id": 1,
  "title": "Clean Code",
  "author": "Robert",
  "price": 499.99
}
```

3) POST http://localhost:8080/books/addBook (send JSON body)



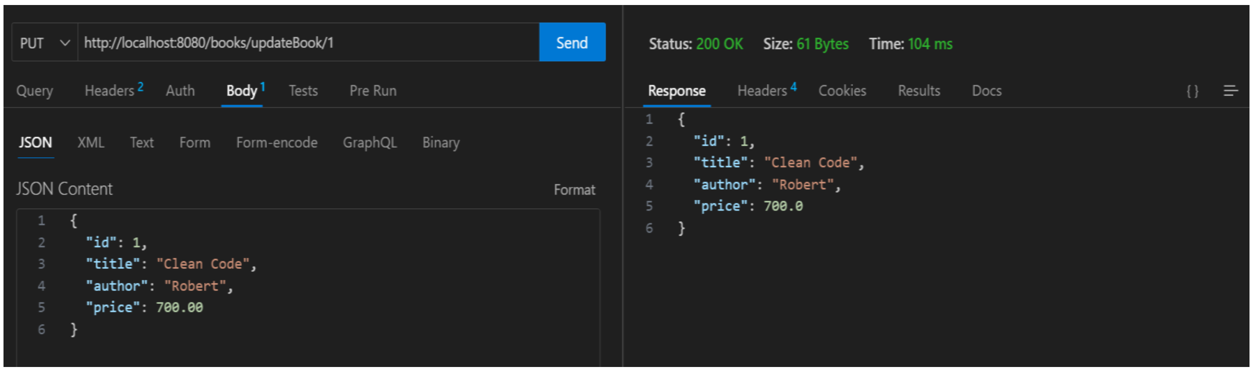
http://localhost:8080/books/addBook

```
{
    "title": "Spring in Action",
    "author": "Craig Walls",
    "price": 799.00
}
```

Status: 200 OK Size: 72 Bytes Time: 337 ms

```
{
  "id": 4,
  "title": "Spring in Action",
  "author": "Craig Walls",
  "price": 799.0
}
```

4) PUT http://localhost:8080/books/updateBook/1 (send JSON body)



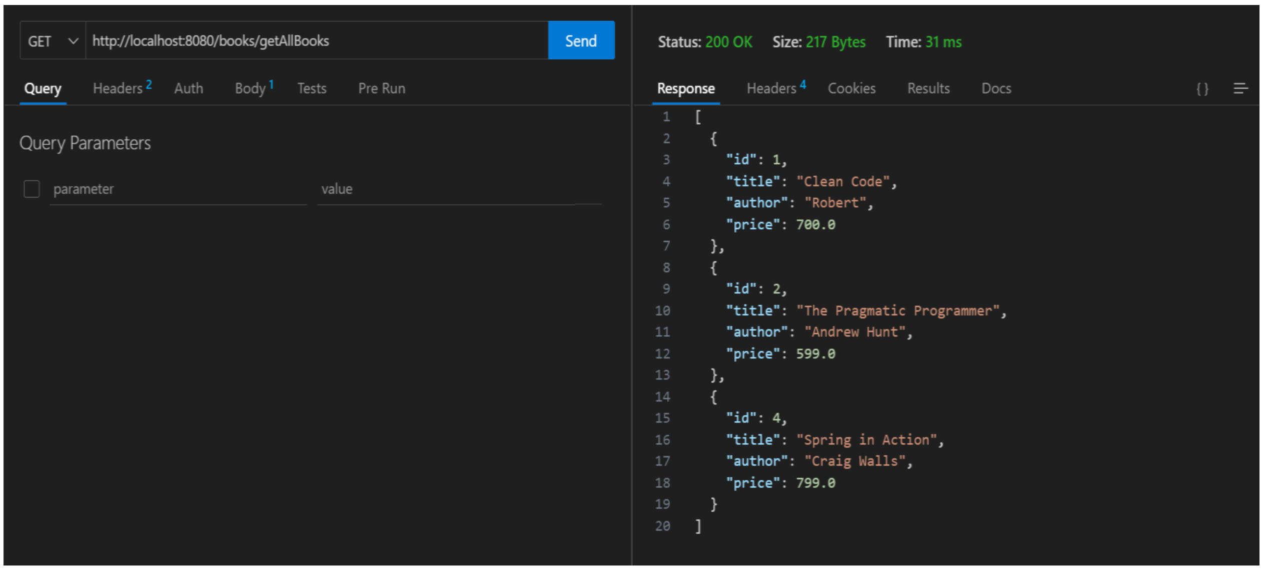
PUT http://localhost:8080/books/updateBook/1

JSON Content

```
{
  "id": 1,
  "title": "Clean Code",
  "author": "Robert",
  "price": 700.00
}
```

Status: 200 OK Size: 61 Bytes Time: 104 ms

```
{
  "id": 1,
  "title": "Clean Code",
  "author": "Robert",
  "price": 700.0
}
```

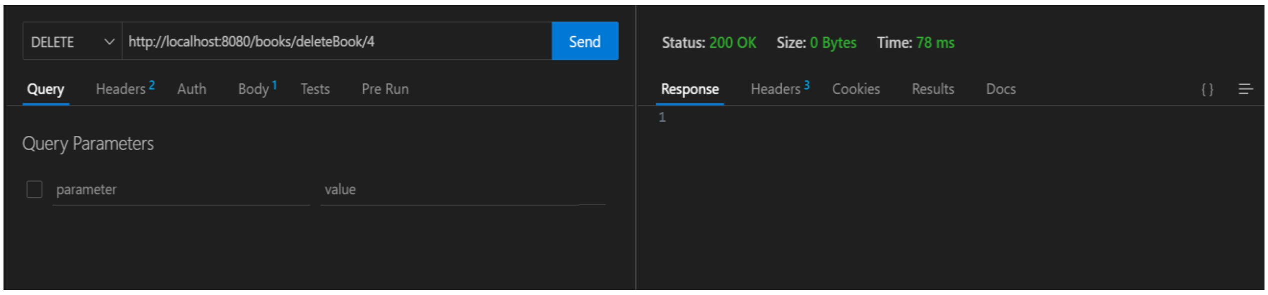


GET http://localhost:8080/books/getAllBooks

Status: 200 OK Size: 217 Bytes Time: 31 ms

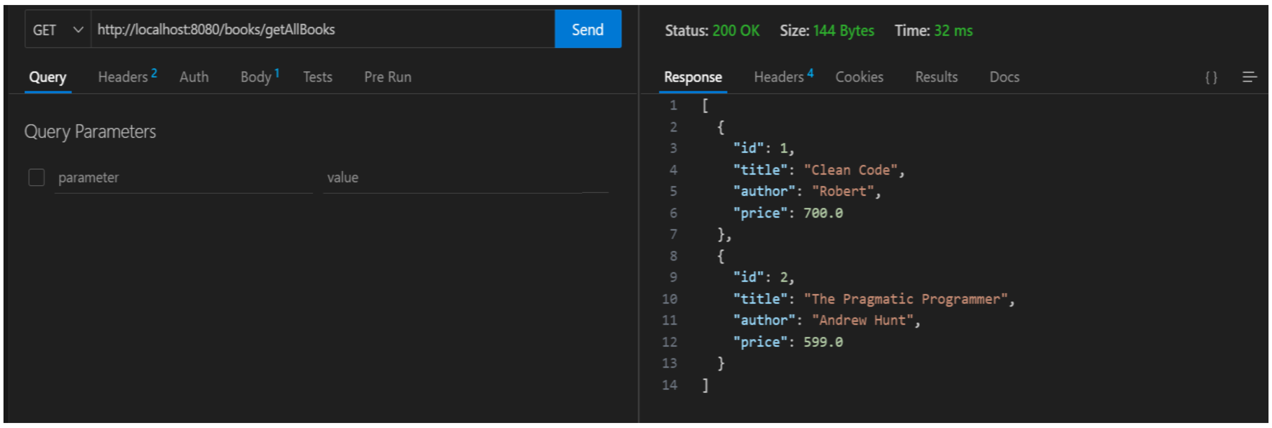
```
[
  {
    "id": 1,
    "title": "Clean Code",
    "author": "Robert",
    "price": 700.0
  },
  {
    "id": 2,
    "title": "The Pragmatic Programmer",
    "author": "Andrew Hunt",
    "price": 599.0
  },
  {
    "id": 4,
    "title": "Spring in Action",
    "author": "Craig Walls",
    "price": 799.0
  }
]
```

5) DELETE http://localhost:8080/books/deleteBook/1



DELETE http://localhost:8080/books/deleteBook/4

Status: 200 OK Size: 0 Bytes Time: 78 ms



GET http://localhost:8080/books/getAllBooks

Status: 200 OK Size: 144 Bytes Time: 32 ms

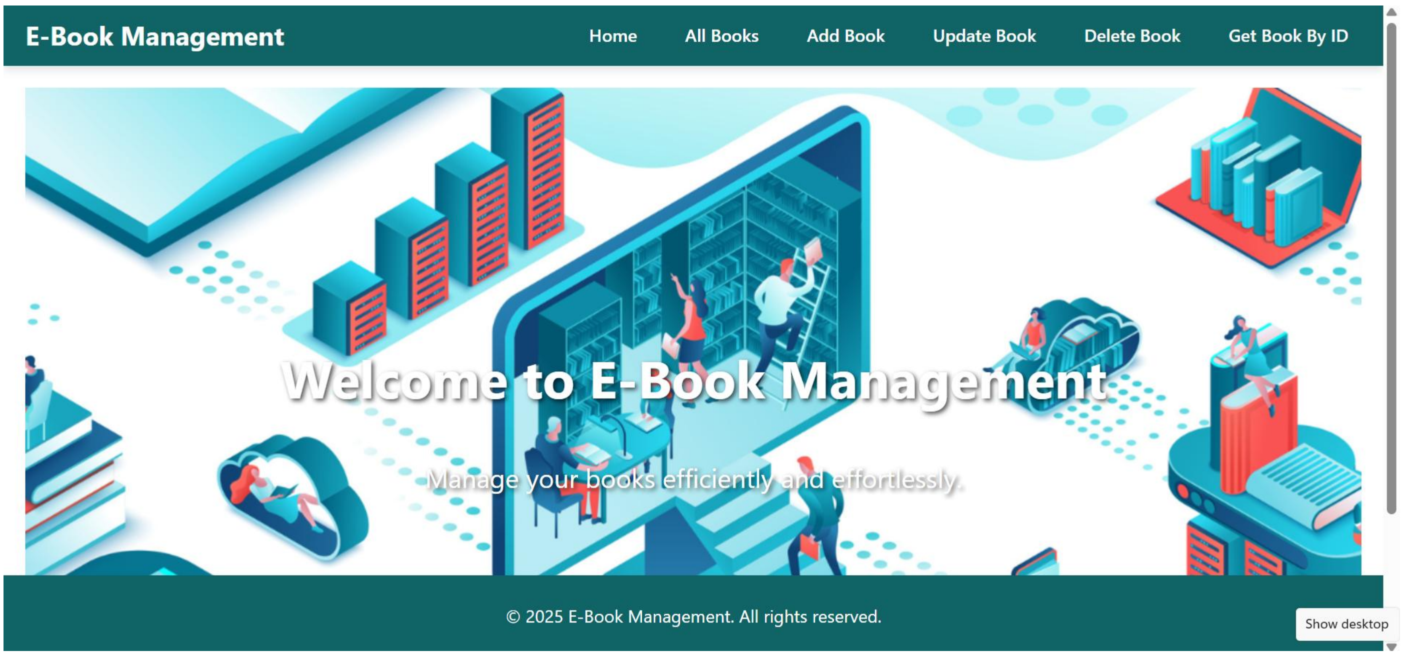
```
[
  {
    "id": 1,
    "title": "Clean Code",
    "author": "Robert",
    "price": 700.0
  },
  {
    "id": 2,
    "title": "The Pragmatic Programmer",
    "author": "Andrew Hunt",
    "price": 599.0
  }
]
```

INTEGRATION & MIGRATION CODE OF REACT FRONTEND & SPRINGBOOT BACKEND – E-BOOK

Home.js

```
import React from 'react';

function Home() {
  return (
    <div style={styles.container}>
      <h1 style={styles.heading}>Welcome to E-Book Management</h1>
      <p style={styles.subheading}>Manage your books efficiently and effortlessly.</p>
    </div>
  );
}
const styles = {
  container: {
    height: '100vh',
    backgroundImage: 'url("/background_ebook.png")', // Replace with a direct image URL
    backgroundSize: 'cover',
    backgroundPosition: 'center',
    display: 'flex',
    flexDirection: 'column',
    justifyContent: 'center',
    alignItems: 'center',
    color: '#fff',
    textShadow: '2px 2px 4px rgba(0, 0, 0, 0.7)',
  },
  heading: {
    fontSize: '48px',
    marginBottom: '20px',
  },
  subheading: {
    fontSize: '24px',
  },
};
export default Home;
```

Output :



Welcome to E-Book Management

Manage your books efficiently and effortlessly.

© 2025 E-Book Management. All rights reserved.

Header.js

```
import React from 'react';
import { Link } from 'react-router-dom';

function Header() {
  return (
    <header style={styles.header}>
      <h1 style={styles.title}>E-Book Management</h1>
      <nav style={styles.nav}>
        <Link to="/" style={styles.link}>Home</Link>
        <Link to="/allBooks" style={styles.link}>All Books</Link>
        <Link to="/addBook" style={styles.link}>Add Book</Link>
        <Link to="/updateBook" style={styles.link}>Update Book</Link>
        <Link to="/deleteBook" style={styles.link}>Delete Book</Link>
        <Link to="/getBookById" style={styles.link}>Get Book By ID</Link>
      </nav>
    </header>
  );
}

const styles = {
  header: {
    backgroundColor: '#116466',
    color: '#fff',
    padding: '10px 20px',
    display: 'flex',
    justifyContent: 'space-between',
    alignItems: 'center',
    boxShadow: '0 4px 8px rgba(0, 0, 0, 0.1)',
  },
  title: {
    margin: 0,
    fontSize: '24px',
    fontWeight: 'bold',
  },
  nav: {
    display: 'flex',
    gap: '20px',
  },
  link: {
    color: '#fff',
    textDecoration: 'none',
    fontSize: '16px',
    fontWeight: '500',
    padding: '8px 12px',
    borderRadius: '4px',
    transition: 'background-color 0.3s',
  },
  linkHover: {
    backgroundColor: '#45a049',
  },
};

export default Header;
```

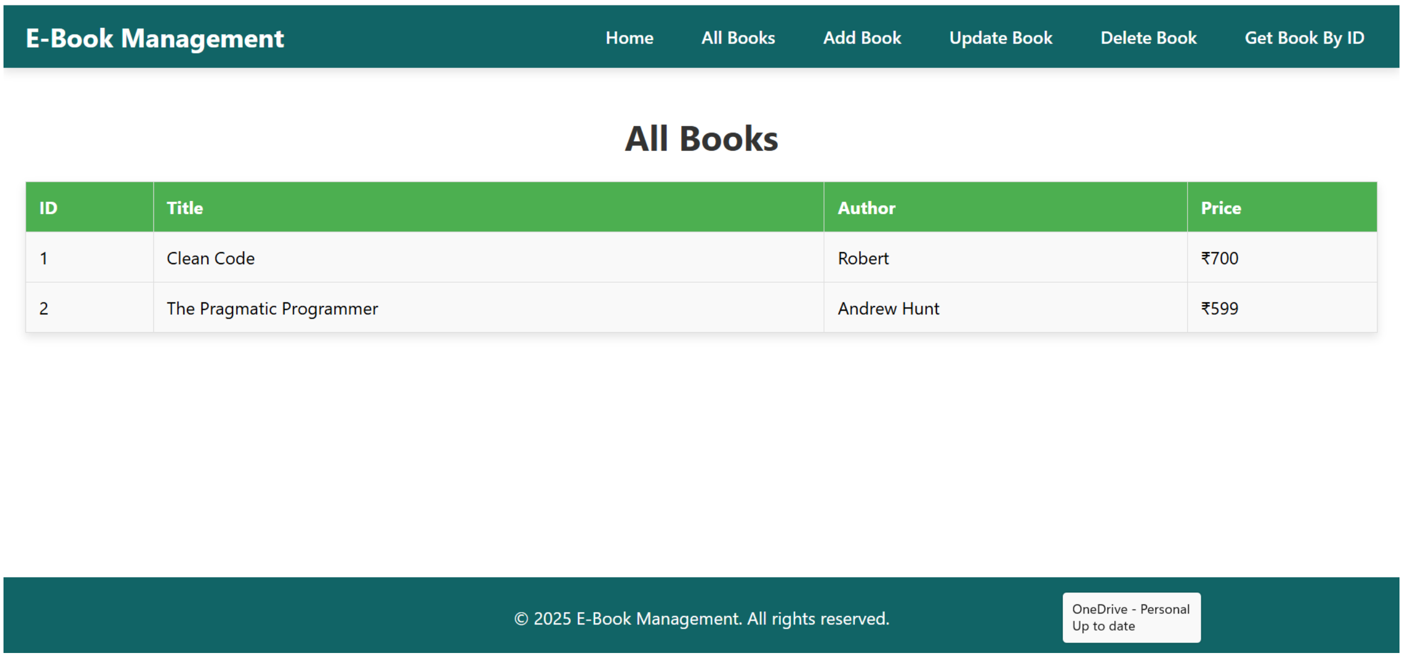
Footer.js

```
import React from 'react';

function Footer() {
  return (
    <footer style={styles.footer}>
      <p>&copy; 2025 E-Book Management. All rights reserved.</p>
    </footer>
  );
}

const styles = {
  footer: {
    backgroundColor: '#116466',
    color: '#fff',
    textAlign: 'center',
    padding: '10px 0',
    position: 'fixed',
    bottom: 0,
    width: '100%',
  },
};

export default Footer;
```

GetAllBooks.js

```
import React, { useState, useEffect } from 'react';
import axiosInstance from '../axiosConfig';

function GetAllBooks() {
  const [books, setBooks] = useState([]);
  const [error, setError] = useState('');

  useEffect(() => {
    const fetchBooks = async () => {
      try {
        const response = await axiosInstance.get('/getAllBooks');
        setBooks(response.data);
      } catch (err) {
        setError('Error fetching books.');
      }
    };
    fetchBooks();
  }, []);

  return (
    <div>
      <h1 style={styles.heading}>All Books</h1>
      {error && <p style={{ color: 'red' }}>{error}</p>}
      {books.length > 0 ? (
        <table style={styles.table}>
          <thead>
            <tr>
              <th style={styles.th}>ID</th>
              <th style={styles.th}>Title</th>
              <th style={styles.th}>Author</th>
              <th style={styles.th}>Price</th>
            </tr>
          </thead>
          <tbody>
            {books.map((book) => (
              <tr key={book.id} style={styles.tr}>
                <td style={styles.td}>{book.id}</td>
                <td style={styles.td}>{book.title}</td>
                <td style={styles.td}>{book.author}</td>
                <td style={styles.td}>₹{book.price}</td>
              </tr>
            ))}
          </tbody>
        </table>
      ) : (
        <p>No books available.</p>
      )}
    </div>
  );
}

const styles = {
  heading: {
    textAlign: 'center',
    marginBottom: '20px',
    color: '#333',
  },
  table: {
    width: '100%',
    borderCollapse: 'collapse',
    marginTop: '20px',
    boxShadow: '0 4px 8px rgba(0, 0, 0, 0.1)',
  },
  th: {
    border: '1px solid #ddd',
    padding: '12px',
    backgroundColor: '#4CAF50',
    color: 'white',
    textAlign: 'left',
  },
  td: {
    border: '1px solid #ddd',
    padding: '12px',
    textAlign: 'left',
  },
  tr: {
    backgroundColor: '#f9f9f9',
  },
  trHover: {
    backgroundColor: '#f1f1f1',
  },
};

export default GetAllBooks;
```

<u>Outputs :</u>



All Books

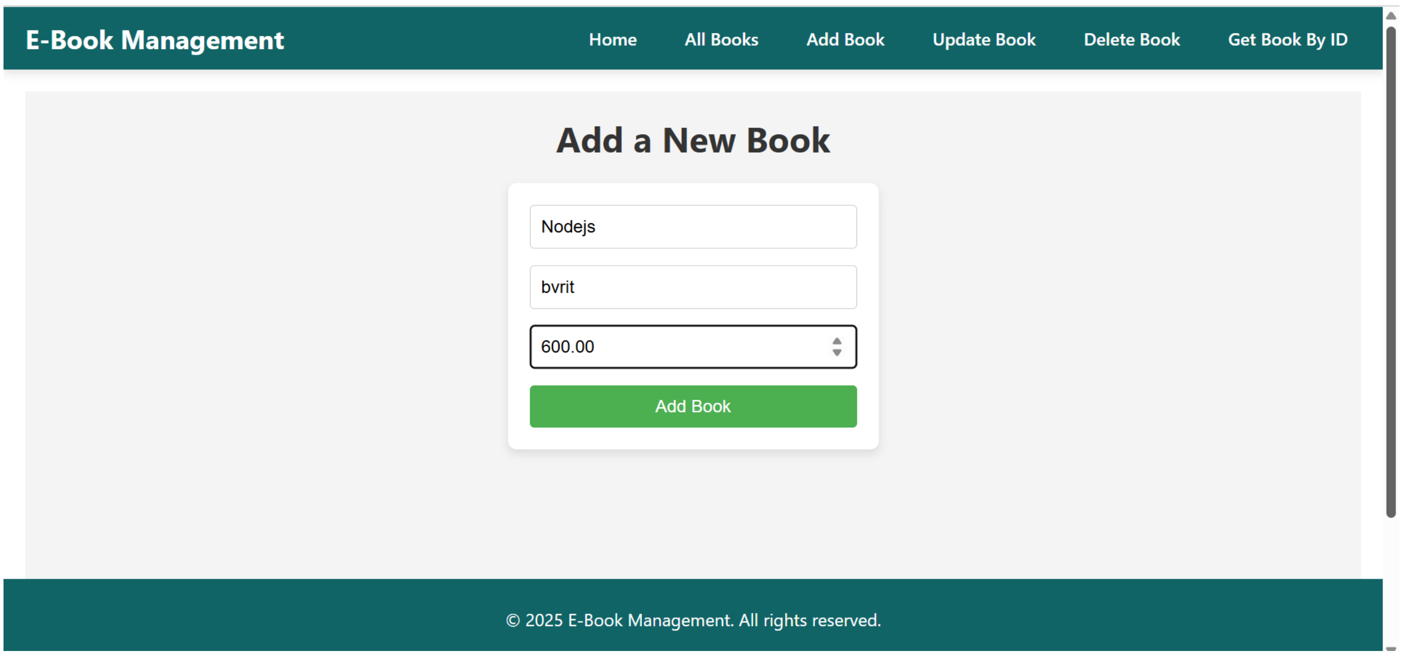
ID	Title	Author	Price
1	Clean Code	Robert	₹700
2	The Pragmatic Programmer	Andrew Hunt	₹599

© 2025 E-Book Management. All rights reserved.

AddBook.js

```
import React, { useState } from 'react';
import axiosInstance from '../axiosConfig';
function AddBook() {
  const [book, setBook] = useState({ title: '', author: '', price: '' });
  const [message, setMessage] = useState('');

  const addBook = async () => {
    try {
      await axiosInstance.post('/addBook', book);
      setMessage('Book added successfully!');
      setBook({ title: '', author: '', price: '' });
    } catch (err) {
      setMessage('Error adding book.');
    }
  };
  return (
    <div style={styles.container}>
      <h1 style={styles.heading}>Add a New Book</h1>
      <div style={styles.form}>
        <input
          type="text"
          placeholder="Title"
          value={book.title}
          onChange={(e) => setBook({ ...book, title: e.target.value })}
          style={styles.input}
        />
        <input
          type="text"
          placeholder="Author"
          value={book.author}
          onChange={(e) => setBook({ ...book, author: e.target.value })}
          style={styles.input}
        />
        <input
          type="number"
          placeholder="Price"
          value={book.price}
          onChange={(e) => setBook({ ...book, price: e.target.value })}
          style={styles.input}
        />
        <button onClick={addBook} style={styles.button}>Add Book</button>
        {message && <p style={styles.message}>{message}</p>}
      </div>
    </div>
  );
}
const styles = {
  container: {
    display: 'flex',
    flexDirection: 'column',
    justifyContent: 'flex-start',
    alignItems: 'center',
    height: '100vh',
    backgroundColor: '#f4f4f4',
  },
  heading: {
    fontSize: '32px',
    marginBottom: '20px',
    color: '#333',
  },
  form: {
    display: 'flex',
    flexDirection: 'column',
    gap: '15px',
    width: '300px',
    padding: '20px',
    backgroundColor: '#fff',
    boxShadow: '0 4px 8px rgba(0, 0, 0, 0.1)',
    borderRadius: '8px',
  },
  input: {
    padding: '10px',
    fontSize: '16px',
    border: '1px solid #ccc',
    borderRadius: '4px',
  },
  button: {
    padding: '10px',
    fontSize: '16px',
    backgroundColor: '#4CAF50',
    color: '#fff',
    border: 'none',
    borderRadius: '4px',
    cursor: 'pointer',
  },
  message: {
    marginTop: '10px',
    color: '#4CAF50',
    fontWeight: 'bold',
  },
};
export default AddBook;
```

Outputs :



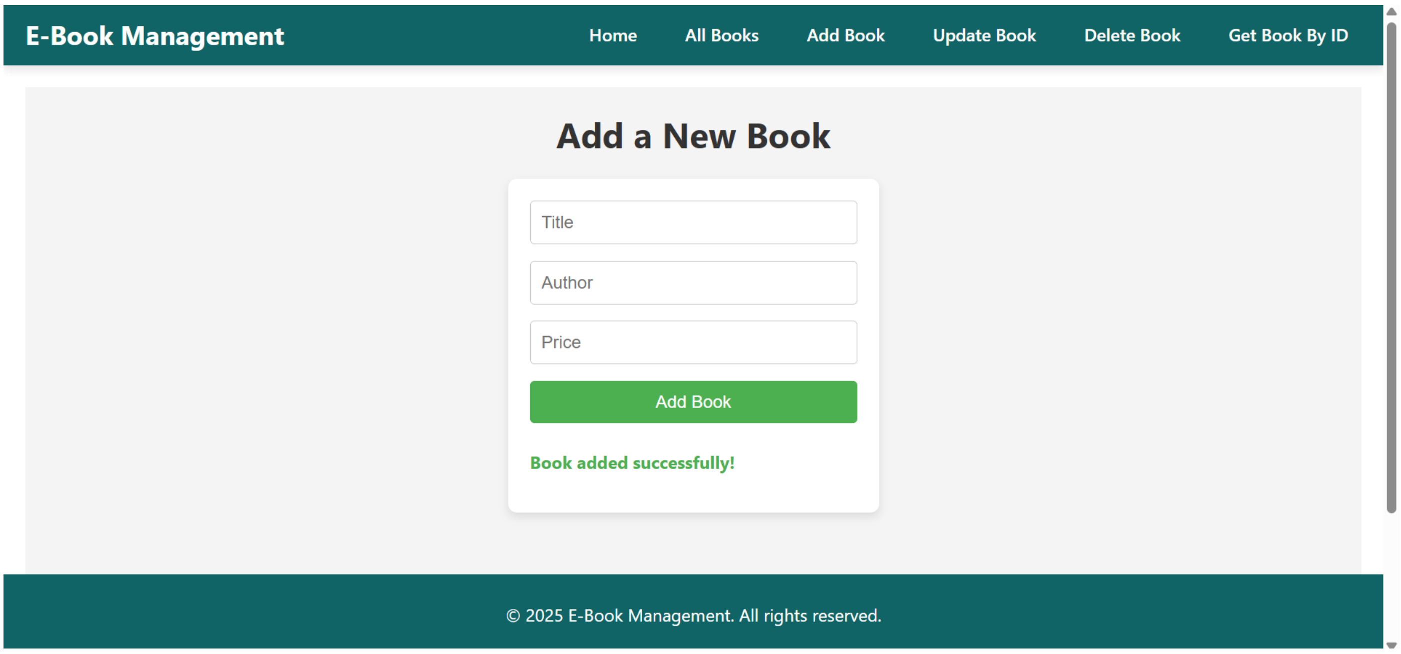
Add a New Book

Nodejs

bvrit

600.00

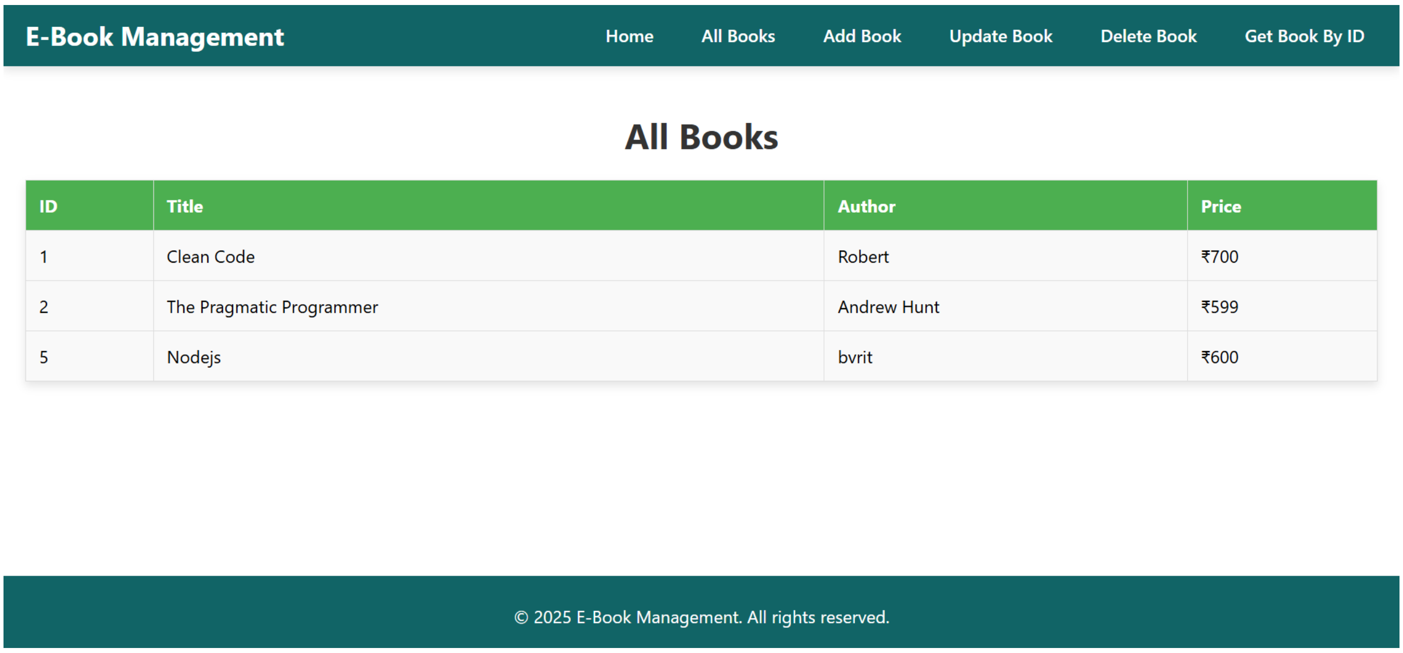
© 2025 E-Book Management. All rights reserved.



Add a New Book

Book added successfully!

© 2025 E-Book Management. All rights reserved.



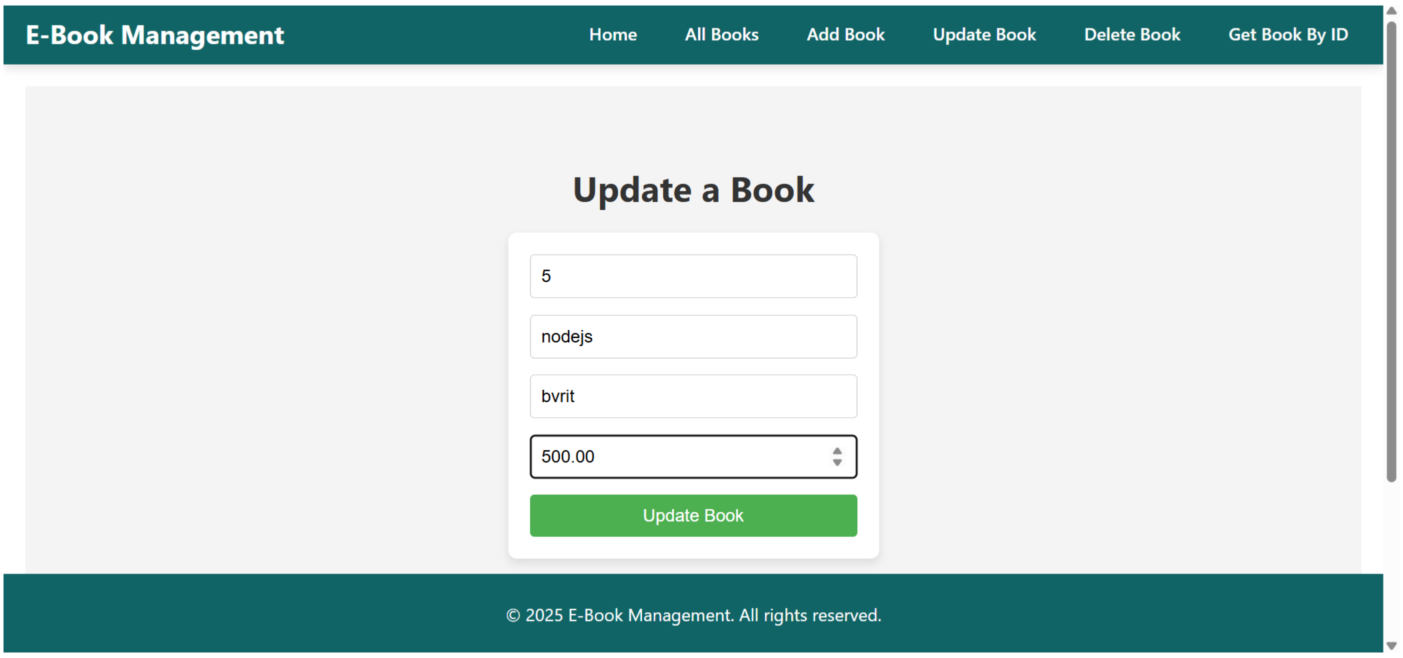
All Books

ID	Title	Author	Price
1	Clean Code	Robert	₹700
2	The Pragmatic Programmer	Andrew Hunt	₹599
5	Nodejs	bvrit	₹600

© 2025 E-Book Management. All rights reserved.

UpdateBook.js

```
import React, { useState } from 'react';
import axiosInstance from '../axiosConfig';

function UpdateBook() {
  const [bookId, setBookId] = useState('');
  const [book, setBook] = useState({ title: '', author: '', price: '' });
  const [message, setMessage] = useState('');

  const updateBook = async () => {
    try {
      await axiosInstance.put(`/updateBook/${bookId}`, book);
      setMessage('Book updated successfully!');
      setBookId('');
      setBook({ title: '', author: '', price: '' });
    } catch (err) {
      setMessage('Error updating book.');
    }
  };

  return (
    <div style={styles.container}>
      <h1 style={styles.heading}>Update a Book</h1>
      <div style={styles.form}>
        <input
          type="number"
          placeholder="Enter Book ID"
          value={bookId}
          onChange={(e) => setBookId(e.target.value)}
          style={styles.input}
        />
        <input
          type="text"
          placeholder="Title"
          value={book.title}
          onChange={(e) => setBook({ ...book, title: e.target.value })}
          style={styles.input}
        />
        <input
          type="text"
          placeholder="Author"
          value={book.author}
          onChange={(e) => setBook({ ...book, author: e.target.value })}
          style={styles.input}
        />
        <input
          type="number"
          placeholder="Price"
          value={book.price}
          onChange={(e) => setBook({ ...book, price: e.target.value })}
          style={styles.input}
        />
        <button onClick={updateBook} style={styles.button}>Update Book</button>
        {message && <p style={styles.message}>{message}</p>}
      </div>
    </div>
  );
}

const styles = {
  container: {
    display: 'flex',
    flexDirection: 'column',
    justifyContent: 'flex-start', // Aligns the content closer to the top
    alignItems: 'center', // Centers the content horizontally
    height: '100vh',
    backgroundColor: '#f4f4f4',
    paddingTop: '50px', // Adds spacing from the top
  },
  heading: {
    fontSize: '32px',
    marginBottom: '20px',
    color: '#333',
  },
  form: {
    display: 'flex',
    flexDirection: 'column',
    gap: '15px',
    width: '300px',
    padding: '20px',
    backgroundColor: '#fff',
    boxShadow: '0 4px 8px rgba(0, 0, 0, 0.1)',
    borderRadius: '8px',
  },
  input: {
    padding: '10px',
    fontSize: '16px',
    border: '1px solid #ccc',
    borderRadius: '4px',
  },
  button: {
    padding: '10px',
    fontSize: '16px',
    backgroundColor: '#4CAF50',
    color: '#fff',
    border: 'none',
    borderRadius: '4px',
    cursor: 'pointer',
  },
  message: {
    marginTop: '10px',
    color: '#4CAF50',
    fontWeight: 'bold',
  },
};

export default UpdateBook;
```



Update a Book

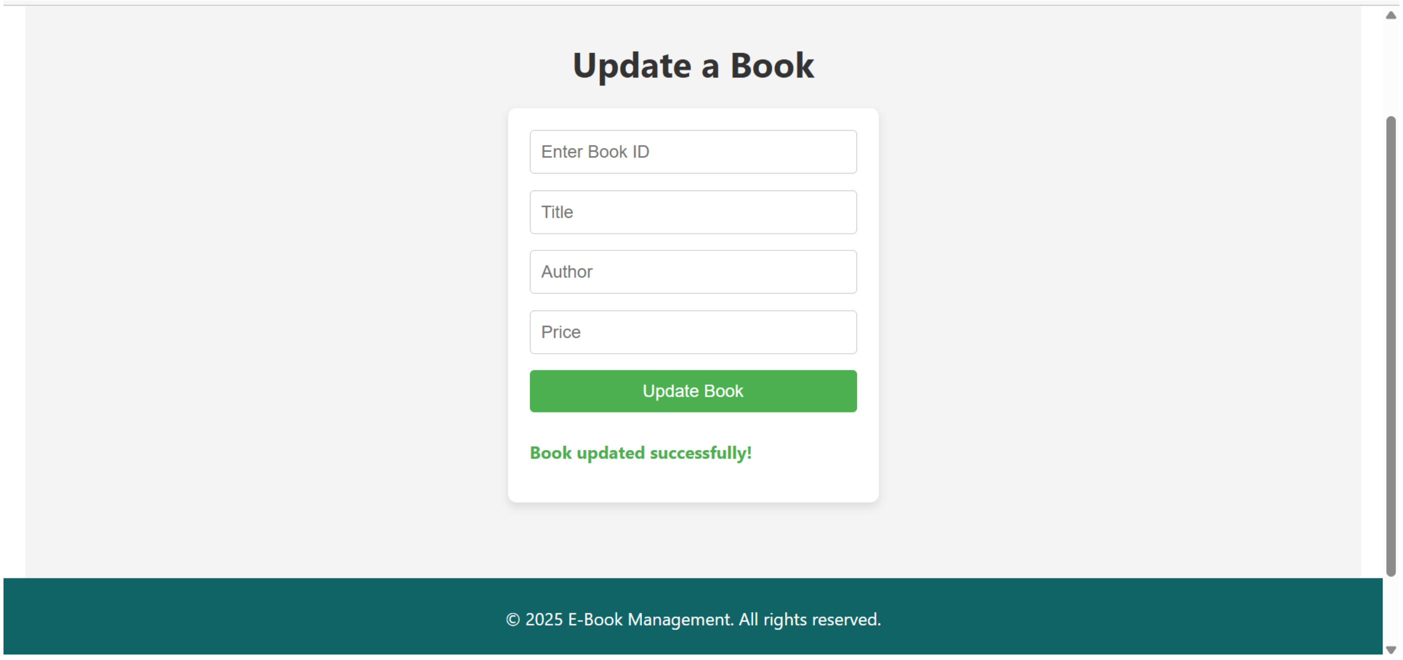
5

nodejs

bvrit

500.00

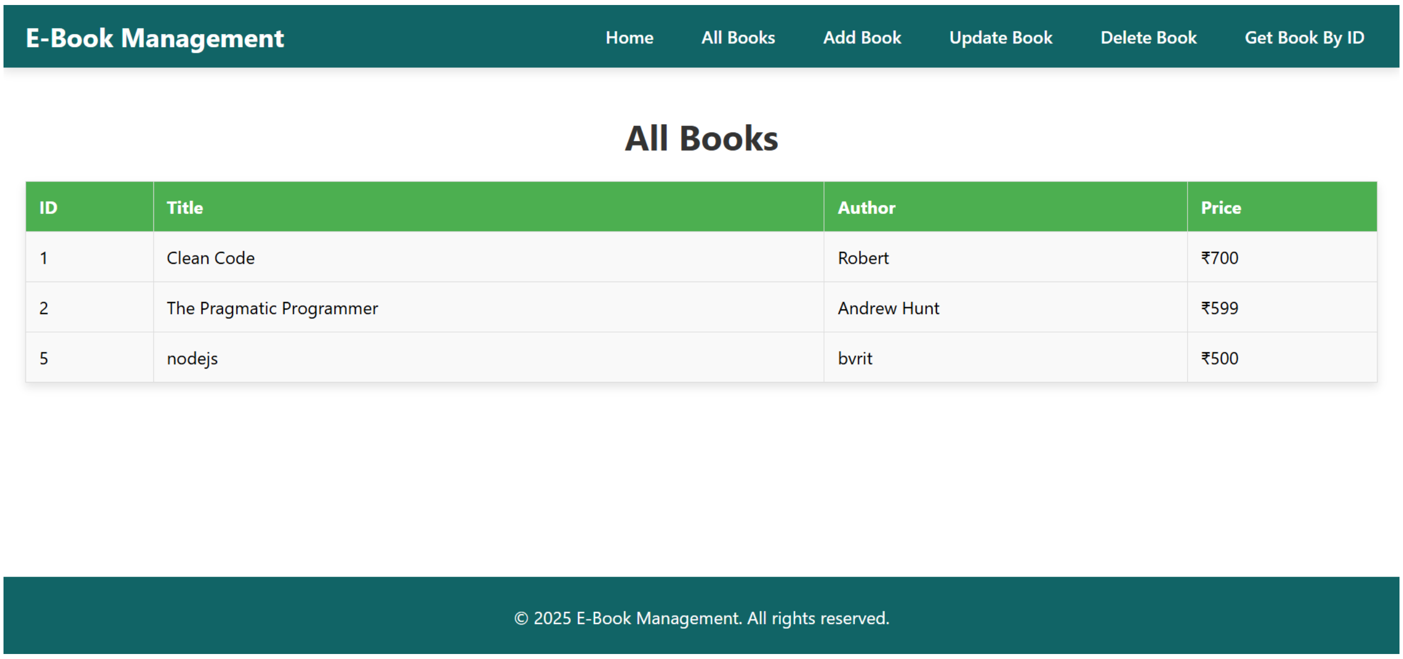
© 2025 E-Book Management. All rights reserved.



Update a Book

Book updated successfully!

© 2025 E-Book Management. All rights reserved.



All Books

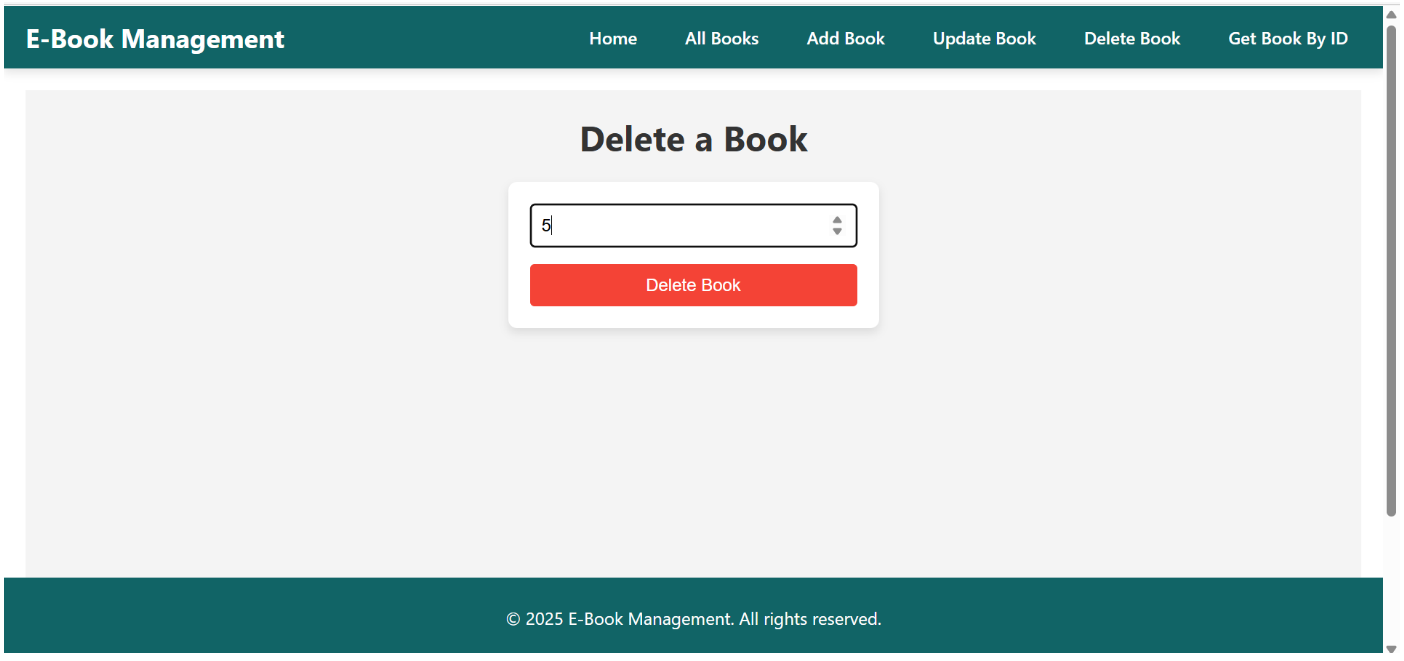
ID	Title	Author	Price
1	Clean Code	Robert	₹700
2	The Pragmatic Programmer	Andrew Hunt	₹599
5	nodejs	bvrit	₹500

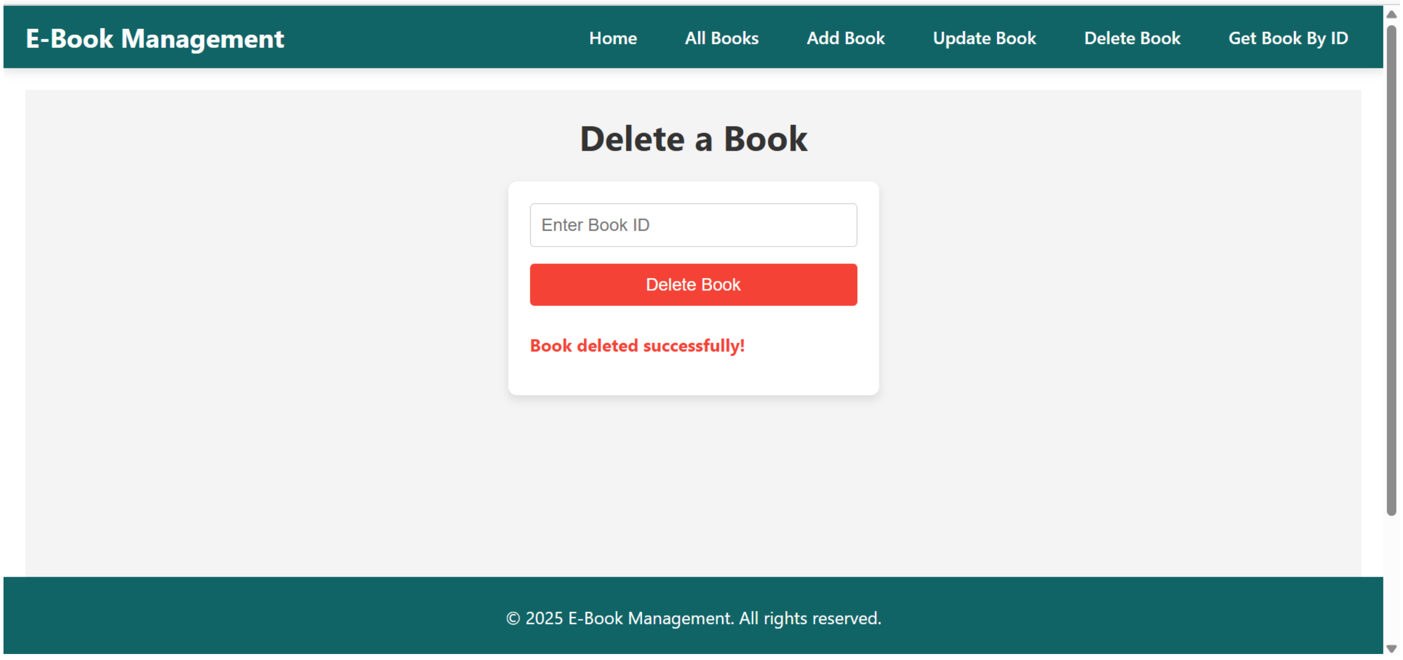
© 2025 E-Book Management. All rights reserved.

DeleteBook.js

```
import React, { useState } from 'react';
import axiosInstance from '../axiosConfig';

function DeleteBook() {
  const [bookId, setBookId] = useState('');
  const [message, setMessage] = useState('');

  const deleteBook = async () => {
    try {
      await axiosInstance.delete(`/deleteBook/${bookId}`);
      setMessage('Book deleted successfully!');
      setBookId('');
    } catch (err) {
      setMessage('Error deleting book.');
    }
  };

  return (
    <div style={styles.container}>
      <h1 style={styles.heading}>Delete a Book</h1>
      <div style={styles.form}>
        <input
          type="number"
          placeholder="Enter Book ID"
          value={bookId}
          onChange={(e) => setBookId(e.target.value)}
          style={styles.input}
        />
        <button onClick={deleteBook} style={styles.button}>Delete Book</button>
        {message && <p style={styles.message}>{message}</p>}
      </div>
    </div>
  );
}

const styles = {
  container: {
    display: 'flex',
    flexDirection: 'column',
    justifyContent: 'flex-start',
    alignItems: 'center',
    height: '100vh',
    backgroundColor: '#f4f4f4',
  },
  heading: {
    fontSize: '32px',
    marginBottom: '20px',
    color: '#333',
  },
  form: {
    display: 'flex',
    flexDirection: 'column',
    gap: '15px',
    width: '300px',
    padding: '20px',
    backgroundColor: '#fff',
    boxShadow: '0 4px 8px rgba(0, 0, 0, 0.1)',
    borderRadius: '8px',
  },
  input: {
    padding: '10px',
    fontSize: '16px',
    border: '1px solid #ccc',
    borderRadius: '4px',
  },
  button: {
    padding: '10px',
    fontSize: '16px',
    backgroundColor: '#f44336',
    color: '#fff',
    border: 'none',
    borderRadius: '4px',
    cursor: 'pointer',
  },
  message: {
    marginTop: '10px',
    color: '#f44336',
    fontWeight: 'bold',
  },
};
export default DeleteBook;
```

Outputs :

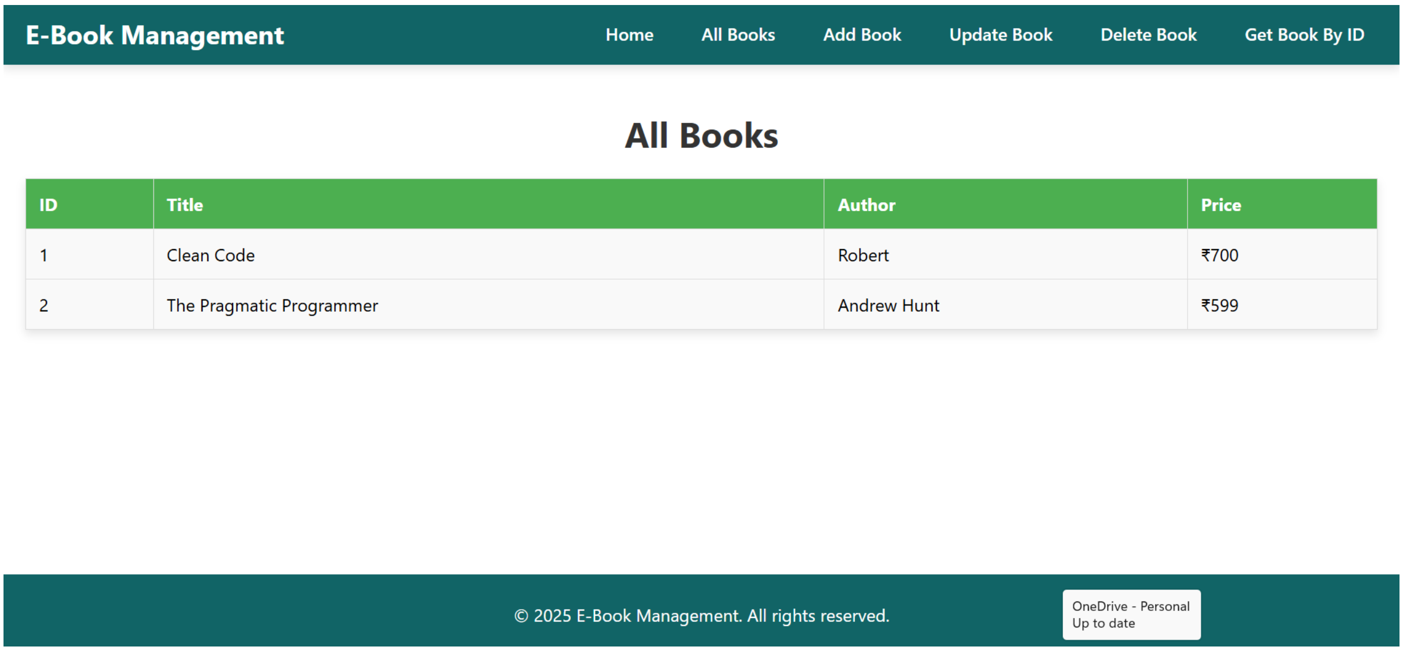




Delete a Book

Book deleted successfully!

© 2025 E-Book Management. All rights reserved.



All Books

ID	Title	Author	Price
1	Clean Code	Robert	₹700
2	The Pragmatic Programmer	Andrew Hunt	₹599

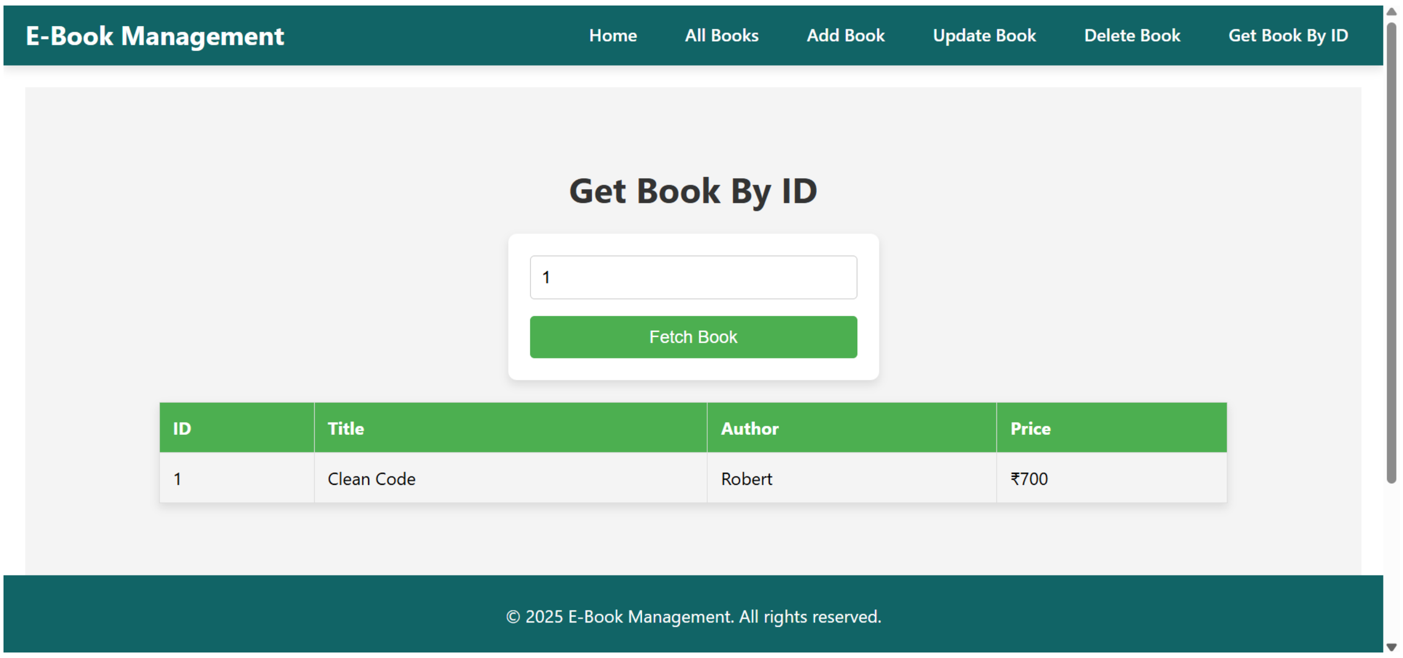
© 2025 E-Book Management. All rights reserved.

GetBookById.js

```
import React, { useState } from 'react';
import axiosInstance from '../axiosConfig';

function GetBookById() {
  const [bookId, setBookId] = useState('');
  const [book, setBook] = useState(null);
  const [error, setError] = useState('');

  const fetchBookById = async () => {
    try {
      const response = await axiosInstance.get(`/getBookById/${bookId}`);
      setBook(response.data);
      setError('');
    } catch (err) {
      setError('Book not found.');
      setBook(null);
    }
  };

  return (
    <div style={styles.container}>
      <h1 style={styles.heading}>Get Book By ID</h1>
      <div style={styles.form}>
        <input
          type="number"
          placeholder="Enter Book ID"
          value={bookId}
          onChange={(e) => setBookId(e.target.value)}
          style={styles.input}
        />
        <button onClick={fetchBookById} style={styles.button}>Fetch Book</button>
      </div>
      {error && <p style={styles.error}>{error}</p>}
      {book && (
        <table style={styles.table}>
          <thead>
            <tr>
              <th style={styles.th}>ID</th>
              <th style={styles.th}>Title</th>
              <th style={styles.th}>Author</th>
              <th style={styles.th}>Price</th>
            </tr>
          </thead>
          <tbody>
            <tr>
              <td style={styles.td}>{book.id}</td>
              <td style={styles.td}>{book.title}</td>
              <td style={styles.td}>{book.author}</td>
              <td style={styles.td}>₹{book.price}</td>
            </tr>
          </tbody>
        </table>
      )}
    </div>
  );
```

```
}

const styles = {
  container: {
    display: 'flex',
    flexDirection: 'column',
    justifyContent: 'flex-start', // Aligns the content closer to the top
    alignItems: 'center', // Centers the content horizontally
    height: '100vh',
    backgroundColor: '#f4f4f4',
    paddingTop: '50px', // Adds spacing from the top
  },
  heading: {
    fontSize: '32px',
    marginBottom: '20px',
    color: '#333',
  },
  form: {
    display: 'flex',
    flexDirection: 'column',
    gap: '15px',
    width: '300px',
    padding: '20px',
    backgroundColor: '#fff',
    boxShadow: '0 4px 8px rgba(0, 0, 0, 0.1)',
    borderRadius: '8px',
  },
  input: {
    padding: '10px',
    fontSize: '16px',
    border: '1px solid #ccc',
    borderRadius: '4px',
  },
  button: {
    padding: '10px',
    fontSize: '16px',
    backgroundColor: '#4CAF50',
    color: '#fff',
    border: 'none',
    borderRadius: '4px',
    cursor: 'pointer',
  },
  error: {
    marginTop: '10px',
    color: '#f44336',
    fontWeight: 'bold',
  },
  table: {
    marginTop: '20px',
    width: '80%',
    borderCollapse: 'collapse',
    boxShadow: '0 4px 8px rgba(0, 0, 0, 0.1)',
  },
  th: {
    border: '1px solid #ddd',
    padding: '12px',
    backgroundColor: '#4CAF50',
    color: 'white',
    textAlign: 'left',
  },
  td: {
    border: '1px solid #ddd',
    padding: '12px',
    textAlign: 'left',
  },
};

export default GetBookById;
```

Outputs :



Get Book By ID

1

ID	Title	Author	Price
1	Clean Code	Robert	₹700

© 2025 E-Book Management. All rights reserved.

App.js

```
import React from 'react';
import { BrowserRouter as Router, Route, Routes } from 'react-router-dom';
import Header from './components/Header';
import Footer from './components/Footer';
import Home from './components/Home';
import GetAllBooks from './components/GetAllBooks';
import AddBook from './components/AddBook';
import UpdateBook from './components/UpdateBook';
import DeleteBook from './components/DeleteBook';
import GetBookById from './components/GetBookById';

function App() {
  return (
    <Router>
      <div style={{ paddingBottom: '50px' }}>
        <Header />
        <main style={styles.main}>
          <Routes>
            <Route path="/" element={<Home />} />
            <Route path="/allBooks" element={<GetAllBooks />} />
            <Route path="/addBook" element={<AddBook />} />
            <Route path="/updateBook" element={<UpdateBook />} />
            <Route path="/deleteBook" element={<DeleteBook />} />
            <Route path="/getBookById" element={<GetBookById />} />
          </Routes>
        </main>
        <Footer />
      </div>
    </Router>
  );
}

const styles = {
  main: {
    padding: '20px',
  },
};

export default App;
```