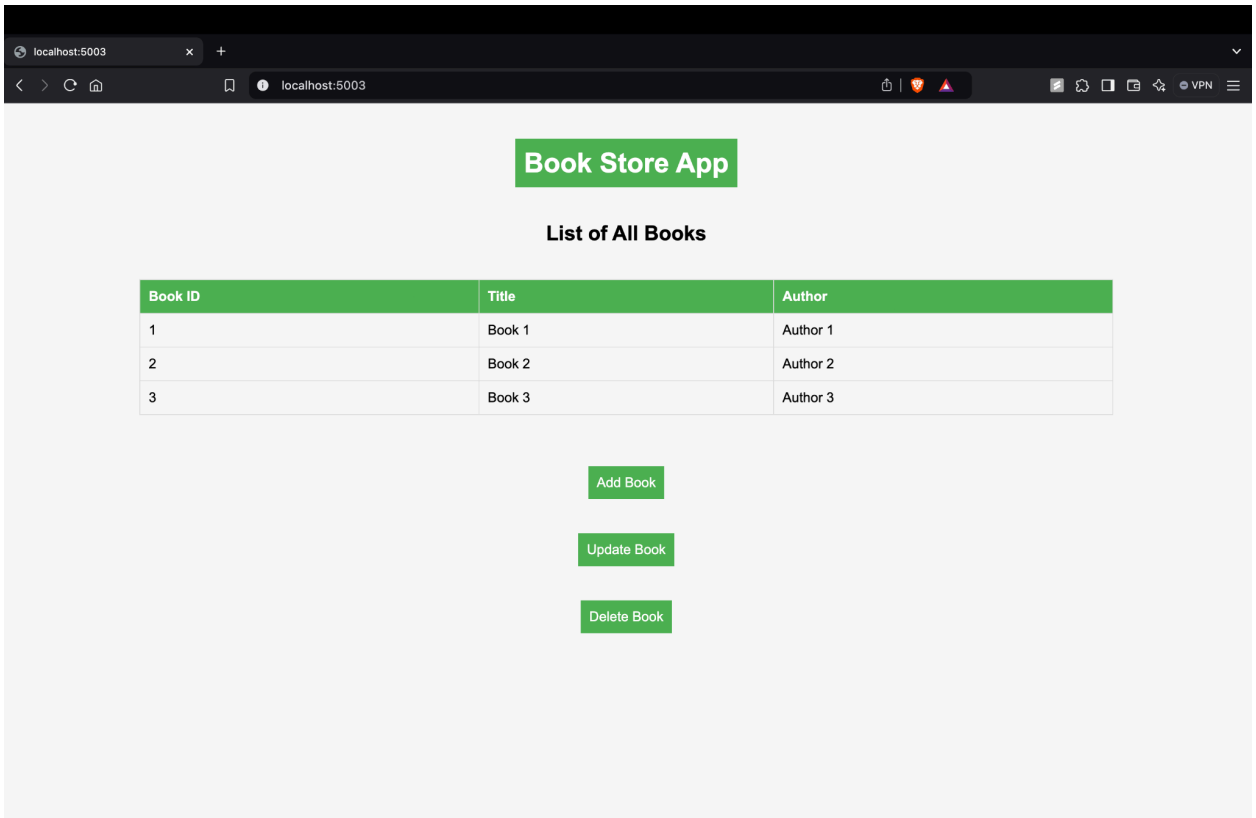
DATA-236 Sec 12 - Distributed Systems for Data Engineering
HOMEWORK 2
Nandhakumar Apparsamy
018190003

GitHub link for full code artifacts -
https://github.com/Nandha951/DATA-236-HW-2-CRUD-using-HTML-CSS-JS

Instructions:
- Please provide the screenshots of code solution for each question along with its intended output. Ensure that the code and corresponding output screenshots are placed together, one below the other.
- Submission should be in PDF Format.
- Please name your submission file as {last_name}_HW2.pdf

Download the hw2_template.zip file from canvas.

Part 1. HTML & CSS (4 points)
1. Center align all the content present on the home page (i.e the headings, text, buttons, table, etc. everything should be centered). (1 point)

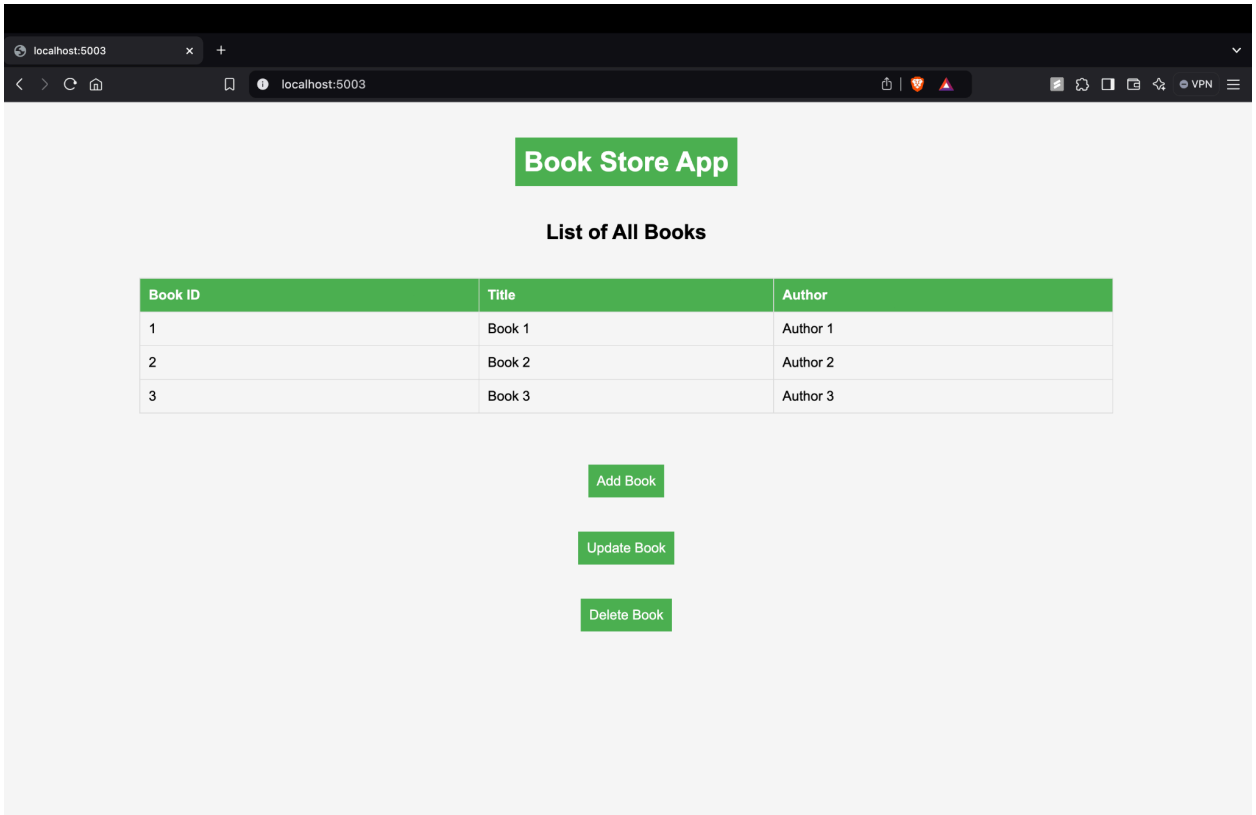


JS index.js | <> home.ejs | <> create.ejs

public > css > # styles.css > ...

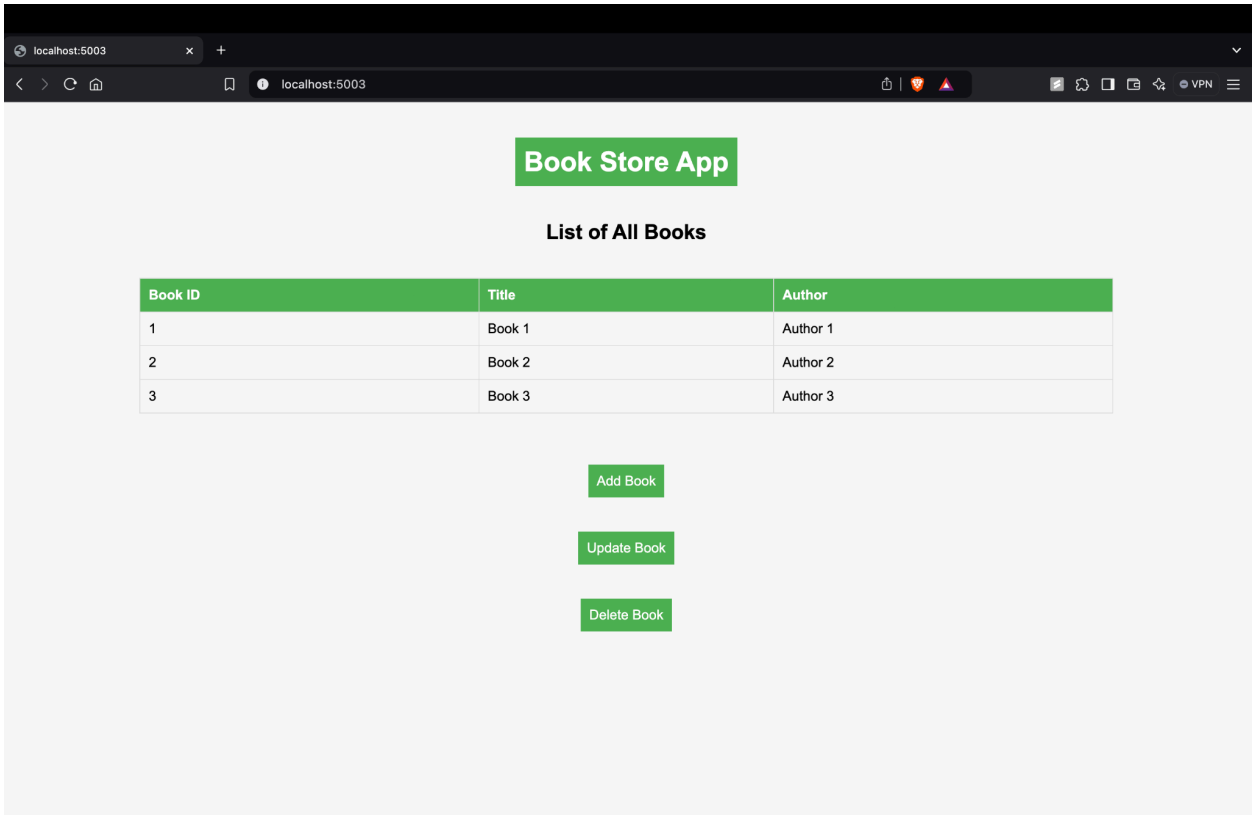
```
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 20px;
    background-color: #f9f9f9;
    display: flex;
    flex-direction: column;
    align-items: center;
    text-align: center;
}
```

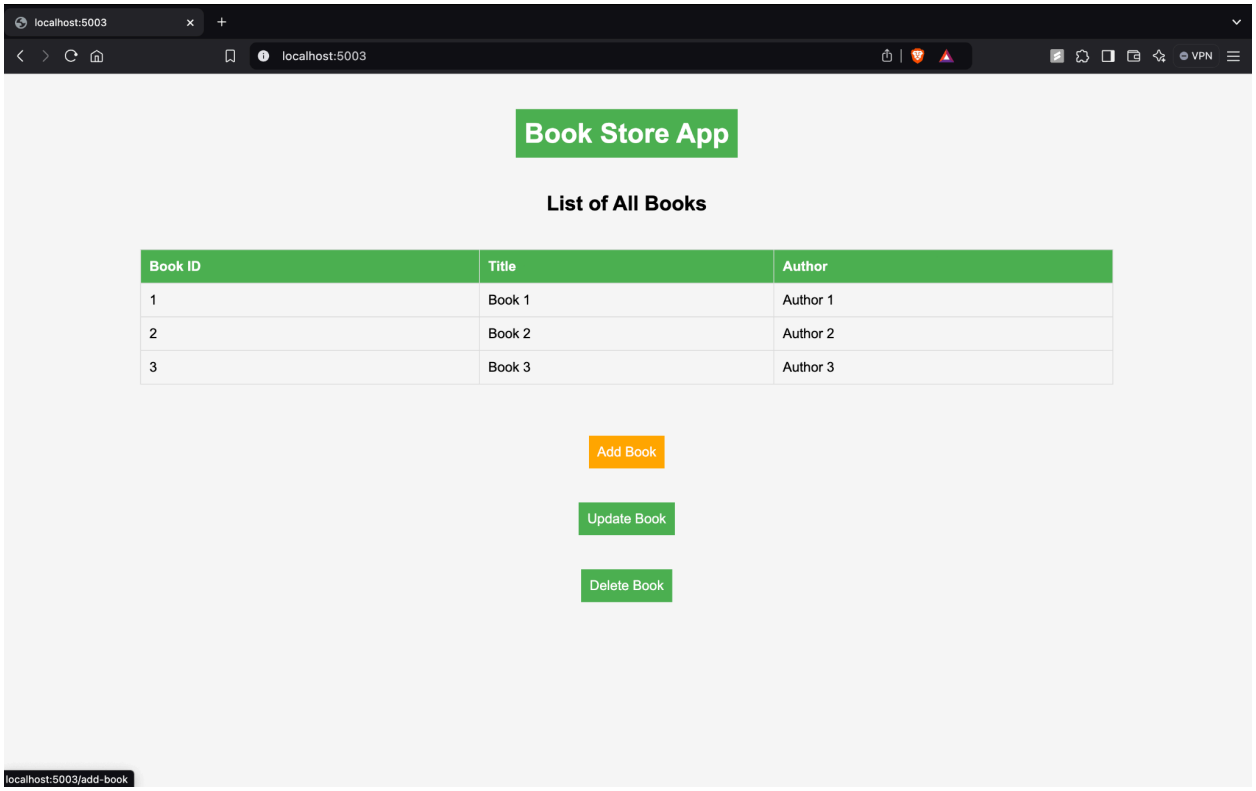
2. Make the background color of the heading, table header, and buttons to ##4CAF50 and set the text color to white for the heading, table header and buttons. (1 point)

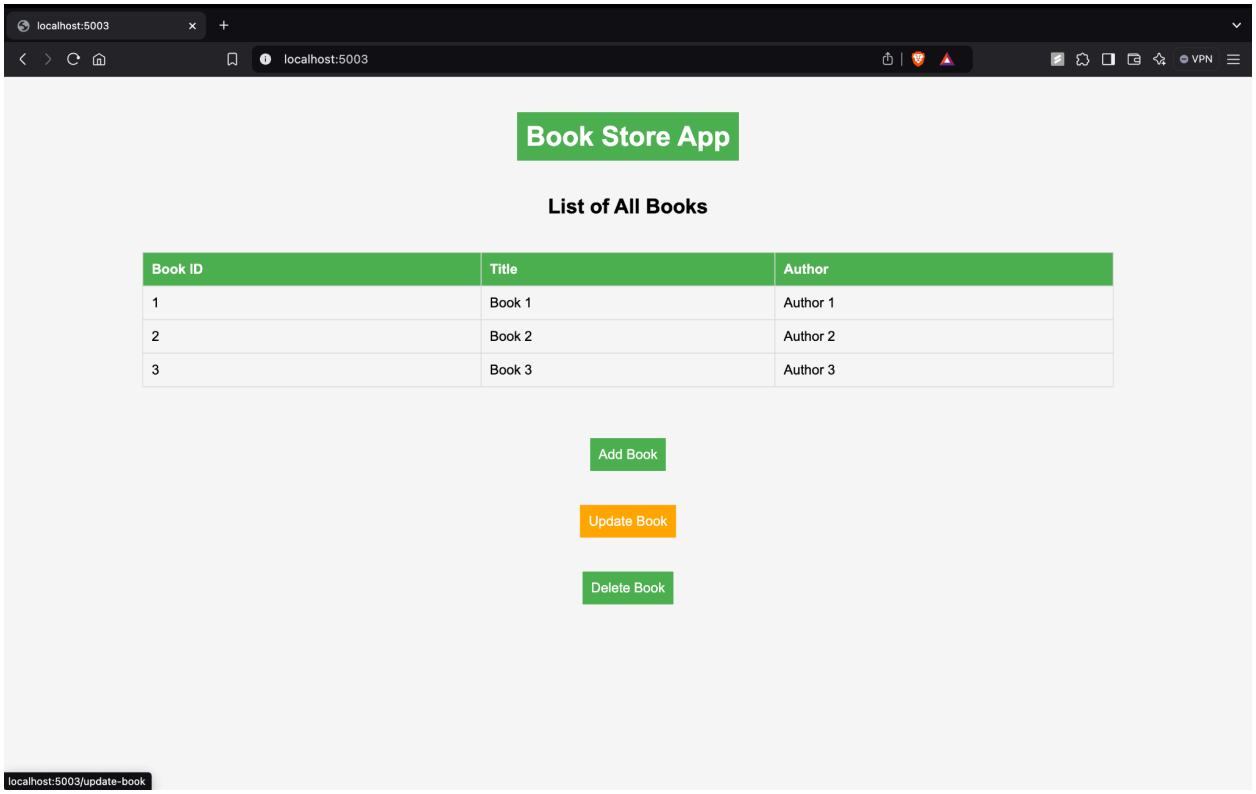


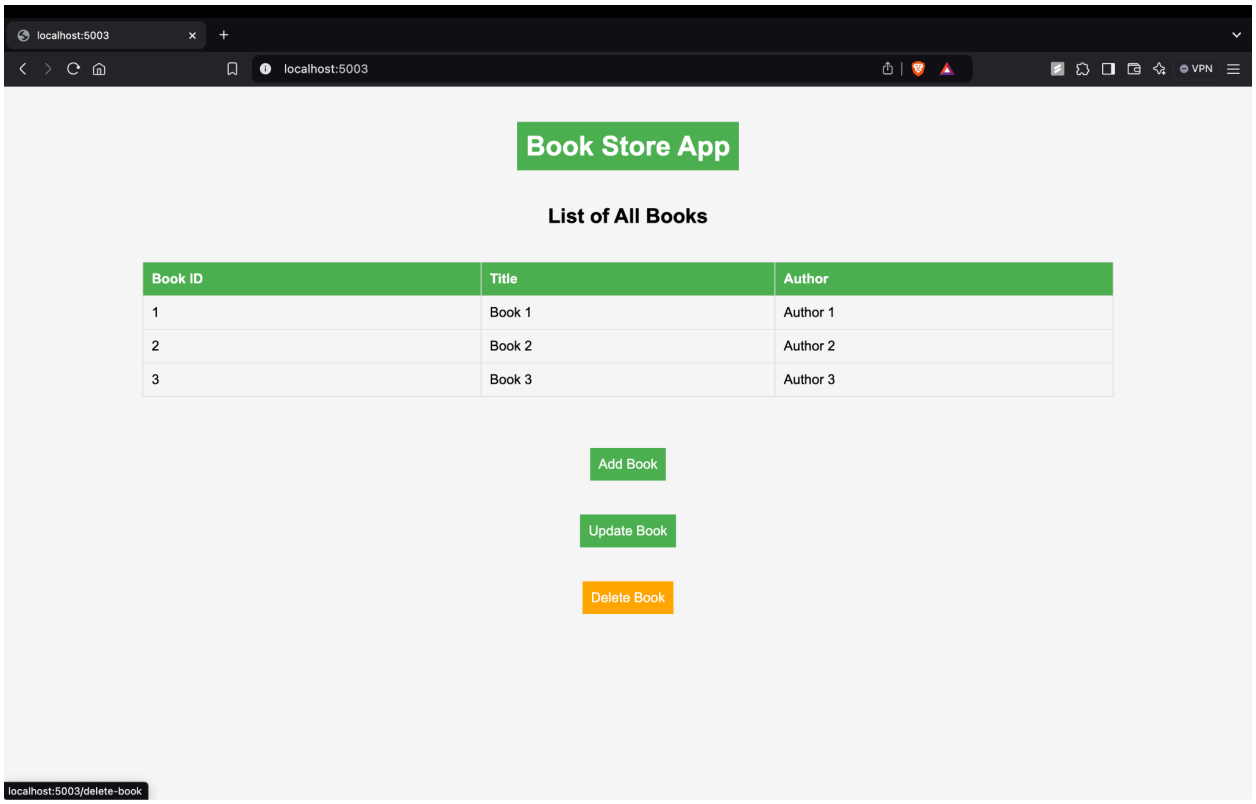

```
11
12  h1.heading,.books-table th, button {
13      background-color: #4CAF50;
14      color: white;
15  }
```

3. Remove any border from the action buttons and change the background of the button to orange on the button's hover, and set the cursor to pointer on hover of the buttons. (1 point)


localhost:5003
localhost:5003
Book Store App
List of All Books
Book ID
Title
Author
1
Book 1
Author 1
2
Book 2
Author 2
3
Book 3
Author 3
Add Book
Update Book
Delete Book

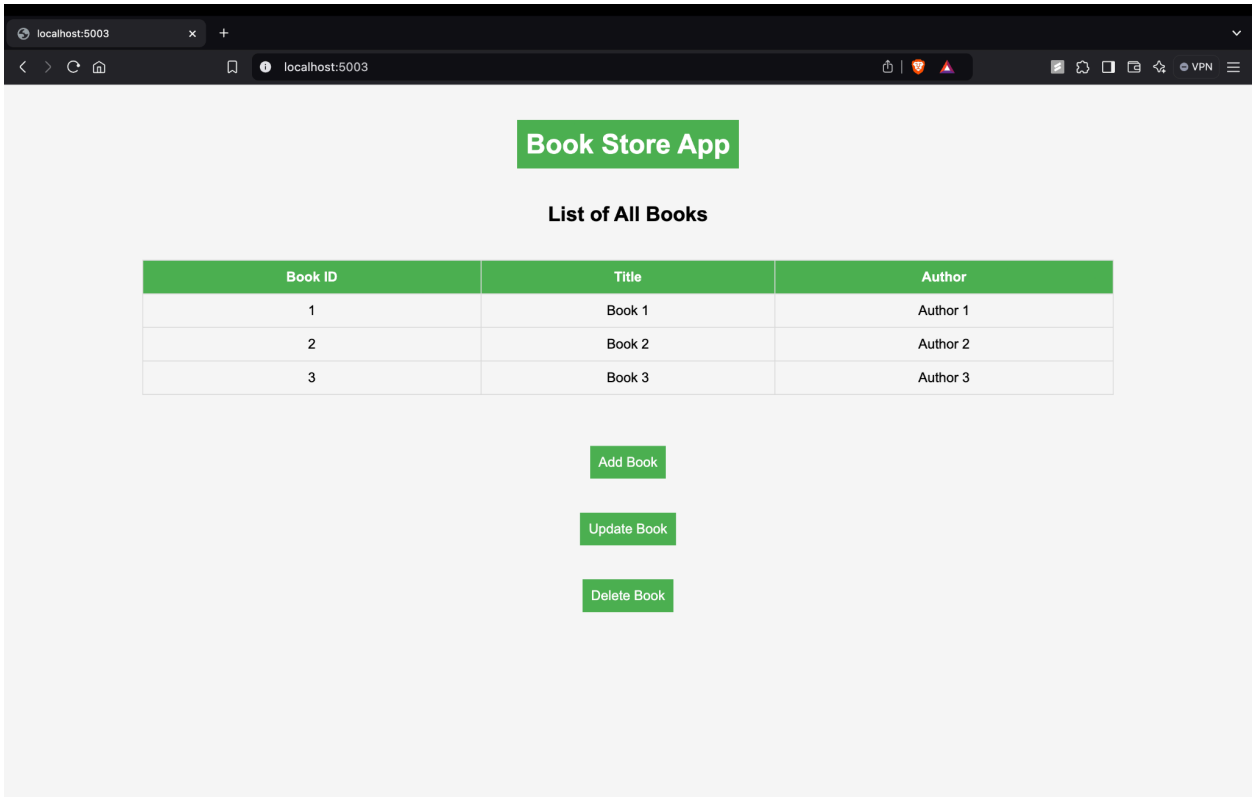

localhost:5003
localhost:5003
Book Store App
List of All Books
Book ID
Title
Author
1
Book 1
Author 1
2
Book 2
Author 2
3
Book 3
Author 3
Add Book
Update Book
Delete Book
localhost:5003/add-book


localhost:5003
localhost:5003
Book Store App
List of All Books
Book ID
Title
Author
1
Book 1
Author 1
2
Book 2
Author 2
3
Book 3
Author 3
Add Book
Update Book
Delete Book
localhost:5003/update-book


localhost:5003
localhost:5003
Book Store App
List of All Books
Book ID
Title
Author
1
Book 1
Author 1
2
Book 2
Author 2
3
Book 3
Author 3
Add Book
Update Book
Delete Book
localhost:5003/delete-book

```
21  button {
22      margin: 10px;
23      padding: 10px;
24      font-size: 16px;
25      border: none;
26  }
27
28  button:hover {
29      background-color: orange;
30      cursor: pointer;
31  }
```

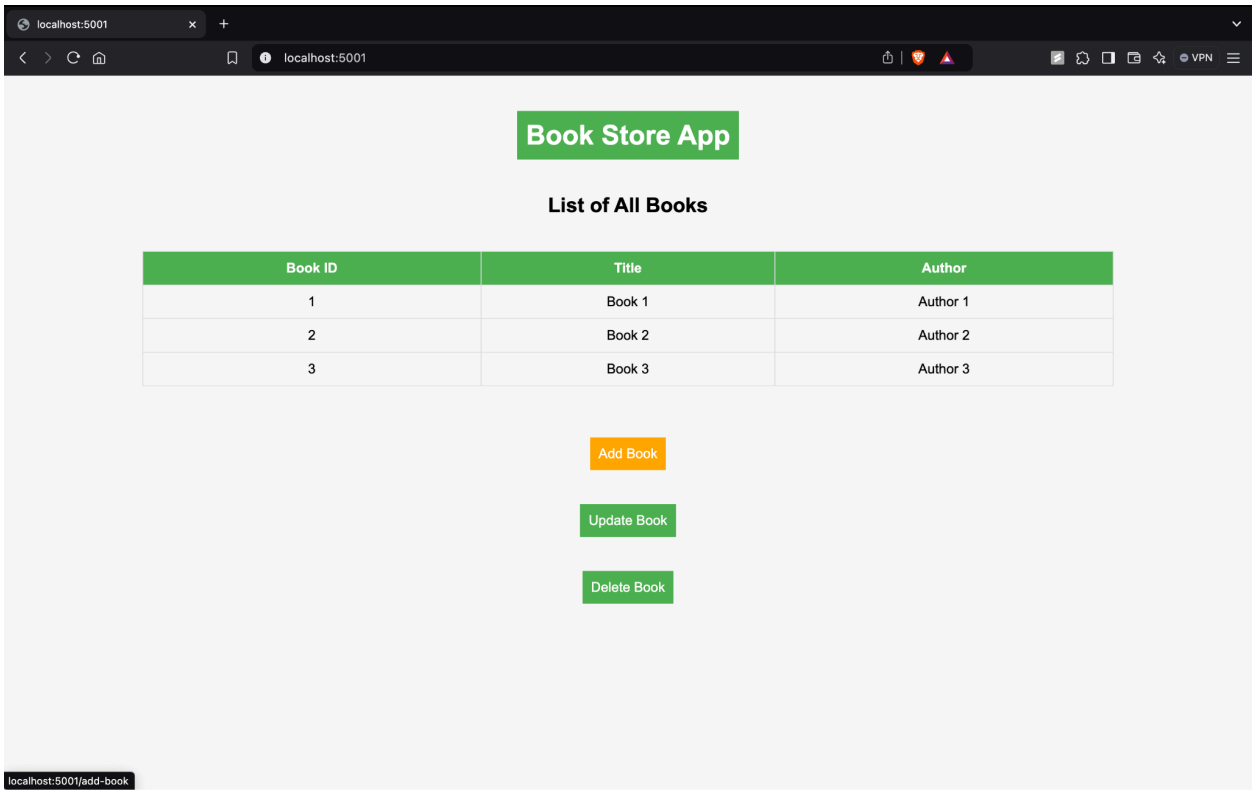
4. Center align the table text of the table. (1 point)

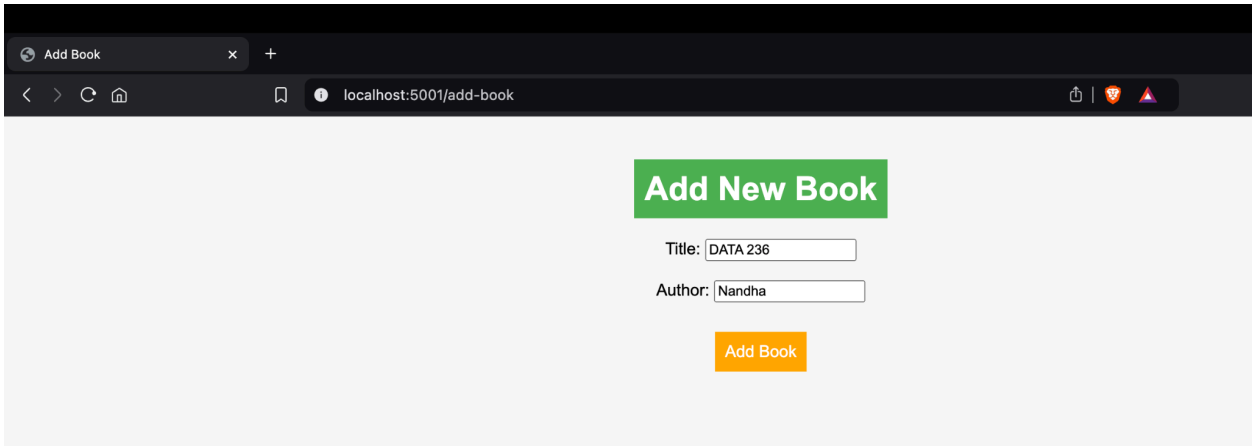


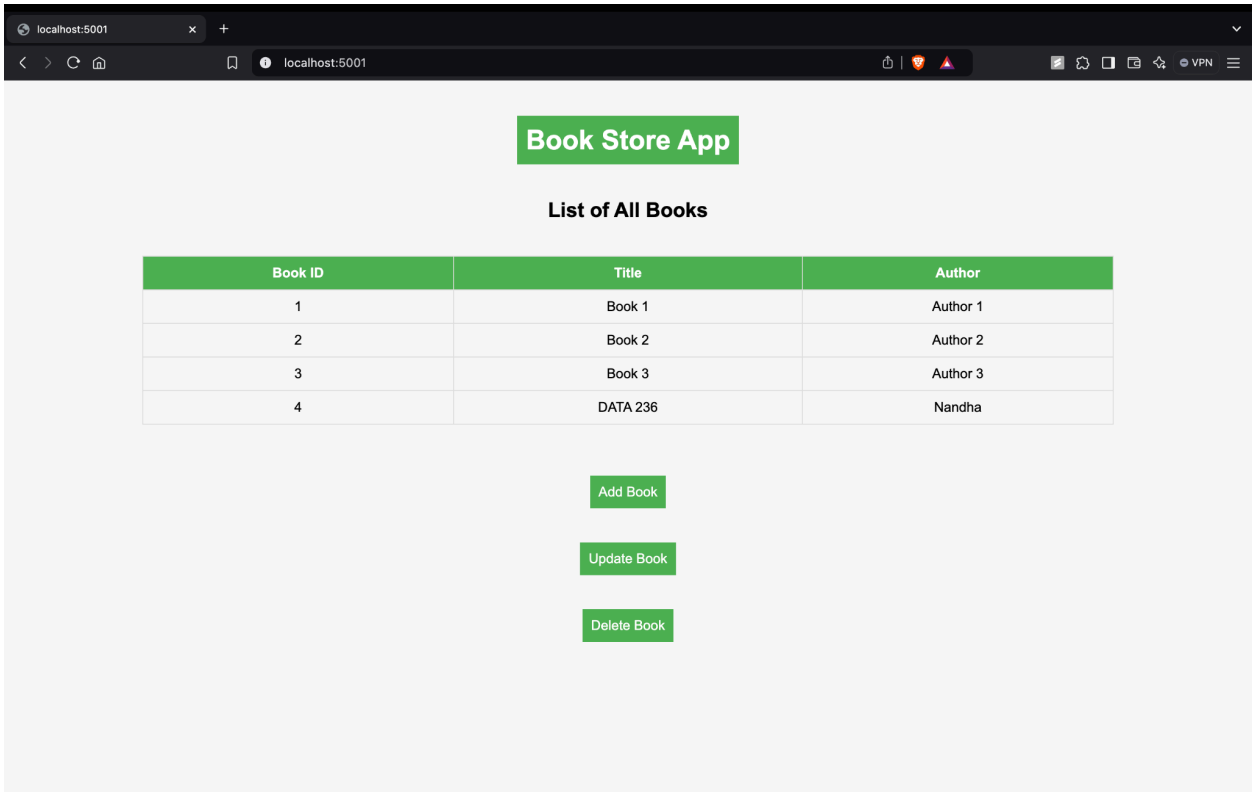
```
51  .books-table {
52      width: 80%;
53      margin: 20px auto;
54      border-collapse: collapse;
55      text-align: center;
56  }
57
```

Part 2. HTTP, Express, NodeJS (6 points)

1. Write the code to add a new book. The user should be able to enter the Book Title and Author Name. Once the user submits the required data, the book should be added and the user should be redirected to the home view showing the updated list of books. (2 points)






localhost:5001
localhost:5001
VPN
Book Store App
List of All Books
Book ID
Title
Author
1
Book 1
Author 1
2
Book 2
Author 2
3
Book 3
Author 3
4
DATA 236
Nandha
Add Book
Update Book
Delete Book

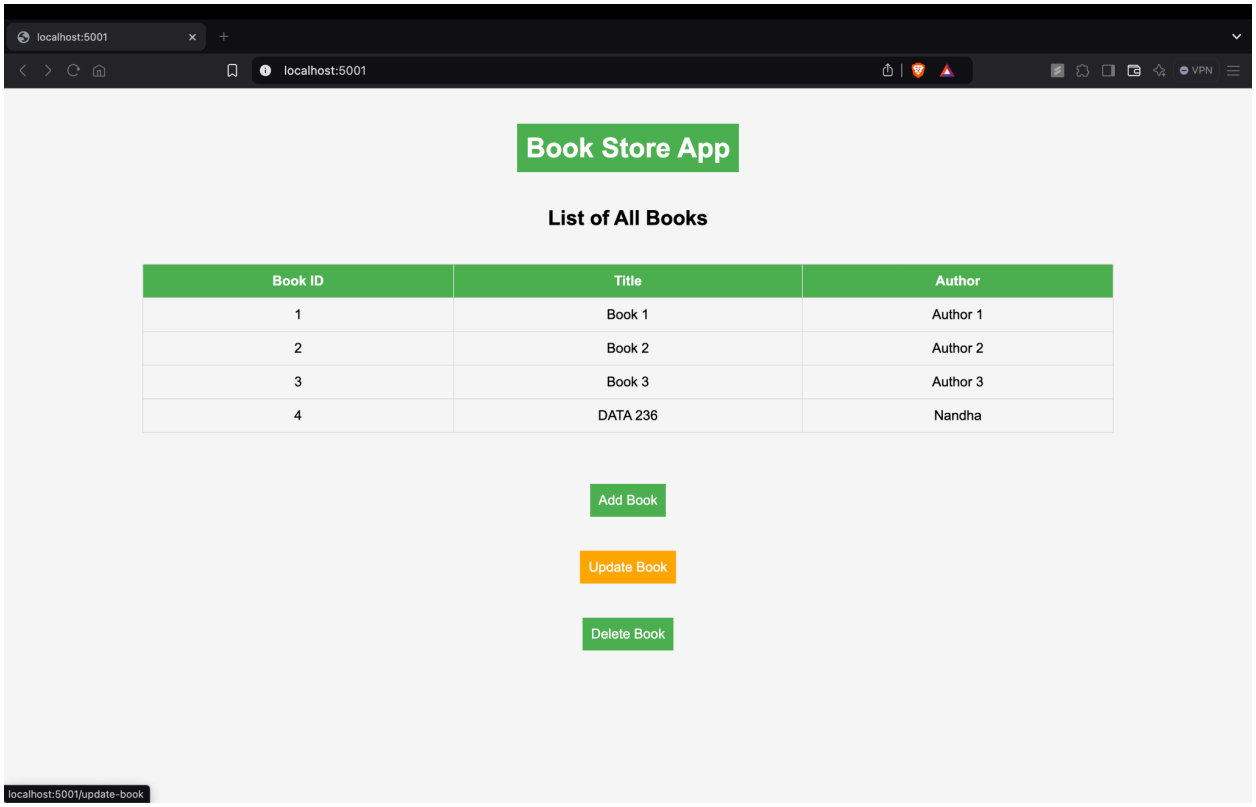
index.js M | .gitignore | home.ejs | create.ejs M

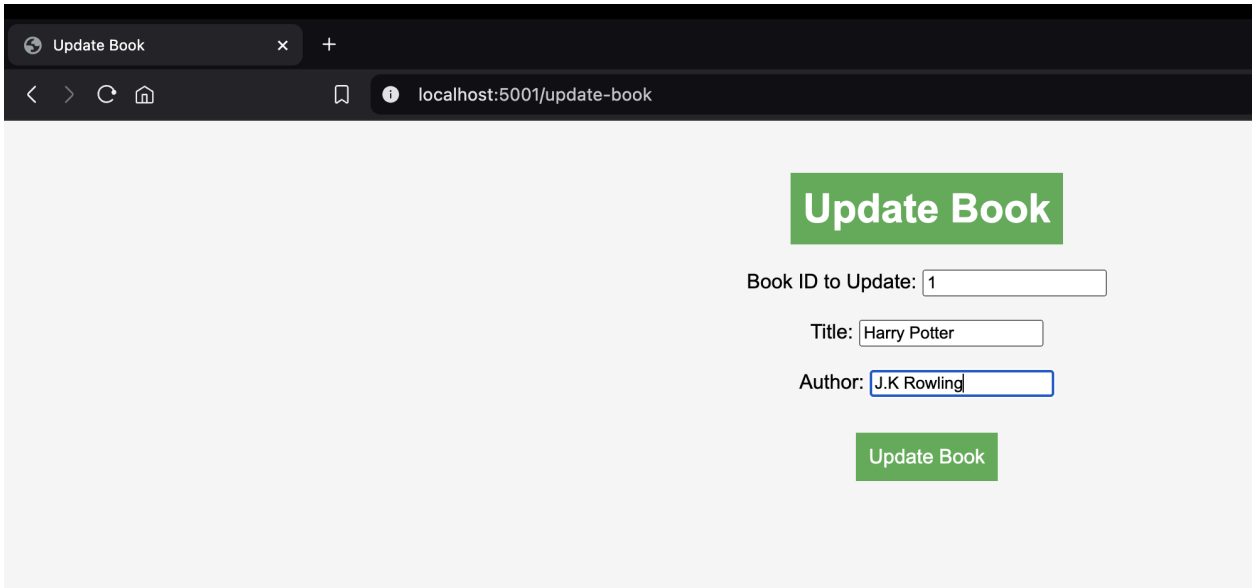
index.js > app.post('/update-book') callback

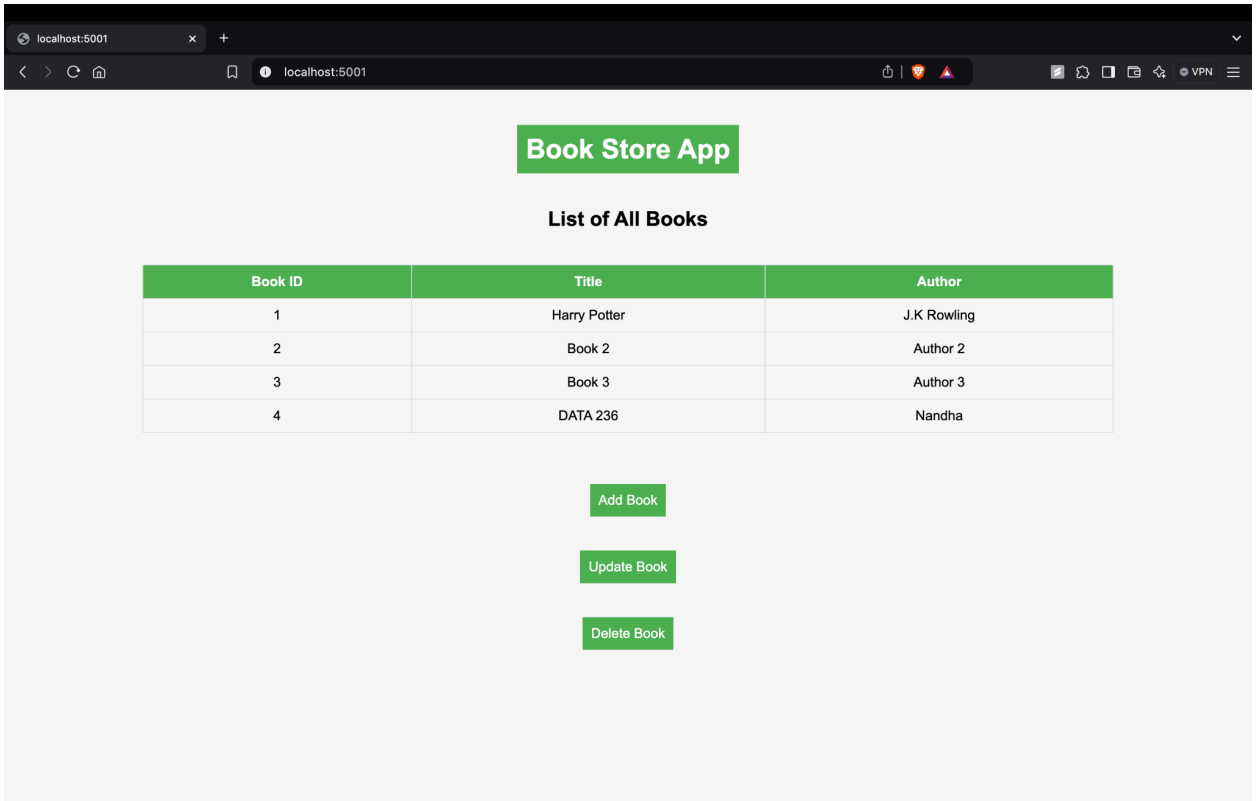
```javascript
const express = require('express');
const app = express();
const bodyParser = require('body-parser');

app.set('view engine', 'ejs');
app.set('views', './views');
app.use(express.static(__dirname + '/public'));

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));

let books = [
    { "BookID": "1", "Title": "Book 1", "Author": "Author 1" },
    { "BookID": "2", "Title": "Book 2", "Author": "Author 2" },
    { "BookID": "3", "Title": "Book 3", "Author": "Author 3" }
];

app.get('/', function (req, res) {
    res.render('home', {
        books: books
    });
});

// Add Book
app.get('/add-book', function (req, res) {
    res.render('create');
});

app.post('/add-book', function (req, res) {
    const newBook = {
        "BookID": (books.length + 1).toString(),
        "Title": req.body.title,
        "Author": req.body.author
    };
    books.push(newBook);
    res.redirect('/');
});


app.get('/update-book', function (req, res) {
    res.render('update-book', { book: null });
});
```

```
43
44  app.post('/update-book', function (req, res) {
45      const bookIdToUpdate = String(req.body.bookId);
46
47      const bookToUpdate = books.find(book => book.BookID === bookIdToUpdate);
48      if (!bookToUpdate) {
49          return res.send("Book not found");
50      }
51
52      const updatedBook = {
53          "BookID": bookIdToUpdate,
54          "Title": req.body.title,
55          "Author": req.body.author
56      };
57
58      books = books.map(book =>
59          book.BookID === bookIdToUpdate ? updatedBook : book
60      );
61
62      res.redirect('/');
63  });
64
65  // Delete Book
66  app.get('/delete-book', function (req, res) {
67      res.render('delete');
68  });
69
70  app.post('/delete-book', function (req, res) {
71      const maxId = Math.max(...books.map(book => parseInt(book.BookID, 10)));
72      books = books.filter(book => parseInt(book.BookID, 10) !== maxId);
73      res.redirect('/');
74  });
75
76
77  app.listen(5001, function () {
78      console.log("Server listening on port 5001");
79  });
```

2. Write the code to update book with id 1 to title:”Harry Potter”, Author Name: “J.K Rowling”. After submitting the data, redirect to the home view and show the updated data in the list of books. (2 points)







Book ID	Title	Author
1	Harry Potter	J.K Rowling
2	Book 2	Author 2
3	Book 3	Author 3
4	DATA 236	Nandha

index.js M | .gitignore | home.ejs | create.ejs M | update-book.ej

index.js > app.post('/update-book') callback

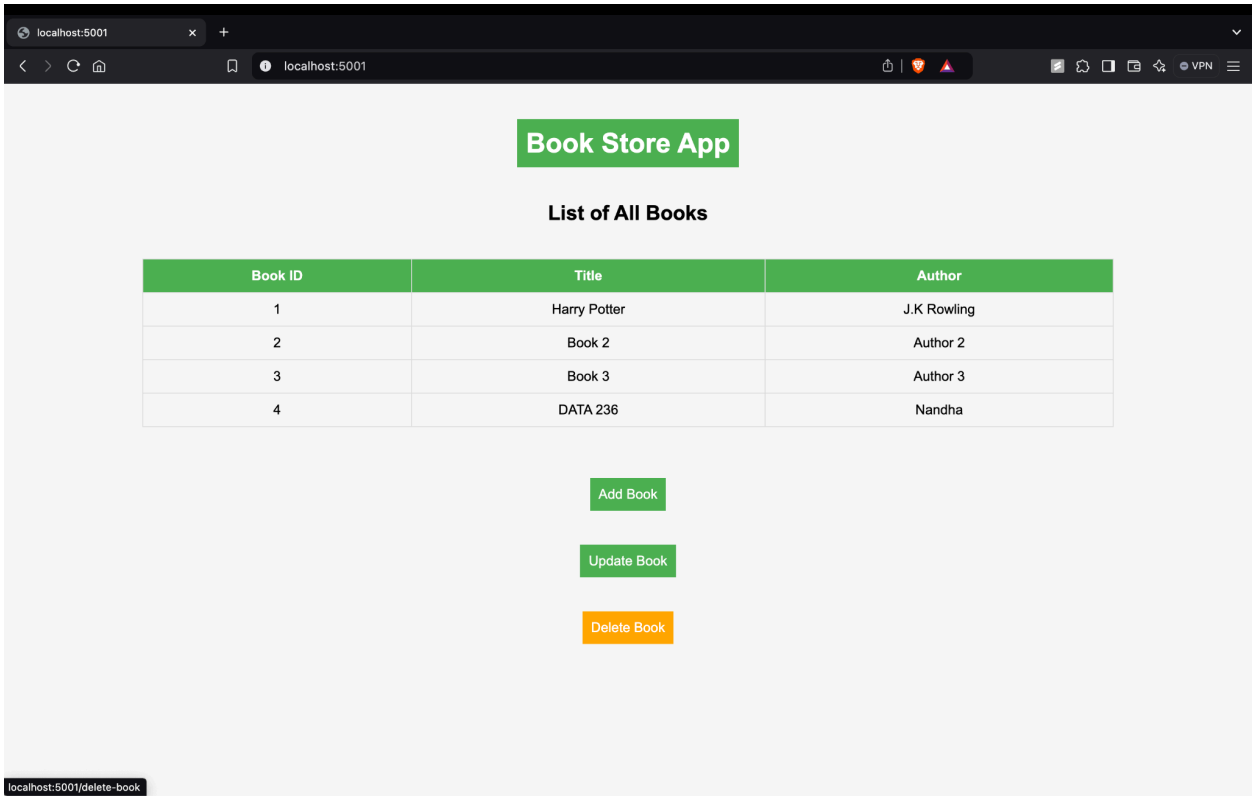
```javascript

app.get('/update-book', function (req, res) {
    res.render('update-book', { book: null });
});

app.post('/update-book', function (req, res) {
    const bookIdToUpdate = String(req.body.bookId);

    const bookToUpdate = books.find(book => book.BookID === bookIdToUpdate);
    if (!bookToUpdate) {
        return res.send("Book not found");
    }

    const updatedBook = {
        "BookID": bookIdToUpdate,
        "Title": req.body.title,
        "Author": req.body.author
    };

    books = books.map(book =>
        book.BookID === bookIdToUpdate ? updatedBook : book
    );

    res.redirect('/');
});
```

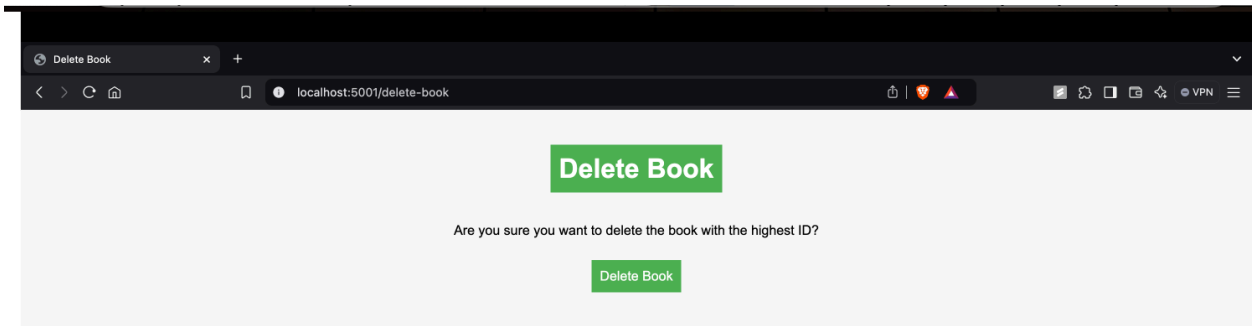
index.js M | .gitignore | home.ejs | create.ejs M | update-book.ejs U

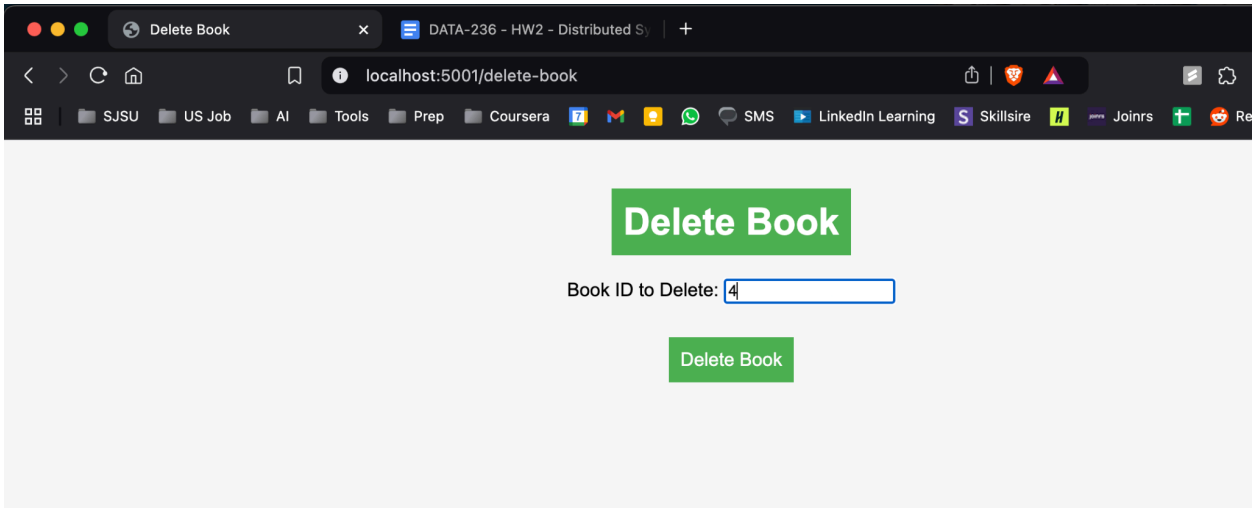
views > update-book.ejs > html > body > form > br

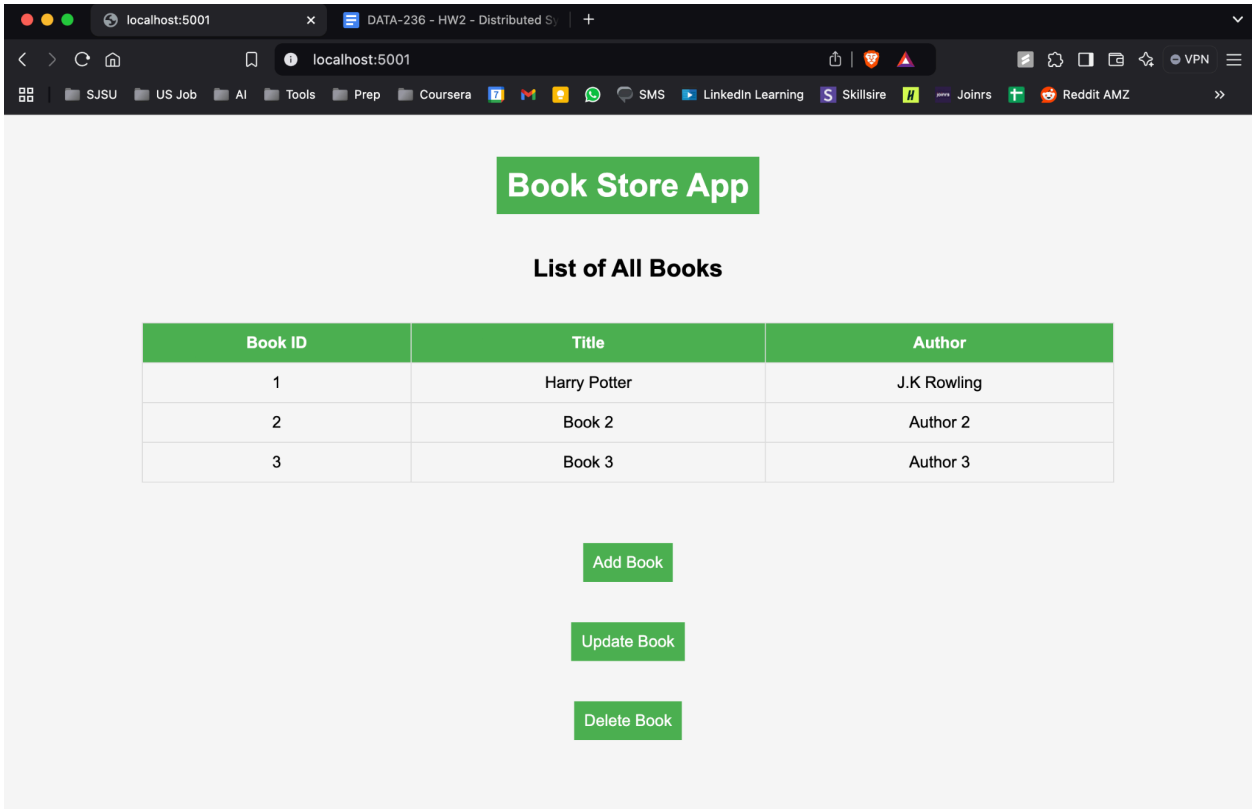
```html
<!DOCTYPE html>
<html>
<head>
    <title>Update Book</title>
    <link rel="stylesheet" href="/css/styles.css">
</head>
<body>
    <div class="container">
        <h1 class="heading">Update Book</h1>
    </div>
    <form action="/update-book" method="post">
        <label for="bookId">Book ID to Update:</label>
        <input type="text" id="bookId" name="bookId" required><br><br>
        <label for="title">Title:</label>
        <input type="text" id="title" name="title" required><br><br>
        <label for="author">Author:</label>
        <input type="text" id="author" name="author" required><br><br>
        <button type="submit">Update Book</button>
    </form>
</body>
</html>
```

3. Write the code to delete the book with the highest id. After submitting the data, redirect to the home view and show the updated data in the list of books. (2 points)


localhost:5001
Book Store App
List of All Books
Book ID | Title | Author
1 | Harry Potter | J.K Rowling
2 | Book 2 | Author 2
3 | Book 3 | Author 3
4 | DATA 236 | Nandha
Add Book
Update Book
Delete Book
localhost:5001/delete-book


Delete Book
localhost:5001/delete-book
Delete Book
Are you sure you want to delete the book with the highest ID?
Delete Book


Delete Book
DATA-236 - HW2 - Distributed Sy
localhost:5001/delete-book
SJSU
US Job
AI
Tools
Prep
Coursera
SMS
LinkedIn Learning
Skillsire
Joinrs
Delete Book
Book ID to Delete: 4
Delete Book




```
app.get('/delete-book', function (req, res) {
    res.render('delete');
});

app.post('/delete-book', function (req, res) {
    const bookIdToDelete = req.body.bookId;

    if (!bookIdToDelete) {
        return res.send("Book ID is required");
    }

    books = books.filter(book => book.BookID !== bookIdToDelete);

    res.redirect('/');
});

```

delete.ejs M × index.js M

views > delete.ejs > html

```html
<!-- Add your code here -->
<!DOCTYPE html>
<html>
<head>
    <title>Delete Book</title>
    <link rel="stylesheet" href="/css/styles.css">
</head>
<body>
    <div class="container">
        <h1 class="heading">Delete Book</h1>
    </div>
    <form action="/delete-book" method="post">
        <label for="bookId">Book ID to Delete:</label>
        <input type="text" id="bookId" name="bookId" required><br><br>
        <button type="submit">Delete Book</button>
    </form>
</body>
</html>
```