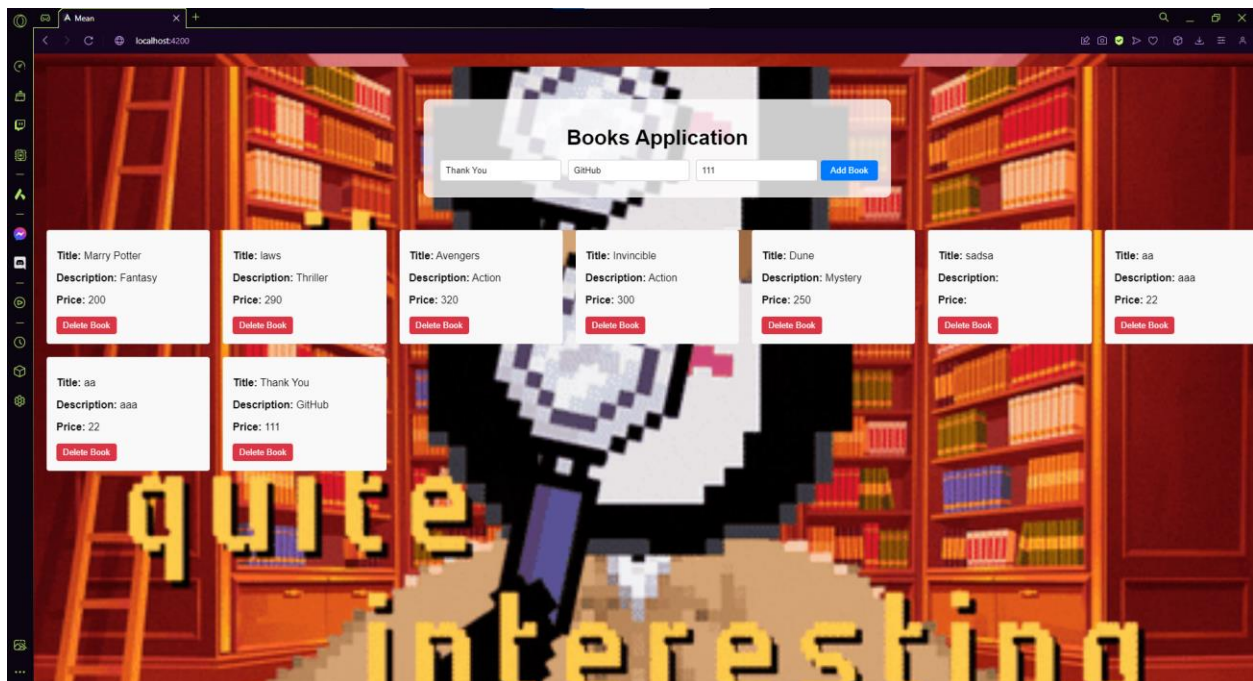
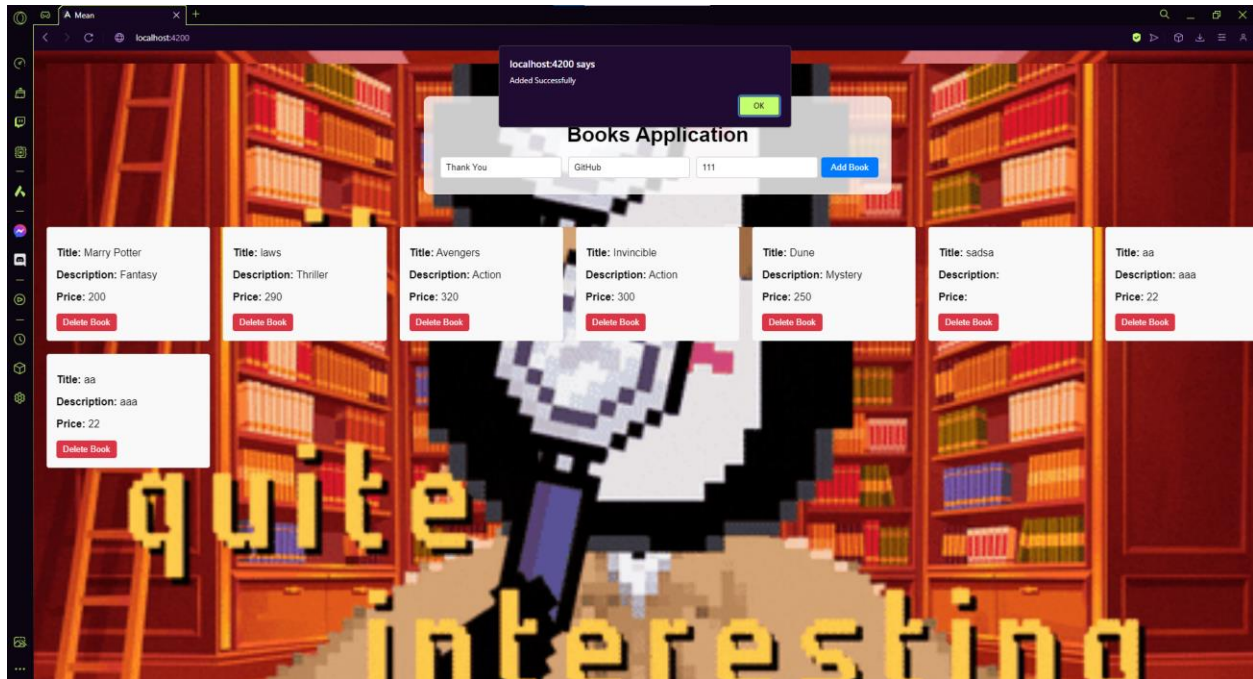
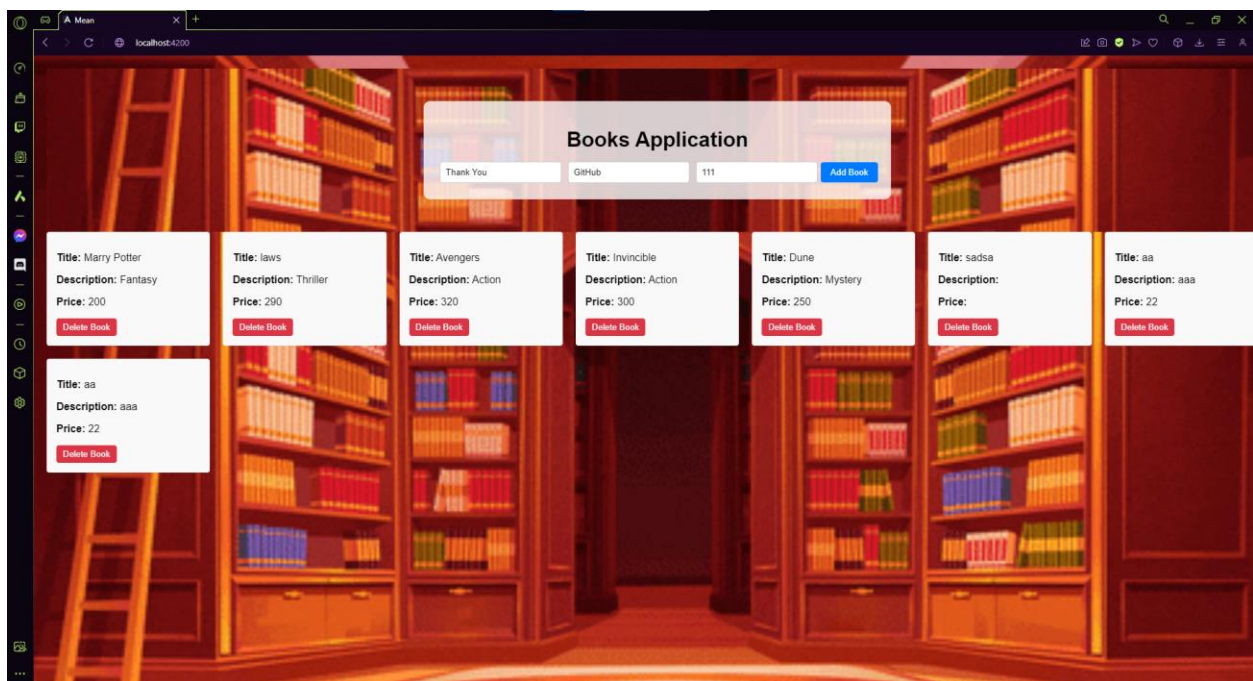
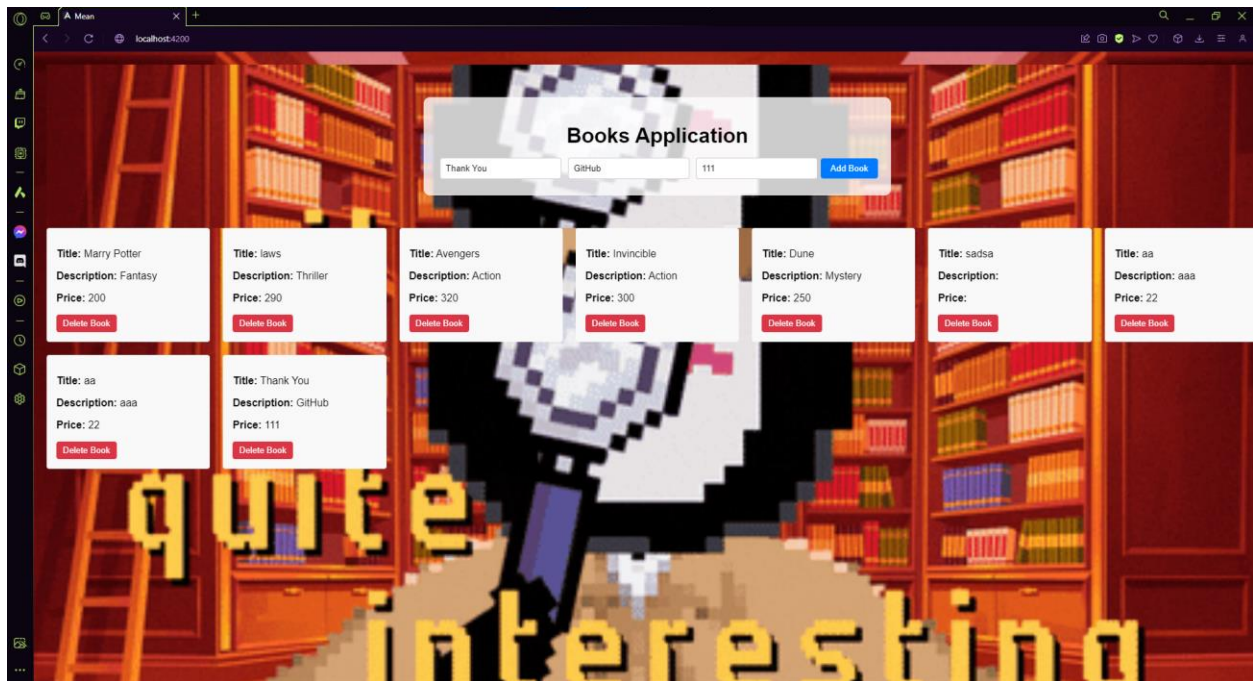


Adding books:



## Deleting books:



This screenshot shows the Visual Studio Code editor with the file explorer on the left displaying the project structure. The main editor window shows the `app.component.html` file. The code defines a container with a header and a list of books. The header contains input fields for title, description, and price, along with an 'Add Book' button. The list of books is rendered using an `*ngFor` loop, displaying the title, description, price, and a 'Delete Book' button for each item.

```
1 <div class="container">
2   <div class="header">
3     <div class="header-content">
4       <div class="input-group">
5         <input id="newBook" placeholder="Title">
6         <input id="newDesc" placeholder="Description">
7         <input id="newPrice" placeholder="Price">
8         <button (click)="addBook()">Add Book</button>
9       </div>
10    </div>
11  </div>
12 </div>
13
14 <div class="books-list">
15   <div *ngFor="let book of books" class="book-item">
16     <div class="book-info">
17       <strong>Title:</strong> {{book.title}}</p>
18       <strong>Description:</strong> {{book.description}}</p>
19       <strong>Price:</strong> {{book.price}}</p>
20     </div>
21     <button (click)="deleteBook(book.id)" class="delete-button">Delete Book</button>
22   </div>
23 </div>
24 </div>
25
```

This screenshot shows the Visual Studio Code editor with the file explorer on the left. The main editor window shows the `app.module.ts` file. The code sets up the Express application, configures the database connection, and defines the REST API endpoints for books. The endpoints include a GET endpoint to retrieve all books, a POST endpoint to add a new book, and a DELETE endpoint to remove a book.

```
1 var express = require('express');
2 var MongoClient = require('mongodb').MongoClient;
3 var cors = require('cors');
4 const multer = require('multer');
5
6 var app = express();
7
8 app.use(cors());
9
10 var CONNECTION_STRING = "mongodb+srv://allencarlo32:psw@cluster0.yyz58lq.mongodb.net/retryWrites=true&majorityOpName=Cluster0";
11 var DATABASENAME = "mydb";
12
13 var database;
14
15 app.listen(5080, () => {
16   MongoClient.connect(CONNECTION_STRING, (error, client) => {
17     database = client.db(DATABASENAME);
18     console.log("MONGOCONNECT SUCCESS!");
19   });
20 });
21
22 app.get('/api/books/getBooks', (req, res) => {
23   database.collection("books").find({}).toArray((error, result) => {
24     res.send(result);
25   });
26 });
27
28 app.post('/api/books/AddBook', multer().none(), async (req, res) => {
29   try {
30     const numOfDocs = await database.collection("books").countDocuments();
31     await database.collection("books").insertOne({
32       id: (numOfDocs + 1).toString(),
33       title: req.body.title,
34       description: req.body.description,
35       price: req.body.price
36     });
37     res.json("Added Successfully");
38   } catch (error) {
39     console.error("Error adding book:", error);
40     res.status(500).json({ error: "Failed to add book" });
41   }
42 });
43
44 app.delete('/api/books/DeleteBook', (req, res) => {
45   database.collection("books").deleteOne({
46     id: req.query.id
47   });
48   res.json("Deleted successfully!");
49 });
50
```