

FINAL PROJECT
ADVANCED DATABASE SYSTEMS DESIGN
BOOK CATALOG MANAGEMENT SYSTEM

GitHub Link: https://github.com/VENUKENT/ADSD_Final_Project

Web Application:

The web application Book Catalog Management System is a platform for managing and organizing information about a collection of books. The web application streamlines the process of managing a library or personal book collection. Users can easily navigate through their catalog, obtain detailed information about books, contribute ratings, and keep the catalog organized and current. This application is particularly useful for book enthusiasts, librarians, or anyone looking to manage and enhance their book collection efficiently.

I have provided some of the key functionalities and their corresponding benefits are as follows.

Viewing and Organizing Books: Users can easily view a list of books in the catalog.

Adding New Books: Allows users to add new books to the catalog, providing a way to expand the collection.

Viewing Book Details: Users can access detailed information about each book, including title, author, and associated ratings. Enables users to make informed decisions about which books to read based on comprehensive details.

Adding Ratings: Users have the ability to add ratings to individual books. Facilitates a user-driven rating system, enhancing the catalog with user feedback.

Updating Book Information: Provides the functionality to update the details of existing books. Ensures that the catalog remains accurate and up-to-date with any changes.

Deleting Books: Allows users to remove books from the catalog.

Why it is Useful:

The application offers a centralized and efficient way to organize a book collection, allowing users to easily find specific books based on titles, authors, or other criteria. Its user-friendly interface is accessible to individuals with varying technical expertise, making it suitable for book enthusiasts, librarians, and anyone with a significant collection of books, addressing the specific needs of individuals or organizations dealing with book management.

Why this web application requires two tables:

In the Book Catalog Management system, we are using two tables, one for storing Book names and the other for ratings which help us to store and organize the data efficiently.

Book table:

The book table stores information about each book in the catalog, including attributes such as id, title, and author.

Ratings table:

The Rating table is likely used to store ratings associated with each book. It includes fields such as book_id, rating_id, and rating_value.

Reason for using two tables:

In database design, the decision to use two tables instead of a single table often arises from the need to manage related but distinct entities or concepts. In the case of "Book Catalog Management System," I have considered the separate tables for `books` and `ratings` for the following reasons:

Normalization: Two tables allow for a more normalized database structure, minimizing redundancy and dependency. Each table addresses a specific concern, leading to efficient storage and reduced data duplication.

Scalability: Separate tables make it easier to scale the application, allowing flexibility in expanding the database schema.

Maintenance: Separating books and ratings into different tables makes the system more maintainable, reducing the risk of unintended consequences.

Performance Optimization: Depending on usage patterns and queries, having separate tables can optimize certain types of queries.

Data Integrity: Using separate tables allows for the enforcement of specific constraints on each table, ensuring data integrity.

Why SQLite:

I selected SQLite for the Book Catalog management System due to its excellent fit for small to medium-sized projects. With its file-based system, lack of server requirements, and seamless integration with Python, SQLite simplifies deployment, administration, and data interaction, making it an ideal choice for this application.

Book Catalog Management System Project Code:

```
# app.py
from bottle import route, post, run, template, redirect, request
import database

# Initialize the database
database.initialize_database()

@route("/")
def get_index():
```

```
        redirect("/books")

@route("/books")
def get_books():
    books = database.get_all_books()
    return template("books.tpl", books=books)

@route("/books/add")
def get_add_book():
    return template("add_book.tpl")

@post("/books/add")
def post_add_book():
    title = request.forms.get("title")
    author = request.forms.get("author")
    database.add_book(title, author)
    redirect("/books")

@route("/books/<book_id>")
def get_book_details(book_id):
    book = database.get_book_details(book_id)
    ratings = database.get_ratings_for_book(book_id)
    return template("book_details.tpl", book=book,
ratings=ratings)

@route("/books/<book_id>/add_rating")
def get_add_rating(book_id):
    return template("add_rating.tpl", book_id=book_id)

@post("/books/<book_id>/add_rating")
def post_add_rating(book_id):
    rating_value = request.forms.get("rating")
    database.add_rating(book_id, rating_value)
    redirect(f"/books/{book_id}")

@route("/books/<book_id>/update")
def get_update_book(book_id):
    book = database.get_book_details(book_id)
    return template("update_book.tpl", book=book)

@post("/books/<book_id>/update")
def post_update_book(book_id):
    title = request.forms.get("title")
    author = request.forms.get("author")
    database.update_book(book_id, title, author)
```

```

        redirect("/books")

@route("/books/<book_id>/delete")
def get_delete_book(book_id):
    database.delete_book(book_id)
    redirect("/books")

run(host='localhost', port=8080)

# database.py
books = []
book_id_counter = 1

def initialize_database():
    global books, book_id_counter
    books = []
    book_id_counter = 1

def get_all_books():
    return books

def add_book(title, author):
    global book_id_counter
    book = {
        'id': book_id_counter,
        'title': title,
        'author': author,
        'ratings': []
    }
    book_id_counter += 1
    books.append(book)

def get_book_details(book_id):
    return next((book for book in books if str(book['id']) ==
book_id), None)

def get_ratings_for_book(book_id):
    book = get_book_details(book_id)
    return book['ratings'] if book else []

def add_rating(book_id, rating_value):
    book = get_book_details(book_id)
    if book:

```

```

        book['ratings'].append(int(rating_value))

def update_book(book_id, title, author):
    book = get_book_details(book_id)
    if book:
        book['title'] = title
        book['author'] = author

def delete_book(book_id):
    global books
    books = [book for book in books if str(book['id']) != book_id]

```

tpl files:

```

<!DOCTYPE html>

<html>

<head>

    <title>Add Book</title>

</head>

<body>

    <h1>Add Book</h1>

    <form action="/books/add" method="post">

        <label for="title">Title:</label>

        <input type="text" id="title" name="title" required><br>

        <label for="author">Author:</label>

        <input type="text" id="author" name="author" required><br>

        <input type="submit" value="Add Book">

    </form>

    <p><a href="/books">Back to Books</a></p>

</body>

</html>

<!DOCTYPE html>

```

```

<html>
<head>
  <title>Add Rating</title>
</head>
<body>
  <h1>Add Rating</h1>
  <form action="/books/{{book_id}}/add_rating" method="post">
    <label for="rating">Rating:</label>
    <input type="number" id="rating" name="rating" min="1" max="5" required><br>

    <input type="submit" value="Add Rating">
  </form>
  <p><a href="/books/{{book_id}}">Back to Book Details</a></p>

```

```

</body>
</html>
<!DOCTYPE html>
<html>
<head>
  <title>{{book['title']}}</title>
</head>
<body>
  <h1>{{book['title']}}</h1>
  <p>Author: {{book['author']}}</p>

  <h2>Ratings</h2>
  <ul>
    % for rating in ratings:
      <li>{{rating}}</li>

```

```

        % end
    </ul>

    <p><a href="/books/{{book['id']}}/add_rating">Add Rating</a></p>
    <p><a href="/books/{{book['id']}}/update">Update Book</a></p>
    <p><a href="/books/{{book['id']}}/delete">Delete Book</a></p>
    <p><a href="/books">Back to Books</a></p>
</body>
</html>

<!DOCTYPE html>
<html>
<head>
    <title>Books</title>
</head>
<body>
    <h1>Books</h1>
    <ul>
        % for book in books:
            <li><a href="/books/{{book['id']}}">{{book['title']}}</a> by {{book['author']}}</li>
        % end
    </ul>
    <p><a href="/books/add">Add a Book</a></p>
</body>
</html>

<!DOCTYPE html>
<html>
<head>

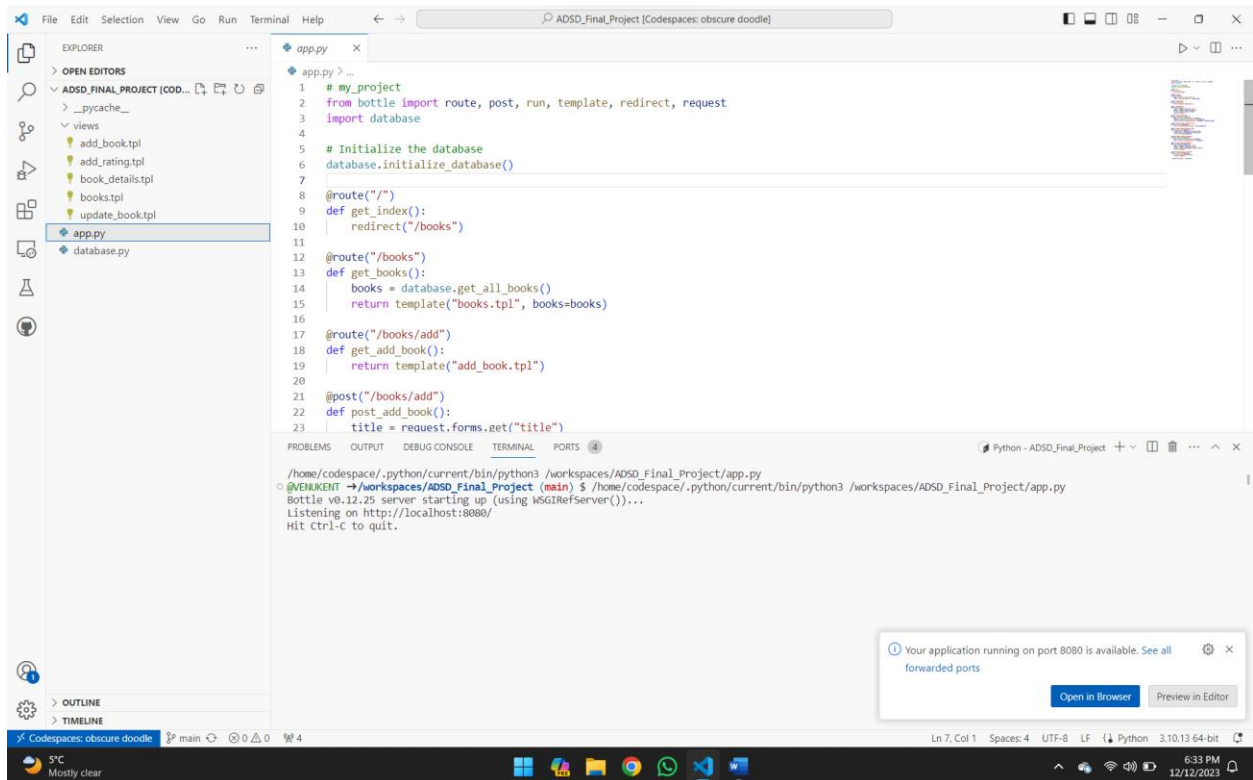
```

```
<title>Update Book</title>
</head>
<body>
  <h1>Update Book</h1>
  <form action="/books/{{book['id']}}/update" method="post">
    <label for="title">Title:</label>
    <input type="text" id="title" name="title" value="{{book['title']}}" required><br>

    <label for="author">Author:</label>
    <input type="text" id="author" name="author" value="{{book['author']}}" required><br>

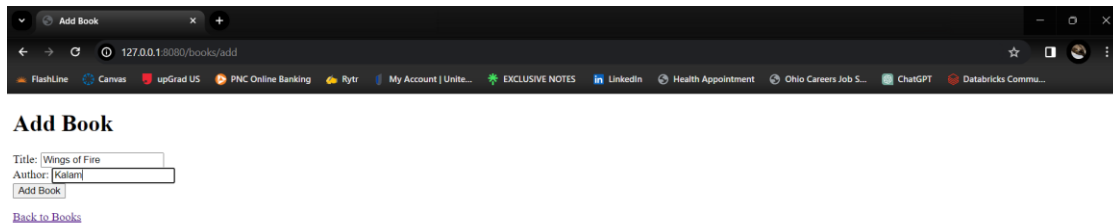
    <input type="submit" value="Update Book">
  </form>
  <p><a href="/books/{{book['id']}}">Back to Book Details</a></p>
</body>
</html>
```

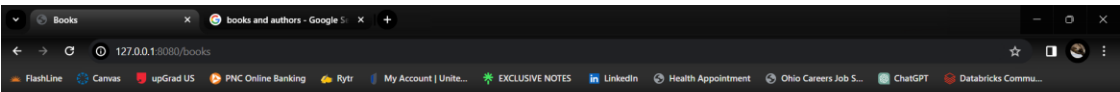

Implementation of the project in Visual Studio:



Performing CRUD Operation on web application:

Adding a Book to Catalog:

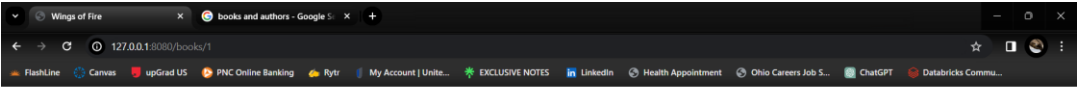




Books

- [Wings of Fire](#) by Kalam
- [Heart of Soldier](#) by James

[Add a Book](#)



Wings of Fire

Author: Kalam

Ratings

[Add Rating](#)

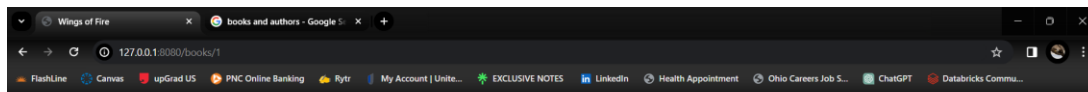
[Update Book](#)

[Delete Book](#)

[Back to Books](#)



Add a Rating to the book in Catalog:



Wings of Fire

Author: Kalam

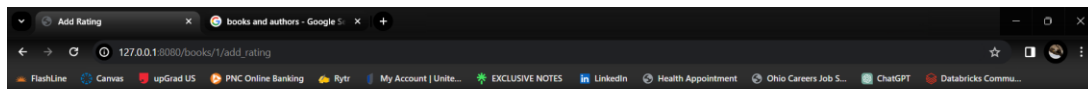
Ratings

[Add Rating](#)

[Update Book](#)

[Delete Book](#)

[Back to Books](#)



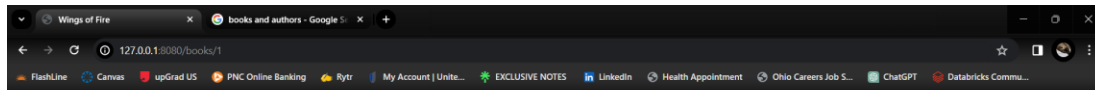
Add Rating

Rating:

[Add Rating](#)

[Back to Book Details](#)





Wings of Fire

Author: Kalam

Ratings

• 5

[Add Rating](#)

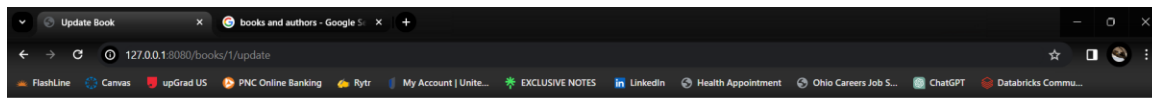
[Update Book](#)

[Delete Book](#)

[Back to Books](#)



Updating a book Information in Catalog:

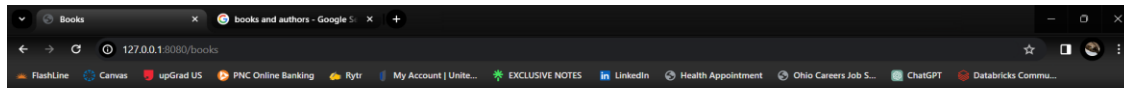


Update Book

Title:
Author:

[Back to Book Details](#)





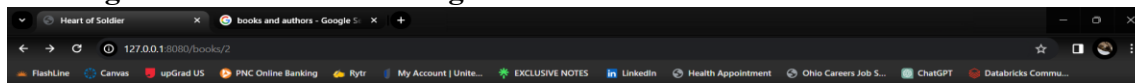
Books

- [Wings of Fire](#) by Abdul Kalam
- [Heart of Soldier](#) by James

[Add a Book](#)



Deleting a Book from the Catalog:



Heart of Soldier

Author: James

Ratings

[Add Rating](#)

[Update Book](#)

[Delete Book](#)

[Back to Books](#)



