FINAL PROJECT REPORT

COMP 3005 – F22

Sean Matute | 101189868

Project was completed solo.

Table of Contents

2.1 Conceptual Design

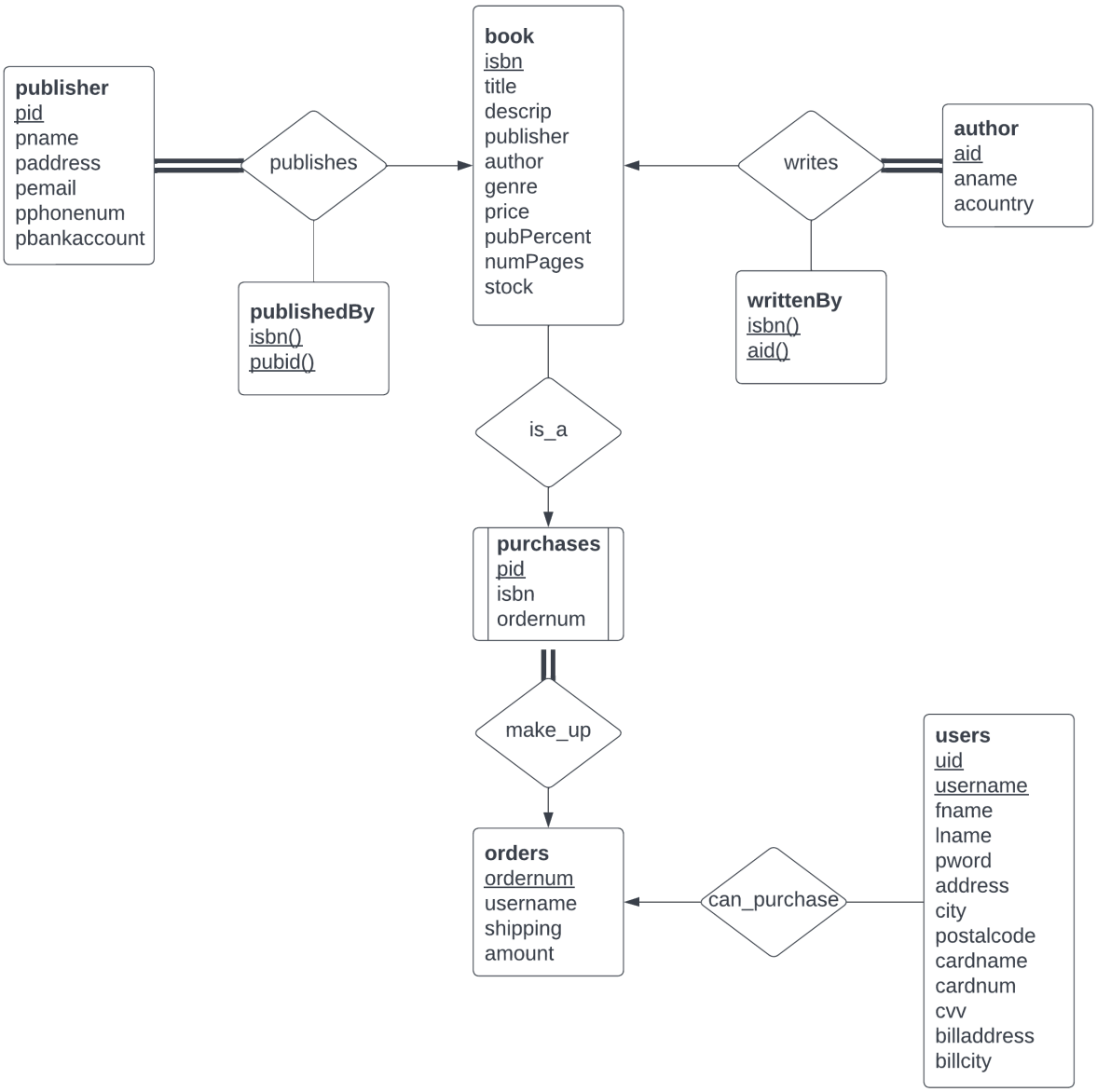


Figure 1: ER-Diagram

Entities

- book:
  Primary Key:
  - Isbn: as no 2 books have the same ISBN number

Composition:

Contains a variety of info that defines the book. These are its title, descript(description), publisher (publisher name), author (author name), genre, price, pubPercent (percent that goes to publisher), numPages (number of pages in the book) and stock.

- publisher:

  Primary Key:

  - pid: unique publisher id

  Composition:

  Contains a variety of info that defines the publisher. Contains their name, address, email address, phone number and bank account info.

- publishedBy:

  Primary Key:

  - pubid: id of a publisher **ALSO A FOREIGN KEY TO PUBLISHER**
  - isbn: isbn of a book **ALSO A FOREIGN KEY TO BOOK**

  Composition:

  This entity exists to easily connect a publisher to the book they published.

- writtenBy:

  Primary Key:

  - aid: id of a author **ALSO A FOREIGN KEY TO AUTHOR**
  - isbn: isbn of a book **ALSO A FOREIGN KEY TO BOOK**

  Composition:

  This entity exists to easily connect an author to the book they wrote.

- author:

  Primary Key:

  - aid: unique author id

  Composition:

  Contains limited info that defines the author. Contains their name and country of origin. If people want to learn more about an author they'll use google not this database.

- purchases:

  Primary Key:

  - pid: unique purchase id

  Composition:

  Contains an isbn to the book and an ordernum to the order. Each tuple in purchases is a purchase of a single book and will be assigned to an ordernum stores in the orders relation.

- orders:
  Primary Key:
  - ordernum: unique order number

  Composition:
  Contains the username of the user that created the order, the shipping address to the user and the total amount of the order.

- users:
  Primary Key:
  - uid: unique user id
  - username: username created by the user at registration

  Composition:
  Contains a variety of info on the individual user. Admin is also a user and will have the username of "admin", giving them the special owner privileges. Other info stored is the fname (first name), lname (last name), username, pword (password), address, city, postalcode, cardname, cardnum, cvv, billaddress and billcity

Cardinalities

- Book with author (and writtenBy):
  - 1-to-many relationship, each book can have only one author (in this db), but an author can write many books.
- Book with publisher (and publishedBy):
  - 1-to-many relationship, each book can have only one publisher (in this db), but a publisher can publish many books.
- Book with purchases:
  - 1-to-many relationship. Each tuple in purchases is a single book being purchased, so each purchase can have only one book. But one book can be purchased multiple times.
- Purchases with orders
  - 1-to-many relationship, as an order can be made up of multiple purchases but each purchase will only be assigned to one ordernum.
- Users with orders
  - 1-to-many relationship, as a user can have many orders but an order can only belong to one user.

Participation

Total Participation:

- Book to "publishes" as a book cannot exist without a publisher

- Book to “writes” as a book cannot exist without an author
- Purchases to “make_up” as an order cannot exist without purchases

Partial Participation:

- Every other relation in the schema as they can all exist without the other relations.

2.2 Reduction to Relation Schemas

- book(isbn, title, descrip, publisher, author, genre, price, pubPercent, numPages, stock)
- users(uid, username, fname, lname, pword, address, city, postalcode, cardname, cardnum, cvv, billaddress, billcity)
- publisher(pid, pname, paddress, pemail, pphonenum, pbankaccount)
- author(aid, name, country)
- writtenBy(isbn, aid)
- publishedBy(isbn, pubid)
- purchases(pid, isbn, ordernum)
- orders(ordernum, username, shipping, amount)

2.3 Normalization of Relation Schemas

- book(isbn, title, descrip, publisher, author, genre, price, pubPercent, numPages, stock)

F={
Isbn -> isbn, title, descrip, publisher, author, genre, price, pubPercent, numPages, stock
}
Thus, $isbn^+$ is a superkey as it determined everything, therefore the relation is in BCNF.

- users(uid, username, fname, lname, pword, address, city, postalcode, cardname, cardnum, cvv, billaddress, billcity)

F={
uid -> uid, username, fname, lname, pword, address, city, postalcode, cardname, cardnum, cvv, billaddress, billcity
username -> uid -> uid, username, fname, lname, pword, address, city, postalcode, cardname, cardnum, cvv, billaddress, billcity

}

Thus, uid$^+$ and username$^+$ are superkeys as they determined everything, therefore the relation is in BCNF.

- publisher(pid, pname, paddress, pemail, pphonenum, pbankaccount)

F={
pid -> pid, pname, paddress, pemail, pphonenum, pbankaccount
}
Thus, pid$^+$ is a superkey as it determined everything, therefore the relation is in BCNF.

- author(aid, name, country)

F={
aid -> aid, name, country
}
Thus, aid$^+$ is a superkey as it determined everything, therefore the relation is in BCNF.

- writtenBy(isbn, aid)

F={
isbn -> isbn, aid
aid -> isbn, aid
}
Thus, isbn$^+$ and aid$^+$ are superkeys as they determined everything, therefore the relation is in BCNF.

- publishedBy(isbn, pubid)

F={
isbn -> isbn, pubid
pubid -> isbn, pubid
}
Thus, isbn$^+$ and pubid$^+$ are superkeys as they determined everything, therefore the relation is in BCNF.

- purchases(pid, isbn, ordernum)

F={
pid -> pid, isbn, ordernum
}
Thus, pid^+ is a superkey as it determined everything, therefore the relation is in BCNF.

- orders(ordernum, username, shipping, amount)

F={
ordernum -> ordernum, username, shipping, amount
}
Thus, $ordernum^+$ is a superkey as it determined everything, therefore the relation is in BCNF.

2.4 Database Schema Diagram

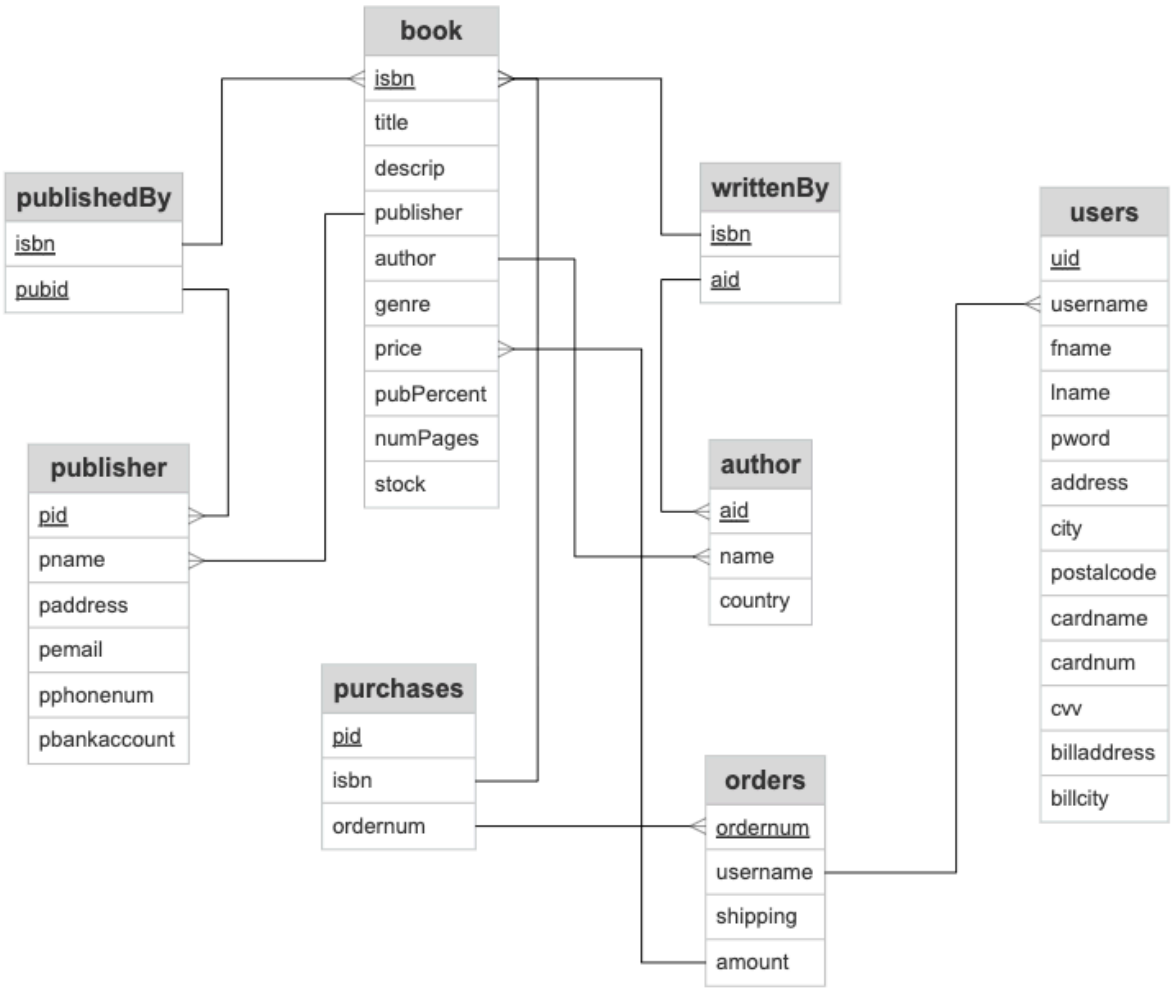


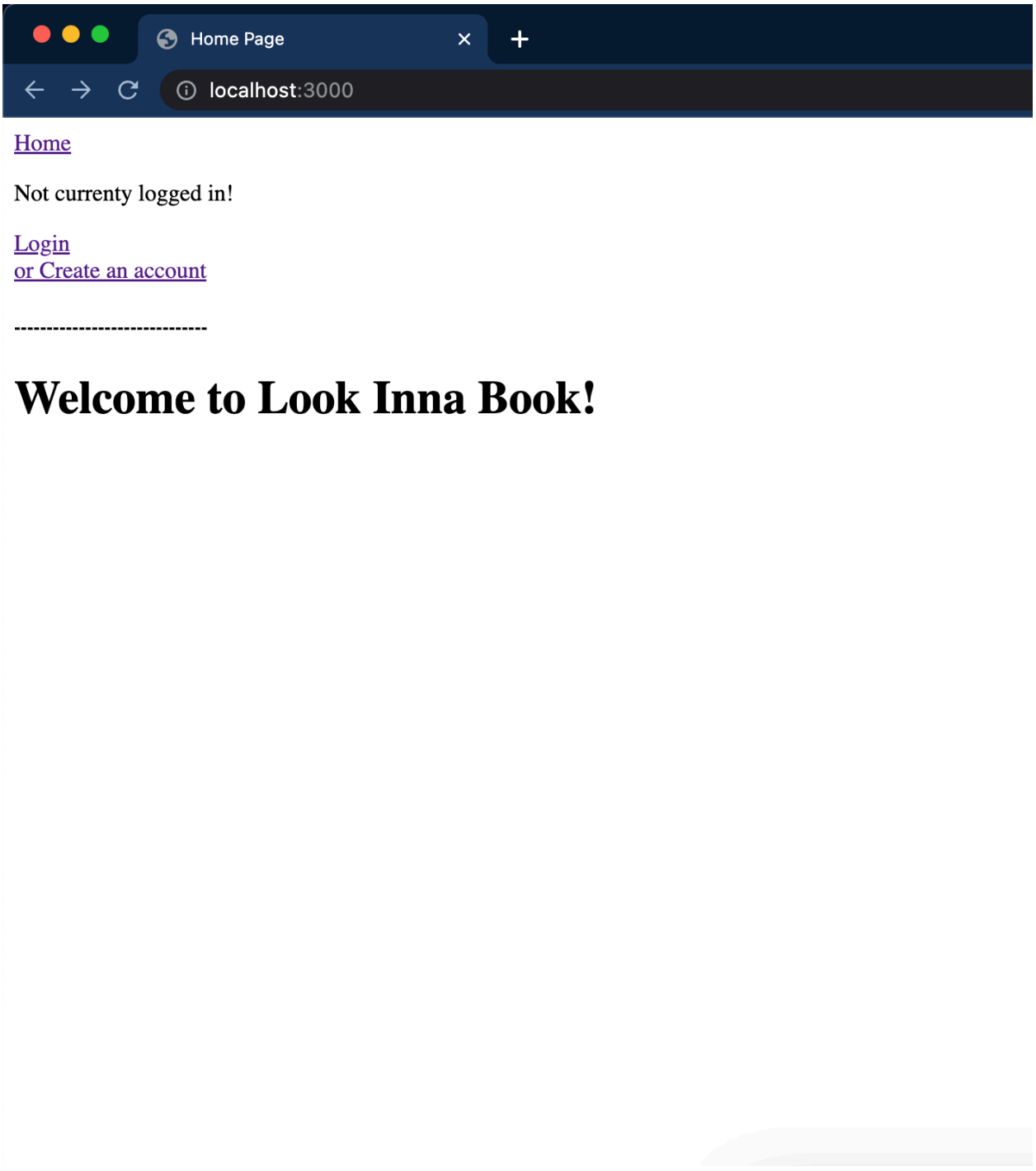
*Figure 2: Database Schema Diagram *The arrows are the unique 3 lined arrows**

2.5 Implementation

Basics

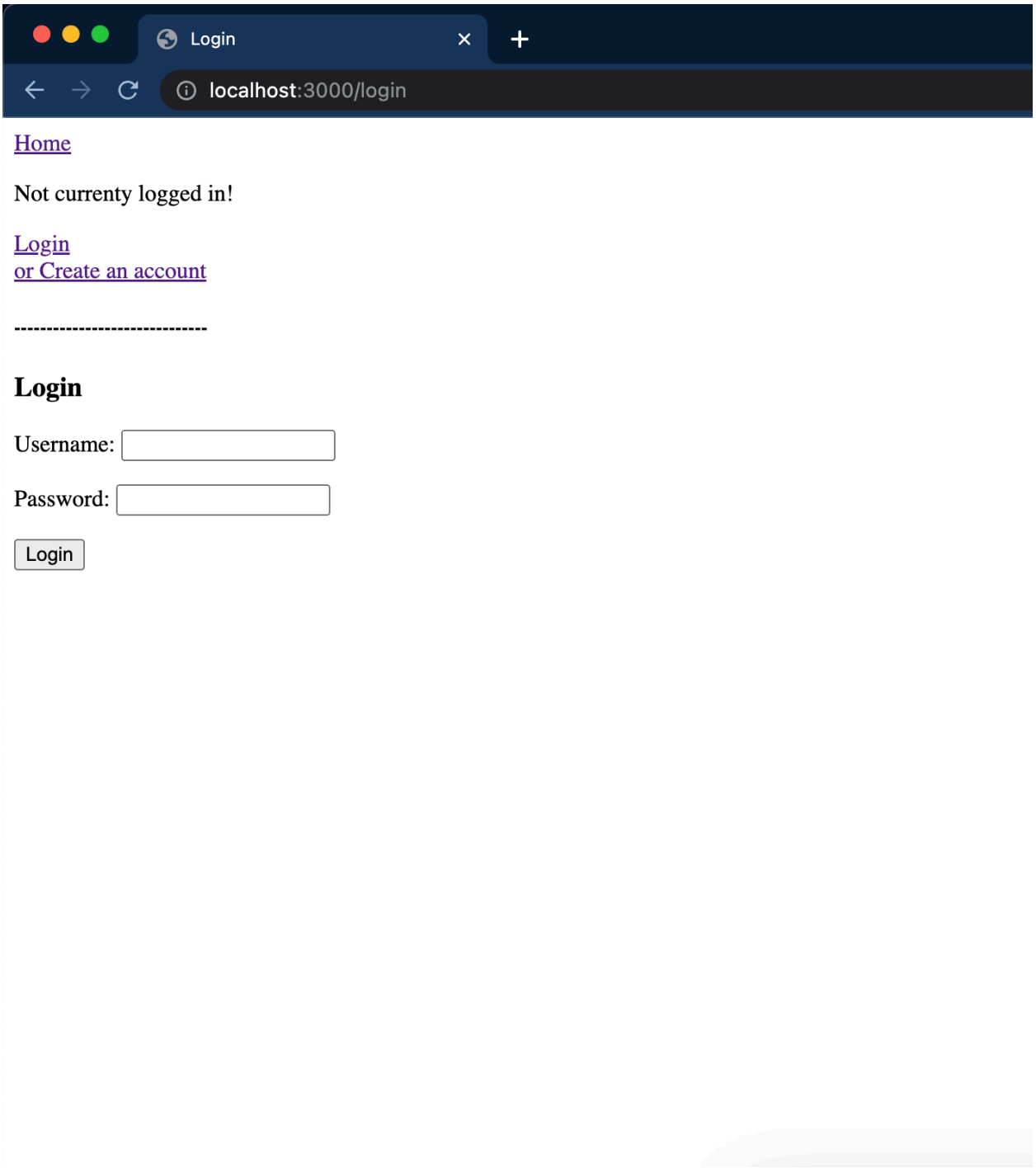
My implementation of the program is a web-based bookstore. The bookstore is written in javascript and everything is contained in 1 server.js file. Node package manager is used and the dependencies are listed in the package.json file. These dependencies are express to run the program through http://localhost:3000/, pg-promise to have async functions await the calls to the database, and pug to make html writing easier. The database connects to Postgres via the call to < pgp('postgres://USERNAME:PASSWORD@localhost:5432/DB_NAME) > on line 436 in server.js.

Home page



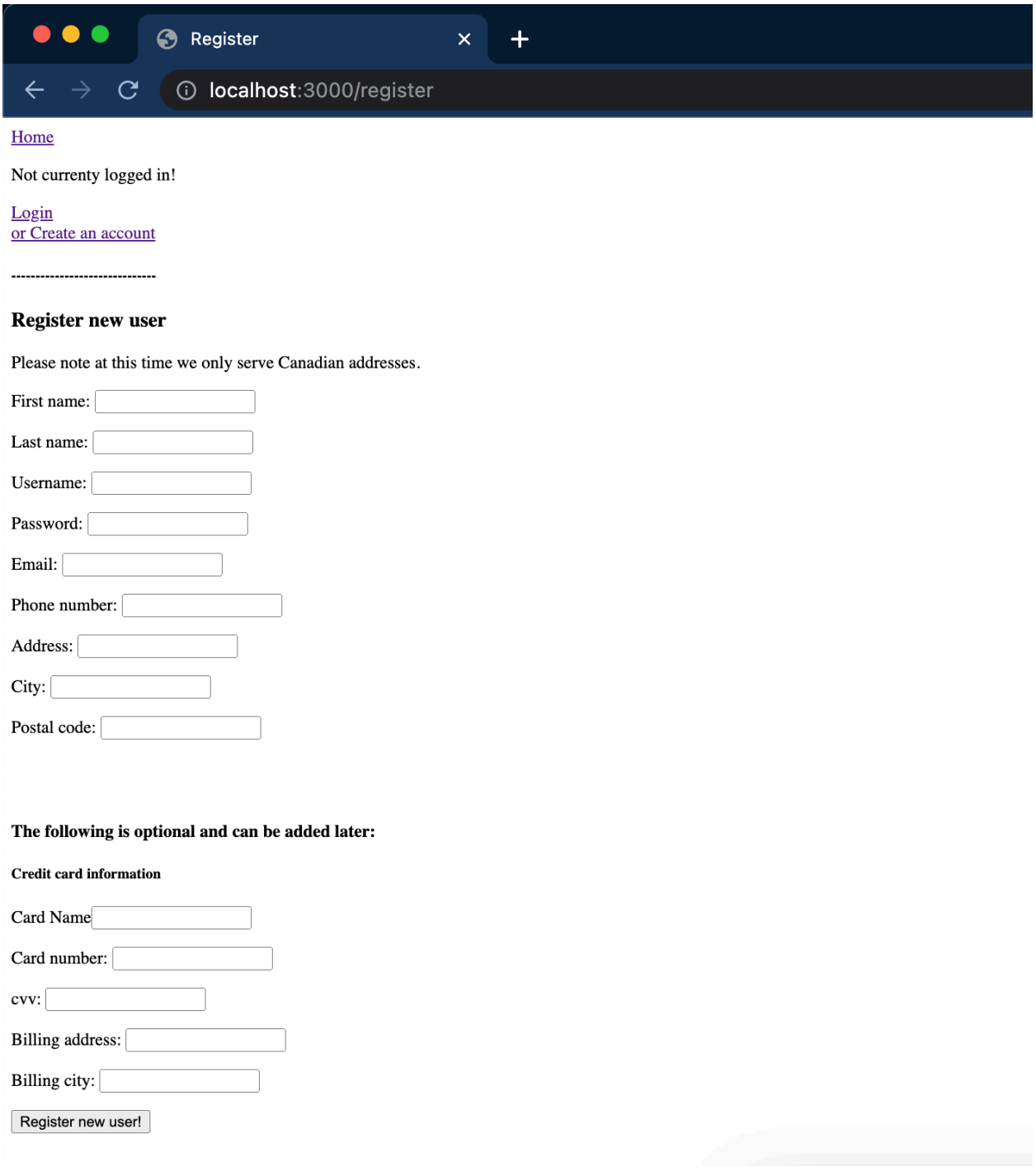
The basic homepage when you first open the website link http://localhost:3000/.

Login



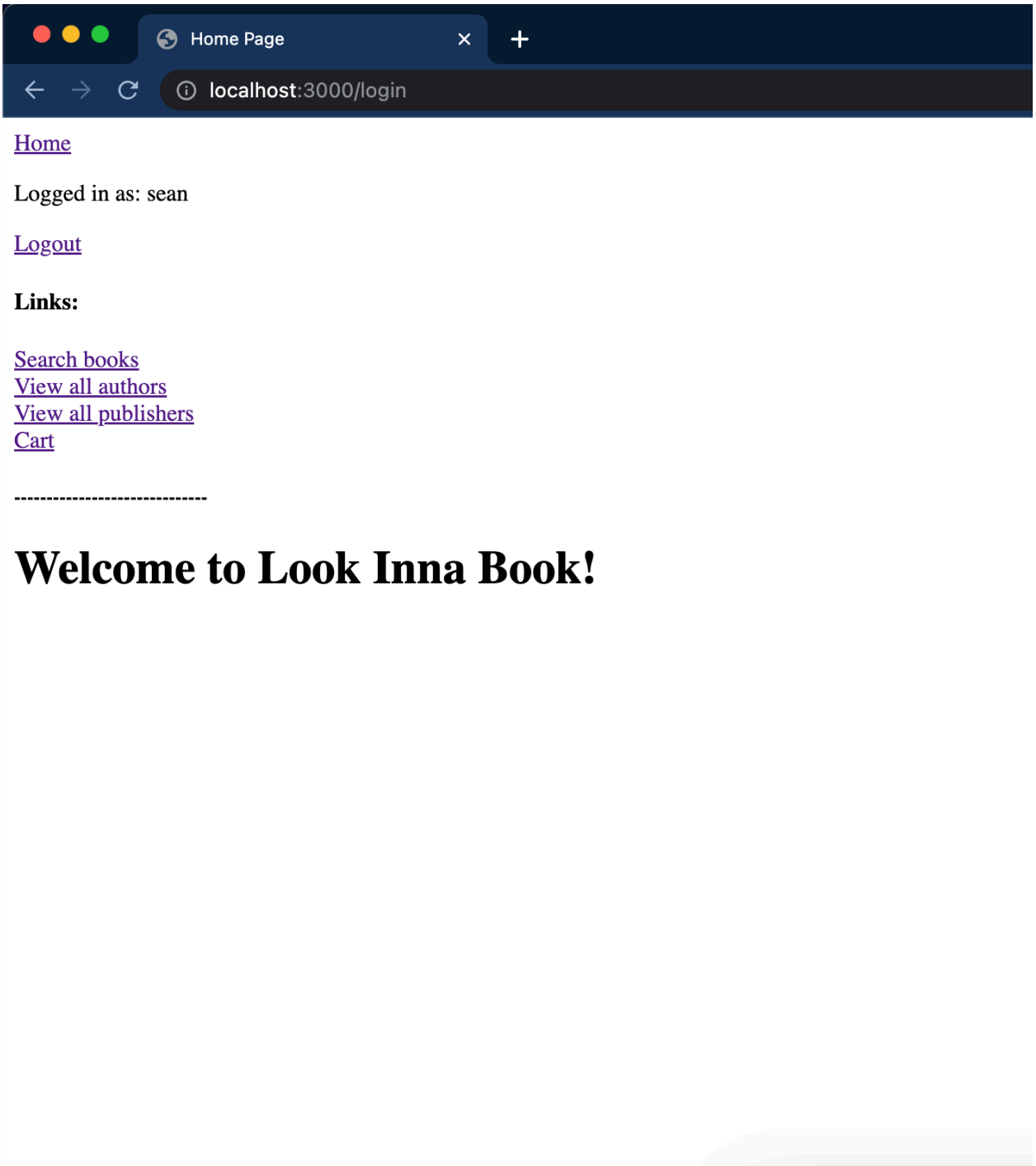
The login page if you already have an account.

Register



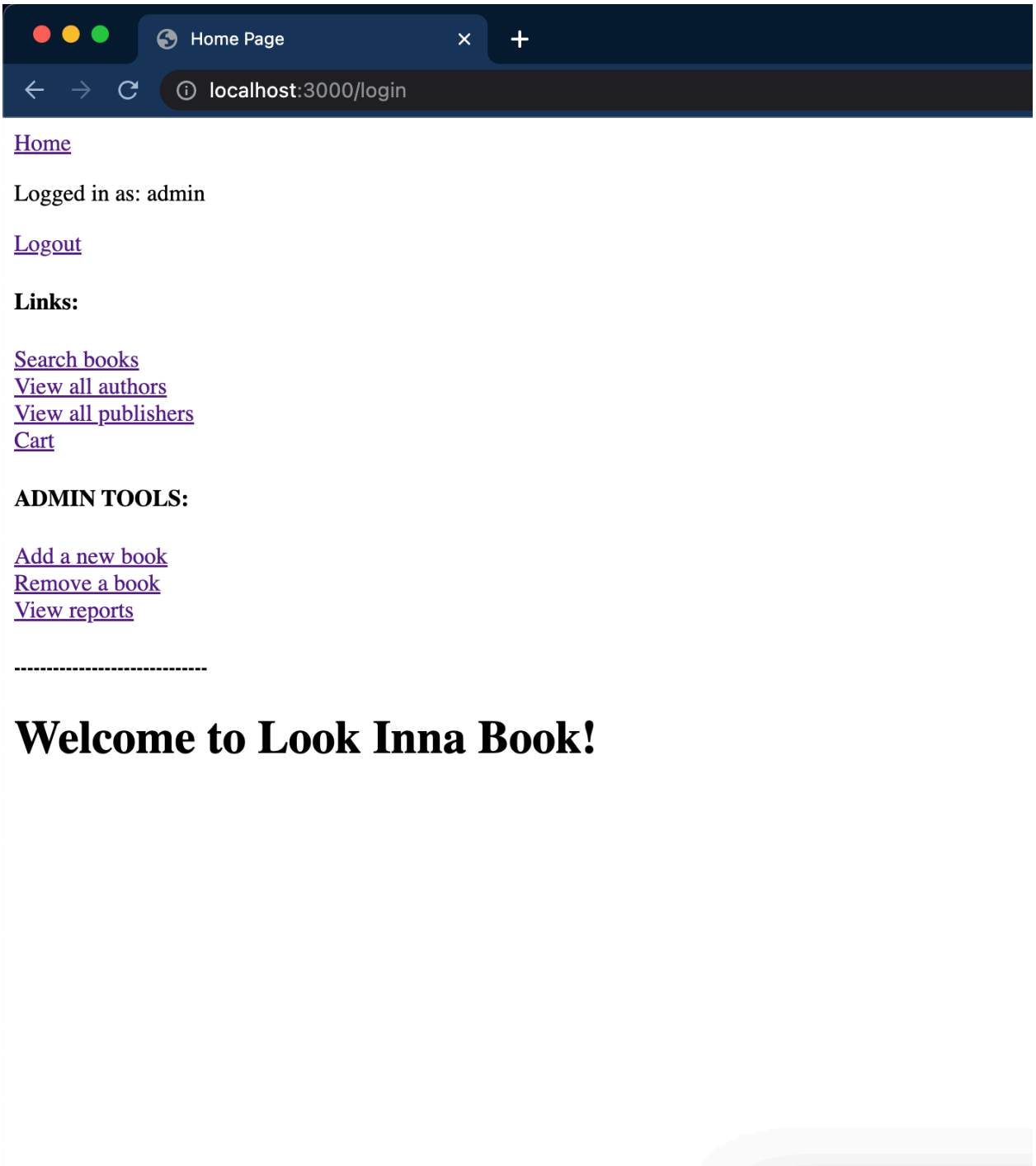
The registration page to create a new account (screenshot is zoomed out for formatting)

Home page when logged in as regular user



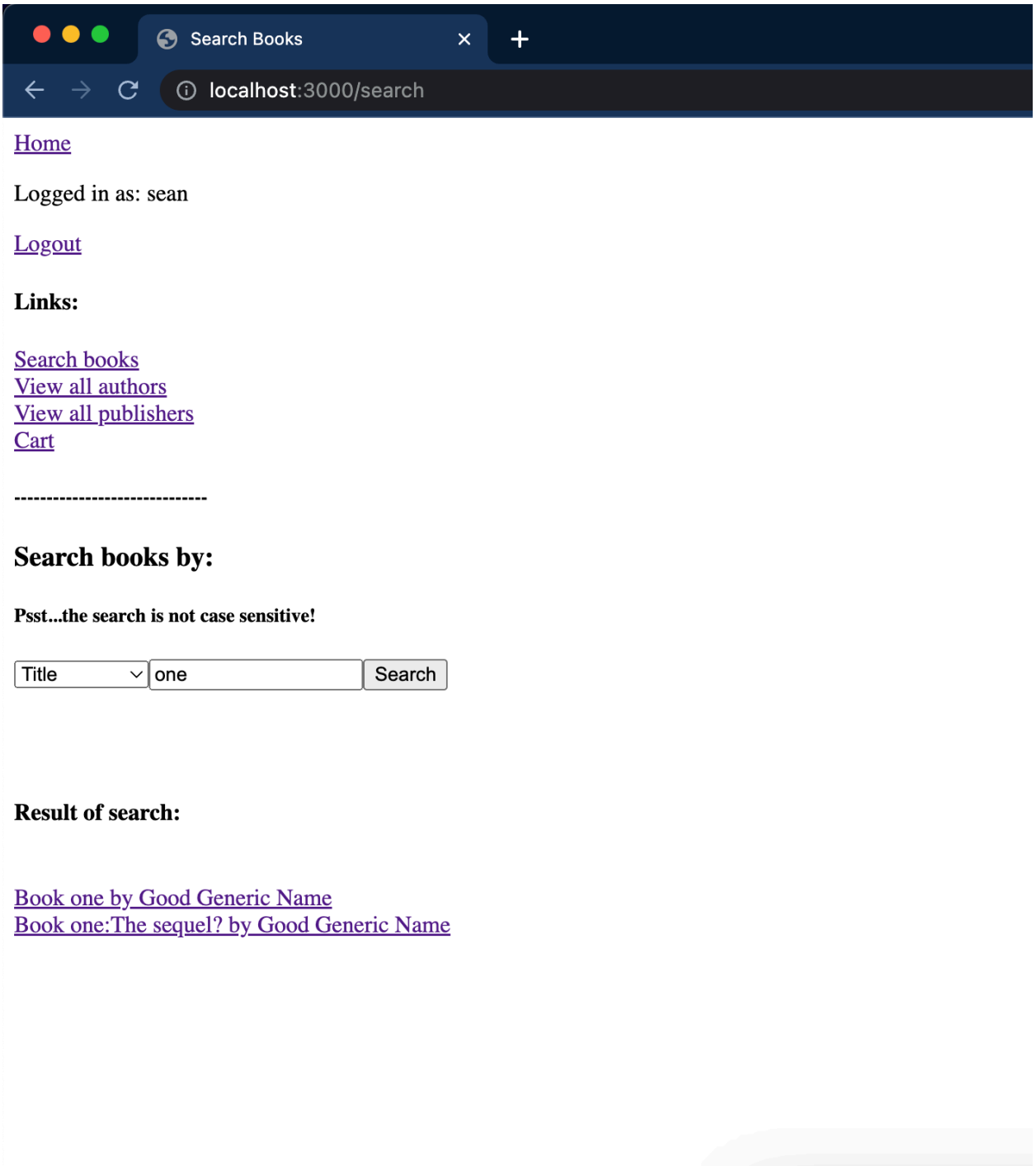
The homepage now has more links in the header and a logout page.

Home page when logged in as admin

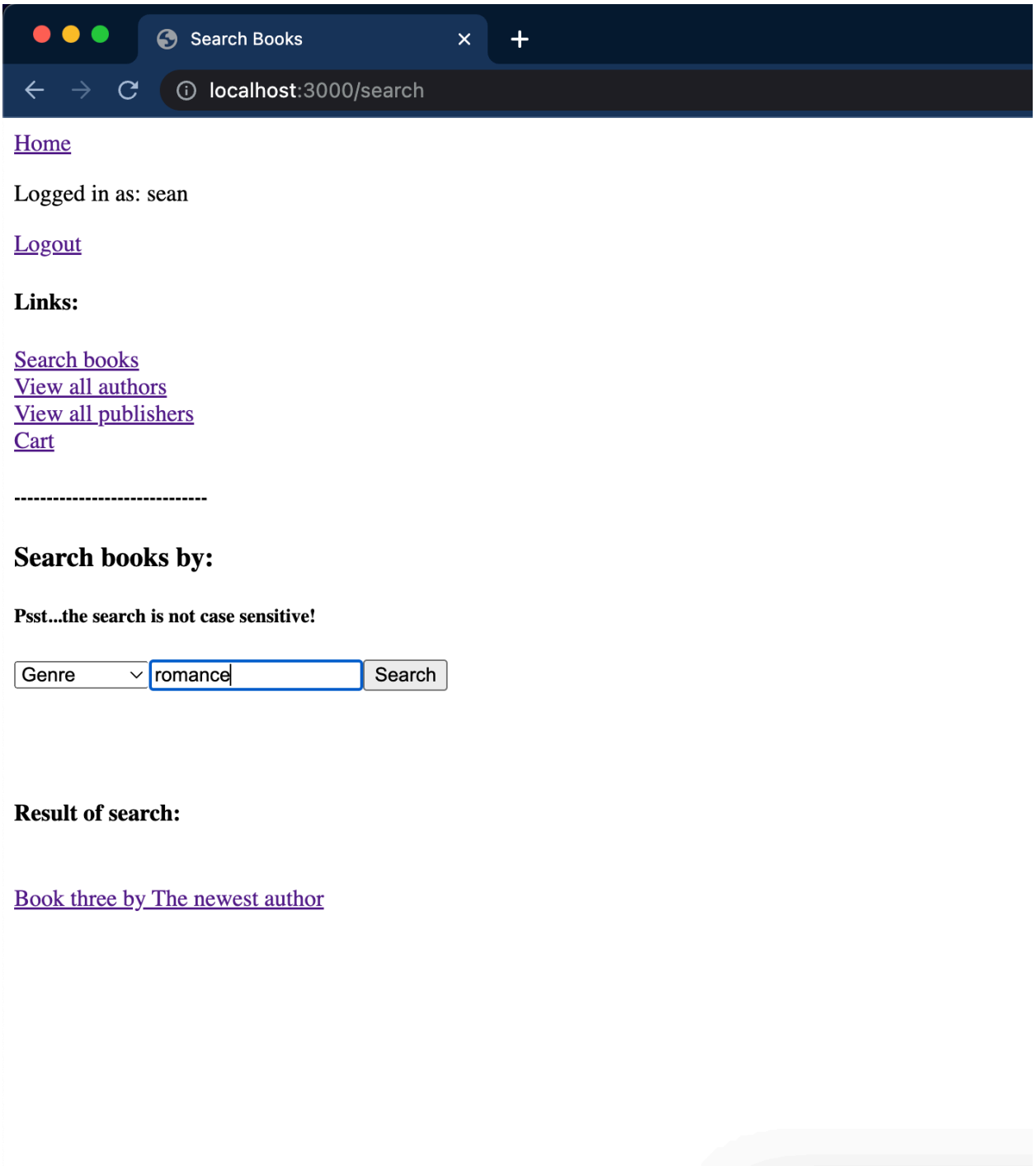


When logged in as an admin you can view admin specific information pages. If you attempt to view these pages not as an admin, it will not let you and simply display “You must be an admin to view this page.”

Search books

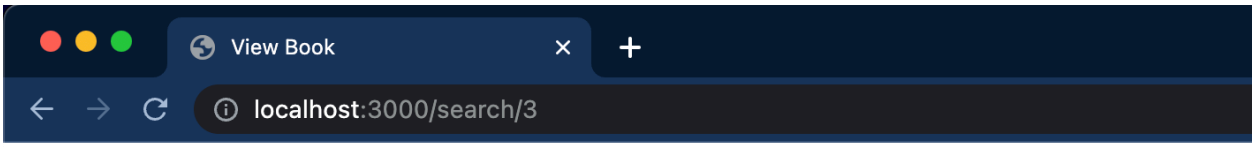


You can search books by a variety of different criteria. These criteria are: ALL(displays all books), Title, Author, ISBN, Genre, Publisher, Max_Pages, Min_Price, and Max_Price. Which are all fully functional.



Another example for genre romance.

View books



Home

Logged in as: sean

Logout

Links:

Search books
View all authors
View all publishers
Cart

Book three

ISBN: 3

Genre: Romance

Page count: 400

By: The newest author

Book description: Help me pls for the love of god

Price: $12.00

Stock: 40

Add to cart

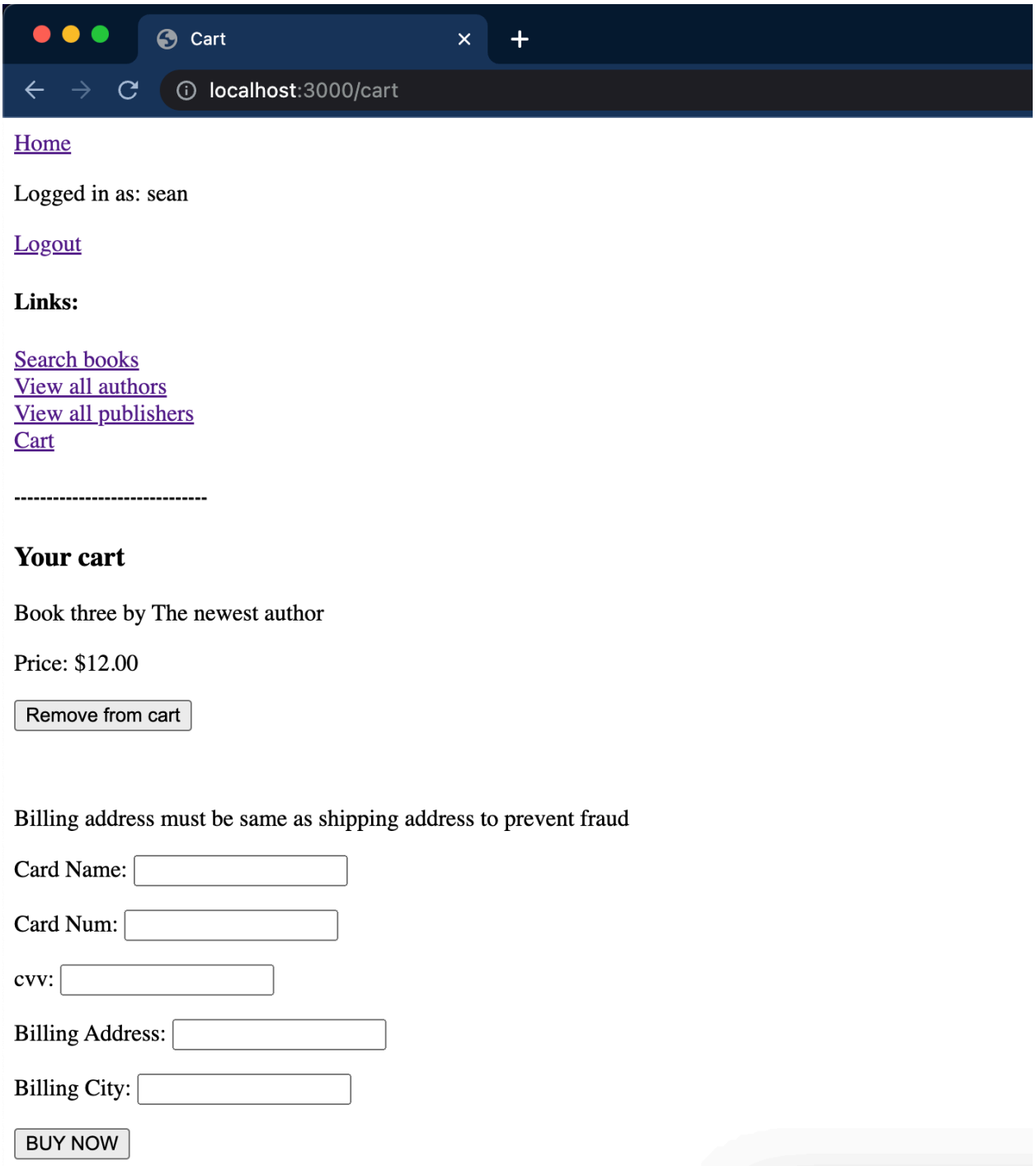
This book was published by: The newest publisher

Click to view more by The newest author

Viewing a specific book will show you info on that book. The link will always be /search/:ISBN. From here you can view the stock in store, add to cart and even click the link to view all books by that author.

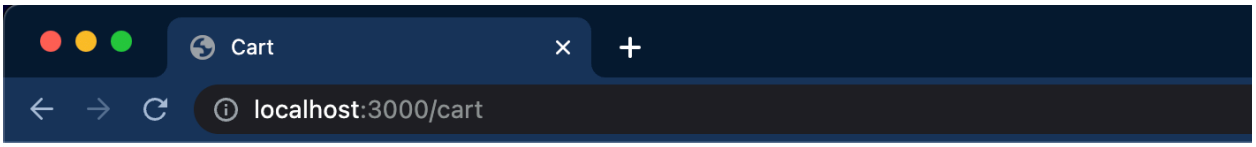
**The Add to cart button will not be displayed if the stock is 0. However that should never happen as the server automatically orders an additional 10 books when the stock is below 10 whenever a purchase is made.

Cart



Upon clicking add to cart it will take you to the cart page where you can view everything in your cart and remove from cart with ease. The cardnum and cvv will also only allow numbers to be inputted and you cannot buy now unless you fill out all the info in the form.

**Upon checking out the db will check if the stock is lower than 10 and if it is will increase whatever stock it is by 10. Using query: update book set stock = stock + 10 where stock<10;



Home

Logged in as: sean

Logout

Links:

Search books
View all authors
View all publishers
Cart

Your cart

Book three by The newest author

Price: $12.00

Remove from cart

Billing address must be same as shipping address to prevent fraud

Card Name:

Card Nu Please fill out this field.

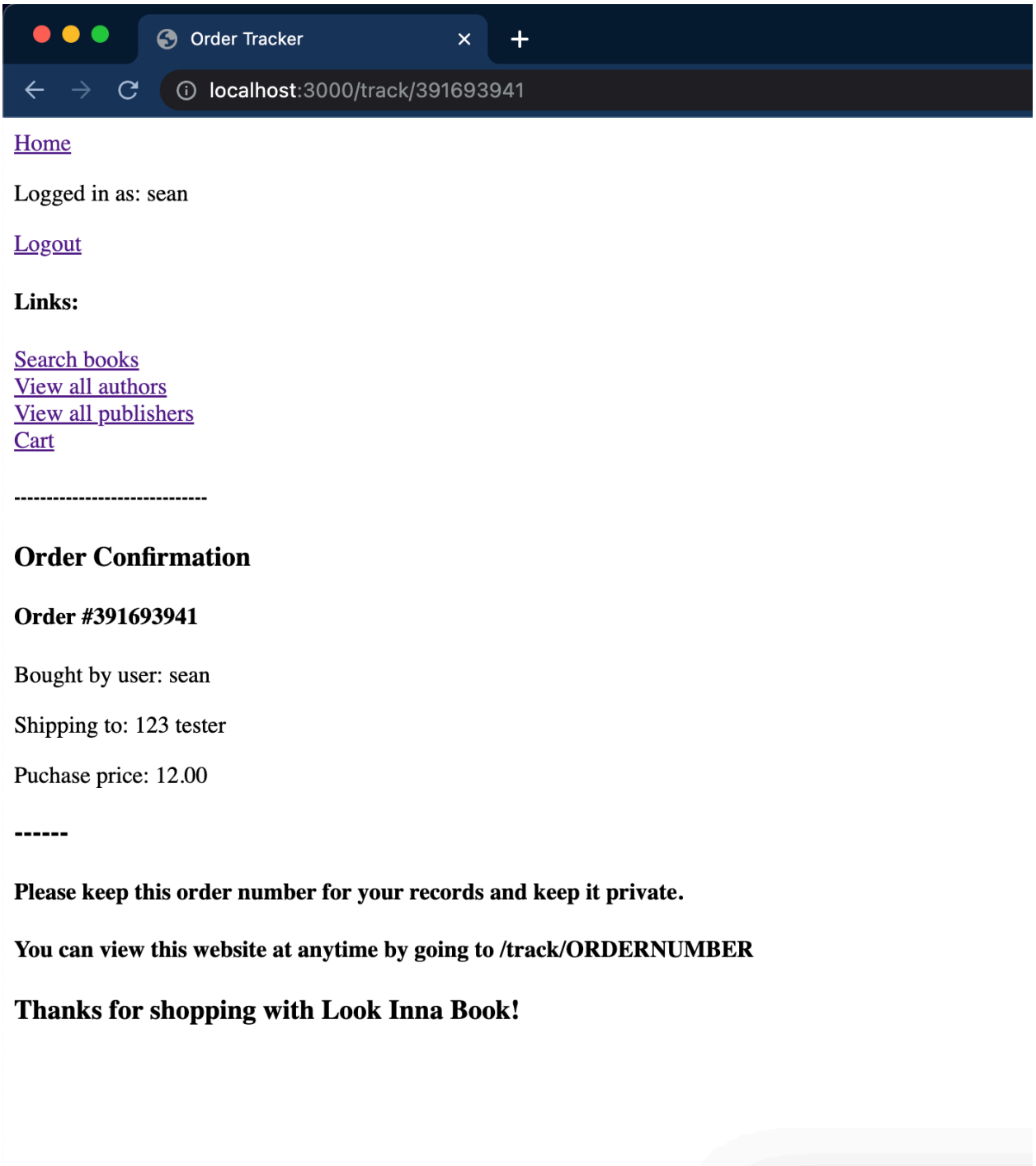
cvv: 1

Billing Address:

Billing City:

BUY NOW

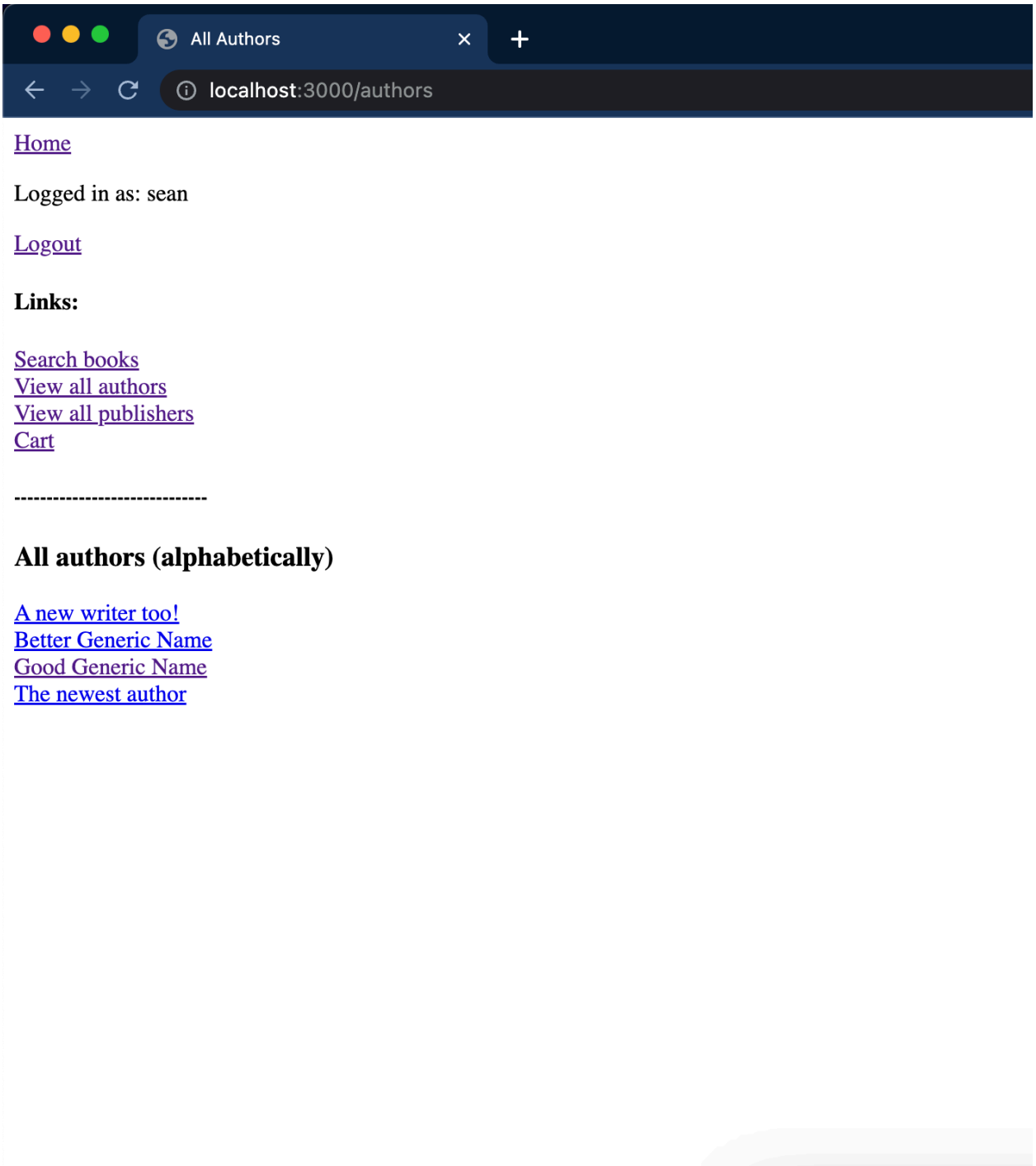
Order tracker



Upon hitting BUY NOW, it will purchase and redirect you to the orders order tracker page. The link will be /track/:ORDERNUM. It will generate a random order number to be your ordernumber.

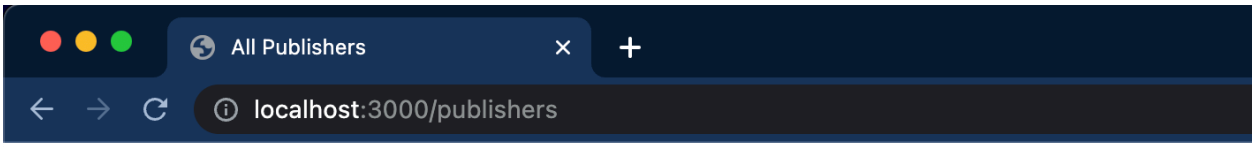
**ADMIN: Can also view all orders by going to /allorders

View all authors



Will display all the authors alphabetically

View all publishers (differs if logged in as admin)



Home

Logged in as: admin

Logout

Links:

Search books
View all authors
View all publishers
Cart

ADMIN TOOLS:

Add a new book
Remove a book
View reports

All publishers (alphabetically)

Name: A new publisher!

Email: p@email.com

Phone number: 6131001234

Admin info:

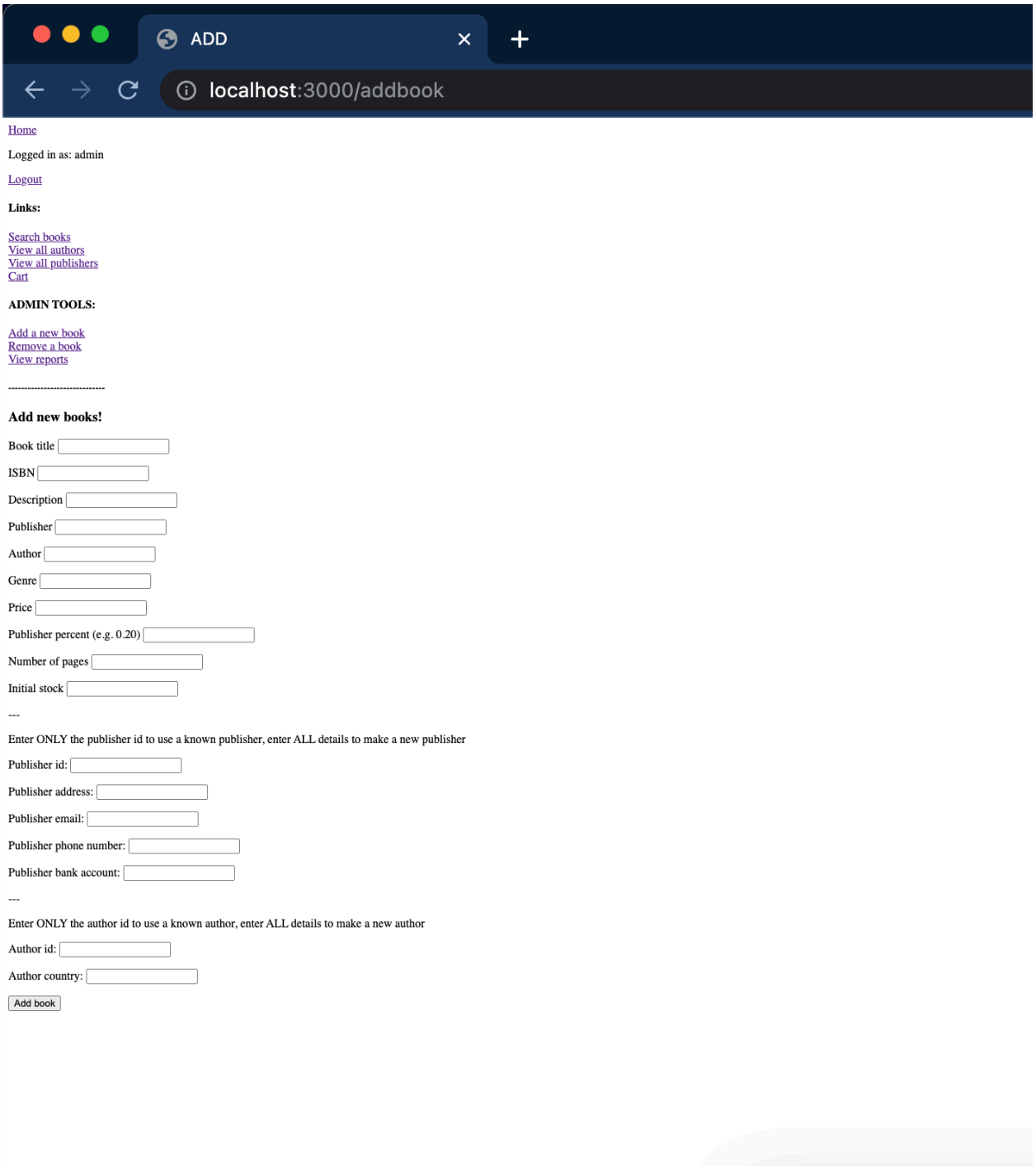
Publisher id: 000003

Address: 123 publish

Bank account info: 1000003

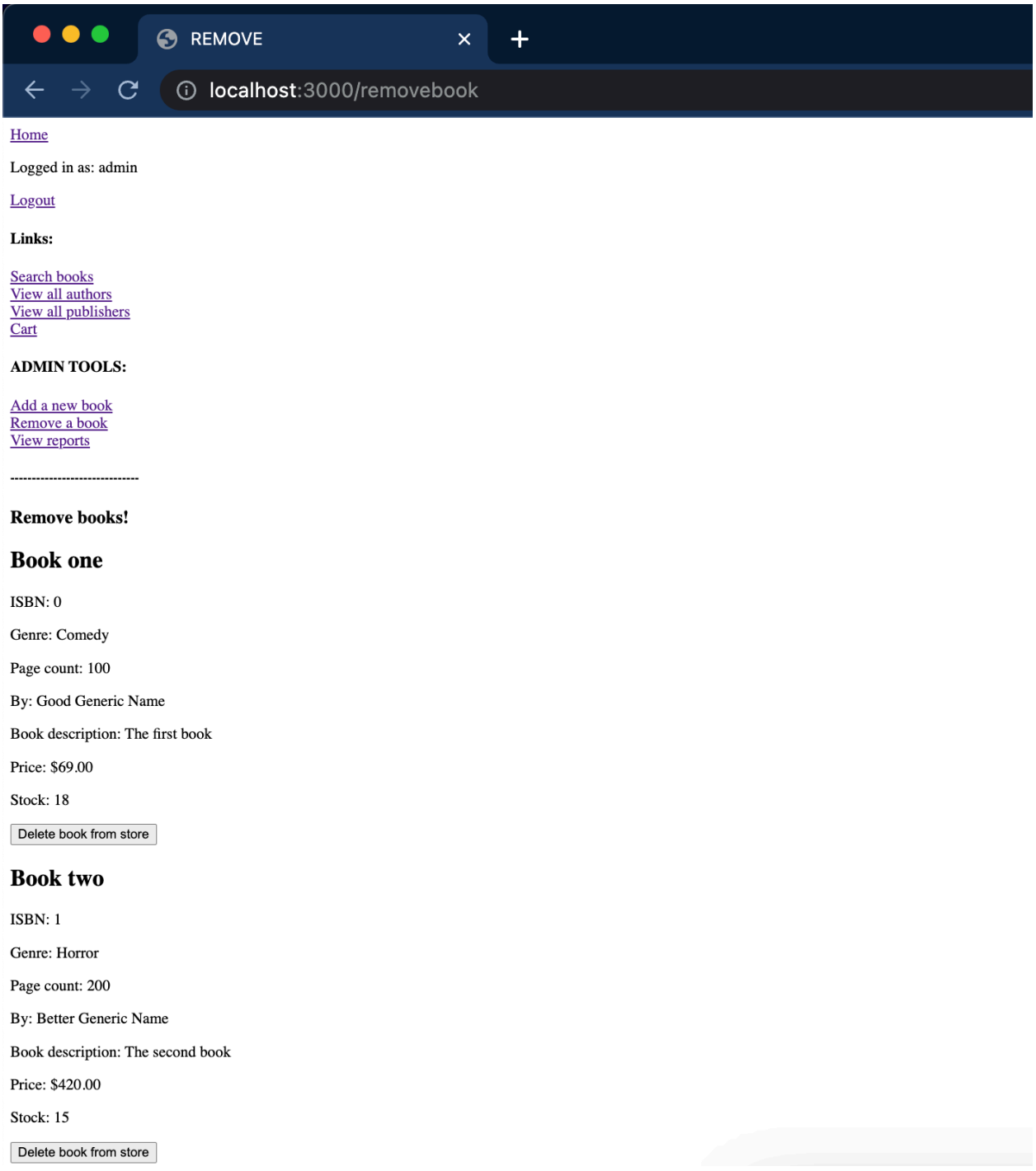
The admin info will not be displayed if not an admin.

Add book



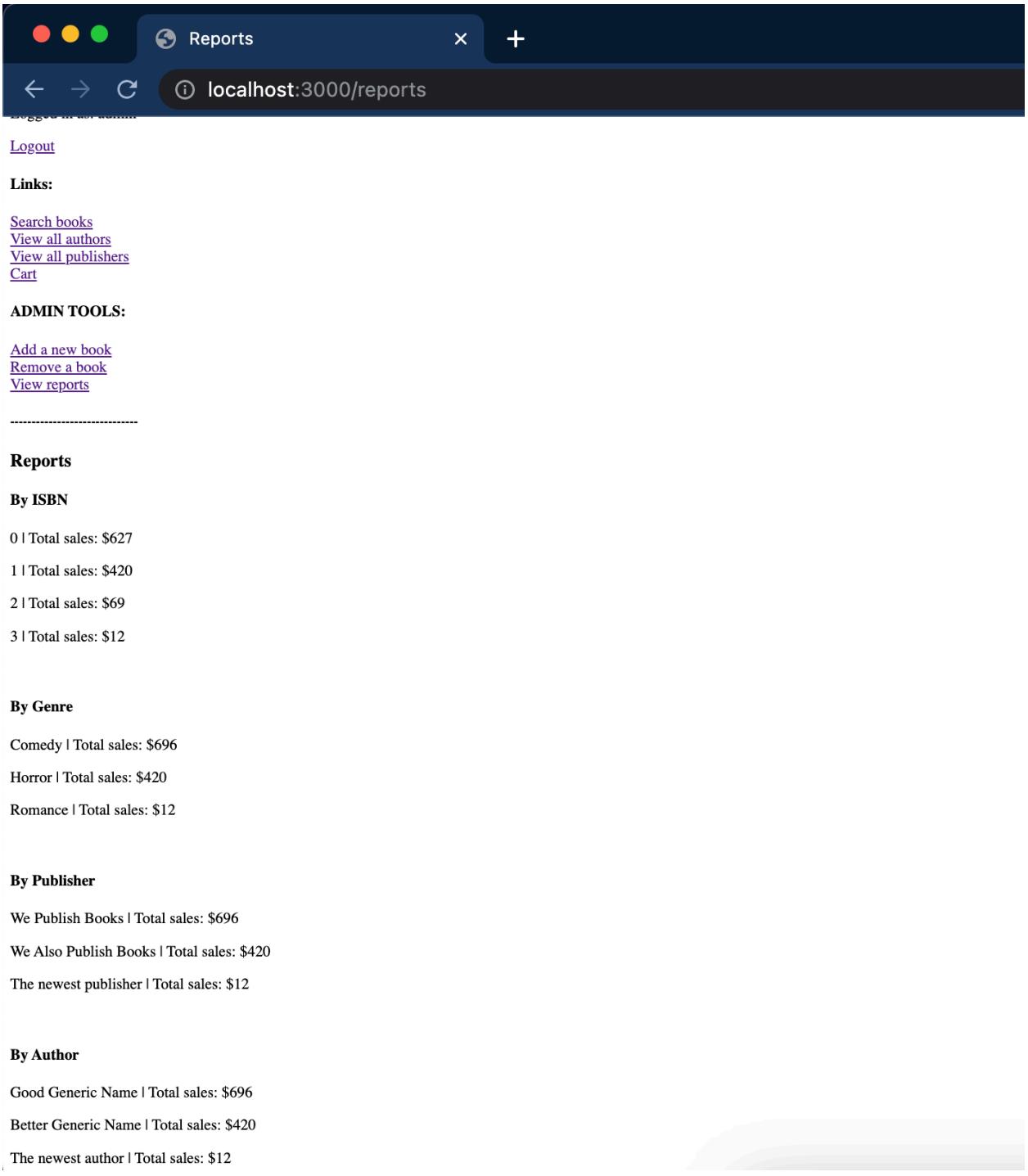
The admin can also add books, if they want to add a book with a known publisher or author, they just have to enter their pid and aid, respectively, and it will automatically add the book and not make new publisher and author and will connect it to preexiting authors and publishers. Or if all the details are filled out it will create a new publisher and author and connect it to them. *Screenshot zoomed out for formatting.

Remove book



Removing books is super easy and all you have to do is click the button to delete it from the db. It will not delete the author or publisher information. *Screenshot zoomed out for formatting.

Reports



The sales reports are all easily accessible from the /reports link. *screenshot zoomed out for formatting.

2.6 Bonus Features

1. Approximate & non-case sensitive searching: Can search by parts of the titles, authors and genres. All searches are not case sensitive.
2. Showing books by same author: Once viewing the book, you can click the link to view all books by that specific author.
3. GUI and redirects: The interactive website that will bring you to the pages you want to go to fast, e.g. once placing an order directing you to the order confirmation page.

2.7 GitHub Repository

https://github.com/seanmatute/comp3005project

2.8 Appendix I (Availability)

Any time after 11am on December 12th.

***

END OF REPORT.