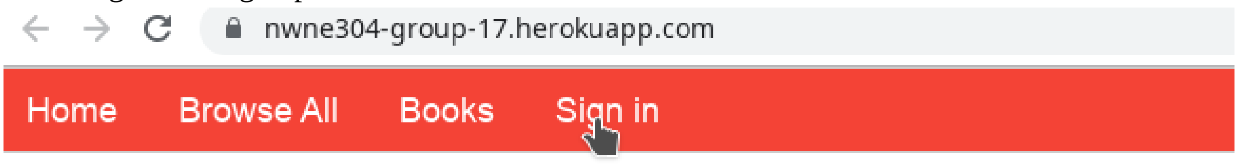
NWEN304 Final Project (Group 17)
Online Book Store
Link: https://nwne304-group-17.herokuapp.com

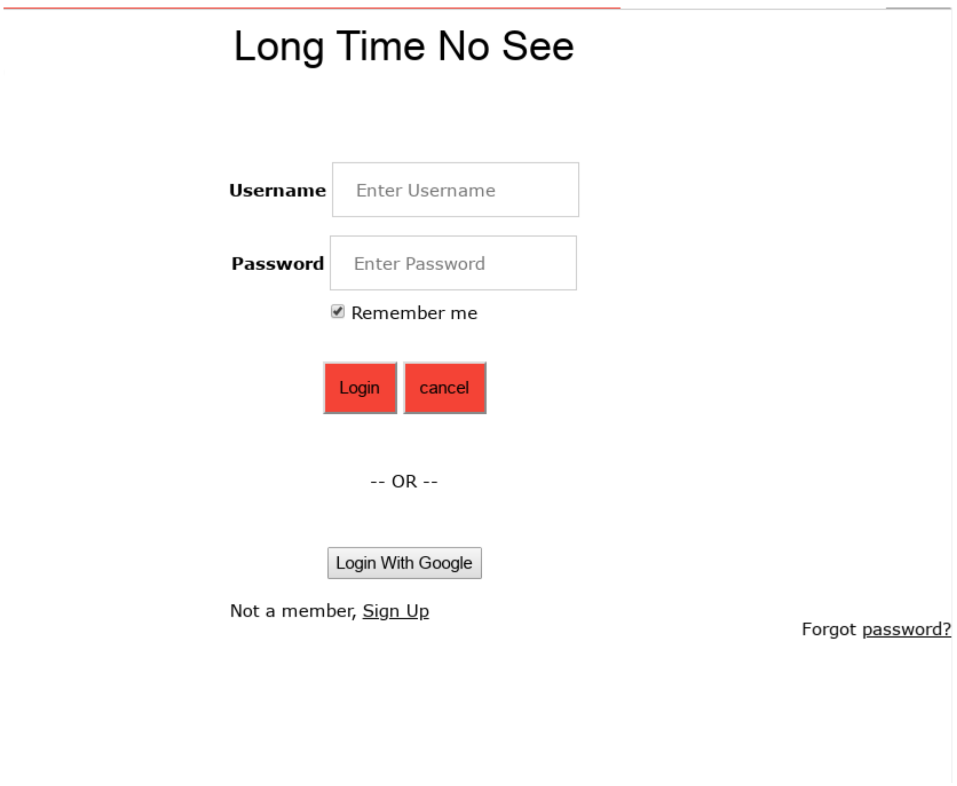
Zane Rawson 300367145
Huaiqu Cai 300434600

1. Using the system:

- Login and SignUp



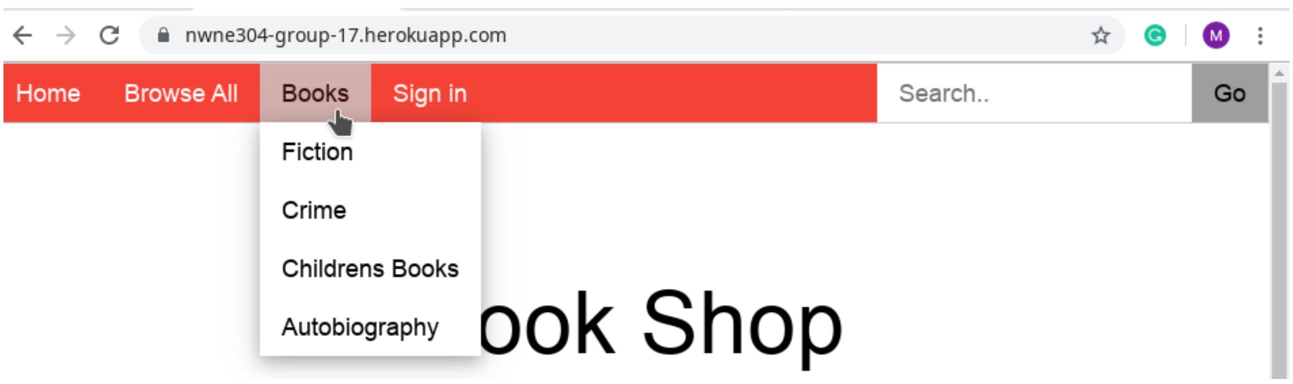
There is a Sign in button on the navigation bar, just click it to jump to the login page. The login page contains functions like directly login for a registered user, or users can login with their Google account. For a new user, it can simply jump to the sign up page from the login page. Same as the password reset. After login, the page will jump to the home page.


Long Time No See

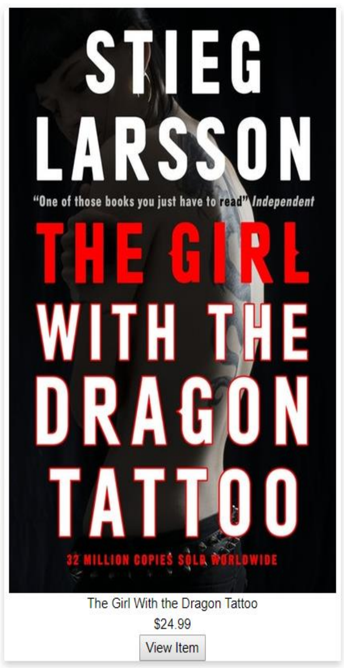
Username Enter Username
Password Enter Password
Remember me
Login cancel
-- OR --
Login With Google
Not a member, Sign Up
Forgot password?

- Recommend, Category and Search.

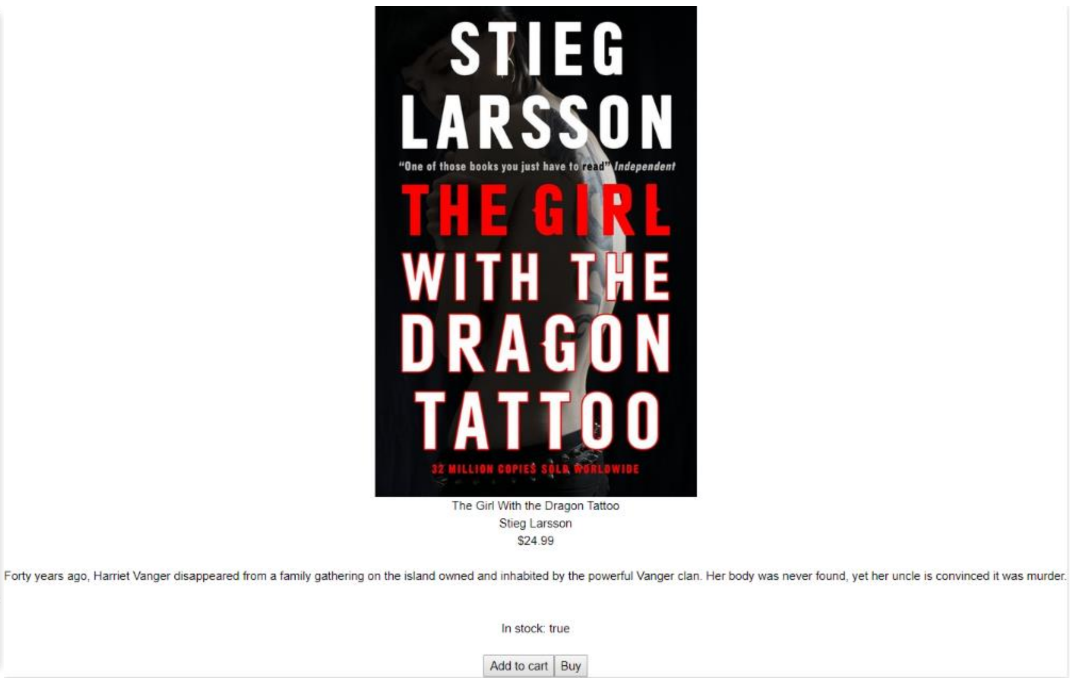
The home page will display the top three books of the sold rank. In the top navigation bar, users can go to the Browse All page to view all books or go to Books and choose one of the categories.



At the right hand side of the navigation bar, there is a Search Bar for user to search books. It will display all books with the key words input by user.



The Girl With the Dragon Tattoo
$24.99

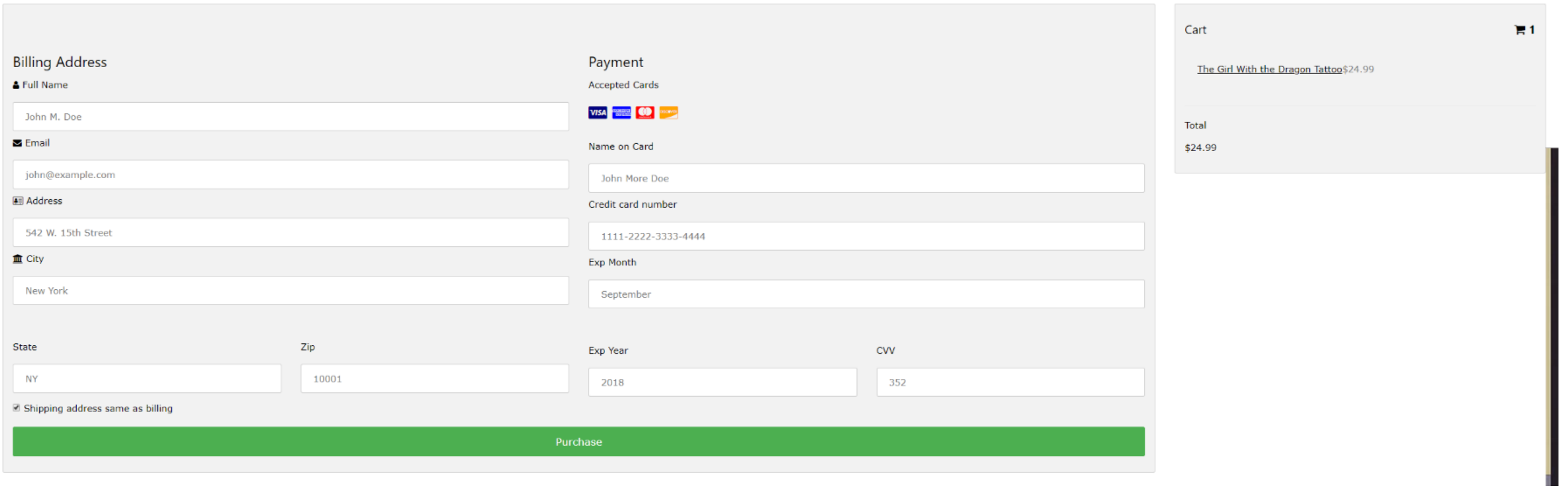


The Girl With the Dragon Tattoo
Stieg Larsson
$24.99

Forty years ago, Harriet Vanger disappeared from a family gathering on the island owned and inhabited by the powerful Vanger clan. Her body was never found, yet her uncle is convinced it was murder.

In stock: true

On any page where book information is displayed the user can view the item which will redirect them to a new page. This page displays more information about the book including the author and a description of the book. It also allows the user to either buy the book directly or add it to the cart. If the users is not signed in then they will be prompted to sign in. The cart page and the page where



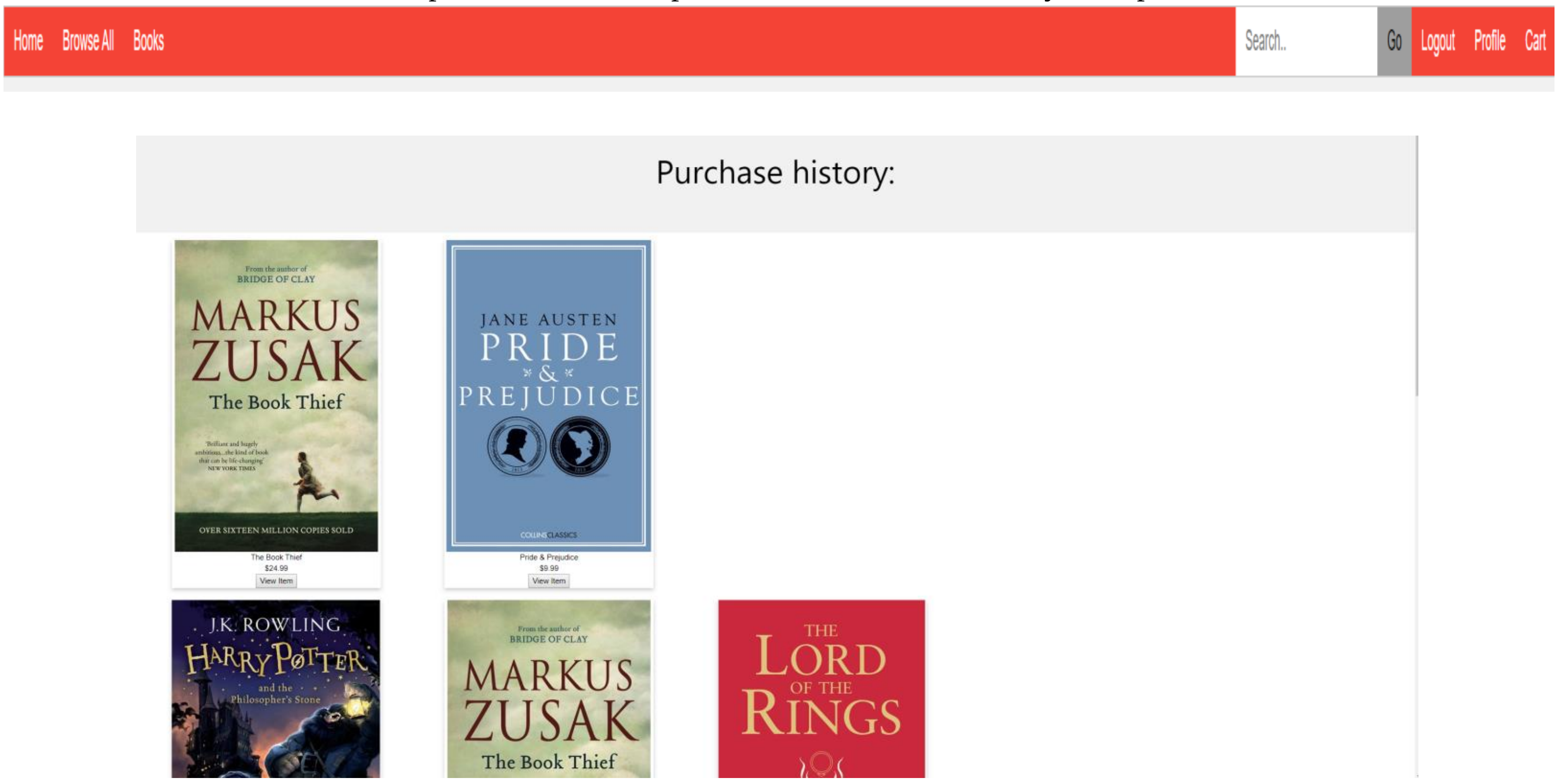
Billing Address
Full Name: John M. Doe
Email: john@example.com
Address: 542 W. 15th Street
City: New York
State: NY
Zip: 10001
☑ Shipping address same as billing

Payment
Accepted Cards
Name on Card: John More Doe
Credit card number: 1111-2222-3333-4444
Exp Month: September
Exp Year: 2018
CVV: 352

Cart 1
The Girl With the Dragon Tattoo $24.99
Total
$24.99

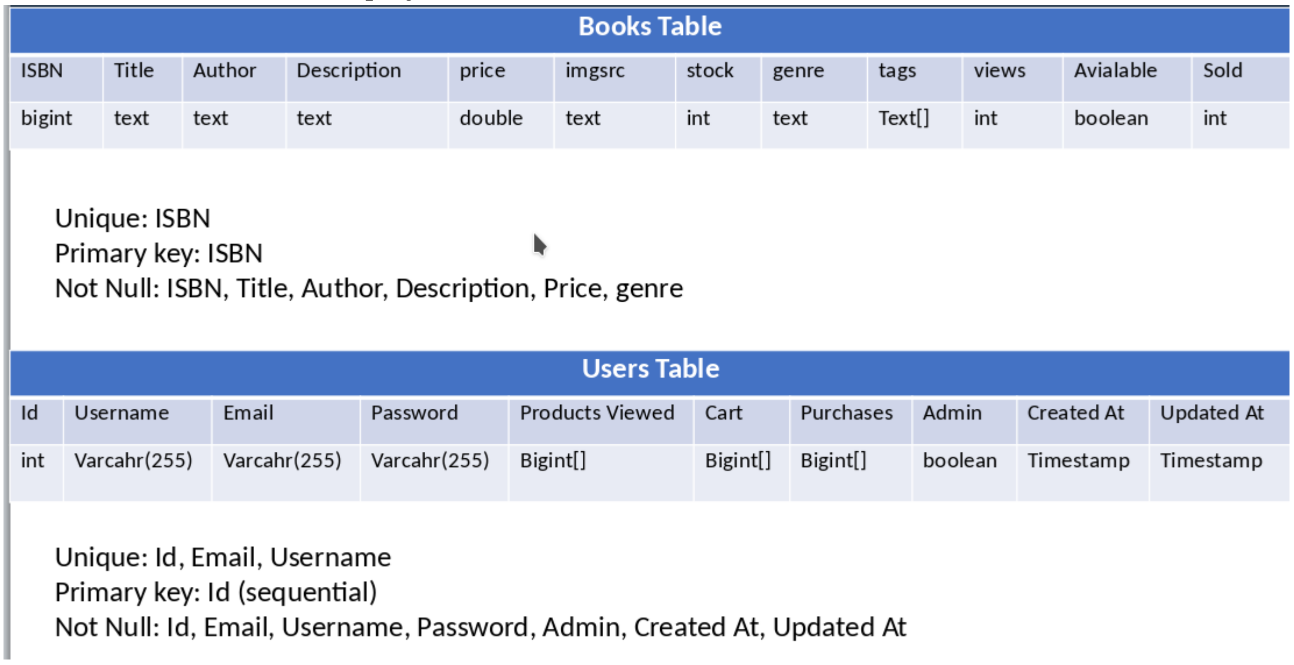
they can buy the book are pretty much identical but the cart page will display all of the items in the users cart. From this page they can purchase the book which will update the database and will redirect them to the store page. The cart page also gives the user the option to remove items from their cart.

The user also has the option to view their profile and the books that they have purchases.



2. RESTful interface

The database of this project has two tables, books and users.



Books Table

ISBN	Title	Author	Description	price	imgsrc	stock	genre	tags	views	Avialable	Sold
bigint	text	text	text	double	text	int	text	Text[]	int	boolean	int

Unique: ISBN
Primary key: ISBN
Not Null: ISBN, Title, Author, Description, Price, genre

Users Table

Id	Username	Email	Password	Products Viewed	Cart	Purchases	Admin	Created At	Updated At
int	Varcahr(255)	Varcahr(255)	Varcahr(255)	Bigint[]	Bigint[]	Bigint[]	boolean	Timestamp	Timestamp

Unique: Id, Email, Username
Primary key: Id (sequential)
Not Null: Id, Email, Username, Password, Admin, Created At, Updated At

The RESTful API we have got:
GET: ('/') ('/browse') ('/Product.html')('/book')('/recommend')('/getCart')('/logout')

POST: (‘/login’) (‘/signUp’)

PUT: (‘/passReset’) (‘/addToCart’) (‘/buyBook’) (‘/passReset’) (‘/forgetPass’)

DELETE: (‘/deleteCart’) (‘/removeFromCart’)

3. Error handing(Mars)

For most of functions of this application, it will simply catch the error as it happens and send the error to the console. For some functions like login, sign up, once users input unexpected words, there will be an alert to notice user that what is wrong and the page will stay there for user to input correct words.

4. List of CURL commands(Zane)

Book Function:
curl -i https://nwne304-group-17.herokuapp.com/book

This will return an array containing all of the books

Recommend Function:
curl -i https://nwne304-group-17.herokuapp.com/recommend

This will return an array containing the top three books based on the number sold

isSignedIn Function:
curl -i https://nwne304-group-17.herokuapp.com/isSignedIn

This will return the user information if they are signed in and 0 if they are not

GetCart Function:
curl -i https://nwne304-group-17.herokuapp.com/getCart

This will return an empty array since a user isn’t logged in.

Logout Function:
curl -i https://nwne304-group-17.herokuapp.com/logout

This will return ‘1’ if it executes successfully.

SignUp Function:
curl -d "username=admin1&email=admin1@admin.org&password=admin1" https://nwne304-group-17.herokuapp.com/signUp

This will return ‘Found, redirecting to login’ as it will have created the new item before checking if it exists.

If it fails it will return ‘redirecting to sign up’

Login Function:
curl -d "username=test&password=test" https://nwne304-group-17.herokuapp.com/login

This will return 1 if the details are correct and will return 0 if the password is incorrect or if the user does not exist.

addToCart Function:

curl -X PUT -d "isbn=9780857054036&username=test" https://nwne304-group-17.herokuapp.com/addToCart

this will return ‘added to cart’ if it succeeds.

addToPurchases Function:

curl -X PUT -d "isbn=9780857054036&username=test" https://nwne304-group-17.herokuapp.com/addToPurchases

this will return ‘Added to purchases’ if it succeeds.

buyBook Function:
curl -X PUT -d "isbn=9780857054036" https://nwne304-group-17.herokuapp.com/buyBook

This will return ‘successfully bought the book’ if it is successful

RemoveFromCart Function:
curl -X DELETE -d "isbn=9780857054036&username=test" https://nwne304-group-17.herokuapp.com/removeFromCart

this will return ‘Removed ISBN from the cart’ if it succeeds.

deleteCart Function:
curl -X DELETE -d "username=test" https://nwne304-group-17.herokuapp.com/deleteCart

this will return ‘The cart is now empty’ if it succeeds.

5. Authentication(Zane)
Our system offers the use of either a local sign in using bcrypt to hash the user’s password or by using their google account to sign in. With the local sign up the user’s password is only hashed on the server side using the base settings in Bcrypt for secure salted hashing. This prevents the user’s password being stored on the database in plain text. In the sign in part the user’s password is also sent to the server in plaintext and compared there.

Our system also allows for the user to reset the password through a page using their old password but also allows for a simple email chain if the users forgotten their password. The biggest downside of the password reset system is that once the user resets their password, they can no longer login as it does not hash their new password. This also present a privacy concern as the new password is stored in plain text.

When a user logs in a cookie is created containing the session id, In theory once this timer runs out(it should last 12 hours) the user should be prompted to sign in a gain however this hasn’t been tested

6. Database management(Zane)
For this project we used postgres sql as we weren’t performing any large queries, and this was the database system that we were most familiar with. Our database consists of two tables with one being for the books and another for the users.

The books table consists of the usual information that you could expect in a database storing books with the ISBN being the only unique field as well as being the primary key. We also have columns storing the imgsrc which contains the name of the image to be used when displaying the book and whether or not the book is available and if the number of books sold.

The users table contains the usual user information with the username, email and the password as well as arrays which contain the ISBN’s of the books the user has bought or added to their cart as well as if they are an admin or not. There are also columns containing information on when the user was created and when the row was last updated. These two columns were a by-product of creating this table using a data model in the user.js file.
The majority of the queries sent to the data base are fairly simple with a lot of them either being simple get methods with various conditions or update methods. The queries related to deleting items from the cart and adding to the cart are a little bit more complex as they use the functions array_remove and array_cat (e.g. "UPDATE users SET purchases=array_cat(purchases, ARRAY["+req.body.isbn+"]) WHERE username='"+req.body.username+"';"). These functions are a part of postgresql and made it very easy to work with the arrays in the database. The biggest downside of using these was that when removing a book from the users cart it will remove all instances of that book if they have more than one.

The code used to set up the database can be seen in the sql inserts text document.

7. Summary of the system(Mars)

This website works similar to most online shopping websites. Users could create an account, log in and buy books. We have many pages to display different contents like different category of books, or recommend books. For each user, there is a cart where stores the book they put. And in the page of each book, there is a buy button for user jump to the purchase page to order their books.