

Mid-Term Summary
CS-489 | Dr. Wang
March 12, 2023

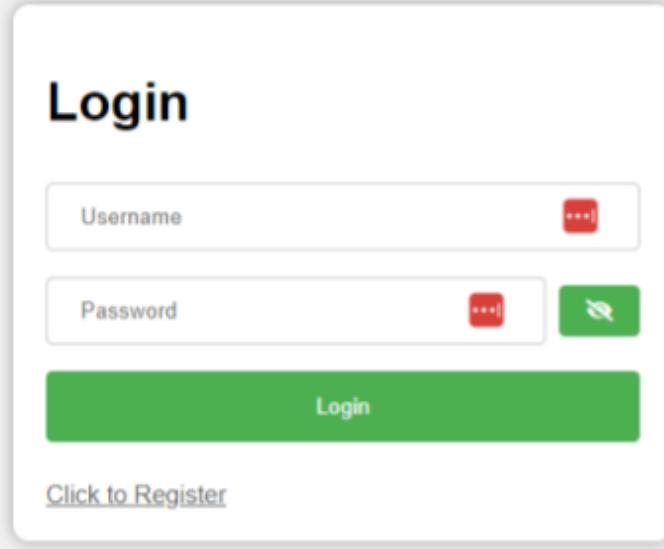
Jacob Wilson

Table of Contents

Table of Contents	1
Project Outline	2
Data	3
Collection of Data	3
Main Code	5
Database	5
Webpages	5
Conclusion	10
Brief Summary	10
Resources	10

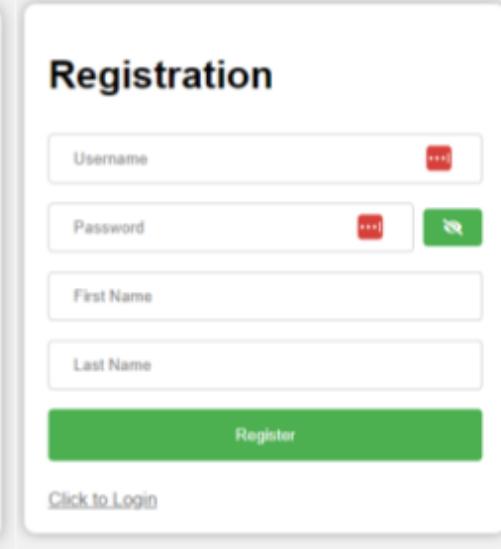
Project Outline

The project is an online bookstore where users can add books to a cart, search for books on the website itself, give reviews for books to show other users what is a good read, allow users to create a wishlist of books they want, and a book list of books they own. This site is coded using bootstrap css, custom Css, Html, Javascript with Jquery, php, and mysql. PhP is used for communicating the queries and transforming that data from the server into data that can be used by the client. This was all completed from January 27, 2023, to March 12, 2023. Below are four screenshots of the main portions of the website that have been finished. Mainly throughout this term the back-end has been worked on the most as it is the most vital part to the website.



Login

This screenshot shows the login interface. It features two input fields: 'Username' and 'Password', each with a red 'forgot password?' link and a green eye icon for password visibility. Below the inputs is a large green 'Login' button. At the bottom left, there is a link 'Click to Register'.



Registration

This screenshot shows the registration interface. It includes four input fields: 'Username', 'Password', 'First Name', and 'Last Name'. Each field has a red 'forgot password?' link and a green eye icon. To the right of the 'Password' field is a green checkmark icon. Below the fields is a green 'Register' button. At the bottom left, there is a link 'Click to Login'.



The Store of Knowledge

Welcome General User

The main website features a grand, ornate library interior with tall bookshelves and a curved balcony. The header includes the site name 'The Store of Knowledge', a navigation bar with links for Home, About, User, and a search bar with a dropdown menu. Below the header, there is a row of four book covers: 'The Martian Chronicles' by Ray Bradbury, 'The God of Small Things' by Arundhati Roy, 'Their Eyes Were Watching God' by Zora Neale Hurston, and 'Perfume' by Patrick Süskind.

Image	Title	Author	Quantity	
	Das Perfum	Patrick Suskind	2	
	Their eyes were watching God	Zora Neale Hurston	1	
	The Jungle	Upton Sinclair	1	
Total:			4	

Data

Collection of Data

The collection of data began after the topic had been decided. The data is taken from <https://openlibrary.org/> a site that is connected to the Internet Archive and house a collection of book data a short list of books was taken and stored inside my own mysql databases in a table called bookList.

Below in Figure 1.1 is an example of how the data is presented in the database table bookList for a book.

record_number	Work_ID	Title	Authors	Purchased_Year	Edition_ID	Rating	Total
1	OL103139W	The Martian Chronicles	Ray Bradbury	1950	OL782470M	0.00	0
2	OL1052138W	The God of Small Things	Joseph Michael	1997	OL31390125M	0.00	0
3	OL10632207W	Their eyes were watching God	Zora Neale Hurston	1996	OL24243977M	0.00	0
4	OL10839W	Das Perfum	Patrick Suskind	1995	OL7848045M	0.00	0
5	OL110971W	Fictiones	Jorge Luis Borges	1951	OL7874550M	0.00	0
6	OL1140109W	The Good Earth	Pearl S. Buck	1931	OL6936001M	0.00	0
7	OL1142292W	The Waste Land	T. S. Eliot	1922	OL6792110M	0.00	0
8	OL114947W	The Jungle	Upton Sinclair	1906	OL4791114M	0.00	0
9	OL116250W	Holes (with "Connections") HNN Library (HNN library)	Louis Sachar	1990	OL7511332M	0.00	0
10	OL1169037W	Animal Farm	George Orwell	1945	OL75717244M	0.00	0
11	OL116933W	Nineteen Eighty-Four	George Orwell	1949	OL2422444M	0.00	0
12	OL1182248W	Paper towns	John Green	2008	OL23924590M	0.00	0
13	OL1190389W	A la recherche du temps perdu	Maurice Proust	1924	OL7941630M	0.00	0
14	OL11914W	The Secret Garden	Frances Hove	1911	OL7490468M	0.00	0
15	OL11929304W	Evermore	Alyson Noel	2009	OL22800975M	0.00	0
16	OL119436W	The Amazing Adventures of Kavalier & Clay	Michael Chabon	1997	OL3787203M	0.00	0
17	OL11947025W	Shaves	Maggie Stiefvater	2009	OL2318070M	0.00	0
18	OL120928W	Im Herzen nichts Neues	Erich Maria Remarque	1928	OL7546371M	0.00	0
19	OL12260912W	... Trotzdem Ja zum Leben sagen	Viktor E. Frankl	1946	OL7944581M	0.00	0
20	OL12791W	My Antonia	Willie Cather	1913	OL30342439M	0.00	0
21	OL13127424W	Black Beauty	Anna Sewell	2001	OL23879403M	0.00	0
22	OL1346931W	Lonely Dove	Larry McMurtry	1985	OL7662107M	0.00	0
23	OL13714952W	Fantastic Beasts and Where to Find Them	J. K. Rowling	2001	OL7511850M	0.00	0
24	OL1389021W	The Once and Future King	T. H. White	1939	OL7524366M	0.00	0
25	OL14372W	Elle Enchanted	Gail Carson Levine	1997	OL4240464M	0.00	0
26	OL14659722W	A prayer for Owen Meany	John Irving	1989	OL4445160M	0.00	0
27	OL14664437W	The heart is a lonely hunter	Carson McCullers	1940	OL3942134M	0.00	0
28	OL14872525W	Pippi Langstrumpf	Astrid Lindgren	1945	OL7360664M	0.00	0
29	OL14933414W	The Fellowship of the Ring	J.R.R. Tolkien	1954	OL6888866M	0.00	0
30	OL14937649W	Walk Two Moons	Sharon Creech	1994	OL235468M	0.00	0

Figure 1.1

In Figure 1.2 you can see what each column name is which indicates the type of data it holds. In Figure 1.3 we can see the API documentation page of

Field	Type	Null	Key	Default	Extra
_record_number	int(11)	NO	PRI	NULL	auto_increment
Work_ID	varchar(50)	YES		NULL	
Title	varchar(500)	YES		NULL	
Authors	varchar(500)	YES		NULL	
First_Publish_Year	varchar(10)	YES		NULL	
Edition_ID	varchar(50)	YES		NULL	
Rating	decimal(4,2)	YES		NULL	
Total	int(11)	YES		NULL	

8 rows in set (0.01 sec)

Figure 1.2

Last edited by [raybb](#)
January 21, 2023 | [History](#) [Edit](#)

[Developer Center \(Home\)](#) | [Web API](#) | [Client Library](#) | [Data Dumps](#) | [Source Code](#) | [Report an Issue](#) | [Licensing](#)

Open Library offers a suite of APIs to help developers get up and running with our data. This includes [RESTful APIs](#), which make Open Library data available in JSON, YAML and RDF/XML formats. There's also an earlier, now deprecated [JSON API](#) which is preserved for backward compatibility.

- [Books API](#) - Retrieve a specific work or edition by identifier
- [Authors API](#) - Retrieve an author and their works by author identifier
- [Subjects API](#) - Fetch books by subject name
- [Search API](#) - Search results for books, authors, and more
- [Search inside API](#) - Search for matching text within millions of books
- [Partner API](#) - Formerly the "Read" API, fetch one or more books by library identifiers (ISBNs, OCLC, LCCNs)
- [Covers API](#) - Fetch book covers by ISBN or Open Library identifier
- [Recent Changes API](#) - Programmatic access to changes across Open Library

Bulk Access

Please do not use our APIs for bulk download of Open Library data because this affects our ability to serve patrons. We make our data [publicly available](#) each month for partners. If you want a dump of complete data, please read about your [Bulk Download](#) options, or email us at openlibrary@archive.org.

More APIs

Did you know, nearly every page on Open Library is or has an API. You can return structured bibliographic data for any page by adding .rdf/.json/.yml extension to the end of any Open Library identifier. For instance: <https://openlibrary.org/works/OL15626917W.json> or <https://openlibrary.org/authors/OL33421A.json>. Many pages, such as the Books, Authors, and Lists, will include links to their RDF and JSON formats.

Questions

We encourage developers to ask questions by opening issues on [GitHub](#) and on our [gitter](#) chat channel.

Figure 1.3

Main Code

Database

The Database was created with 3 initial tables: bookList, cart, and users. Each table holds a different responsibility within the site. The bookList table which was shown earlier in Figure 1.1 and 1.2 holds all the current list of books that have been added to the database. The cart table which can be seen in Figure 2.1 holds all the books that each user has added to their carts. Finally the users table holds all relevant data pertaining to each site user this can be seen in Figure 2.2.

Field	Type	Null	Key	Default	Extra
username	varchar(50)	NO		NULL	
edition_id	varchar(50)	NO		NULL	
title	varchar(500)	NO		NULL	
authors	varchar(500)	NO		NULL	
quantity	int(11)	NO		NULL	
created_at	timestamp	NO		CURRENT_TIMESTAMP	
updated_at	timestamp	NO		CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP

7 rows in set (0.00 sec)

Figure 2.1

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
username	varchar(50)	NO		NULL	
password	varchar(50)	NO		NULL	
firstName	varchar(50)	NO		NULL	
lastName	varchar(50)	NO		NULL	
type	varchar(50)	NO		NULL	
loginAttempts	int(11)	YES		NULL	
lastLoginTime	timestamp	NO		CURRENT_TIMESTAMP	
created_at	timestamp	NO		CURRENT_TIMESTAMP	
updated_at	timestamp	NO		CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP

10 rows in set (0.00 sec)

Figure 2.2

Webpages

The main code for the website revolves around HTML. HTML is used to create headers, navigation bars, containers, and elements on each of the webpages. Some examples of this are as follows. Below in Figure 2.3 is the code for the base login page that the users will see as the first time they go to the site.

```

<div class="form-login">
    <form class="form" method="post" name="login">
        <h1 class="login-title">Login</h1>
        <input type="text" class="login-input" name="username"
placeholder="Username" autofocus="true"/>
        <div class="password-container">
            <input type="password" class="login-input" name="password"
placeholder="Password" id="password-field"/>
            <button id="show-password-btn" class="show-password-btn"><i class="fas
fa-eye-slash"></i></button>
        </div>
        <input type="submit" value="Login" name="submit" class="login-button"/>
        <p class="link"><a href="registration.php">Click to Register</a></p>
    </form>
</div>

```

Figure 2.3

This code translates to Figure 2.4 on the user's end when visiting the webpage

The form is contained within a light gray rounded rectangle. At the top center, the word "Login" is displayed in a large, bold, black font. Below it are two horizontal input fields. The first input field has the placeholder "Username" and a small red square button with three white dots to its right. The second input field has the placeholder "Password" and a similar red square button. To the right of the password field is a small green square button with a white eye icon. Below these fields is a large green rectangular button with the word "Login" in white. At the very bottom of the form, there is a link "Click to Register" in a smaller, gray font.

Figure 2.4

The next main portion of HTML is the navigation bar for the website. Figure 2.5 shows a screenshot of the navigation bar. This code allows for the user to easily navigate the web pages within the website. This code has dropdown menus to avoid cluttering the UI with too many buttons all visible at once. The “hamburger menu” checkbox is used when the user is on mobile. This is done through CSS.

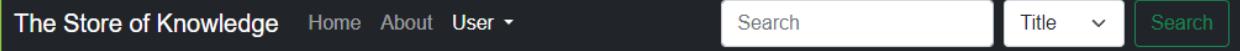


Figure 2.5

On the main page of the website there is a hero section that includes the title of the project and a welcome message to the user with their first and last names entered in the registration process. A screenshot of the hero section of the site is included in Figure 2.6.

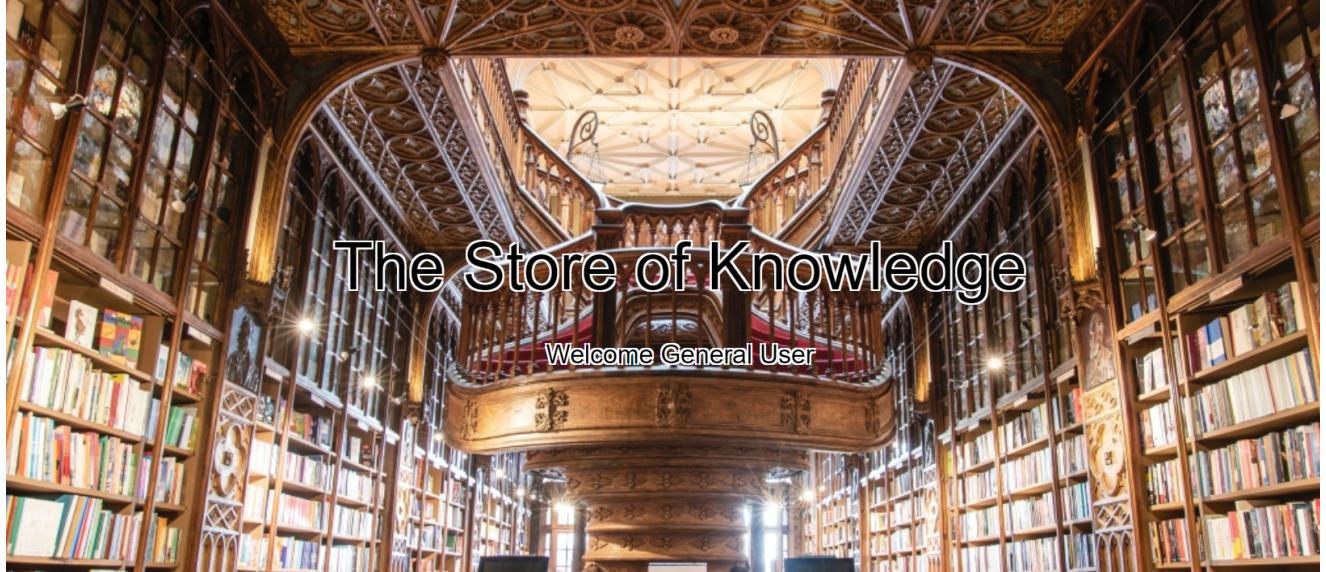
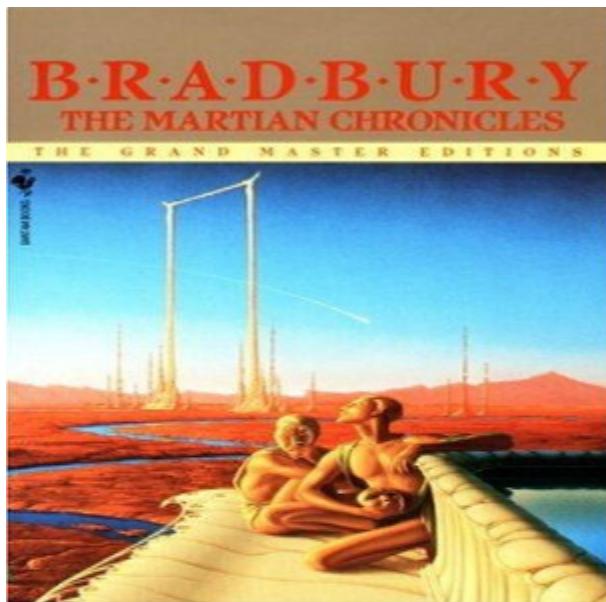


Figure 2.6

On the main page of the website the top 16 books on the site are displayed in a grid format inside bootstrap cards. They each look like the screenshot in Figure 2.7.



The Martian Chronicles by Ray
Bradbury



Add To Cart

Figure 2.7

Clicking the “Add To Cart” on one of the main pages book cards it will add the book displayed on that card to the user’s cart. This is done through the use of php and forms with the post method the code that shows the form and php is displayed in figure 2.8.

```

<form method='post'>
    <input type='hidden' value='$olid_id' name='input_name'>
    <button type='submit' class='btn btn-primary mt-auto'
name='bookBtn'>Add To Cart</button>
</form>
<?php
if (isset($_POST['bookBtn'])) {
    $olid_id = $_POST['input_name'];
    $query_info = "SELECT * FROM bookList WHERE Edition_ID='$olid_id'";
    $result_info = mysqli_query($con, $query_info);
    $info = array();
    while( $row = mysqli_fetch_assoc($result_info)) {
        $info[] = $row;
    }
    $title = $info[0]["Title"];
    $author = $info[0]["Authors"];
    $query_val = "SELECT quantity FROM `cart` WHERE username='$username'
AND edition_id='$olid_id'";
    $result_val = mysqli_query($con, $query_val) or die(mysql_error());
    $quantity = 1;
    if (mysqli_num_rows($result_val) > 0) {
        $quantitys = array();
        while($row = mysqli_fetch_assoc($result_val)){
            $quantitys[] = $row;
        }
        $quatnity = $quantitys[0]['quantity'];
        $quantity = (int)$quantity;
        $quantity++;
        $cart_q = "UPDATE cart SET quantity='$quantity' WHERE
username='$username' AND edition_id='$olid_id'";
    } else {
        $cart_q = "INSERT INTO cart (username, edition_id, title, authors,
quantity) VALUES ('$username', '$olid_id', '$title', '$author',
'$quantity')";
    }
    $cart_r = mysqli_query($con, $cart_q) or die(mysql_error());
    unset($_POST['bookBtn']);
    header("Location: cart.php");
}
?>
```

Figure 2.8

The next page of this site is the cart page of the site that shows all the books that have been added to the cart. A screenshot of the cart page is shown below in Figure 2.9

Image	Title	Author	Quantity	
	Das Parfum	Patrick Suskind	2	<input type="button" value="▲"/> <input type="button" value="▼"/> <input type="button" value="X"/>
Total:			2	<input type="button" value="Checkout"/>

Figure 2.9

On this page there are 3 buttons that each book will have where they can increase or decrease the quantity of that specific book in their cart and the last button where it removes all of that book from their cart. There is also a button called Checkout on the page that when clicked it will try to add all of the books in the user's cart to an amazon cart so that this site can be more useful for actually finding books to buy. The code for this button is attached below in Figure 2.10 also below is the php code for the 3 button mentioned earlier in Figure 2.11

```
<form method="GET" action="https://www.amazon.com/gp/aws/cart/add.html"
target="_blank">
    <input type="hidden" name="AssociateTag" value="Associate Tag" />
    <?php
        for ($i = 0; $i < count($content); $i++) {
            $cartValue = $i + 1;
            $cartBookOlidID = $content[$i]["edition_id"];
            $cartBookQuantity = $content[$i]["quantity"];
            $cartBookApi =
"https://openlibrary.org/books/".$cartBookOlidID.".json";
            $cartBookJson = file_get_contents($cartBookApi);
            $cartBookJsonData = json_decode($cartBookJson,true);
            $cartBookIsbn = $cartBookJsonData["isbn_10"][0];
            echo "<input type='text' class='hidden' name='ASIN.$cartValue'
value='".$cartBookIsbn.'>";
            echo "<input type='text' class='hidden'
name='Quantity.$cartValue' value='".$cartBookQuantity.'>";
        }
    ?>
    <input type="submit" class="btn btn-primary w-100" name="add"
value="Checkout" />
</form>
```

Figure 2.10

```

if (isset($_POST['up'])) {
    $solid_id = $_POST['input_name'];
    $username = $_SESSION["username"];
    $up_q = "UPDATE cart set quantity=quantity+1 WHERE username='$username' AND
edition_id='$solid_id'";
    $up_r = mysqli_query($con, $up_q);
    header("Location: cart.php");
}

if (isset($_POST['down'])) {
    $solid_id = $_POST['input_name'];
    $username = $_SESSION["username"];
    $down_q = "UPDATE cart set quantity=quantity-1 WHERE username='$username' AND
edition_id='$solid_id'";
    $down_r = mysqli_query($con, $down_q);
    header("Location: cart.php");
}

if (isset($_POST['del'])) {
    $solid_id = $_POST['input_name'];
    $username = $_SESSION["username"];
    $del_q = "DELETE FROM cart WHERE username='$username' AND
edition_id='$solid_id'";
    $del_r = mysqli_query($con, $del_q);
    header("Location: cart.php");
}

```

Figure 2.11

Conclusion

Brief Summary

To conclude, the majority of the project so far has been both front-end and back-end development. I have developed the look and feel of each page while working on the back-end side of each page to make the designing aspect easier. The toughest part of this project so far has been the profile editor page because of the image upload portion to the point where I had to just remove that feature from my site.

Resources

- <https://openlibrary.org/developers/api>
- <https://www.php.net/manual/en/book.session.php>
- <https://getbootstrap.com/>
- <https://jquery.com/>
- <https://www.w3schools.com/>
- <https://stackoverflow.com/>