Web Development 2 Assignment

List of all PHP pages

Login.php
Registration.php
Search.php
Category.php
BookTitle_Author.php
Display_Category.php
Show_BookTitle_Author.php
Connect.php
Reserve.php
Show_Reserve.php
Unreserve.php
Logout.php

Login.php

Login.php is the login page of the website. This is always the first page the user sees. It is where the user must sign in to the website with their username and password to use the website. If they have not signed in and they try to access any of the other pages except for Registration.php they are sent to this page to sign in. This page makes sure that the username and password the user has entered matches a username and password in the database. If it does not, it will output the message Incorrect Password or Username on screen. If it does match a username and password the user will be logged in, it will set their session and take the user to Search.php. If they have not created an account before there is a register button which will send them to Registeration.php so they can register their details.

Login.php only interacts with three pages which are Search.php, Registeration.php and Connect.php. For Search.php it interacts with it by sending the user to it when they enter a correct username and password. For Registeration.php, it interacts with it through the register button by sending the user to it when they click the button so they can register their details. It also interacts with Connect.php to connect to the database.

Registration.php

Registeration.php is the register page for the website. It is where the user goes to register their details if they have not done so before. When they click register, it validates that all details are entered and that all are entered correctly. The mobile number must be 10 digits long, the password must be 6 characters in length, the password and confirm password must match and finally the username must be unique. If all details that the user entered are correct the system will take this information and put it into the database and then send the user back to the login page. On Registeration.php there is also a button to go back to the login page.

Registeration.php only interacts with two pages which is Login.php and Connect.php. It interacts with it when all details the user entered are correct by sending the user back to it and also when the user clicks the button to go back to the login page. It also interacts with Connect.php to connect to the database.

Search.php

Search.php is the homepage of my website. Here the user can see the two different ways they can search for books which are by category or by book title and/or author. This page allows the user to decide which way they want to search for a book. There are two buttons, one for search by category and one for search by book title and/or author. Depending on which one they click it will take the user to that search's page. For search by category, it takes you to Category.php and for search by book title and/or author it takes you to BookTitle_Author.php.

Search.php interacts with many pages. It interacts with Category.php and BookTitle_Author.php. As above, it interacts with them through their certain buttons which takes the user to those pages. It also interacts with Login.php at the beginning of the page by checking if the user is logged by confirming the session is set and if not sending them to Login.php to log in. As well, it also interacts with itself, Show_Reserve.php and Logout.php. These are all in the navbar so many of the pages I talk about will interact with these pages. In the navbar there is a search button which brings the user to search.php, a reserved books button which brings the user to Show_Reserve.php and passes the variable position to it and a logout button which brings the user to Logout.php. It also interacts with Connect.php to connect to the database.

Category.php

Category.php is where the user can select what category they want to search for a book in through a dropdown menu. All of these categories are retrieved from the database. Once the user selects a category and clicks search, the page will send the user to Display_Category.php. On Category.php page there is also a button which allows the user to switch to search by book title and/or author.

Category.php interacts with many pages. It interacts with Display_Category.php and BookTitle_Author.php. As above, it interacts with Display_Category.php when the user selects a category and clicks search, the user is sent to that page. It passes the category the user selected and the variable position to Display_Category.php. For BookTitle_Author.php it interacts with it through the button to switch search by book title and/or author. Again, exactly like Search.php it also interacts with Search.php, Show_Reserve.php (passes the variable position to it) and Logout.php through the navbar, Connect.php to connect to the database and Login.php at the beginning to check if the user is logged in.

BookTitle_Author.php

BookTitle_Author.php is where the user can search for books by book title and/or author. The user can search with both entered, just one entered or none. When the user clicks search, the user is sent to Show_BookTitle_Author.php. On this page, there is also a button that allows the user to switch to search by category.

BookTitle_Author.php interacts with many pages. It interacts with Show_BookTitle_Author.php and Category.php. As above, it interacts with Show_BookTitle_Author.php when the user clicks search, the user is sent to that page. It passes the book title and author the user entered and the variable position to Display_Category.php. For Category.php it interacts with it through the button to switch to search by category. Again, like the others it also interacts with Search.php, Show_Reserve.php (passes the variable position to it) and Logout.php through the navbar, Connect.php to connect to the database and Login.php at the beginning to check if the user is logged in.

Display_Category.php

Display_Category.php displays all the books that are in the category the user selected. It displays only 5 books on the screen at a time and displays them in table format. There is a next page and previous page button so the user can traverse between the pages if there are more than 5 books to display. On this page the user can also click the reserve button beside the book to reserve the book. There is also a button that allows the user to go back to search by category.

Display_Category interacts with many pages. It interacts with Category.php and Reserve.php. It interacts with Category.php through getting the category the user selected and the position variable from it when you come from that page and it interacts with it through the go back to search button that when the user clicks it will send the user to Category.php. It interacts with Reserve.php when you click the reserve button beside a book it sends the user to Reserve.php and passes through the ISBN of that book to it. Again, exactly like the other pages it also interacts with Search.php, Show_Reserve.php (passes the variable position to it) and Logout.php through the navbar, Connect.php to connect to the database and Login.php at the beginning to check if the user is logged in.

Show_BookTitle_Author.php

Show_BookTitle_Author.php displays all the books that match or are similar to what the user entered. It will first look for an exact match to what the user entered and if it can not find an exact match it will do a partial search. The partial search will see if what the user entered is similar to another book and/or author in the database. If the user enters no book title and author all the books in the database will be displayed. It displays only 5 books on the screen at a time and displays them in table format. There is a next page and previous page button so the user can traverse between the pages if there are more than 5 books to display. On this page the user can also click the reserve button beside the book to reserve the book. There is also a button that allows the user to go back to search by book title and/or author.

Show_BookTitle_Author.php interacts with many pages. It interacts with BookTitle_Author.php and Reserve.php. It interacts with BookTitle_Author.php through getting the book title and author the user entered and the position variable from it and through the go back to search button that when the user clicks it will send the user to BookTitle_Author.php. It interacts with Reserve.php when you click the reserve button beside a book it sends the user to Reserve.php and passes through the ISBN of that book to it. Again, exactly like the other pages it also interacts with Search.php, Show_Reserve.php (passes the variable position to it) and Logout.php through the navbar, Connect.php to connect to the database and Login.php at the beginning to check if the user is logged in.

Connect.php

Connect.php is used to connect to the database. On any page that this needs to connect to the databse connect.php is called. By connecting to the database the page is able to access the data in the database and run SQL statements. The user will not see this page it will just be called by the page to connect to the database.

It does not interact with any other pages itself but every page except for Logout.php interacts with it so they can connect to the database.

Reserve.php

Reserve.php is used to reserve a book in the database. The user is sent to this page when they click the reserve button on any book. The user will not see this page but will pass through it as once it completes reserving the book it sends the user to Show_Reserve.php. The only time the user may see the page is if it failed to reserve the book through one of the SQL statements failing.

It interacts with many pages. It interacts with Show_Reserve.php. As I said above, it interacts with it by sending the user to it when it completes reserving the book. Again, exactly like the other pages it also interacts with Search.php, Show_Reserve.php (passes the variable position to it) and Logout.php through the navbar, Connect.php to connect to the database and Login.php at the beginning to check if the user is logged in. However, the only time the navbar will be seen is when it fails to reserve the book.

Show_Reserve.php

Show_Reserve.php displays all the books the user has reserved. It displays only 5 books on the screen at a time and displays them in table format. There is a next page and previous page button so the user can traverse between the pages if there are more than 5 books to display. On this page the user can also click the unreserve button beside the book to unreserve the book. There is also a button that allows the user to go back to search.

Show_Reserve.php interacts with many pages. It interacts with Search.php and Reserve.php. It interacts with Search.php when you click the go back to search button it sends the user to Search.php and it interacts with Reserve.php when you click the unreserve button beside a book it sends the user to Unreserve.php and passes through the ISBN of that book to it. Again, exactly like the other pages it also interacts with Search.php, Show_Reserve.php (passes the variable position to it) and Logout.php through the navbar, Connect.php to connect to the database and Login.php at the beginning to check if the user is logged in.

Unreserve.php

Unreserve.php is used to unreserve a book that the user wants to unreserve. The user is sent to this page when they click the unreserve button on any book that they have reserved. The user will not see this page but will pass through it as once it has unreserved the book it sends the user to Show_Reserve.php. The only time the user may see the page is if it failed to unreserve the book through one of the SQL statements failing.

It interacts with many pages. It interacts with Show_Reserve.php. As I said above, it interacts with it by sending the user to it when it has unreserved the book. Again, exactly like the other pages it also interacts with Search.php, Show_Reserve.php (passes the variable position to it) and Logout.php through the navbar, Connect.php to connect to the database and Login.php at the beginning to check if the user is logged in. However, the only time the navbar will be seen is when it fails to unreserve the book.

Logout.php

Logout.php logs the user out of their session/account. When the logout button in the navbar is clicked it will run. The user will never see this page but pass through it very quickly as all it does is destroy their session and immediately sends them back to the login page.

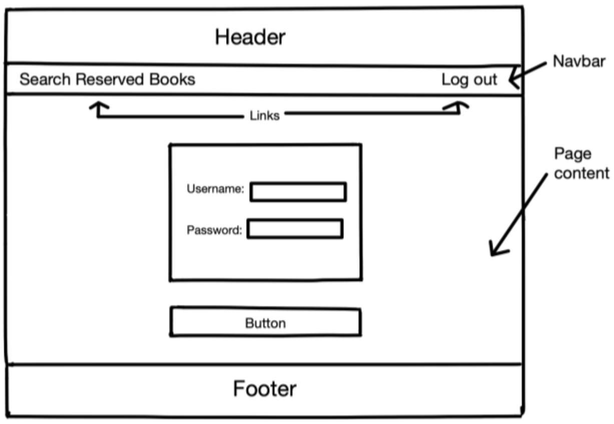
It does not interact with any other pages itself but most pages are able to interact with Logout.php through the logout button in the navbar.

Design Document

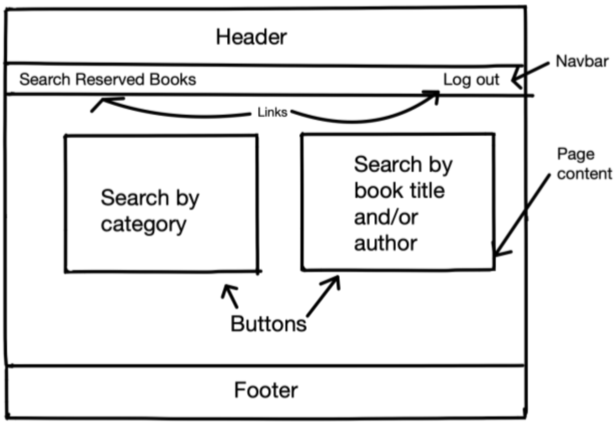
Planning

Sketches

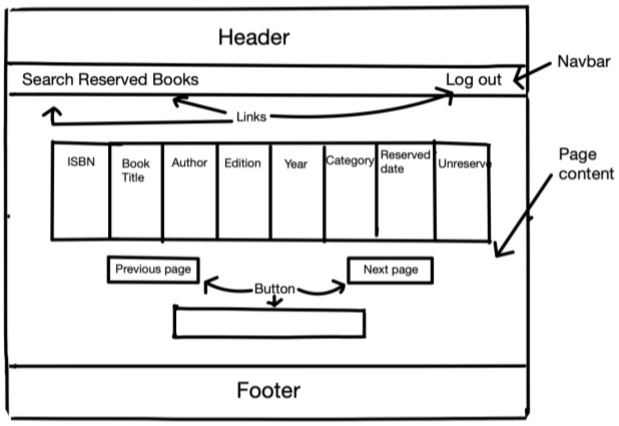
Sketch 1



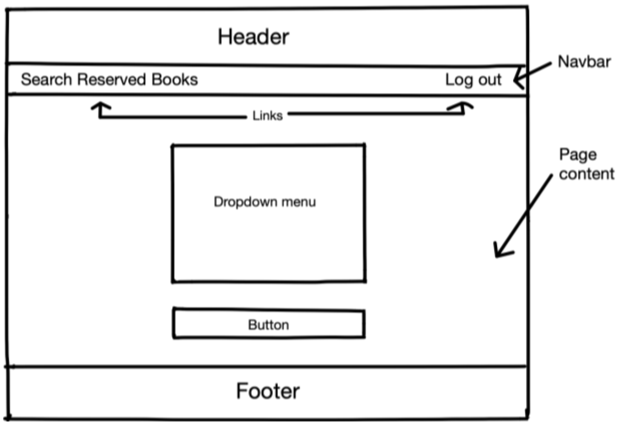
Sketch 2



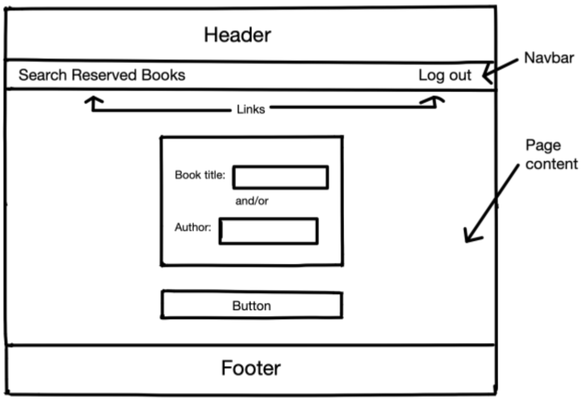
Sketch 3



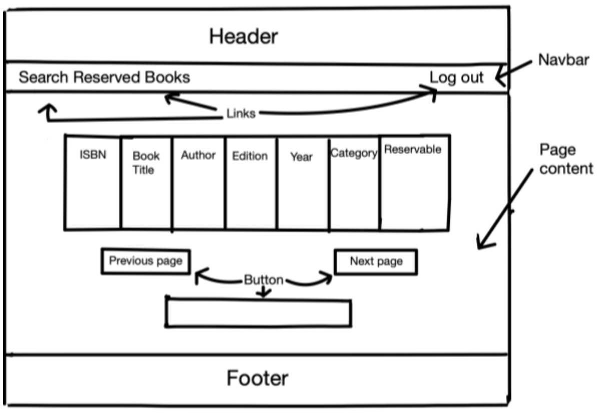
Sketch 4



Sketch 5



Sketch 6



Plan for design - Objectives

Login page
- User enters username and password
- Make sure they can’t leave them blank
- Check if username and password that user entered matches a username and password combination in the database – if not display error
-If it matches set the session and send the user to the homepage
-Register button that allows the user to go to the register page
- Look like sketch 1

Registration Page
- Gets to here from the login page
- Allows user to register details
- Details include: Username, Password, Confirm Password, First Name, Last Name, Address Line 1, Address Line 2, City, Telephone, Mobile Number
- Username has to be unique
- Password and confirm password need to be the same
- Password must be 6 characters in length
- Mobile number must be digits and 10 digits in length
- Nothing can be left blank
- If any of the details are not correct output an error message
- If all details are correct insert it into the users table in the database and send the user back to the login page
- Create a login button to go back to the login page incase the user needs to go back
-Look like sketch 1

Search page
- Homepage
- Gets to here after logging in
- Cannot access if not logged in
- Two buttons – one for search by category and one for search by book title and/or author
- Look like sketch 2

Search by Category
- Cannot access if not logged in
- Allow the user to select the categories in the database through a dropdown menu
- Retrieve book categories from the database

- Make nothing happen when the user clicks the search button but hasn't clicked a category
- Add a button that allows the user to switch to search by book title and/or author
- Look like sketch 4

Search by Book Title and/or Author

- Cannot access if not logged in
- Allow the user to enter a book title and/or author
- Add a button that allows the user to switch to search category
- Look like sketch 5

Display page for Search by category

- Cannot access if not logged in
- Get all results that relate to the category the user selected from the database
- Output all results from users search in a table
- Only allow 5 items to be displayed at a time
- Create a reserve button that allows the user to reserve a book
- Create next and previous page buttons so the user can look through multiple pages if there are multiple pages
- Add a button that allows the user to switch back to search by category
- Look like sketch 6

Display page for search by book title and/or author

- Cannot access if not logged in
- Get all results that are an exact match or similar to the book title and/or author the user entered from the database
- Output all results from users search in a table
- Only allow 5 items to be displayed at a time
- Create a reserve button that allows the user to reserve a book
- Create next and previous page buttons so the user can look through multiple pages if there are multiple pages
- Add a button that allows the user to switch back to search by book title and/or author
- Look like sketch 6

Reserve Page

-User only sees the page if there is an error reserving book
- Reserves book for user
- Puts the reserve into the reservations table and changes the books reserved status to Y in the books table
- Cannot access if not logged in

Show reserved books page

- Cannot access if not logged in
- Get all books the user has reserved
- Output all results from users search in a table
- Only allow 5 items to be displayed at a time
- Create an unreserve button that allows the user to unreserve a book
- Create next and previous page buttons so the user can look through multiple pages if there are multiple pages
- Add a button that allows the user to switch back to the search page
- Look like sketch 3

Logout page

- Cannot access if not logged in
- Log user out and send them back to the login page.

Connect page

- Connect to the database

CSS Design

- Common header or footer across pages
- Sketches for each page are accurately created

Navbar

- Navbar looks like the navbar in the sketches
- Allows user to go to the search and reserved books page
- Allows user to log out of account

Final Design

Login Page

The login page is the first page the user will always be greeted with. It is a requirement for the user to log in before they are able to access any of the system except the registration page. To begin, I created a form that allowed the user to enter their username and password. I removed the possibility of the user potentially leaving one of them blank by putting required in the input tags in the form. Once I was getting the data from the user through POST handling, I was able to begin creating my check to see if what the user entered matched a username and password in my database. I achieved this by firstly getting all usernames and passwords from the Users table in the database using a SQL statement. I then looped through all the data I got from it and checked every row’s username and password against the username and password the user entered to see if it matched. If a match was found the session username and success is set and the user is sent to the search page which is my homepage. If no match is found, it displays Incorrect Password or Username. To achieve the registration button, I created a button that says register that if clicked sends you to the registration page.

Registration Page

The registration page began with creating the form that allowed the user to enter all the details the system needed for the account. Again, I removed the possibility of the user potentially leaving one of them blank by putting required in the input tags in the form. I also removed the possibility of the user not entering digits in the mobile number box by making it’s type equal to number. Once I had this data through the POST method, I started working on the checks on the details. Firstly, I checked if the mobile number was 10 digits in length by using the strlen function on the mobile number the user entered. If it was correct it continued onto the next check and if it was not it displays “Mobile Phone number is not 10 digits in length”. Secondly, I checked to see if the password was 6 characters in length by again using the strlen function on it. If it was not it displays “Password is not 6 characters in length”. Thirdly, I checked to see if the password and confirm password entered was the same by just checking if they were equal to each other. If they are not it displays “Passwords do not match”. Finally, I checked if the username was unique by getting all the usernames in the users table from the database and seeing if it was equal to any of them. If it was it sets a variable called exists to True and displays “Username already exists”. If all checks are passed successfully and the variable exists is false it means all details the user entered are correct. It then puts all these details into their own variables and using a SQL statement inserts them into the users table in the database. If the SQL statement is successful, the user is sent back to the login page and if it is not an error is displayed on screen. To achieve the login button, I created a button that says login that if clicked sends you to the login page.

Search Page

The search page is my homepage. It is where the user is sent after they login. This was a very simple design. I immediately started working on the two buttons to send the user to the different types of searches. I created two buttons, using the same methods as I had used for the login page and registration page, one for search by category and one for search by book title and/or author which when clicked send the user to their respective page. It was not until after I had the two buttons created that I created the way to stop the user from being able to enter the page without logging in. I achieved this by checking if the session username was set and if not sending the user back to the login page and if it was allowing all the code for the page to run.

Search by Category

This is one of the ways a user can search for a book in the system. I began the design of this part with creating a dropdown menu so I could see how they work and understand how I may achieve retrieving the categories from the database. I noticed that in the dropdown menu all items had a value associated with them that was used if the user selected that item so I decided to create an array that could hold all the categories in the database. The item would be the category and the key would be the index. This would mean that even if I added another category to the database it would still appear. To achieve this I first created a category array and using an SQL statement got all the categories from the category table in the database. I then looped through all of the rows of categories and inserted them into the array. I then created a check to see if the search has been clicked and if it was it sends the user to the Display page for search by category and passes the category the user selected and the position variable which will be used to display only 5 books at a time on screen and will be explained later on. However, I quickly realised that if the user clicked search and did not click a category that it would still go to this page and try to display the categories which would cause the system to break. To fix this problem I set the value of the initial text in the dropdown menu to 9 and created another check to see if the category selected was equal to 9 and if it was it does nothing and if it was not it sends the user to the display page with the passed variables as I said above. To achieve the button to switch to search by book title and/or author, I again created a button that says “Search by Book Title and/or Author” and if clicked it sends the user to the search by book title and/or author page. Finally, to finish the page I created the way to stop the user from being able to enter the page without logging in. I achieved this using the exact same method as the search page which is by checking if the session username was set and if not sending the user back to the login page and if it was allowing all the code for the page to run.

Search by Book Title and/or Author

This was one of the easier pages to create. Using what I had done and learned about forms on the login and registration page, I first created a form that allowed the user to enter a book title and/or an author. The user is able to enter both, just one or none if they want. Once I had the form working I created a check to see if the search button was clicked and if so it sent the user to the display page for this search and passed the book title and author the user entered and the position variable (explained later on) to the page. Finally, to achieve the button to switch to search by category, I again created a button like in search by category that says “Search by category” and if clicked sends you to the search by category page. Again, to finish the page I created the way to stop the user from being able to enter the page without logging in. I again achieved this by using the same method as the search page and the search by category page.

Display page for Search by category

To explain my design of this page I need to explain the variables that are being passed into this page. There are two variables being passed, one is the category the user selected and the second is the position variable. The category the user selected will be used to find the results relating to this category. The position variable is used to display 5 items on the screen at a time. When the user is first sent to the page after clicking search in search by category the position variable will get passed as equal to 0 meaning the position variable on this page when it loads first will be equal to 0. I will later explain how this allows the system to display 5 items at a time on screen.

Before that, I want to talk about how I achieved getting the results that relate to the category the user selected and how I got them to display on screen. So using the category variable that was passed into the page I created an SQL statement that will get all the details the system required of the books that are in that category. These details were ISBN, Book Title, Author, Edition, Year, Category and if it is reservable or not. I had to add 1 to the category variable that was passed in the SQL statement because in the dropdown menu my first category is equal to a value of 0 while in my database the first category is equal to 1. I also used a join statement so that I could get the name of the category the books are in for the output. Once I had this SQL statement I was able to loop through all of these rows of data and output them on screen and also check if the book is reserved or not and if it is it outputs “not reservable” and if it is not reserved it outputs the reserve button which allows the user to reserve that specific book and if clicked takes the user to the reserve page and passes the ISBN of the book the user wants to reserve to the page.

Once I had this working I starting to move on to how can I only display 5 items at a time on screen and I came up with a solution. The first step of my solution was getting each detail we are going to get about the books from the database into their own individual array. As I said above, These details were ISBN, Book Title, Author, Edition, Year, Category and if it is reservable or not. I then created an index variable which I called x. Now that I had these, inside my loop that loops through all the rows of data from the SQL statement, I inserted all the details we got from each row into their specific array and then incremented the index variable x. This keeps looping and inserting the details into their specific arrays until there is no more rows of data. We now have all the rows of data and the details split into multiple arrays.

The second step was being able to display the data contained inside all of these arrays. So I began with creating a check to see if any results were found in that category by seeing if the amount of items in the book title array was greater than 0 by using the count function on the array. If it was not it will display no results found and if it was it will display the results that were found. This is where the position variable starts to be used. Using the position variable and a for loop I can loop through all the elements in each array using the position variable as my index and until position is less then how many items are in the author array which will output all the rows of data the SQL statement found.

Now that I had that working, I began the third step making sure only 5 items are displayed on screen. To achieve this, I created a new variable called new_positon which is equal to the position variable plus 5 and put that in my for loop statement so that it would keep looping while the position variable is less than new_position and the position variable is less than how many items in the array author.

The fourth and final step was now creating next page and previous page buttons so the user could traverse multiple pages it there was multiple pages. We will begin with the next page button. I created a check to see if the position variable after the for loop was still less than the number of items in the author array using count. If it was it meant there was still more items to be displayed meaning that there will be another page to display and will display the next page anchor link. If not it just displays next page but just as text with no button function. So when the next page anchor link is clicked this page is reloaded and the position variable, what it is set to at this point, and the category the user selected is passed into the page. So now when the page loads this time the position variable will be set to whatever the position variable was when we outputted the other items and will run all the array code again and will then output the next 5 items in the results. Next the previous page button, which was more difficult to implement. I first created a check to see if the position variable after the for loop has displayed all elements was greater than 5 and if it was it meant that we were not on the first page meaning there is a previous page that was displayed and will display the previous page anchor link. If not it just displays previous page but just as text with no button function. This is when an issue came up. I realised that if you just pass what position is – 5 back into the page you just get the same page so I needed a way of keeping track how many was outputted so I could subtract that off position as well to get me my previous page’s first position. To do this I created the count variable which starts at 0 when the page is loaded and keeps adding 1 to it in the for loop so we know how many items was outputted. So using this when the previous page anchor link is clicked this page is reloaded and the position variable, what it is set to at this point, subtract the count variable + 5 and the category the user selected is passed into the page. So now when the page loads this time the position variable will be set to whatever the position variable was when we outputted the items on the previous page and will run all the array code again and will then output these 5 items in the results.

Once I had this all working, I finished off the page by creating the button to switch back to search by category, I again created a button like the last few buttons that says “Go back to Search” and if clicked sends you to the Search by Category page. Finally, to finish the page I created the way to stop the user from being able to enter the page without logging in. I achieved this using the exact same method as the search page and the search by category page.

Display page for book title and/or author

The design for this page was the exact same design as the display page for search by category but with a few different variables and changes. To begin, in this page three variables are passed into it. Instead of passing category and the position variable into the page like in search by category, the book title and author the user entered and the position variable is passed. The book title and author the user entered will be used to find the results that exactly match or are similar to what the user entered. The position variable is again used to display 5 items on the screen at a time. When the user is first sent to the page after clicking search in search by book title and/or author the position variable will get passed as equal to 0 meaning the position variable on this page when it loads first will be equal to 0. This will then be used in the exact same 4 step solution to display the data as 5 items at a time on screen as search by category with just a few slight differences which I will explain later on.

Before that, I want to talk about how I achieved getting the results that are similar or are an exact match to the book title and/or author the user entered and how I got them to display on screen. So using the book title and author variable that was passed into the page I first created an SQL statement that will get the details of books and authors that are an exact match to the book title and author the user entered. These details are ISBN, Book Title, Author, Edition, Year, Category and if it is reservable or not. I used a join statement so that I could get the name of the category the books are in for the output. To know if any books were found with the exact book title and author as what the user entered I created a variable called found on this page. At the beginning of the page it is initialized to False and if there are any books that are an exact match from the SQL statement to what the user entered, found will be set to true. I then started working on what if there is no exact match and the user wants to do a partial search on both book title and author, just one or none. I solved this with another SQL statement. I first check if found is equal to False and if it is it means that no exact match was found and the partial search needs to run and if it is not it means an exact match was found and the partial search does not need to run. I created an SQL statement which is able to fulfil all the partial search requirements. It gets all the details of the books that have a book title and/or author that contains what the user entered for each. If both are entered it will get books with a book title and author similar to what the user entered. If only one is entered it will get books with a book title or author that is similar to what the user entered. For example if the user only enters the author and clicks search, book title will be set to blank causing the SQL statement to only get the books with an author similar to what the user entered. If the user left the book title and author blank all books will be displayed on screen. If results are found in the partial search, the found variable is set to True. Once I had both of these SQL statements I was able to just loop through all of the rows of data in each SQL statement to output all the results on screen and also check if the book is reserved or not and if it is it outputs "not reservable" and if it is not reserved it outputs the reserve button which allows the user to reserve that specific book and if clicked takes the user to the reserve page and passes the ISBN of the book that the user wants to reserve.

Once I had this working, I then turned my attention to how do I get 5 items to display on screen at a time. I used the exact same solution as in the Display page for search by category with a few small differences. Firstly, before anything is displayed the system checks if the found variable is False instead of checking if the amount of items in the book title array is greater than 0. If it is it means there was no results found from the SQL statements and displays "No results found" on the screen. Secondly, in the for loop instead of checking if position is less than the amount of items in the author array it checks if position is less than the amount of items in the book title array. Thirdly, in the anchor link for next page and previous page instead of passing the category variable it passes the book title and author the user entered into the page alongside the position variable in the next page anchor link and the position variable subtract count + 5 in the previous page anchor link. The final slight difference is in the next page anchor link. It checks to see if the position variable after the for loop is still less than the number of items in the book title array using count instead of the author array. These are all the differences, otherwise the solution to get 5 items to display on screen at a time is the exact same. The second and final difference which relates to which array we check can be changed to any array as all arrays are going to be the exact same length as each will have the amount of rows of results we found from the SQL statement.

Once I had implemented all of this, I finished off the page by creating the button to switch back to search by book title and/or author. I again created a button like the display page for search by category that says “Go back to Search” and if clicked sends you to the Search by book title and/or author page. Finally, to finish the page I created the way to stop the user from being able to enter the page without logging in. I achieved this using the exact same method as the search page and the search by category page.

Reserve page

This page is only seen by the user if there is an error reserving a book. The user gets to this page by clicking the reserve button displayed beside each book on the display pages for search by category and search by book title and/or author. This page has one variable passed into it from other pages which is the ISBN of the book the user wants to reserve. I also create a username variable which gets the username of the user from the session, a date variable for today’s date, a success variable which is set to False in the beginning so we know if the system fails to reserve the book at any point and finally the position variable which Is set to 0 and will be passed to the display reserved books page.

To reserve a book for the user the system needs to change the reserved status in the books table for that book to Y and then put the ISBN, date and username variables into the reservations table. The design begins by first using an SQL statement to update the reserved column for that book with that ISBN to Y in the books table. If this SQL statement successfully does that it will set success to True and move on to the next piece of code. If it does not and an error occurs, it will display there was a problem reserving the book. The next piece of code begins with a check to see if the success variable is equal to True. If it is not it means the first SQL statement failed and an error occurred meaning the system should not continue with reserving the book. If it is equal to True it means the first SQL statement was successful and we can continue with the next step of reserving the book. The next step was to put the ISBN, Username and date variables into the reservations table. The system needs this to happen so it can keep track of who has reserved what. To do this I used another SQL statement which inserts the variables ISBN, Username and date that we created at the start into the reservations table. If it is not successful, it displays an error and runs another SQL statement which updates the reserved column for that book with that ISBN back to N in the books table. If it is successful, it sends the user to the show reserved books page and passes the position variable to it. Once I had this working, I created the way to stop the user from being able to enter the page without logging in. I achieved this using the exact same method as the search page and the search by category page.

Show Reserved books page

The design for this page is the exact same design as the display page for search by category and search by book title and/or author but with a few different variables and changes. To begin, in this page, only one variable is being passed into it. The position variable is the only variable that is passed into this page. The position variable will be again used to display 5 items on the screen at a time. When the user is first sent to this page either by clicking the reserved books button in the navbar or through them clicking the reserve button and the reserve page successfully reserving the book, the position variable will get passed as equal to 0 meaning the position variable on this page when it loads first will be equal to 0. This will then be used in the exact same 4 step solution to display the data as 5 items at a time on screen as the display page for search by category and search by book title and/or author with just a few slight differences which I will explain later on.

Before that, I want to talk about how I achieved getting all the books the user has reserved and how I got them to display. To get the books the user has reserved I created a variable called username which stores the username of the user. Using this I then created an SQL statement which gets the details ISBN and reserved date from the reservations table where the username is the same. Once I had this SQL statement, I was able to loop through all of these rows of data and output them on screen. Alongside outputting the rows I also created an another SQL statement inside the loop which gets all the details of the books the user has reserved by getting all details of the books where the ISBN is equal to that ISBN at that loop. These details are ISBN, Book Title, Author, Edition, Year and Category. I used a join statement so that I could get the name of the category the books are in for the output. Once I had this working I created another loop which then outputted these rows of data alongside an unreserve button beside them that allows the user to unreserve a book and if clicked will send the user to the unreserve page and pass the ISBN of the book the user wants to unreserve.

Once I had this working, I then turned my attention to how do I get 5 items to display on screen at a time. I used the exact same solution as in the Display page for search by category with a few small differences. Firstly, instead of the reserved array I created a reserved date array which holds the detail the reserved date from each row from the SQL statement. Secondly, the details that need to go into their own individual array are ISBN, Book Title, Author, Edition, Year, Category and Reserved date. ISBN and reserved date are put into their own individual arrays in the first loop after the first SQL statement and the rest are put into their own individual arrays in the second loop after the second SQL statement. Thirdly, in the for loop instead of checking if position is less than the amount of items in the author array it checks if position is less than the amount of items in the ISBN array. Fourthly, in the anchor link for next page and previous page it only passes the position variable in the next page anchor link and the position variable subtract count + 5 in the previous page anchor link and nothing else. The final slight difference is in the next page anchor link. It checks to see if the position variable after the for loop is still less than the number of items in the ISBN array using count instead of the author array. The third and final difference which relate to which array we check can be changed to any array as all arrays are going to be the exact same length as each will have the amount of rows of results we found from the SQL statement.

Once I had implemented all of this, I finished off the page by creating the button to switch back to search. I again created a button like the display page for search by category that says "Go back to Search" and if clicked sends you to the Search page. Finally, to finish the page I created the way to stop the user from being able to enter the page without logging in. I achieved this using the exact same method as the search page and the search by category page.

Unreserve page

This page is only seen by the user if there is an error trying to unreserve a book. The user gets to this page by clicking the unreserve button beside a book on the show reserved books page and has only one variable passed into it from other pages which is the ISBN of the book the user wants to unreserve. I also create a username variable which gets the username of the user from the session, a success variable which is set to False in the beginning so we know if the system fails to reserve the book at any point and finally the position variable which is set to 0 and will be passed to the display reserved books page.

To unreserve a book for the user the system needs to change the reserved status in the books table for that book to N and then delete the reservation/row for that book from the reservations table. The design is the exact same as the reserve but there are a few changes with the SQL statements. The first SQL statement updates the reserved column for that book with that ISBN to N in the books table instead of Y. The second SQL statement deletes the row/reservation for that book from the reservations table. The third SQL statement that runs if the second SQL statement is not successful updates the reserved column for that book with that ISBN back to Y in the books table instead of N. Other than that, and the error messages that appear being slightly different, the code is the same as the reserve page. Like the reserve page, if it is successful, it sends the user to the show reserved books page and passes the position variable to it. Once I had this working, I created the way to stop the user from being able to enter the page without logging in. I achieved this using the exact same method as the search page and the search by category page.

Logout page

This page is never seen by the user. The user gets to this page by clicking the logout button in the navbar. This page’s sole purpose is to log the user out and send them back to the login page. The page design begins with first checking if the user is logged in. I achieved this using the exact same method as the search page and the search by category page. If the user is logged in their session will be destroyed and they will be sent to the login page.

Connect page

The user never sees or gets sent to this page. This is used in every page and its sole purpose is to establish the connection with the database. It is in every page and called to run using a require_once statement.

CSS Design

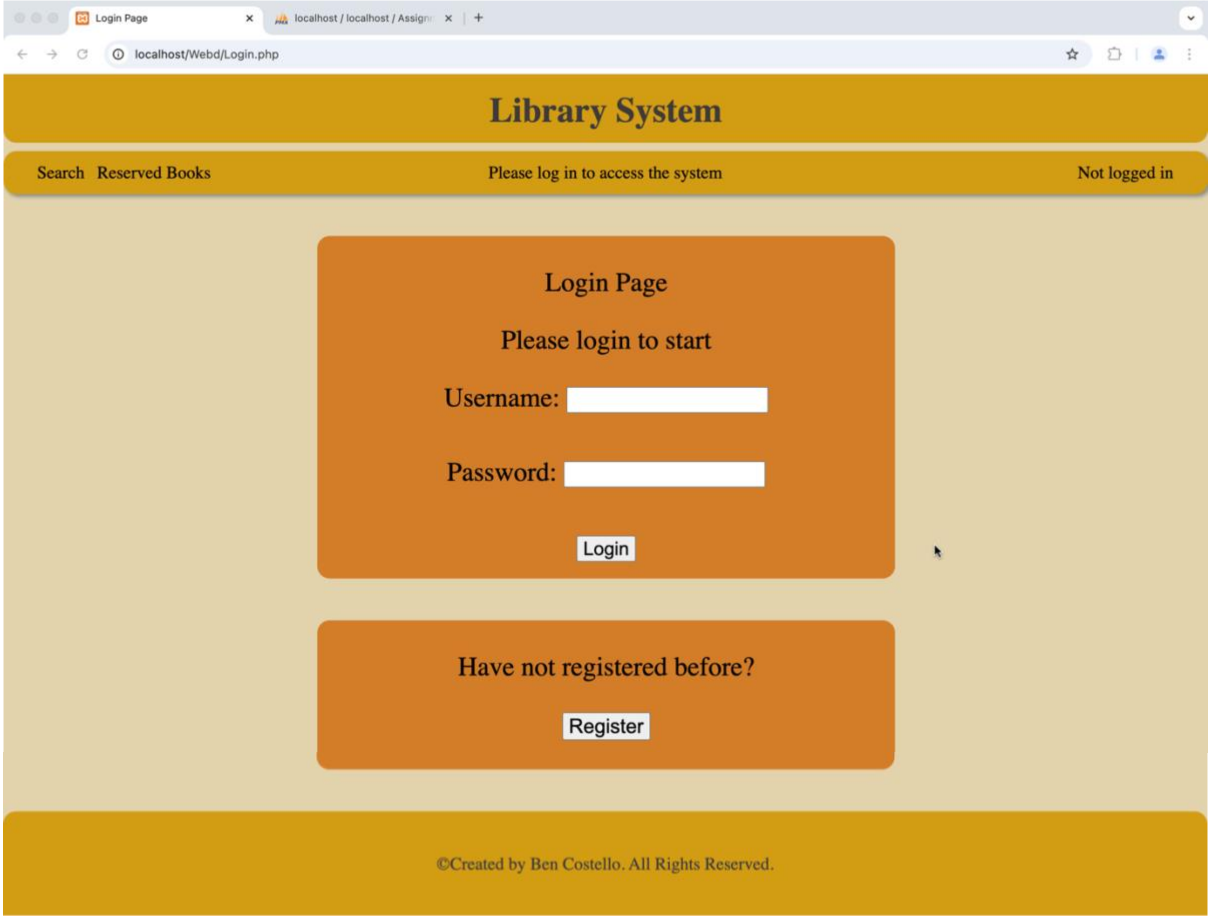
In all my pages the colours and visual design of the page are done by the CSS. I believe that I achieved an accurate representation of the sketches I wanted each page to look like.

Navbar

The navbar can be found on nearly every page except for the connect and logout. On any page that the user can potentially get to the navbar can be found. It is the way for the user to traverse the different aspects of the website. On the left-hand side of the navbar are two buttons which are the search button and the reserve books button. The search button sends the user to the search page and the reserved books button sends the user to the display reserved books page and passes the variable position to it. In the middle the navbar shows the user who they are logged in as when they are logged in and if they are on the login page or registration page that they are not logged in. Finally, on the right hand side is the logout button which sends the user to the logout page so they can be logged out.

Copy Sample of each screen

Login.php



Library System

Please log in to access the system

Not logged in

Login Page

Please login to start

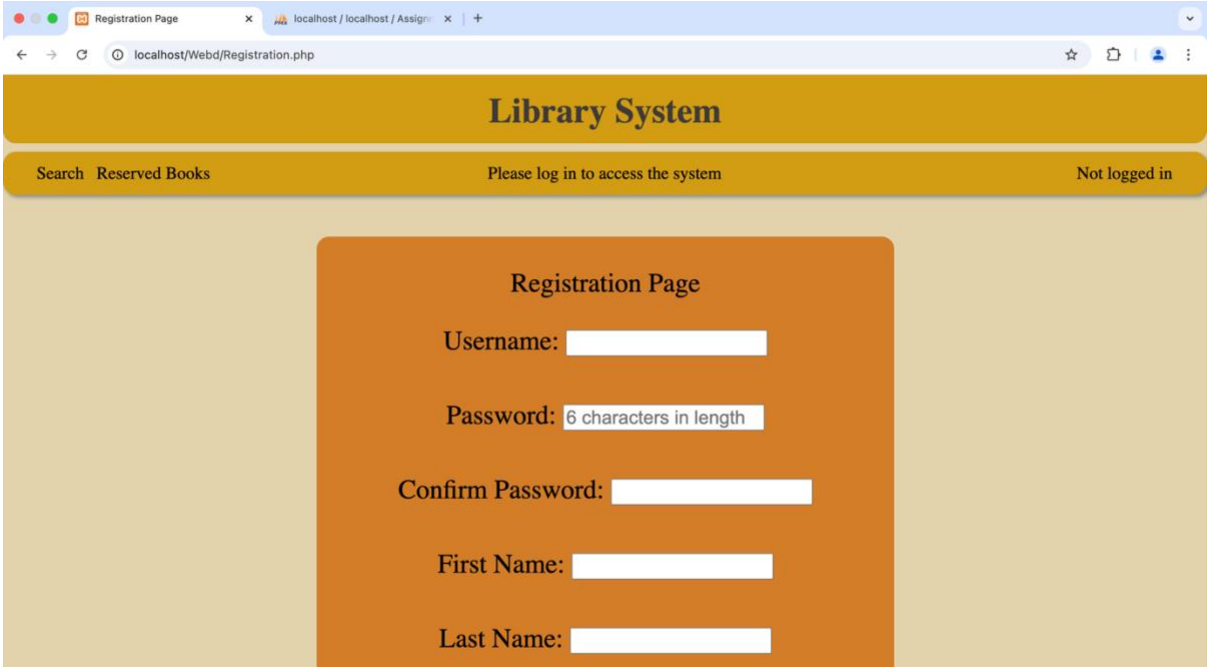
Username:

Password:

Have not registered before?

©Created by Ben Costello. All Rights Reserved.

Registration.php


Registration Page
localhost / localhost / Assignm
localhost/Webd/Registration.php
Search
Reserved Books

Library System

Please log in to access the system

Not logged in

Registration Page

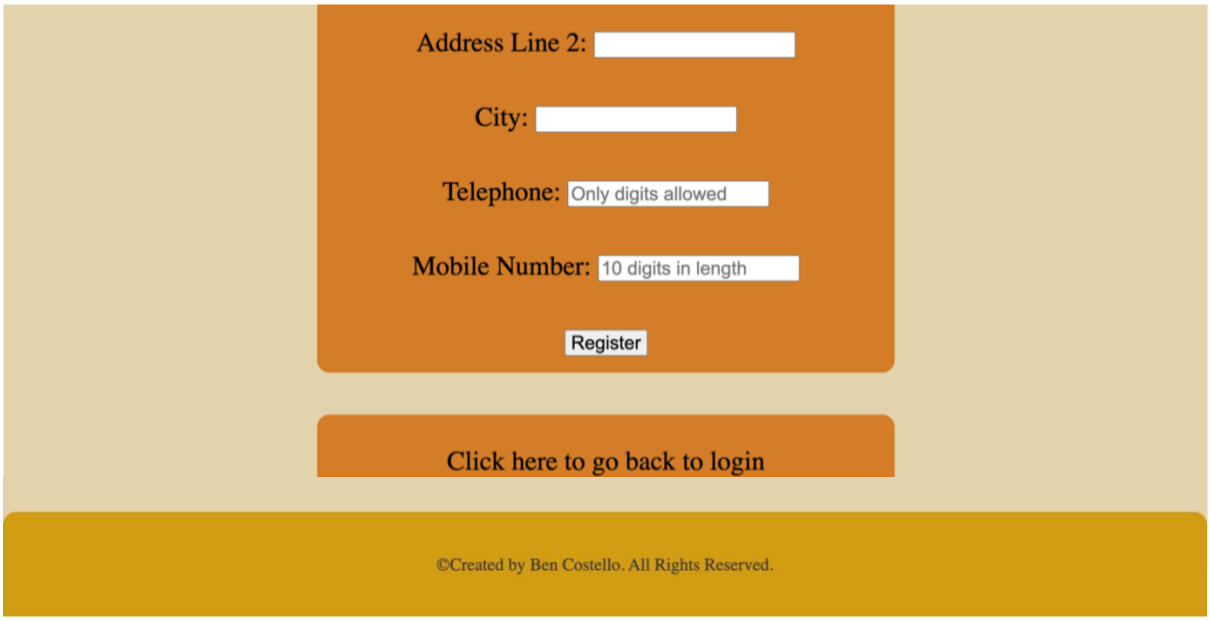
Username:

Password: 6 characters in length

Confirm Password:

First Name:

Last Name:



Address Line 2:

City:

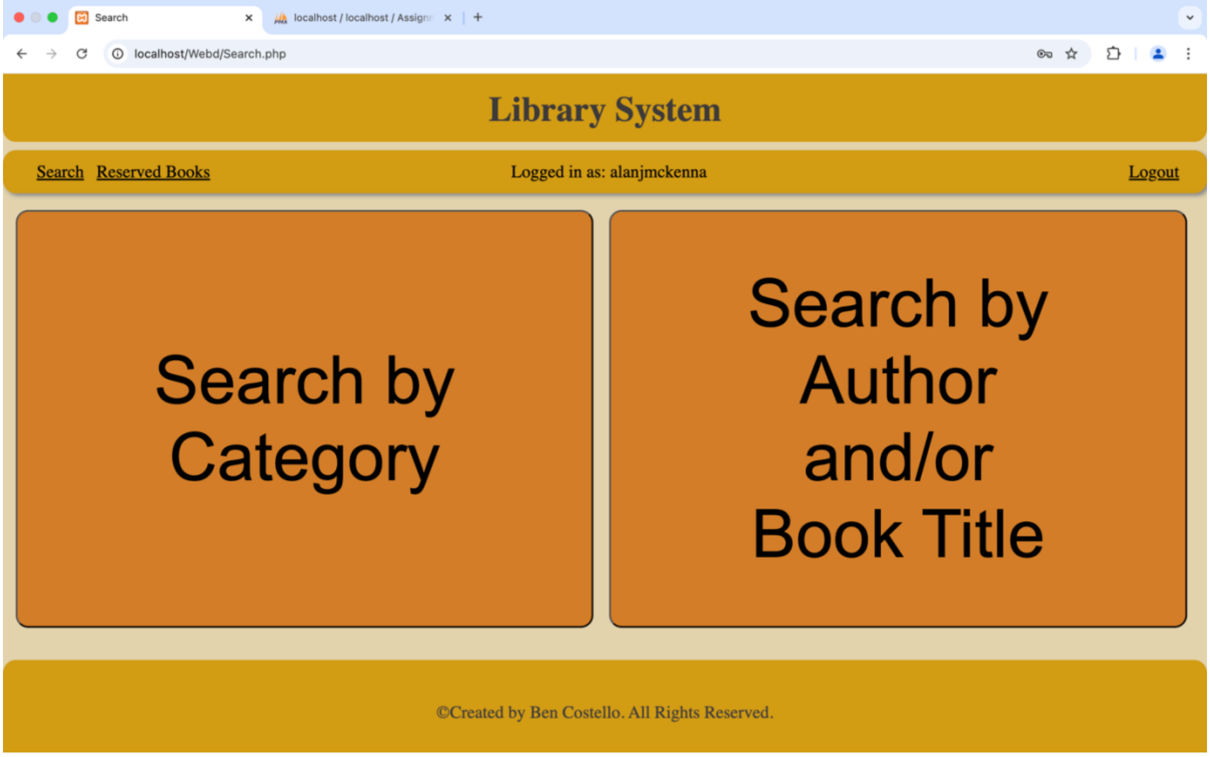
Telephone: Only digits allowed

Mobile Number: 10 digits in length

Click here to go back to login

©Created by Ben Costello. All Rights Reserved.

Search.php



Library System

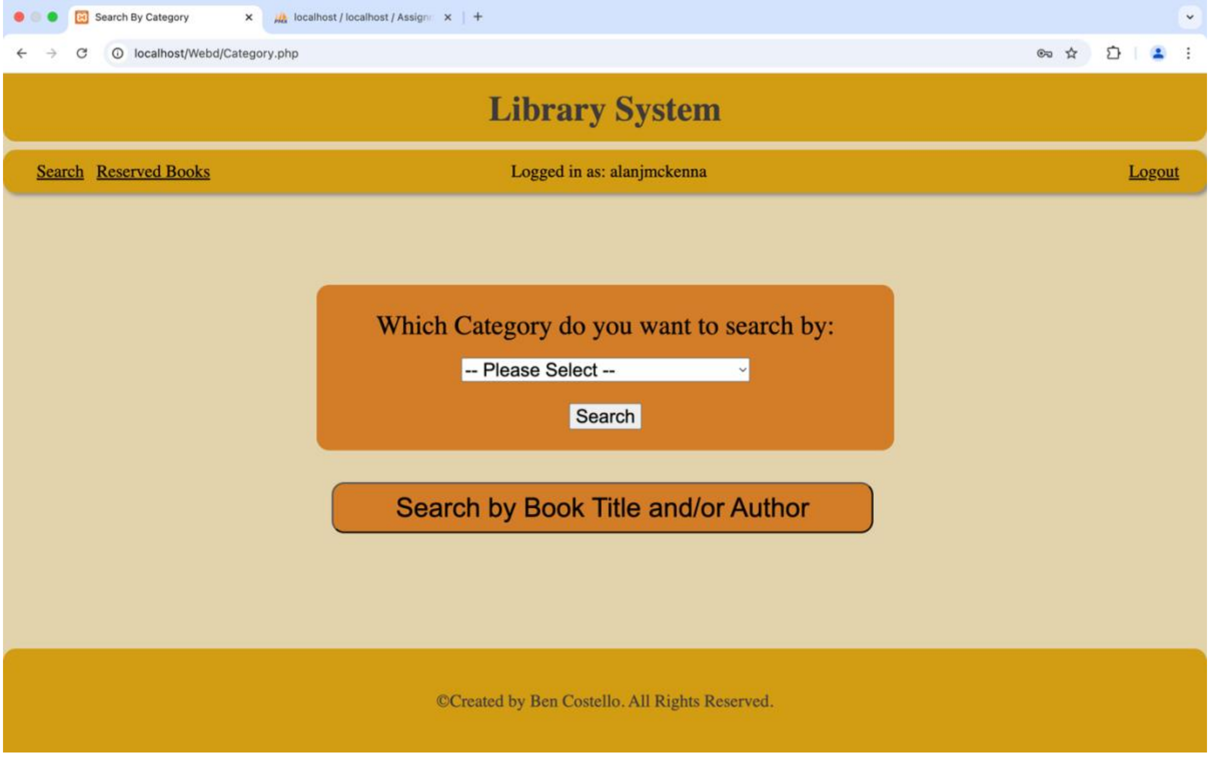
Search Reserved Books Logged in as: alanjmckenna Logout

Search by Category

Search by Author and/or Book Title

©Created by Ben Costello. All Rights Reserved.

Category.php



Search By Category
localhost / localhost / Assign
localhost/Webd/Category.php

Library System

Search Reserved Books

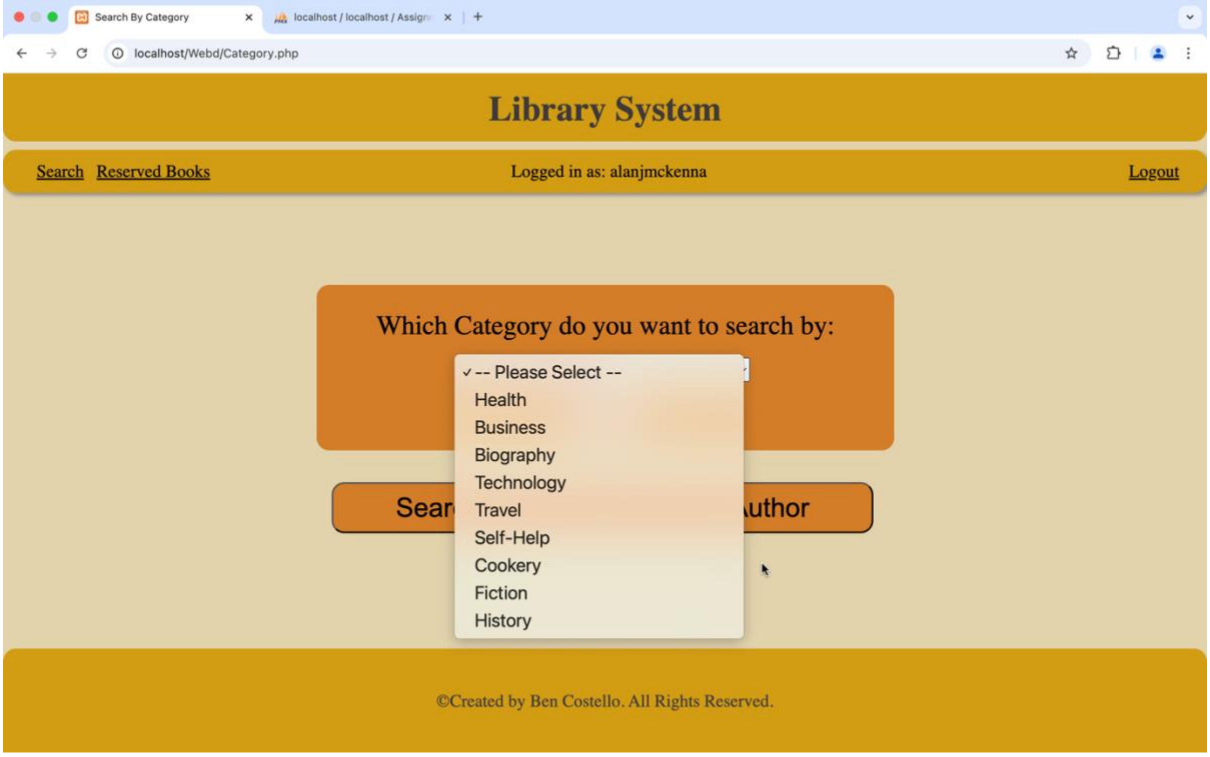
Logged in as: alanjmckenna

Logout

Which Category do you want to search by:

-- Please Select --
Search
Search by Book Title and/or Author

©Created by Ben Costello. All Rights Reserved.



Library System

Search Reserved Books

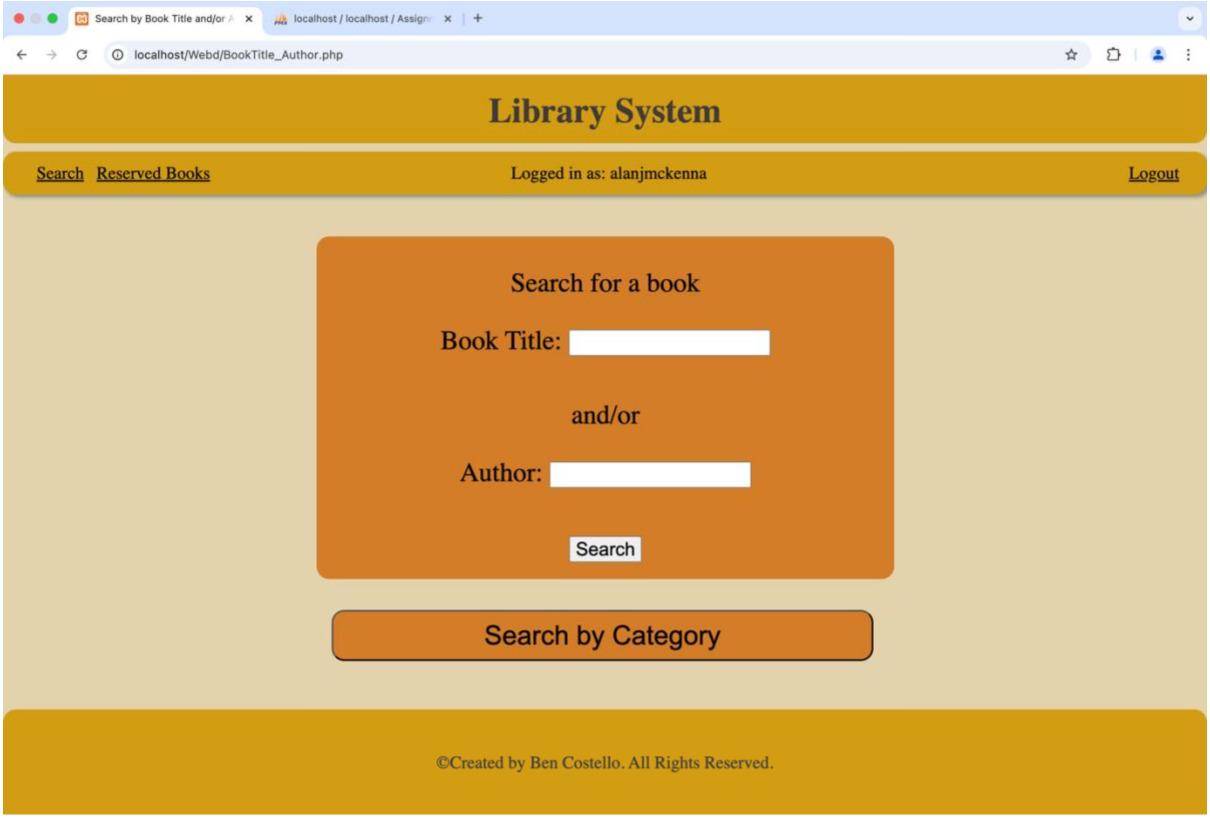
Logged in as: alanjmckenna

Logout

Which Category do you want to search by:

©Created by Ben Costello. All Rights Reserved.

BookTitle_Author.php



Search by Book Title and/or A
localhost / localhost / Assign
localhost/Webd/BookTitle_Author.php

Library System

Search Reserved Books Logged in as: alanjmckenna Logout

Search for a book

Book Title:

and/or

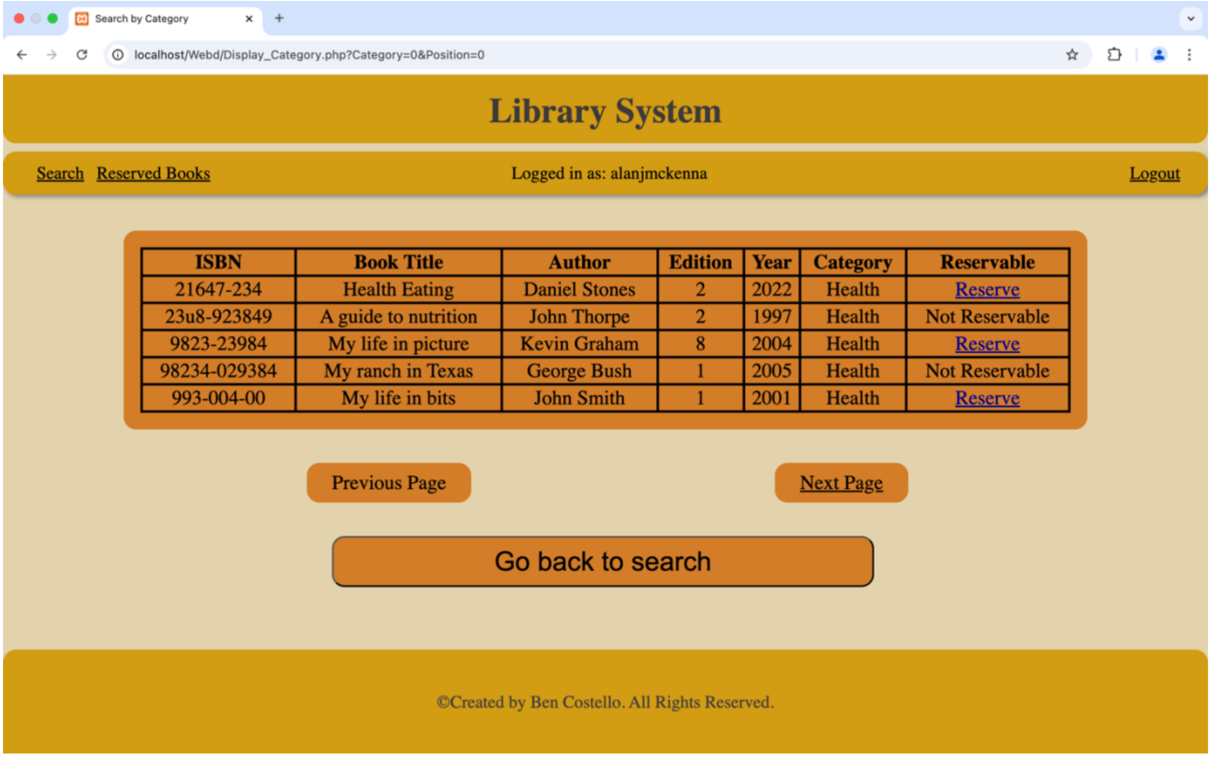
Author:

Search

Search by Category

©Created by Ben Costello. All Rights Reserved.

Display_Category.php



Library System

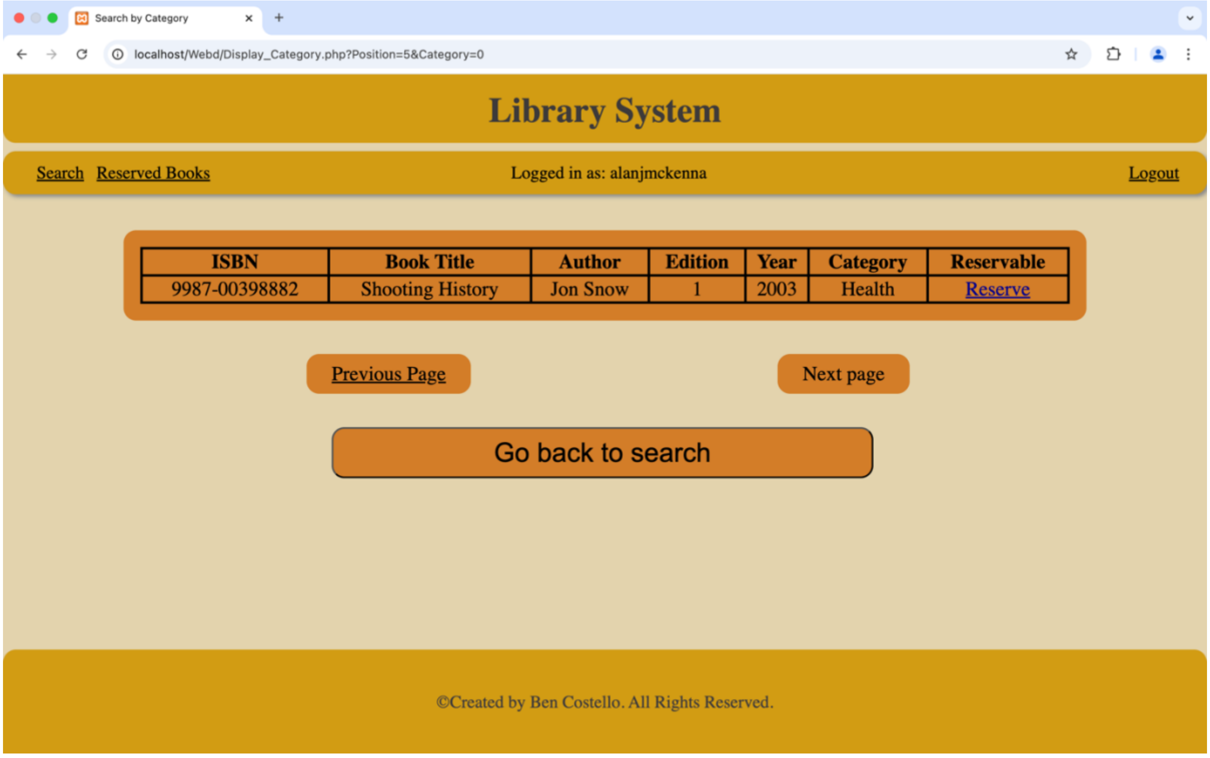
Search Reserved Books Logged in as: alanjmckenna Logout

ISBN	Book Title	Author	Edition	Year	Category	Reservable
21647-234	Health Eating	Daniel Stones	2	2022	Health	Reserve
23u8-923849	A guide to nutrition	John Thorpe	2	1997	Health	Not Reservable
9823-23984	My life in picture	Kevin Graham	8	2004	Health	Reserve
98234-029384	My ranch in Texas	George Bush	1	2005	Health	Not Reservable
993-004-00	My life in bits	John Smith	1	2001	Health	Reserve

Previous Page Next Page

Go back to search

©Created by Ben Costello. All Rights Reserved.



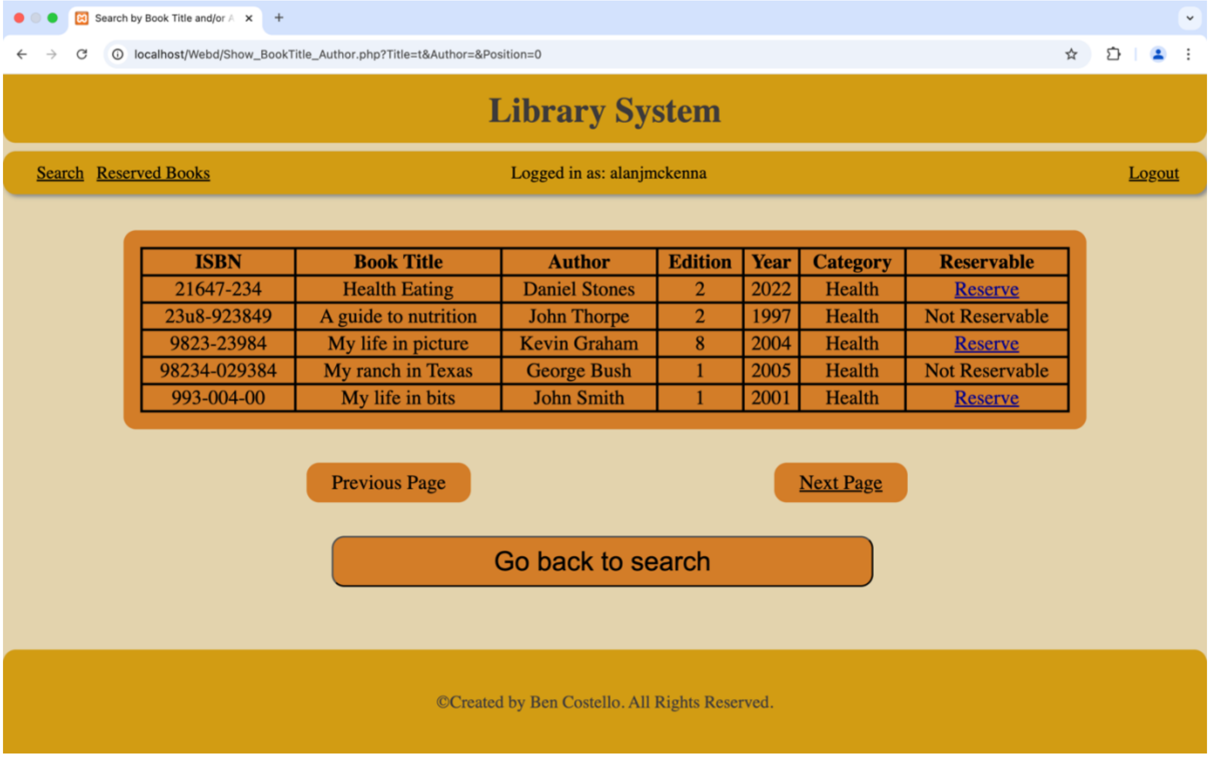
Library System

Search Reserved Books Logged in as: alanjmckenna Logout

ISBN	Book Title	Author	Edition	Year	Category	Reservable
9987-00398882	Shooting History	Jon Snow	1	2003	Health	Reserve

©Created by Ben Costello. All Rights Reserved.

Show_BookTitle_Author.php

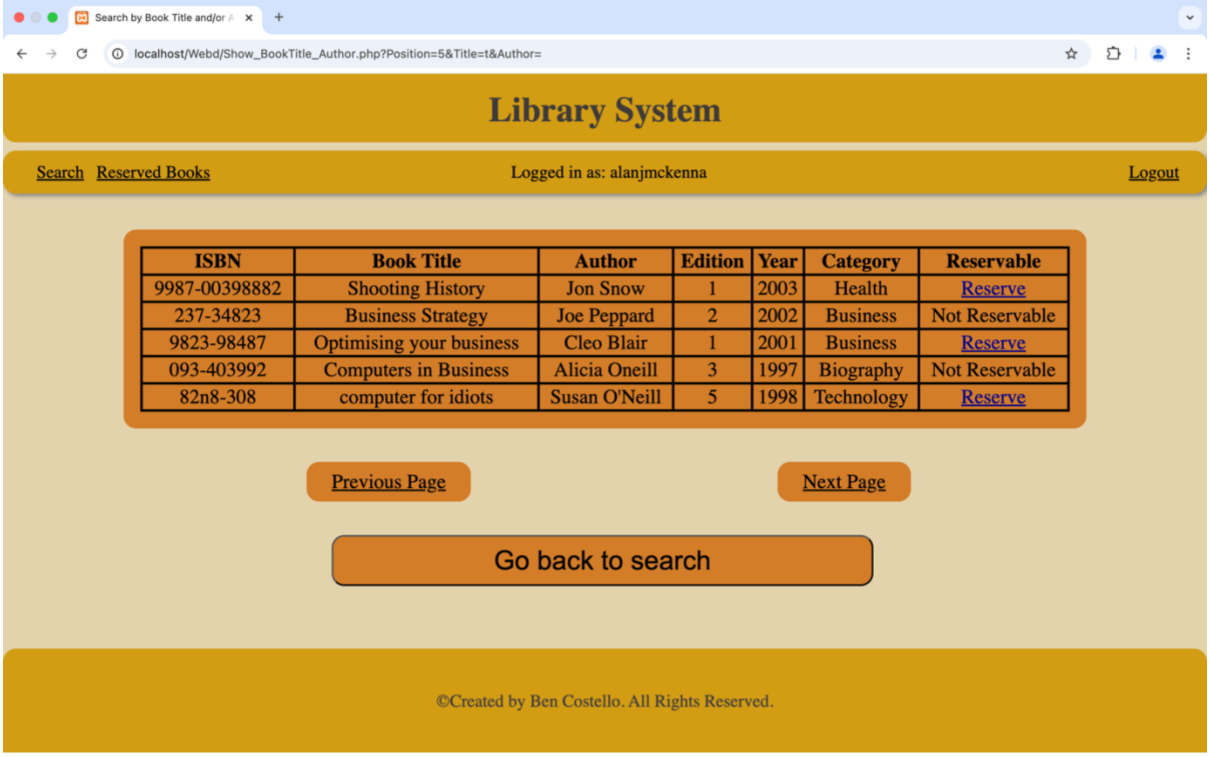

Search by Book Title and/or A
localhost/Webd/Show_BookTitle_Author.php?Title=t&Author=&Position=0

Library System

Search Reserved Books Logged in as: alanjmckenna Logout

ISBN	Book Title	Author	Edition	Year	Category	Reservable
21647-234	Health Eating	Daniel Stones	2	2022	Health	Reserve
23u8-923849	A guide to nutrition	John Thorpe	2	1997	Health	Not Reservable
9823-23984	My life in picture	Kevin Graham	8	2004	Health	Reserve
98234-029384	My ranch in Texas	George Bush	1	2005	Health	Not Reservable
993-004-00	My life in bits	John Smith	1	2001	Health	Reserve

©Created by Ben Costello. All Rights Reserved.


Search by Book Title and/or A
localhost/Webd/Show_BookTitle_Author.php?Position=5&Title=t&Author=

Library System

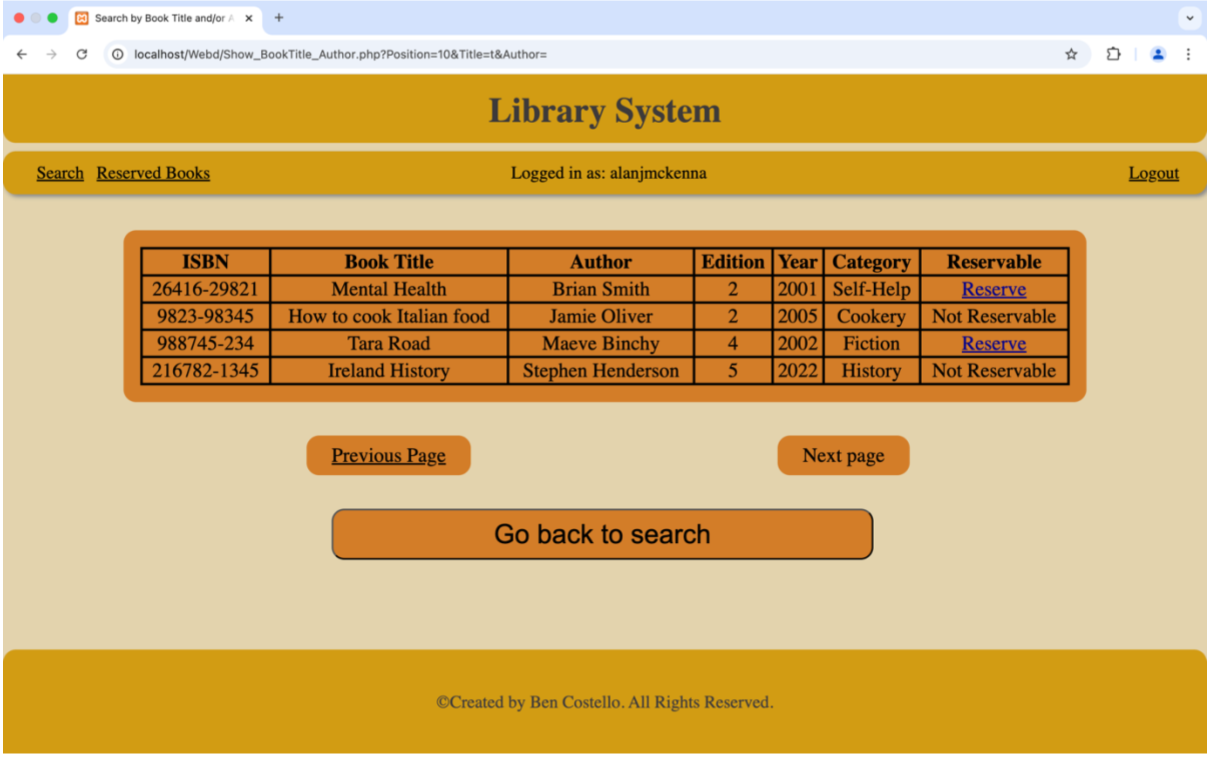
Search Reserved Books Logged in as: alanjmckenna Logout

ISBN	Book Title	Author	Edition	Year	Category	Reservable
9987-00398882	Shooting History	Jon Snow	1	2003	Health	Reserve
237-34823	Business Strategy	Joe Peppard	2	2002	Business	Not Reservable
9823-98487	Optimising your business	Cleo Blair	1	2001	Business	Reserve
093-403992	Computers in Business	Alicia Oneill	3	1997	Biography	Not Reservable
82n8-308	computer for idiots	Susan O'Neill	5	1998	Technology	Reserve

Previous Page Next Page

Go back to search

©Created by Ben Costello. All Rights Reserved.


Search by Book Title and/or A
localhost/Webd/Show_BookTitle_Author.php?Position=10&Title=t&Author=

Library System

Search Reserved Books Logged in as: alanjmckenna Logout

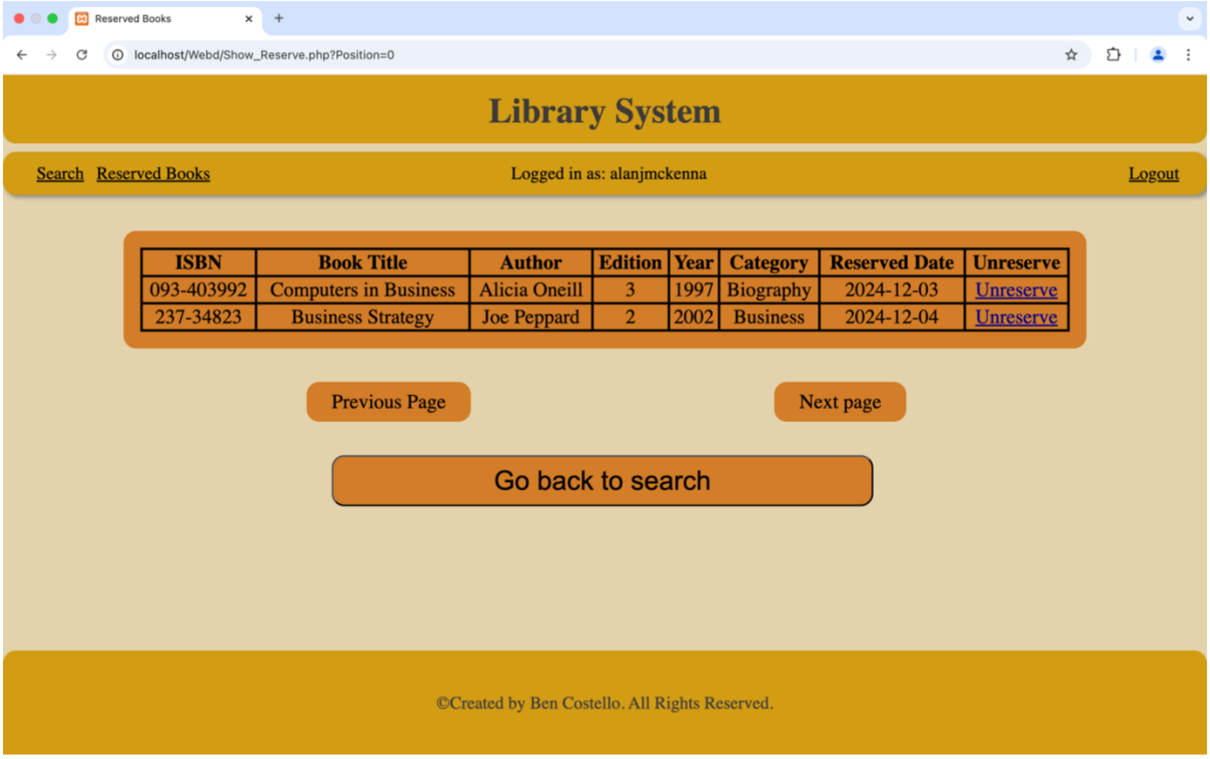
ISBN	Book Title	Author	Edition	Year	Category	Reservable
26416-29821	Mental Health	Brian Smith	2	2001	Self-Help	Reserve
9823-98345	How to cook Italian food	Jamie Oliver	2	2005	Cookery	Not Reservable
988745-234	Tara Road	Maeve Binchy	4	2002	Fiction	Reserve
216782-1345	Ireland History	Stephen Henderson	5	2022	History	Not Reservable

Previous Page Next page

Go back to search

©Created by Ben Costello. All Rights Reserved.

Show_Reserve.php



Library System

Logged in as: alanjmckenna

ISBN	Book Title	Author	Edition	Year	Category	Reserved Date	Unreserve
093-403992	Computers in Business	Alicia Oneill	3	1997	Biography	2024-12-03	Unreserve
237-34823	Business Strategy	Joe Peppard	2	2002	Business	2024-12-04	Unreserve

©Created by Ben Costello. All Rights Reserved.

Copy Listing of my Source Code

Login.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Page</title>
    <link rel="stylesheet" href="Design.css"> <!-- Links to the css file --->
</head>
<body>

    <!-- The header --->
    <header>
        <h1>Library System</h1>
    </header>

    <!-- The buttons in the navbar will do nothing on this page as the user is not logged in --->
    <nav>
        <div class="Options">
```

```
            <a style = "text-decoration: none; color: black;" class = "Options" href = "Search.php">Search</a>
            <?php
                echo ('<a style = "text-decoration: none; color: black;" href="Show_Reserve.php?Position='.htmlentities($Position).'">Reserved Books</a>');
            ?>
        </div>
        <div id = "message_login_register">
            <?php
                echo "Please log in to access the system";
            ?>
        </div>
        <div class = "Logout">
            <?php
                echo "Not logged in";
            ?>
        </div>

    </nav>

    <?php
        session_start(); //Starts the users session
        if (isset($_SESSION["error"])) //If the user enters an incorrect password or username it displays the error on screen.
        {
            echo '<div class = "Login_Search_Register_box">';
            echo $_SESSION["error"];
            echo '</div>';
            unset($_SESSION["error"]);
        }
    ?>

    <!-- This is my login form that gets the user's login information from them --->
    <!-- Required stops the user from leaving that text box blank --->
    <section class = "Login_Search_Register_box">
        <p> Login Page </p>
        <p> Please login to start </p>
        <form method="post">
        <p>Username:
        <input type="text" name="Username" required></p>
        <p>Password:
        <input type="Password" name="Password" required></p>
        <p><input type="submit" name = "Login" value="Login"/></p>
        </form>
    </section>

    <!-- This is a button on screen that takes the user to teh registration page if they need to register --->
    <div class = "Login_Search_Register_box" style = "margin-bottom: 50px;">
        <p>Have not registered before?</p>
        <a id = "Inside_Login_Register_button" href="Registration.php">
            <button type = "button">
                Register
            </button>
        </a>
    </div>
    <?php
        unset($_SESSION["Username"]);

        //Checks if the login button (submit button) has been clicked on the form and if so will run the code
        if (isset($_POST["Login"]))
        {
            //Connects the page to the database
            require_once("Connect.php");
            unset($_SESSION["error"]);

            //This sql statement gets all the usernames and passwords from the users table in the database
            $sql = "SELECT Username, Password FROM Users";
            $result = $conn->query($sql);
            if ($result->num_rows > 0) {
            // output data of each row
            while($row = $result->fetch_assoc()) {
                //Checks if the username and password the user entered is equal to any username and passwords in the users table
                if ($row["Username"] == $_POST['Username'] && $row["Password"] == $_POST['Password'])
                {
                    //If the username and password matches it will set the session so we know they are logged in and take them to the search page
                    $_SESSION["Username"] = htmlentities($_POST["Username"]);
                    $_SESSION["success"] = "Logged in";
                    header("Location: Search.php");
                    return;
                }
                else
                {
                    //If it matches no username and password it will set the error in the session and reload the login page
                    $_SESSION["error"] = "Incorrect Password or Username";
                    header("Location: Login.php");

                }
                }
            }

            //Closes the database connection
            $conn->close();
        }

    ?>



    <footer>
        <p>©Created by Ben Costello. All Rights Reserved.</p>
    </footer>
```

```
</body>
</html>
```

Registration.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Registration Page</title>
    <link rel="stylesheet" href="Design.css"> <!-- Links to the css file --->
</head>
<body>

    <!-- The header --->
    <header>
        <h1>Library System</h1>
    </header>

    <!-- The buttons in the navbar will do nothing on this page as the user is not logged in --->
    <nav>
        <div class="Options">
            <a style = "text-decoration: none; color: black;" class = "Options" href = "Search.php">Search</a>
            <?php
                echo ('<a style = "text-decoration: none; color: black;" href="Show_Reserve.php?Position='.htmlentities($Position).'">Reserved
Books</a>');
            ?>
        </div>
        <div id = "message_login_register">
            <?php
                echo "Please log in to access the system";
            ?>
        </div>
        <div class = "Logout">
        <?php
            echo "Not logged in";
        ?>
        </div>

    </nav>

    <?php

        //Checks if the register button (submit button) has been clicked on the form and if so will run the code
        if (isset($_POST['submit']))
        {
            //Connects the page to the database
            require_once "Connect.php";
            //This variable is used to check if the username already exists late on in the code
            $exists = FALSE;

            //Checks if the Mobile number the user entered is 10 digits in length
            if (strlen($_POST['Mobile']) == 10 )
            {
                //Checks if the password the user entered is 6 character in length
                if(strlen($_POST['Password']) == 6)
                {
                    //Checks if the password and confirm password matches
                    if ($_POST['Password'] == $_POST['ConfirmPassword'])
                    {
                        //This sql statement gets all the usernames and passwords from the users table in the database
                        $sql = "SELECT Username FROM Users";
                        $result = $conn->query($sql);
                        if ($result->num_rows > 0) {
                        // output data of each row
                            while($row = $result->fetch_assoc())
                            {
                                //This checks if the username the user entered already exists by seeing if it matches with any of the usernames in the user
table
                                if ($row["Username"] == $_POST['Username'])
                                {
                                    //If it does, it means it already exists and outputs that to the user and sets the variable exists to true
                                    echo '<div class = "Login_Search_Register_box">';
                                    echo "Username already exists";
                                    echo '</div>';
                                    $exists = TRUE;
                                    break;
                                }
                            }
                        }
                        //If the variable exists is false it means the username doesn't already exist and if it is true it means it does already exist
                        if ($exists == FALSE)
                        {
                            //Gets all the data from the from and puts them into variables
                            $n = $conn -> real_escape_string(htmlentities($_POST['Username']));
                            $p = $conn -> real_escape_string(htmlentities($_POST['Password']));
                            $fn = $conn -> real_escape_string(htmlentities($_POST['Firstname']));
                            $sn = $conn -> real_escape_string(htmlentities($_POST['Surname']));
                            $a1 = $conn -> real_escape_string(htmlentities($_POST['Address1']));
                            $a2 = $conn -> real_escape_string(htmlentities($_POST['Address2']));
                            $c = $conn -> real_escape_string(htmlentities($_POST['City']));
                            $t = $conn -> real_escape_string(htmlentities($_POST['Telephone']));
                            $m = $conn -> real_escape_string(htmlentities($_POST['Mobile']));

                            //This sql statement inserts all the data we got from the from into the table users in the database
                            $sql = "INSERT INTO Users (Username, Password, FirstName, Surname, AddressLine1, AddressLine2,City, Telephone, Mobile
) VALUES ('$n', '$p', '$fn', '$sn', '$a1','$a2','$c','$t','$m')";
                            if ($conn->query($sql) === TRUE) {
                                //If it successfully inserted all the data the user gets sent back to the login page
                                header("Location: Login.php");
                            } else {
                                //If it is unsuccessful an error message will appear for the user
                                echo '<div class = "Login_Search_Register_box">';
                                echo "Error: " . $sql . "<br>" . $conn->error;
                                echo '</div>';
                            }
                        }
                    }
                    else
                    {
                        //If the password and confirm password do not match it displays an error
                        echo '<div class = "Login_Search_Register_box">';
                        echo "Passwords do not match";
                        echo '</div>';
                    }//End password matching check
                }
                else
                {
                    //If the password is not 6 characters in length it displays an error
                    echo '<div class = "Login_Search_Register_box">';
                    echo "Password is not 6 characters in length";
                    echo '</div>';
                }//End password length check
            }
            else
            {
                //If the mobile number is not 10 characters in length it displays an error
                echo '<div class = "Login_Search_Register_box">';
                echo "Mobile Phone number is not 10 digits in length";
                echo '</div>';
            }//End mobile number length check

            $conn->close();
        }
    ?>


    <!-- This is my registration from --->
    <!-- The value on all of them is set to output the data the user has already entered there so they don't lose what they entered if they do something wrong --->
    <!-- Required stops the user from leaving that text box blank --->
    <section class = "Login_Search_Register_box">
        <p> Registration Page </p>
        <form method="post">
        <p>Username:
        <input required type="text" name="Username" value=<?php echo htmlentities($_POST["Username"])?>></p>
        <p>Password:
        <input required type="Password" placeholder = "6 characters in length" name="Password"></p>
        <p>Confirm Password:
        <input required type="Password" name="ConfirmPassword"></p>
```

```
        <p>First Name:
        <input required type="text" name="Firstname" value=<?php echo htmlentities($_POST["Firstname"])?>></p>
        <p>Last Name:
        <input required type="text" name="Surname" value=<?php echo htmlentities($_POST["Surname"])?>></p>
        <p>Address Line 1:
        <input required type="text" name="Address1" value=<?php echo htmlentities($_POST["Address1"])?>></p>
        <p>Address Line 2:
        <input required type="text" name="Address2" value=<?php echo htmlentities($_POST["Address2"])?>></p>
        <p>City:
        <input required type="text" name="City" value=<?php echo htmlentities($_POST["City"])?>></p>
        <p>Telephone:
        <!-- Input type being equal to number allows the user to be only able to enter numbers into the input box --->
        <input required type="number" name="Telephone" placeholder = "Only digits allowed" value=<?php echo htmlentities($_POST["Telephone"])?>></p>
        <p>Mobile Number:
        <input required type="number" name="Mobile" placeholder = "10 digits in length" value=<?php echo htmlentities($_POST["Mobile"])?>></p>
        <p><input type="submit" name="submit" value="Register"/></p>
        </form>
    </section>

    <!-- A button to allow the user to go back to login page if needed --->
    <div class = "Login_Search_Register_box" style = "margin-bottom: 50px;">
        <p>Click here to go back to login</p>
        <a id = "Inside_Login_Register_button" href="Login.php">
            <button type = "button">
                Login
            </button>
        </a>
    </div>



    <footer>
        <p>©Created by Ben Costello. All Rights Reserved.</p>
    </footer>



</body>
</html>
```

Search.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Search</title>
    <link rel="stylesheet" href="Design.css"> <!-- Links to the css file --->
```

```
</head>

<body>
<?php
    session_start(); //Starts the user's session on this page

    //Checks if the user has logged in or not
    if( !isset($_SESSION["Username"]))
    {
        //If the user has not logged in they are sent back to the login page
        header("Location:Login.php");
    }
    else //If the user has logged in the code below will run and the page will load - End bracket at end of page
    {
        //This is a variable that will be passed into the Show_Reserve page to display 5 books from the user's search on screen at a  time
        $Position = 0
?>

        <!-- The header --->
        <header>
            <h1>Library System</h1>
        </header>

        <nav>
            <div class = "Options">
                <!-- Search button - brings the user to the search page --->
                <a style = "color: black;" class = "Options" href = "Search.php">Search</a>
                <?php
                    //Reserved books button - brings the user to the show_reserve page and passes the position variable to it it
                    echo ('<a style = "color: black;" href="Show_Reserve.php?Position='.htmlentities($Position).'">Reserved Books</a>');
                ?>
            </div>
            <div class = "Text">
                <?php
                    echo "Logged in as: " . $_SESSION["Username"];
                ?>
            </div>
            <div class = "Logout">
                <?php
                    //Logout button - brings the user to the logout page
                    echo ('<a style = "color: black;" href="Logout.php">Logout</a>');
                ?>
            </div>

        </nav>



        <section>
            <!-- This is the search by category button --->
            <a class = "Selection" href="Category.php">
                <button type = "button">
```

```
                    Search by <br>
                    Category
                </button>
                <br></br>
            </a>

            <!-- This is the search by Author and/or Book Title --->
            <a class = "Selection" href="BookTitle_Author.php">
                <button type = "button">
                    Search by <br>
                    Author <br> and/or <br> Book Title
            </button>
            <br></br>
            </a>
        </section>

        <footer>
            <p>©Created by Ben Costello. All Rights Reserved.</p>
        </footer>

<!-- Closes the database connection --->
<?php $conn->close();} ?> <!-- Closes the else part of the check if the user is logged in --->



</body>
</html>
```

Category.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Search By Category</title>
    <link rel="stylesheet" href="Design.css"> <!-- Links to the css file --->
</head>
<body>

<?php
    session_start(); //Starts the user's session on this page

    //Checks if the user has logged in or not
    if( !isset($_SESSION["Username"]))
    {
        //If the user has not logged in they are sent back to the login page
        header("Location:Login.php");
    }
    else //If the user has logged in the code below will run and the page will load - end bracket at the end of page
    {
```

```
    //This is a variable that will be passed into the Show_Category page to display 5 books from the user's search on screen at a time
    $Position = 0;

?>
    <!-- The header --->
    <header>
      <h1>Library System</h1>
    </header>

    <nav>
      <div class = "Options">
        <!-- Search button - brings the user to the search page --->
        <a style = "color: black;" class = "Options" href = "Search.php">Search</a>
        <?php
          //Reserved books button - brings the user to the show_reserve page and passes the position variable to it it
          echo ('<a style = "color: black;" href="Show_Reserve.php?Position='.htmlentities($Position).'">Reserved Books</a>');
        ?>
      </div>
      <div class = "Text">
        <?php
          echo "Logged in as: " . $_SESSION["Username"];
        ?>
      </div>
      <div class = "Logout">
        <?php
          //Logout button - brings the user to the logout page
          echo ('<a style = "color: black;" href="Logout.php">Logout</a>');
        ?>
      </div>
    </nav>

    <?php
      //Connects the page to the database
      require_once("Connect.php");
      $categories = array(); //This array will hold all the categories in the database so we can display them
      $x = 0; //Index variable for the array
      $sql = "SELECT CategoryDescription FROM Category"; //This sql statement gets all the category's from the category table in the
database
      $result = $conn->query($sql);
      if ($result->num_rows > 0) {
      // output data of each row
        while($row = $result->fetch_assoc())
        {
          //Puts all the categories into the category array
          $categories[$x] = $row["CategoryDescription"];
          $x += 1;
        }
      }

      if (isset($_POST['Search'])) //Checks to see if the search button was clicked
      {
        //Checks to make sure that a category has been selected
        //The not equal to 9 makes sure the user didn't click the search button without selecting a category
        if (isset($_POST["Category"]) && $_POST["Category"] != 9)
        {
          //If they have selected a category it sends the user to the Display_Category page to display all books that are in that category and
          //passes the position variable to that page
          header("Location: Display_Category.php?Category=".htmlentities($_POST["Category"])."&Position=".htmlentities($Position));
          return;
        }
      }
      //Closes the database connection
      $conn->close();
    ?>

    <!-- This is the dropdown menu for the user to select a category to search by --->
    <div>
      <form id = "Search_Category" method="post">
        <label for="category">Which Category do you want to search by: <br>
        <select id = "Menu" name="Category">
          <option value="9">-- Please Select -- </option> <!-- This has a value of 9 so the program knows if the user hasn't selected a category
--->
          <?php
            //Loops through the categories array and inserts their index as the key and the category name as the item and creates the
dropdown menu
            foreach ($categories as $k => $v)
            {
              echo "<option value=$k>$v</option>";
            }
          ?>
        </select>
        <br>
        <input type="submit" name = "Search" value="Search"/>
      </form>
    </div>

    <!-- Button - Allows the user to switch to the search by author and/or book title -->
    <div>
      <a id = "InsideCategory" href="BookTitle_Author.php">
        <button type = "button">
          Search by Book Title and/or Author
        </button>
      </a>
    </div>


    <footer>
      <p>©Created by Ben Costello. All Rights Reserved.</p>
    </footer>

<?php } ?> <!-- Closes the else part of the check if the user is logged in --->
```

```
</body>
</html>
```

BookTitle_Author.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Search by Book Title and/or Author</title>
    <link rel="stylesheet" href="Design.css"> <!-- Links to the css file --->
</head>
<body>

<?php
    session_start(); //Starts the user's session on this page

    //Checks if the user has logged in or not
    if( !isset($_SESSION["Username"]))
    {
        //If the user has not logged in they are sent back to the login page
        header("Location:Login.php");
    }
    else//If the user has logged in the code below will run and the page will load - End bracket at end of page
    {
        //This is a variable that will be passed into the Show_BookTitle_Author page to display 5 books from the user's search on screen at a time
        $Position = 0;
?>
        <!-- The header --->
        <header>
            <h1>Library System</h1>
        </header>

        <nav>
            <div class = "Options">
                <!-- Search button - brings the user to the search page --->
                <a style = "color: black;" class = "Options" href = "Search.php">Search</a>
                <?php
                    //Reserved books button - brings the user to the show_reserve page and passes the position variable to it it
                    echo ('<a style = "color: black;" href="Show_Reserve.php?Position='.htmlentities($Position).'">Reserved Books</a>');
                ?>
            </div>
            <div class = "Text">
                <?php
                    echo "Logged in as: " . $_SESSION["Username"];
                ?>
            </div>
            <div class = "Logout">
                <?php
                    //Logout button - brings the user to the logout page
                    echo ('<a style = "color: black;" href="Logout.php">Logout</a>');
                ?>
            </div>
        </nav>

        <!-- This form allows the user to enter the book title and/or author they want to search for --->
        <div class = "Login_Search_Register_box">
            <p>Search for a book </p>
            <form method="post">
            <p>Book Title:
            <input type="text" name="Title"></p>
            <p id = "padding_and_or"> and/or </p>
            <p>Author:
            <input type="text" name="Author"></p>
            <p><input type="submit" name = "Search" value="Search"/></p>
            </form>
        </div>

        <!-- Button - Allows the user to switch to search by category --->
        <div>
            <a id = "InsideTitleAuthor" href="Category.php">
                <button type = "button">
                    Search by Category
                </button>
            </a>
        </div>

        <?php
            //Checks if the search button has been clicked
            if (isset($_POST['Search']))
            {
                //If it has been clicked
                //Connects the page to the database
                require_once("Connect.php");

                //Sends the user to the Show_BookTitle_Author page so the books that match their search can be displayed
                //It passes the title,author they searched for and the position variable so we can display 5 items on the screen at a time
                header("Location:
Show_BookTitle_Author.php?Title=".htmlentities($_POST["Title"])."&Author=".htmlentities($_POST["Author"])."&Position=".htmlentities($Position))
;
                return;

                //Closes the database connection
                $conn->close();
            }

        ?>

        <footer>
            <p>©Created by Ben Costello. All Rights Reserved.</p>
        </footer>

<?php } ?>
</body>
</html>
```

Display_Category.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Search by Category</title>
    <link rel="stylesheet" href="Design.css"> <!-- Links to the css file --->
</head>
<body>

<?php
    session_start(); //Starts the user's session on this page

    //Checks if the user has logged in or not
    if( !isset($_SESSION["Username"]))
    {
        //If the user has not logged in they are sent back to the login page
        header("Location:Login.php");
    }
    else //If the user has logged in the code below will run and the page will load - End bracket at end of page
    {
        //Connects the page to the database
        require_once("Connect.php");
        $Position = 0; //This is a variable that will be passed into the Show_Reserve page to display 5 books from the user's search on screen at a
time
?>
        <!-- The header --->
        <header>
            <h1>Library System</h1>
        </header>

        <nav>
            <div class = "Options">
                <!-- Search button - brings the user to the search page --->
                <a style = "color: black;" class = "Options" href = "Search.php">Search</a>
        <?php
          //Reserved books button - brings the user to the show_reserve page and passes the position variable to it it
          echo ('<a style = "color: black;" href="Show_Reserve.php?Position='.htmlentities($Position).'">Reserved Books</a>');
        ?>
      </div>
      <div class = "Text">
        <?php
          echo "Logged in as: " . $_SESSION["Username"];
        ?>
      </div>
      <div class = "Logout">
        <?php
          //Logout button - brings the user to the logout page
          echo ('<a style = "color: black;" href="Logout.php">Logout</a>');
        ?>
      </div>
    </nav>

<?php
    $Category = $conn -> real_escape_string($_GET['Category']); //Gets the category the user selected which is passed into the page
    $Position = $conn -> real_escape_string($_GET['Position']); //Gets the position variable which is passed into the page
    $New_Position = $Position + 5; //This will be used so only 5 books are displayed and it stops at that position

    //Creates an array for all the columns in the books table in the database we are gonna display
    $BookTitle = array();
    $Author = array();
    $ISBN = array();
    $Reserved = array();
    $Edition = array();
    $Year = array();
    $Category_array = array();
    $x = 0; //Index variable for the arrays
    $count = 0; //This keeps track of how many books we have displayed on the page and will be used to go to the previous page of books

    //This sql statement gets all the rows of books that are in the category the user selected - You add 1 to the category variable as the first
category is equal to 1 in the dropdown menu
    //while in the database the first category is equal to 1
    //Columns it gets from the rows are Book title, author, edition, year and reserved from the book table and the category description from the
category table
    $sql = "SELECT ISBN, BookTitle,Author, Edition, Year, Reserved, CategoryDescription FROM Books join Category ON
Books.CategoryCode = Category.CategoryID where CategoryCode = $Category + 1";
    $result = $conn->query($sql);
    if ($result->num_rows > 0)
    {
      while($row = $result->fetch_assoc())
      {
        //Puts all the columns of the rows of books that were found into the correct arrays.
        $BookTitle[$x] = $row["BookTitle"];
        $Author[$x] = $row["Author"];
        $ISBN[$x] = $row["ISBN"];
        $Reserved[$x] = $row["Reserved"];
        $Edition[$x] = $row["Edition"];
        $Year[$x] = $row["Year"];
        $Category_array[$x] = $row["CategoryDescription"];
        $x = $x + 1; //Increments the index
    }
}

//Checks if  any books were found in that category by checking if the amount of items in the array book title is above 0
if (count($BookTitle) > 0)
{
    //If there was books it will display the rows of books found in a table
    ?>

    <!-- Start creation of table which will display the result --->
    <div class = "display_box" style = "margin-top: 3%;">
    <table style = "border: 1px solid black;">
        <tr>
            <!-- The headings for each column in the table --->
            <th>ISBN </th>
            <th>Book Title</th>
            <th>Author</th>
            <th>Edition</th>
            <th>Year</th>
            <th>Category</th>
            <th>Reservable</th>
        </tr>

    <?php

    //This for loop uses the position variable and will keep output the rows of results until 5 results have been outputted or there is no  more results left
    for($Position = $Position; $Position < $New_Position && $Position < count($Author); $Position++)
    {
        echo ('<tr><td>');
        echo $ISBN[$Position];
        echo ("</td><td>");
        echo $BookTitle[$Position];
        echo ("</td><td>");
        echo $Author[$Position];
        echo ("</td><td>");
        echo $Edition[$Position];
        echo ("</td><td>");
        echo $Year[$Position];
        echo ("</td><td>");
        echo $Category_array[$Position];
        echo ("</td><td>");

         //Checks if the book is reserved or not
        if($Reserved[$Position] == "N")
        {
            //If the book is not reserved the reserve button will appear and allow the user to reserve the book
```

```
                    //When clicked it will send the user to the reserve page and pass the ISBN variable of the book that the user wants to reserve to that page
                    echo ('<a style = "color: #00008B;" href="Reserve.php?ISBN='.htmlentities($ISBN[$Position]).'">Reserve</a>'. "<br>");
                    echo ("</td></tr>");
                }
                else //If the is reserved it will display not reservable
                {
                    echo " Not Reservable <br> ";
                    echo ("</td></tr>");
                }
                $count = $count + 1; //Adds one to count to keep track of how many was outputted
            }
            echo "</table>";
            echo "</div>";
            //End of creation of the table

            //At this point this position variable is now whatever it is after displaying all the items in the for loop
            //Checks if position is greater then 5 which means that it is not on the first page of results meaning it needs to print previous page
            if($Position > 5)
            {
                //This is the previous page button - When clicked it wil load this page again and pass the position variable subtract how many results were outputted + 5 and the category the user selected back to the page
                //By subtracting how many results were outputted + 5 off he position variable it resets it back to what it is was when it started outputting the results on the previous page
                echo ('<a class = "page" href="Display_Category.php?Position='.htmlentities($Position-($count + 5)).'&Category='.htmlentities($Category).'">Previous Page</a>'. " ");
            }
            else //If the position is not greater then 5 the user is on the first page of results and the previous page button is not needed
            {
                //It just displays the text Previous page but with no button
                echo ('<a class = "page">Previous Page</a>');
            }

            //Checks if the position is less then how many rows of results there are by checking if it is less then the amount of items in the Author array
            if($Position < count($Author))
            {
                //If it less then it it means there are more results to display and the next page button needs to display
                //When clicked it will reload this page and pass the position variable and the category back to the page
                echo ('<a class = "page" href="Display_Category.php?Position='.htmlentities($Position).'&Category='.htmlentities($Category).'">Next Page</a>'. " ");
            }
            else //If the position is not less then it it means there is no more results and their is no more pages needed to be displayed
            {
                //It just displays the text next page but with no button
                echo ('<a class = "page">Next page</a>');
            }

        }
        else //If no results are found it will display "There are no books in the category"
        {
            echo '<div class = "display_box" style = "margin-bottom: 3%;">';
            echo "There are no books in this category";
            echo '</div>';
        }

        //Closes the database connection
        $conn->close();

    ?>

        <!-- This button allows the user to go back to the search page --->
        <div>
        <a class="Inside_Show_Category_TitleAuthor_Reserve" href="Category.php">
                <button style = "margin-top: 0px;" type = "button">
                    Go back to search
            </button>
        </a>
        </div>

        <footer style="position: fixed; bottom: 0;">
                <p>©Created by Ben Costello. All Rights Reserved.</p>
        </footer>

<?php } ?> <!-- Closes the else part of the check if the user is logged in --->
</body>
</html>
```

Show_BookTitle_Author.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Search by Book Title and/or Author</title>
    <link rel="stylesheet" href="Design.css"> <!-- Links to the css file --->
</head>
<body>

<?php
    session_start(); //Starts the user's session on this page

    //Checks if the user has logged in or not
    if( !isset($_SESSION["Username"]))
    {
        //If the user has not logged in they are sent back to the login page
        header("Location:Login.php");
    }
    else //If the user has logged in the code below will run and the page will load - End bracket at end of page
    {
        require_once("Connect.php");  //Connects the page to the database
        $Position = 0; //This is a variable that will be passed into the Show_Reserve page to display 5 books from the user's search on screen at a time
?>

        <!-- The header --->
        <header>
            <h1>Library System</h1>
        </header>

        <nav>
            <div class = "Options">
                <!-- Search button - brings the user to the search page --->
                <a style = "color: black;" class = "Options" href = "Search.php">Search</a>
                <?php
                    //Reserved books button - brings the user to the show_reserve page and passes the position variable to it it
                    echo ('<a style = "color: black;" href="Show_Reserve.php?Position='.htmlentities($Position).'">Reserved Books</a>');
                ?>
            </div>
            <div class = "Text">
                <?php
                    echo "Logged in as: " . $_SESSION["Username"];
                ?>
            </div>
            <div class = "Logout">
                <?php
                    //Logout button - brings the user to the logout page
                    echo ('<a style = "color: black;" href="Logout.php">Logout</a>');
                ?>
            </div>
        </nav>

    <?php

        //Gets the book title and author the user entered which is passed into the page
        $Search_BookTitle = $conn -> real_escape_string($_GET['Title']);
        $Search_Author = $conn -> real_escape_string($_GET['Author']);
        $Position = $conn -> real_escape_string($_GET['Position']); //Gets the position variable that is passed into the page
        $New_Position = $Position + 5; //This will be used so only 5 books are displayed and it stop at that position

        //Creates an array for all the columns in the books table in the database we are gonna display
        $BookTitle = array();
        $Author = array();
        $ISBN = array();
        $Reserved = array();
        $Edition = array();
        $Year = array();
        $Category_array = array();
        $x = 0; //Index variable for the arrays
        $count = 0; //This keeps track of how many books we have displayed on the page and will be used to go to the previous page of books
        $found = FALSE; //This is used to check if we found books that match/partially match what the user entered

        //This sql statement gets all the details of the books with a book title and author that are an exact match with what the user entered
        //Columns/Details it gets are - Book title, author, edition, year and reserved from the book table and the category description from the category table
        $sql = "SELECT ISBN,BookTitle,Author,Edition,Year,Reserved,CategoryDescription FROM Books join Category ON Books.CategoryCode = Category.CategoryID WHERE BookTitle = '$Search_BookTitle' AND Author = '$Search_Author'";
        $result = $conn->query($sql);
        if ($result->num_rows > 0)
        {
            while($row = $result->fetch_assoc())
            {
                //Puts all the details of the rows of books that were found into the correct arrays.
                $BookTitle[$x] = $row["BookTitle"];
                $Author[$x] = $row["Author"];
                $ISBN[$x] = $row["ISBN"];
                $Reserved[$x] = $row["Reserved"];
                $Edition[$x] = $row["Edition"];
                $Year[$x] = $row["Year"];
                $Category_array[$x] = $row["CategoryDescription"];
                $found = True; //Sets it to true as we found an exact match
                $x = $x + 1; //Increments the index
            }
        }

        //If there is not an exact match, found will at this point still false and we will run a partial search through the database
        //If there is an exact match, false will be true and it will not run the partial search
        if ($found == FALSE)
        {
            //This sql statement gets all the details of the books with a book title and author that partially have/contain the book title and author the user searched
            //Columns/Details it gets are - Book title, author, edition, year and reserved from the book table and the category description from the category table
            $sql2 = "SELECT ISBN,BookTitle,Author,Edition,Year,Reserved,CategoryDescription FROM Books join Category ON Books.CategoryCode = Category.CategoryID WHERE BookTitle like '%$Search_BookTitle%' AND Author like '%$Search_Author%'";
            $result2 = $conn->query($sql2);
            if ($result2->num_rows > 0)
            {
                while($row2 = $result2->fetch_assoc())
                {
                    //Puts all the details of the rows of books that were found into the correct arrays.
                    $BookTitle[$x] = $row2["BookTitle"];
                    $Author[$x] = $row2["Author"];
                    $ISBN[$x] = $row2["ISBN"];
                    $Reserved[$x] = $row2["Reserved"];
                    $Edition[$x] = $row2["Edition"];
                    $Year[$x] = $row2["Year"];
                    $Category_array[$x] = $row2["CategoryDescription"];
                    $x = $x + 1; //Increments the index
                    $found = True; //Sets it to true as we found an exact match
                }
            }
        }


        //Checks if found is false and if it ise it means there was no results found and will display this on screen
        if ($found == FALSE)
        {
            echo '<div class = "display_box" style = "margin-bottom: 3%;">';
            echo"No Results found";
            echo '</div>';
        }
        else //If there was results found, this code will display them on screen in a table
        {
            ?>


            <!-- Start creation of table which will display the result --->
            <div class = "display_box" style = "margin-top: 3%;">
            <table style = "border: 1px solid black;">
                <tr>
                    <!-- The headings for each column in the table --->
                    <th>ISBN </th>
                    <th>Book Title</th>
                    <th>Author</th>
                    <th>Edition</th>
                    <th>Year</th>
                    <th>Category</th>
                    <th>Reservable</th>
                </tr>

            <?php

            //This for loop uses the position variable and will keep output the rows of results until 5 results have been outputted or there is no  more
results left
            for($Position = $Position; $Position < $New_Position && $Position < count($BookTitle); $Position++)
            {
                //Uses the position variable as the index
                //Outputs the data at that index in all the arrays into the table
                echo ('<tr><td>');
                echo $ISBN[$Position];
                echo ("</td><td>");
                echo $BookTitle[$Position];
                echo ("</td><td>");
                echo $Author[$Position];
                echo ("</td><td>");
                echo $Edition[$Position];
                echo ("</td><td>");
                echo $Year[$Position];
                echo ("</td><td>");
                echo $Category_array[$Position];
        echo ("</td><td>");

        //Checks if the book is reserved or not
        if($Reserved[$Position] == "N")
        {
          //If the book is not reserved the reserve button will appear and allow the user to reserve the book
          //When clicked it will send the user to the reserve page and pass the ISBN variable to that page
          echo ('<a style = "color: #00008B;" href="Reserve.php?ISBN='.htmlentities($ISBN[$Position]).'">Reserve</a>'. "<br>");
          echo ("</td></tr>");
        }
        else //If the is reserved it will display not reservable
        {
          echo " Not Reservable <br> ";
          echo ("</td></tr>");
        }
        $count = $count + 1; //Adds one to count to keep track of how many was outputted
      }

      echo "</table>";
      echo "</div>";
      //End of creation of the table

      //At this point this position variable is now whatever it is after displaying all the items in the for loop
      //Checks if position is greater then 5 which means that it is not on the first page of results meaning it needs to print previous page
      if($Position > 5)
      {
        //This is the previous page button - When clicked it wil reload the this page and pass the position variable subtract how many results
were outputted + 5, the book title and the author the user selected back to the page
        //By subtracting how many results were outputted + 5 off he position variable it resets it back to what it is was when it started outputting
the results on the previous page
        echo ('<a class = "page" href="Show_BookTitle_Author.php?Position='.htmlentities($Position-($count +
5)).'&Title='.htmlentities($Search_BookTitle).'&Author='.htmlentities($Search_Author).'">Previous Page</a>'. " ");
      }
      else //If the position is not greater then 5 the user is on the first page of results and the previous page button is not needed
      {
        //It just displays the text Previous page but with no button
        echo ('<a class = "page">Previous Page</a>');
      }

      //Checks if the position is less then how many rows of results there are by checking if it is less then the amount of items in the book title
array
      if($Position < count($BookTitle))
      {
        //If it less then it it means there are more results to display and the next page button needs to display
        //When clicked it will reload this page and pass the position variable, the book title and the author back to the page
        echo ('<a class = "page"
href="Show_BookTitle_Author.php?Position='.htmlentities($Position).'&Title='.htmlentities($Search_BookTitle).'&Author='.htmlentities($Search_Au
thor).'">Next Page</a>'. " ");
      }
      else //If the position is not less then it it means there is no more results and their is no more pages needed to be displayed
      {
```

```
                //It just displays the text next page but with no button
                echo ('<a class = "page">Next page</a>');
            }

        }
        //Closes the database connection
        $conn->close();

    ?>

        <!-- This button allows the user to go back to the search page --->
        <div>
        <a class = "Inside_Show_Category_TitleAuthor_Reserve" href="BookTitle_Author.php">
                <button style = "margin-top: 0px;" type = "button">
                    Go back to search
            </button>
        </a>
        </div>

        <footer style="position: fixed; bottom: 0;">
                <p>©Created by Ben Costello. All Rights Reserved.</p>
        </footer>

<?php } ?> <!-- Closes the else part of the check if the user is logged in --->
</body>
</html>
```

Connect.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Connecting to database</title>
</head>
<body>
    <?php


        //This piece of code connects whatever page it is called in to the database
        $servername = "localhost";
        $username = "root";
        $password = "";
        $dbname = "Assignment";
        // Create connection
        $conn = new mysqli($servername,$username, $password, $dbname);
        // Check connection
        if ($conn->connect_error) {
            die("Connection failed: " . $conn->connect_error);
        }

    ?>
</body>
</html>
```

Reserve.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Reserving</title>
</head>
<body>

<?php

    session_start(); //Starts the users session on this page

    //Checks if the user has logged in or not
    if( !isset($_SESSION["Username"]))
    {
        //If the user has not logged in they are sent back to the login page
        header("Location:Login.php");
    }
    else //If the user has logged in the code will run and the page will load
    {

?>

        <!-- The header --->
        <header>
            <h1>Library System</h1>
        </header>

        <nav>
            <div class = "Options">
                <!-- Search button - brings the user to the search page --->
                <a style = "color: black;" class = "Options" href = "Search.php">Search</a>
                <?php
                    //Reserved books button - brings the user to the show_reserve page and passes the position variable to it it
                    echo ('<a style = "color: black;" href="Show_Reserve.php?Position='.htmlentities($Position).'">Reserved Books</a>');
                ?>
            </div>
            <div class = "Text">
                <?php
```

```
                    echo "Logged in as: " . $_SESSION["Username"];
                ?>
            </div>
            <div class = "Logout">
                <?php
                    //Logout button - brings the user to the logout page
                    echo ('<a style = "color: black;" href="Logout.php">Logout</a>');
                ?>
            </div>
        </nav>

<?php

        //Connects the page to the database
        require_once("Connect.php");
        $Username = $_SESSION["Username"]; //Gets the username of the user
        $ISBN = $conn -> real_escape_string($_GET['ISBN']); //Gets the ISBN of the book we are gonna reserve from the page the user came from
        $date = date('Y-m-d'); //Gets today's date in the format year-month-day
        $success = FALSE; //This is to check that the book was reserved successfully
        $Position = 0; //This is a variable that will be passed into the Show_Reserve page to display 5 books on screen at a time

        //This sql statement updates the reserved column of the specific book in the books table to Y to reserve the book
        $sql = "UPDATE Books Set Reserved = 'Y' WHERE ISBN = '$ISBN'";
        if ($conn->query($sql) === TRUE) {
            $success = TRUE;
        }
        else
        {
            echo '<div class = "display_box">';
            echo "There was a problem reserving the book";
            echo '</div>';

        }

        //Checks if the book was reserved successfully
        if ($success == TRUE)
        {
            //If successful this sql statement will insert the reservation that has been made into the reservation table
            $sql1 = "INSERT INTO Reservations (ISBN,Username,ReservedDate) VALUES ('$ISBN','$Username','$date')";
            if ($conn->query($sql1) === TRUE) {
                //If the statement is successful the user will be taken to the show_reserve page and the Position variable will be passed to it.
                echo "Book was reserved Successfully";
                header("Location: Show_Reserve.php?Position=".htmlentities($Position));
                return;
            }
            else
            {
                //If the statement is unsuccessful an error will display and the reserved column for tha specific book in the books table will be reset to N
                echo '<div class = "display_box">';
                echo "There was a problem reserving the book";
                echo '</div>';
        $sql2 = "UPDATE Books Set Reserved = 'N' WHERE ISBN = '$ISBN'";
        $conn->query($sql2);
      }
    }

    //Closes the database connection
    $conn->close();
?>
    <footer style="position: fixed; bottom: 0;">
      <p>©Created by Ben Costello. All Rights Reserved.</p>
    </footer>

<?php } ?> <!-- Closes the else part of the check if the user is logged in --->

?>



</body>
</html>
```

Show_Reserve.php

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Reserved Books</title>
  <link rel="stylesheet" href="Design.css"> <!-- Links to the css file --->
</head>
<body>

<?php
  session_start(); //Starts the user's session on this page

  //Checks if the user has logged in or not
  if( !isset($_SESSION["Username"]))
  {
    //If the user has not logged in they are sent back to the login page
    header("Location:Login.php");
  }
  else //If the user has logged in the code below will run and the page will load - End bracket at end of page
  {
    require_once("Connect.php"); //Connects the page to the database
    $Username = $_SESSION["Username"]; //Gets the username of the user

     //Creates an array for all the columns in the books table in the database we are gonna display
    $Author = array();
    $BookTitle = array();
```

```
$ISBN = array();
$reservedDate = array();
$Edition = array();
$Year = array();
$Category = array();
$x = 0; //Index variable for the arrays
$count = 0; //This keeps track of how many books we have displayed on the page and will be used to go to the previous page of books
$Position = $conn -> real_escape_string($_GET['Position']); //Gets the position variable which is passed into the page
$New_Position = $Position + 5; //This will be used so only 5 books are displayed and it stop at that position

//This code below will find all the books the user has reserved
//This sql statement gets the ISBN and reserved date from the reservations table where the username is the same as the user's  username in the database
$sql = "SELECT ISBN, ReservedDate FROM Reservations WHERE Username = '$Username'";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
// output data of each row
    while($row = $result->fetch_assoc())
    {
        //Puts the ISBN and ReservedDate variables from the sql statement into their appropriate arrays
        $ISBN[$x] = $row["ISBN"];
        $ReservedDate[$x] = $row["ReservedDate"];

        //This sql statement gets the book that has that ISBN number
        //Columns/Details it gets are - Book title, author, edition, year and reserved from the book table and the category description from the category table
        $sql2 = "SELECT BookTitle, Author, Edition, Year, CategoryDescription FROM Books join Category ON Books.CategoryCode = Category.CategoryID WHERE ISBN = '$ISBN[$x]'";
        $result2 = $conn->query($sql2);
        if ($result2->num_rows > 0) {
            while($row2 = $result2->fetch_assoc())
            {
                //Puts all the columns of the  book that was found into the correct arrays.
                $BookTitle[$x] = $row2['BookTitle'];
                $Author[$x] = $row2['Author'];
                $Edition[$x] = $row2['Edition'];
                $Year[$x] = $row2['Year'];
                $Category[$x] = $row2['CategoryDescription'];
                $x = $x + 1; //Increments the index
            }
        }
    }
}
//End of finding all the books the user has reserved

?>

<!-- The header --->
<header>
    <h1>Library System</h1>
</header>
```

```
<nav>
<div class = "Options">
<!-- Search button - brings the user to the search page --->
<a style = "color: black;" class = "Options" href = "Search.php">Search</a>
<?php
  //Reserved books button - brings the user to the show_reserve page and passes the position variable to it it
  echo ('<a style = "color: black;" href="Show_Reserve.php?Position='.htmlentities($Position).'">Reserved Books</a>');
?>
</div>
<div class = "Text">
<?php
  echo "Logged in as: " . $_SESSION["Username"];
?>
</div>
<div class = "Logout">
<?php
  //Logout button - brings the user to the logout page
  echo ('<a style = "color: black;" href="Logout.php">Logout</a>');
?>
</div>

</nav>
<?php

//Checks if the user has any books reserved by checking if the amount of items in the array book title is above 0
if (count($BookTitle) > 0)
{
  //If there was books it will display the rows of books found in a table
  ?>

  <!-- Start creation of table which will display the result --->
  <div class = "display_box" style = "margin-top: 3%;">
  <table style = "border: 1px solid black;">
    <tr>
      <!-- The headings for each column in the table --->
      <th>ISBN</th>
      <th>Book Title</th>
      <th>Author</th>
      <th>Edition</th>
      <th>Year</th>
      <th>Category</th>
      <th>Reserved Date</th>
      <th>Unreserve</th>
    </tr>

  <?php

  //This for loop uses the position variable and will keep output the rows of results until 5 results have been outputted or there is no more results left
  for($Position = $Position; $Position < $New_Position && $Position < count($ISBN); $Position++)
```

```
{
    echo ('<tr><td>');
    echo $ISBN[$Position];
    echo ("</td><td>");
    echo $BookTitle[$Position];
    echo ("</td><td>");
    echo $Author[$Position];
    echo ("</td><td>");
    echo  $Edition[$Position];
    echo ("</td><td>");
    echo  $Year[$Position];
    echo ("</td><td>");
    echo  $Category[$Position];
    echo ("</td><td>");
    echo  $ReservedDate[$Position];
    echo ("</td><td>");
    //Outputs the unreserve button to allow the user to remove/unreserve a book they have reserved
    //When clicked it sends the user to the unreserve page and passes the ISBN variable the user wants to unreserve to that page
    echo ('<a style = "color: #00008B;" href="Unreserve.php?ISBN='.htmlentities($ISBN[$Position]).'">Unreserve</a>');
    echo ("</td></tr>");
    $count = $count + 1; //Adds one to count to keep track of how many was outputted
}
echo "</table>";
echo "</div>";
 //End of creation of the table


//At this point this position variable is now whatever it is after displaying all the items in the for loop
//Checks if position is greater then 5 which means that it is not on the first page of results meaning it needs to print previous page
if($Position > 5)
{
    //This is the previous page button - When clicked it wil load this page again and pass the position variable subtract how many results were outputted + 5 back to the page
    //By subtracting how many results were outputted + 5 off he position variable it resets it back to what it is was when it started outputting the results on the previous page
    echo ('<a class = "page" href="Show_Reserve.php?Position='.htmlentities($Position-($count + 5)).'">Previous Page</a>');
}
else //If the position is not greater then 5 the user is on the first page of results and the previous page button is not needed
{
    //It just displays the text Previous page but with no button
    echo ('<a class = "page">Previous Page</a>');
}

//Checks if the position is less then how many rows of results there are by checking if it is less then the amount of items in the ISBN array
if($Position < count($ISBN))
{
    //If it less then it it means there are more results to display and the next page button needs to display
    //When clicked it will reload this page and pass the position variable back to the page
    echo ('<a class = "page" href="Show_Reserve.php?Position='.htmlentities($Position).'">Next Page</a>');
}
else //If the position is not less then it it means there is no more results and their is no more pages needed to be displayed
```

```
            {
                //It just displays the text next page but with no button
                echo ('<a class = "page">Next page</a>');
            }
        }
        else//If the user has no books reserved it will display "You have no books reserved"
        {
            echo '<div class = "display_box" style="margin-bottom: 40px;">';
            echo "You have no books reserved";
            echo '</div>';
        }

        //Closes the database connection
        $conn->close();
?>

        <!-- This button allows the user to go back to the search page --->
        <div>
        <a class = "Inside_Show_Category_TitleAuthor_Reserve" href="Search.php">
                <button style = "margin-top: 0px;" type = "button">
                    Go back to search
            </button>
        </a>
        </div>


        <footer style="position: fixed; bottom: 0;">
                <p>©Created by Ben Costello. All Rights Reserved.</p>
        </footer>




<?php } ?> <!-- Closes the else part of the check if the user is logged in --->
</body>
</html>
```

Unreserve.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Unreserving book</title>
</head>
<body>
```

```
<?php
    session_start(); //Starts the user's session on this page

    //Checks if the user has logged in or not
    if( !isset($_SESSION["Username"]))
    {
        //If the user has not logged in they are sent back to the login page
        header("Location:Login.php");
    }
    else //If the user has logged in the code below will run and the page will load - End bracket at end of page
    {
?>

        <!-- The header --->
        <header>
            <h1>Library System</h1>
        </header>

        <nav>
            <div class = "Options">
                <!-- Search button - brings the user to the search page --->
                <a style = "color: black;" class = "Options" href = "Search.php">Search</a>
                <?php
                    //Reserved books button - brings the user to the show_reserve page and passes the position variable to it it
                    echo ('<a style = "color: black;" href="Show_Reserve.php?Position='.htmlentities($Position).'">Reserved Books</a>');
                ?>
            </div>
            <div class = "Text">
                <?php
                    echo "Logged in as: " . $_SESSION["Username"];
                ?>
            </div>
            <div class = "Logout">
                <?php
                    //Logout button - brings the user to the logout page
                    echo ('<a style = "color: black;" href="Logout.php">Logout</a>');
                ?>
            </div>
        </nav>


<?php
        //This is a variable that will be passed into the Show_Reserve page to display 5 books from the user's search on screen at a  time
        require_once("Connect.php");
        $Username = $_SESSION["Username"]; //Gets the username of the user
        $ISBN = $conn -> real_escape_string($_GET['ISBN']); //Gets the ISBN of the book we are gonna reserve from the page the user came from
        $success = FALSE; //This is to check that the book was reserved successfully
        $Position = 0; //This is a variable that will be passed into the Show_Reserve page to display 5 books on screen at a time

        //This sql statement updates the reserved column of the specific book in the books table to N to reserve the book
        $sql = "UPDATE Books Set Reserved = 'N' WHERE ISBN = '$ISBN'";
    if ($conn->query($sql) === TRUE) {
      $success = TRUE;
    }
    else
    {
      echo '<div class = "display_box">';
      echo "There was a problem trying to unreserve the book";
      echo '</div>';
    }

    //Checks if the book was unreserved successfully
    if ($success == TRUE)
    {
      //If successful this sql statement will remove the reservation from the reservation table
      $sql1 = "DELETE FROM Reservations WHERE ISBN = '$ISBN'";
      if ($conn->query($sql1) === TRUE)
      {
        //If the statement is successful the user will be taken to the show_reserve page and the Position variable will be passed to it.
        echo "Book was unreserved Successfully";
        header("Location: Show_Reserve.php?Position=".htmlentities($Position));
      }
      else
      {
        //If the statement is unsuccessful an error will display and the reserved column for the specific book in the books table will be reset to Y
        echo '<div class = "display_box">';
        echo "There was a problem trying to unreserve the book";
        echo '</div>';
        $sql2 = "UPDATE Books Set Reserved = 'Y' WHERE ISBN = '$ISBN'";
        $conn->query($sql2);
      }
    }

    //Closes the database connection
    $conn->close();
?>

    <footer style="position: fixed; bottom: 0;">
        <p>©Created by Ben Costello. All Rights Reserved.</p>
    </footer>

<?php } ?> <!-- Closes the else part of the check if the user is logged in --->

?>



</body>
</html>
```

Logout.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Logging out</title>
</head>
<body>
    <?php

        session_start(); //Starts the user's session on this page

        //Checks if the user has logged in or not
        if( !isset($_SESSION["Username"]))
        {
            //If the user has not logged in they are sent back to the login page
            header("Location:Login.php");
        }
        else //If the user has logged in the code, they will be logged out
        {
            //Deletes their session and then sends the user to the login page
            session_destroy();
            header("Location: Login.php");
        }
    ?>
</body>
</html>
```

Design.css

```
html, body
{
    padding: 0px;
    margin: 0px;
    background-color: rgb(227, 211, 173);
    font-size: 16px;
}


header{
    text-align: center;
    color: rgb(60, 58, 58);
    background-color:rgb(210, 156, 18);
    border-radius: 0 0 15px 15px;
    width:100%vw;
    font-size: 1.3em;
    height: 2rem;
    padding: 20px 0px 30px 0px;
    margin-bottom: 10px;
```

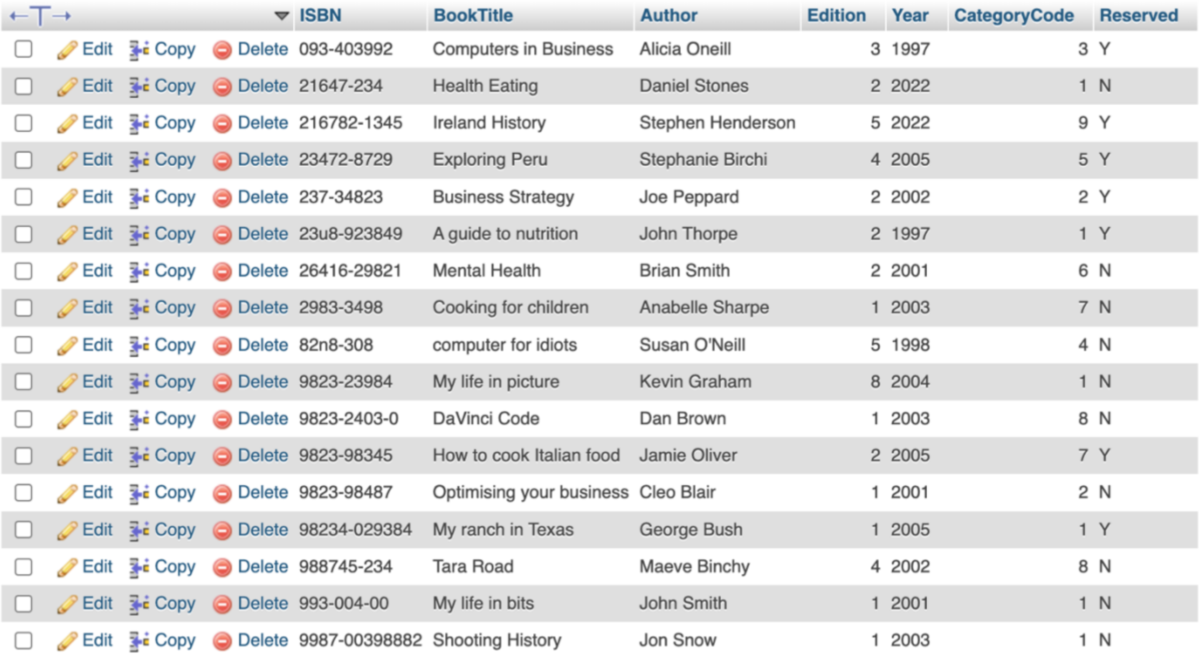
```
}

footer
{
    text-align: center;
    color: rgb(60, 58, 58);
    background-color:rgb(210, 156, 18);
    border-radius: 15px 15px 0px 0px;
    width:100%;
    font-size: 1.3em;
    height: 4rem;
    padding: 30px 0px 30px 0px;
}

h1
{
    margin: 0;
}


nav{
    background-color:  rgb(210, 156, 18);
    padding: 14px 0px;
    position: sticky;
    font-size: 1.3em;
    width: 100%vw;
    border-radius: 15px;
    box-shadow: 3px 3px 5px #888;
    overflow: hidden;
}


.Options
{
    display: inline;
    padding: 0px 10px 0px 20px;
}

.Text
{
    padding: 0px 500px 0px 345px;
    display: inline;
}

#message_login_register
{
    padding: 0px 420px 0px 317px;
    display: inline;
}
.Logout
{
    display:inline;
    padding: 0px 20px 0px 0px;;
    margin: 0;
}

.Selection button
{
    text-align: center;
    font-size: 5em;
    background-color:rgb(211, 125, 40);
    color: rgb(0, 0, 0);
    cursor: pointer;
    width: 48vw;
    height: 500px;
    border-radius: 15px;
}

.Selection
{
    padding: 20px 0px 20px 15px;
    display: inline-block;
}

#Search_Category
{
    text-align: center;
    font-size: 2em;
    background-color:rgb(211, 125, 40);
    color: rgb(0, 0, 0);
    width: 48vw;
    border-radius: 15px;
    margin: 110px 0px 33px 375px;
    padding: 30px 0px 5px 0px;
}


#Menu
{
    width: 50%;
    height: 50%;
    font-size: 70%;
    margin: 20px 0px 20px 0px;
}


form input
{
    font-size: 70%;
    margin: 0px 0px 20px 0px;
}
```

```
#InsideCategory button
{
    text-align: center;
    font-size: 2em;
    background-color:rgb(211, 125, 40);
    color: rgb(0, 0, 0);
    cursor: pointer;
    width: 45%;
    border-radius: 15px;
    padding: 10px 10px;
    margin: 5px 0px 138px 393px;
}

#InsideTitleAuthor button
{
    text-align: center;
    font-size: 2em;
    background-color:rgb(211, 125, 40);
    color: rgb(0, 0, 0);
    cursor: pointer;
    width: 45%;
    border-radius: 15px;
    padding: 10px 10px;
    margin: 5px 0px 58px 393px;
}

.Inside_Show_Category_TitleAuthor_Reserve button
{
    text-align: center;
    font-size: 2em;
    background-color:rgb(211, 125, 40);
    color: rgb(0, 0, 0);
    cursor: pointer;
    width: 45%;
    border-radius: 15px;
    padding: 10px 10px;
    margin: 5px 0px 0px 393px;

}
.Login_Search_Register_box
{
    text-align: center;
    font-size: 2em;
    background-color:rgb(211, 125, 40);
    color: rgb(0, 0, 0);
    width: 48vw;
    border-radius: 15px;
    margin: 50px 0px 0px 375px;
    padding: 5px 0px 0px 0px;
}
```

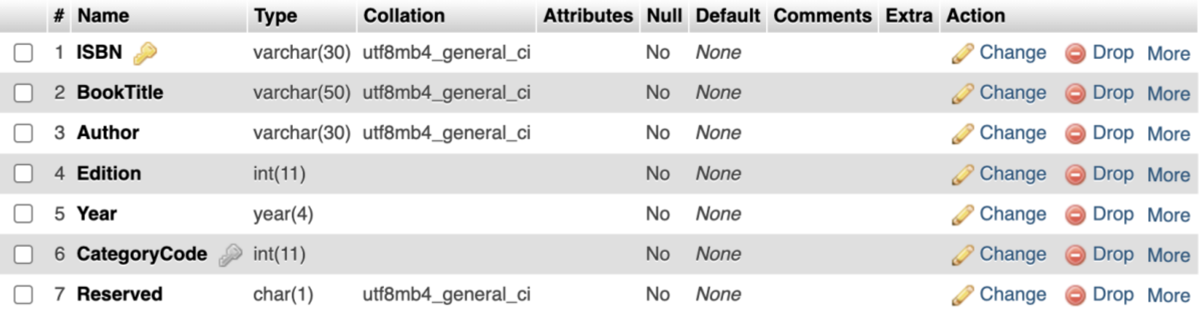
```
#padding_and_or
{
    margin-bottom: 0.1;
    margin-top: 0.1;
    padding-bottom: 0.5;
    font-size: 1e;
}


#InsideTitleAuthor button
{
    text-align: center;
    font-size: 2em;
    background-color:rgb(211, 125, 40);
    color: rgb(0, 0, 0);
    cursor: pointer;
    width: 45%;
    border-radius: 15px;
    padding: 10px 10px;
    margin: 5px 0px 58px 393px;
}


#Inside_Login_Register_button button
{
    text-align: center;
    font-size: 1.5rem;
    color: rgb(0, 0, 0);
    margin: 0px 0px 30px 0px;
}

table
{
    width: 100%;
}

.display_box
{
    text-align: center;
    font-size: 150%;
    background-color:rgb(211, 125, 40);
    color: rgb(0, 0, 0);
    border-radius: 15px;
    margin: 7% 10% 0% 10%;
    padding: 20px;
}

table, th,td
{
    border: 3px solid black;
    border-collapse: collapse;
}

.page, .page a
{
    text-align: center;
    display: inline-block;
    background-color:rgb(211, 125, 40);
    color:black;
    font-size: 1.5rem;
    padding: 10px 30px;
    border-radius: 15px;
    margin: 40px 0px 40px 363px;
}
```

Database

Books table

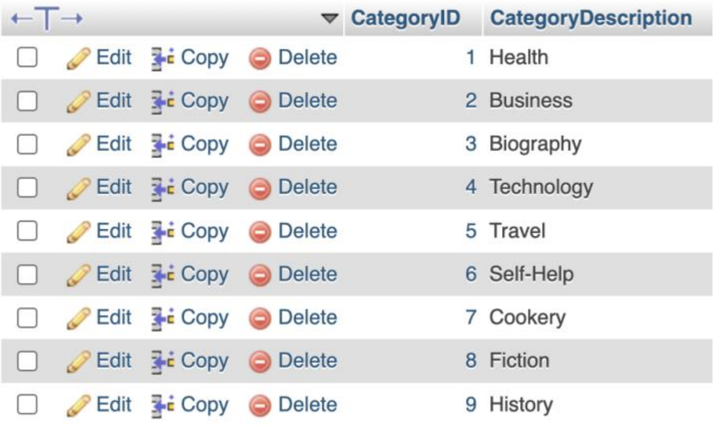


←T→			▼	ISBN	BookTitle	Author	Edition	Year	CategoryCode	Reserved
□	Edit	Copy	Delete	093-403992	Computers in Business	Alicia Oneill	3	1997	3	Y
□	Edit	Copy	Delete	21647-234	Health Eating	Daniel Stones	2	2022	1	N
□	Edit	Copy	Delete	216782-1345	Ireland History	Stephen Henderson	5	2022	9	Y
□	Edit	Copy	Delete	23472-8729	Exploring Peru	Stephanie Birchi	4	2005	5	Y
□	Edit	Copy	Delete	237-34823	Business Strategy	Joe Peppard	2	2002	2	Y
□	Edit	Copy	Delete	23u8-923849	A guide to nutrition	John Thorpe	2	1997	1	Y
□	Edit	Copy	Delete	26416-29821	Mental Health	Brian Smith	2	2001	6	N
□	Edit	Copy	Delete	2983-3498	Cooking for children	Anabelle Sharpe	1	2003	7	N
□	Edit	Copy	Delete	82n8-308	computer for idiots	Susan O'Neill	5	1998	4	N
□	Edit	Copy	Delete	9823-23984	My life in picture	Kevin Graham	8	2004	1	N
□	Edit	Copy	Delete	9823-2403-0	DaVinci Code	Dan Brown	1	2003	8	N
□	Edit	Copy	Delete	9823-98345	How to cook Italian food	Jamie Oliver	2	2005	7	Y
□	Edit	Copy	Delete	9823-98487	Optimising your business	Cleo Blair	1	2001	2	N
□	Edit	Copy	Delete	98234-029384	My ranch in Texas	George Bush	1	2005	1	Y
□	Edit	Copy	Delete	988745-234	Tara Road	Maeve Binchy	4	2002	8	N
□	Edit	Copy	Delete	993-004-00	My life in bits	John Smith	1	2001	1	N
□	Edit	Copy	Delete	9987-00398882	Shooting History	Jon Snow	1	2003	1	N

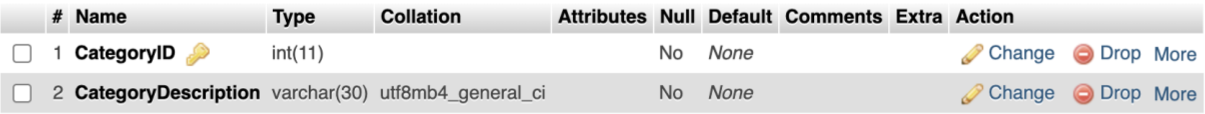


	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**ISBN** 🔑	varchar(30)	utf8mb4_general_ci		No	*None*			Change Drop More
□	2	**BookTitle**	varchar(50)	utf8mb4_general_ci		No	*None*			Change Drop More
□	3	**Author**	varchar(30)	utf8mb4_general_ci		No	*None*			Change Drop More
□	4	**Edition**	int(11)			No	*None*			Change Drop More
□	5	**Year**	year(4)			No	*None*			Change Drop More
□	6	**CategoryCode** 🔑	int(11)			No	*None*			Change Drop More
□	7	**Reserved**	char(1)	utf8mb4_general_ci		No	*None*			Change Drop More

Category Table

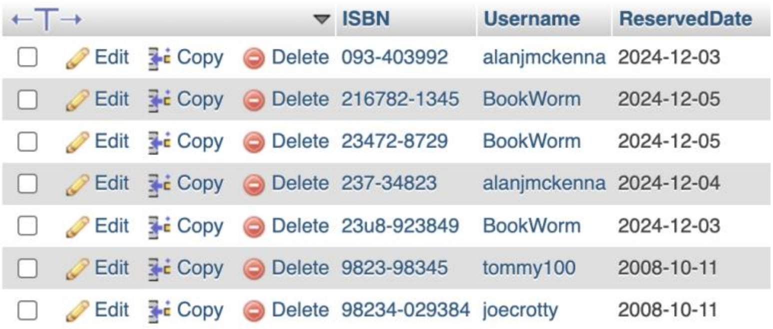


←T→			▼	CategoryID	CategoryDescription
□	Edit	Copy	Delete	1	Health
□	Edit	Copy	Delete	2	Business
□	Edit	Copy	Delete	3	Biography
□	Edit	Copy	Delete	4	Technology
□	Edit	Copy	Delete	5	Travel
□	Edit	Copy	Delete	6	Self-Help
□	Edit	Copy	Delete	7	Cookery
□	Edit	Copy	Delete	8	Fiction
□	Edit	Copy	Delete	9	History

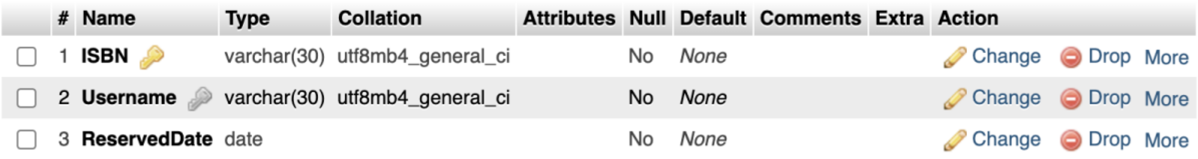


	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**CategoryID** 🔑	int(11)			No	*None*			Change Drop More
□	2	**CategoryDescription**	varchar(30)	utf8mb4_general_ci		No	*None*			Change Drop More

Reservations table

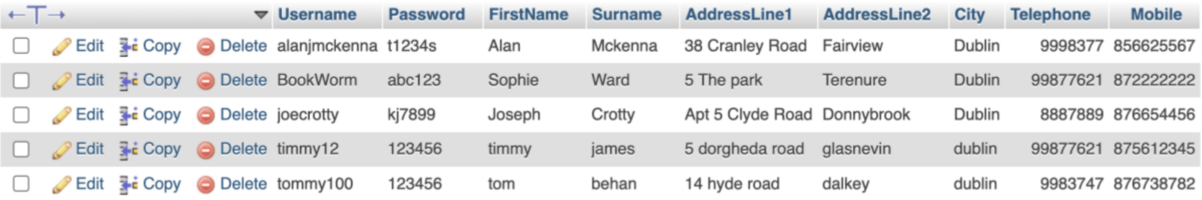


←T→				ISBN	Username	ReservedDate
□	Edit	Copy	Delete	093-403992	alanjmckenna	2024-12-03
□	Edit	Copy	Delete	216782-1345	BookWorm	2024-12-05
□	Edit	Copy	Delete	23472-8729	BookWorm	2024-12-05
□	Edit	Copy	Delete	237-34823	alanjmckenna	2024-12-04
□	Edit	Copy	Delete	23u8-923849	BookWorm	2024-12-03
□	Edit	Copy	Delete	9823-98345	tommy100	2008-10-11
□	Edit	Copy	Delete	98234-029384	joecrotty	2008-10-11

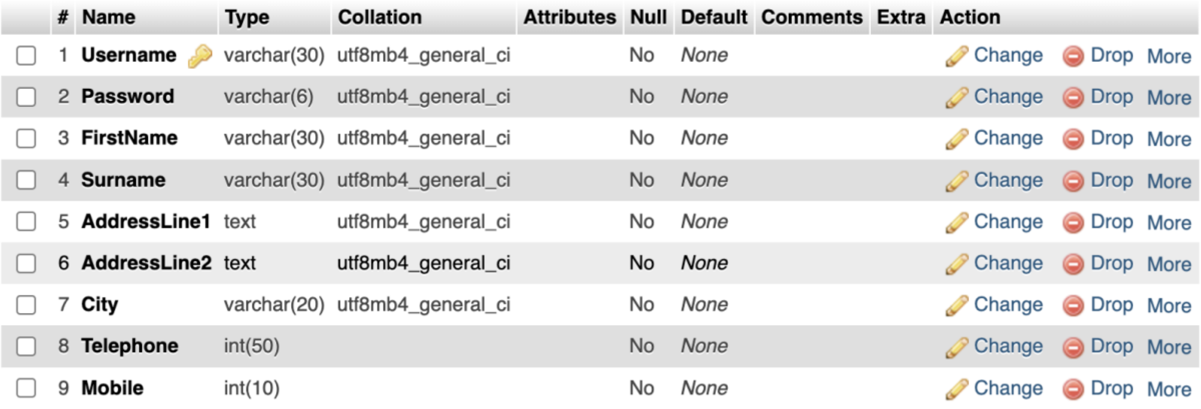


	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
□	1	**ISBN**	varchar(30)	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	2	**Username**	varchar(30)	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	3	**ReservedDate**	date			No	*None*			Change	Drop	More

Users table



←T→				Username	Password	FirstName	Surname	AddressLine1	AddressLine2	City	Telephone	Mobile
□	Edit	Copy	Delete	alanjmckenna	t1234s	Alan	Mckenna	38 Cranley Road	Fairview	Dublin	9998377	856625567
□	Edit	Copy	Delete	BookWorm	abc123	Sophie	Ward	5 The park	Terenure	Dublin	99877621	872222222
□	Edit	Copy	Delete	joecrotty	kj7899	Joseph	Crotty	Apt 5 Clyde Road	Donnybrook	Dublin	8887889	876654456
□	Edit	Copy	Delete	timmy12	123456	timmy	james	5 dorgheda road	glasnevin	dublin	99877621	875612345
□	Edit	Copy	Delete	tommy100	123456	tom	behan	14 hyde road	dalkey	dublin	9983747	876738782



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
□	1	**Username**	varchar(30)	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	2	**Password**	varchar(6)	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	3	**FirstName**	varchar(30)	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	4	**Surname**	varchar(30)	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	5	**AddressLine1**	text	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	6	**AddressLine2**	text	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	7	**City**	varchar(20)	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	8	**Telephone**	int(50)			No	*None*			Change	Drop	More
□	9	**Mobile**	int(10)			No	*None*			Change	Drop	More