

IS ELEC 02 | WEB SYSTEM

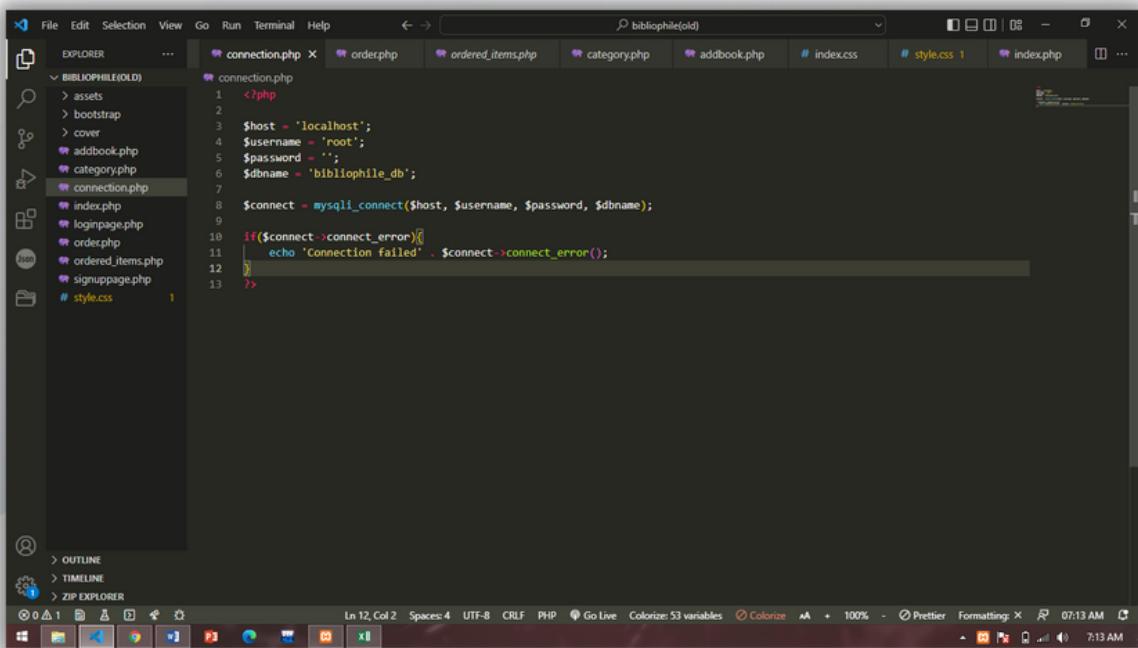
PHP SCRIPTS

MIDTERM PROJECT

GROUP 4:

Maricar Romero
Reggie Reblando
Alyssa Ragos
Jayzielyn Oguira

CONNECTING TO DATABASE



The screenshot shows a code editor window with the title "bibliophile(OLD)". The left sidebar displays a file tree for a project named "BIBLIOPHILE(OLD)" containing files like assets, bootstrap, cover, addbook.php, category.php, connection.php, index.php, loginpage.php, order.php, ordered_items.php, and signuppage.php, along with a style.css file. The main editor area contains the following PHP code:

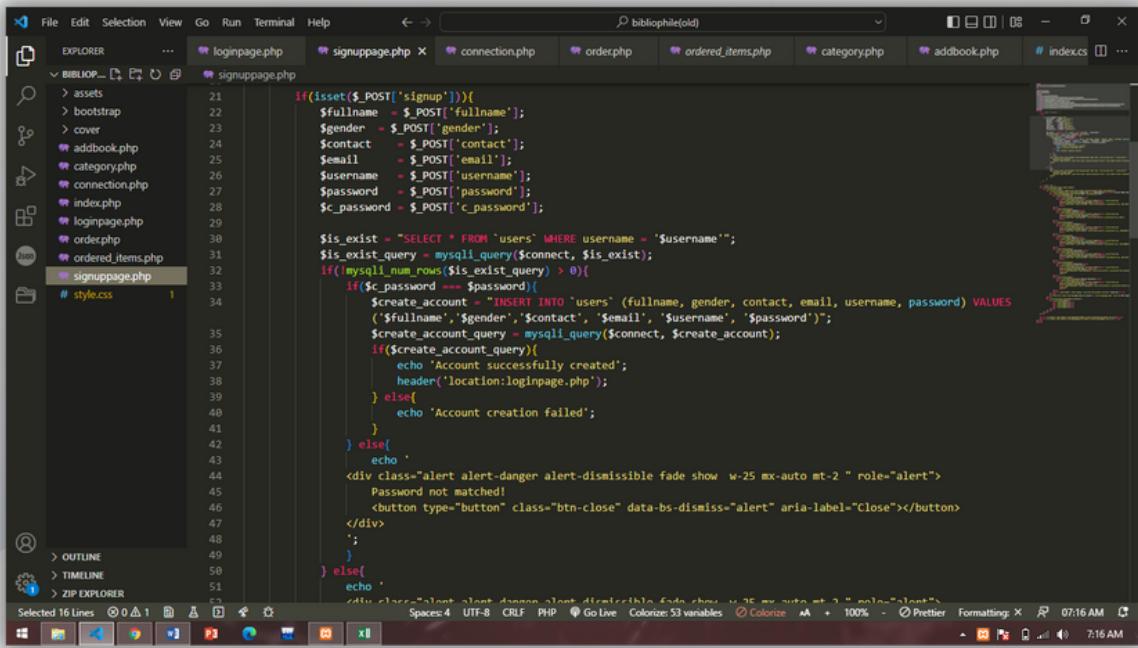
```
<?php
$host = 'localhost';
$username = 'root';
$password = '';
$dbname = 'bibliophile_db';

$connect = mysqli_connect($host, $username, $password, $dbname);
if($connect >connect_error){
    echo 'Connection failed' . $connect->connect_error();
}
?>
```

The status bar at the bottom shows "Ln 12, Col 2" and "Formatting: 07:13 AM".

First we declared 4 variables with server and database configuration. In order to connect to the database, we used the `mysqli_connect()` function and passed the 4 variables inside the function. Then we used an if condition to display an error if the MySQL connection and configuration were incorrect.

INSERTING DATA TO DATABASE (SIGN-UP)



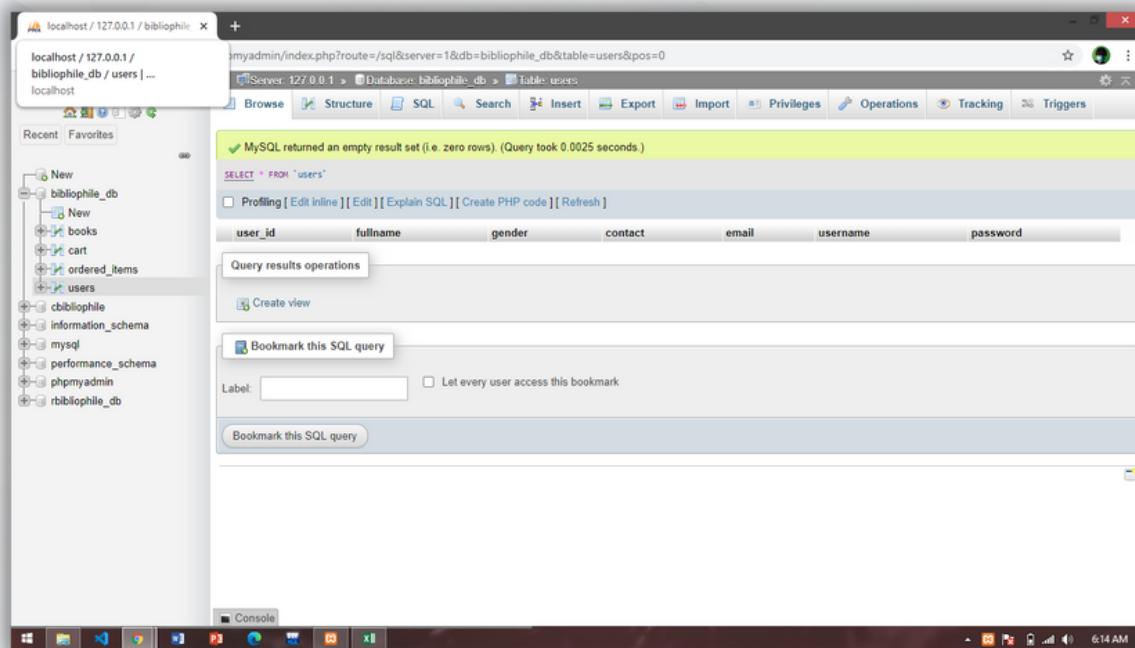
The screenshot shows a code editor window with multiple tabs open. The active tab is 'signuppage.php'. The code within this tab implements a sign-up process. It starts by checking if the 'signup' button has been pressed via the `$_POST` superglobal. If true, it initializes variables for fullname, gender, contact, email, username, password, and c_password. It then performs a `SELECT` query to check if the username already exists in the 'users' database table. If no user is found (rows > 0), it checks if the password matches the confirmation password. If they match, it executes an `INSERT INTO` query to add a new user record. Finally, it outputs a success message or an error message indicating that the password did not match.

```
if(isset($_POST['signup'])){
    $fullname = $_POST['fullname'];
    $gender = $_POST['gender'];
    $contact = $_POST['contact'];
    $email = $_POST['email'];
    $username = $_POST['username'];
    $password = $_POST['password'];
    $c_password = $_POST['c_password'];

    $is_exist = "SELECT * FROM `users` WHERE username = '$username'";
    $is_exist_query = mysqli_query($connect, $is_exist);
    if(mysqli_num_rows($is_exist_query) > 0){
        if($c_password === $password){
            $create_account = "INSERT INTO `users` (fullname, gender, contact, email, username, password) VALUES
                ('$fullname', '$gender', '$contact', '$email', '$username', '$password')";
            $create_account_query = mysqli_query($connect, $create_account);
            if($create_account_query){
                echo 'Account successfully created';
                header('location:loginpage.php');
            } else{
                echo 'Account creation failed';
            }
        } else{
            echo '
                <div class="alert alert-danger alert-dismissible fade show w-25 mx-auto mt-2" role="alert">
                    Password not matched!
                </button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
                </div>
            ';
        }
    } else{
        echo '
            <div class="alert alert-success alert-dismissible fade show w-25 mx-auto mt-2" role="alert">
                User successfully registered!
            </button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
            </div>
        ';
    }
}
```

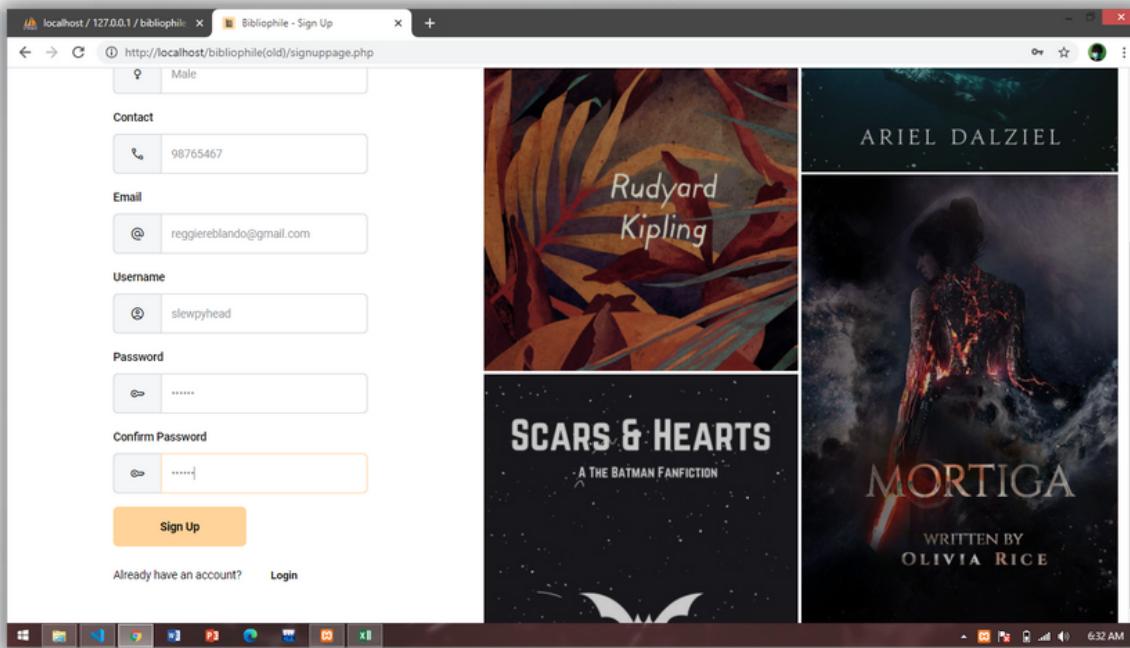
On the sign up page, we use the `isset()` function to detect if the sign up button is pressed or clicked by the user. If the condition is true, we declare seven variables and pass the information entered by the user to the signup field. Then we used a `SELECT` query to see if the users' information was in the database. If the information exists, we display a notification error to the users. If the information does not exist, we use `if` and `else if` statements to validate the information entered by the users, like if all the fields are not empty, check the username if valid, and make sure the password matches. If all the validations are passed, then we use the `INSERT` query to insert the record into the database and to register the users.

DATABASE WITHOUT RECORDS:



SIGN-UP FORM:

A screenshot of a web browser displaying a sign-up page. The URL is http://localhost/bibliophile/old/signuppage.php. The page has a heading 'Create Your Bibliophile Account Right Now!'. On the left, there are input fields for 'Fullname' (Reggie Reblado), 'Gender' (Male), 'Contact' (98765467), 'Email' (reggiereblando@gmail.com), 'Username' (slewyhead), and 'Password'. To the right of the form are two book covers: 'the JUNGLE BOOK' by Rudyard Kipling and 'CROWN NEST' by Ariel Dalziel. The bottom status bar shows the time as 6:32 AM.

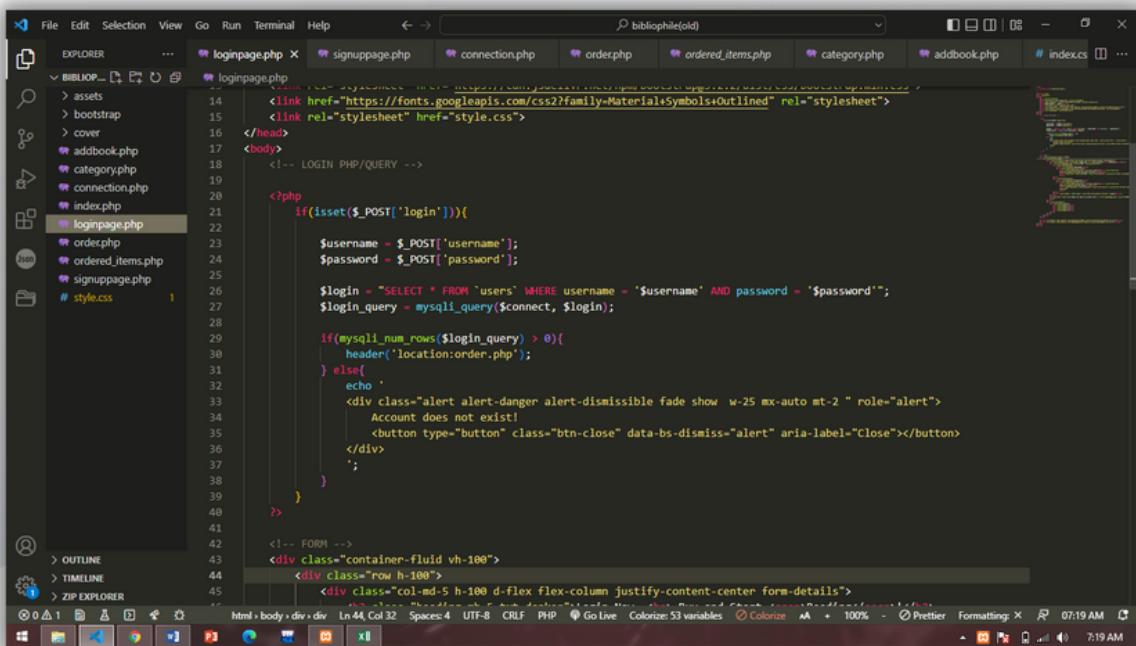


DATABASE WITH RECORDS:

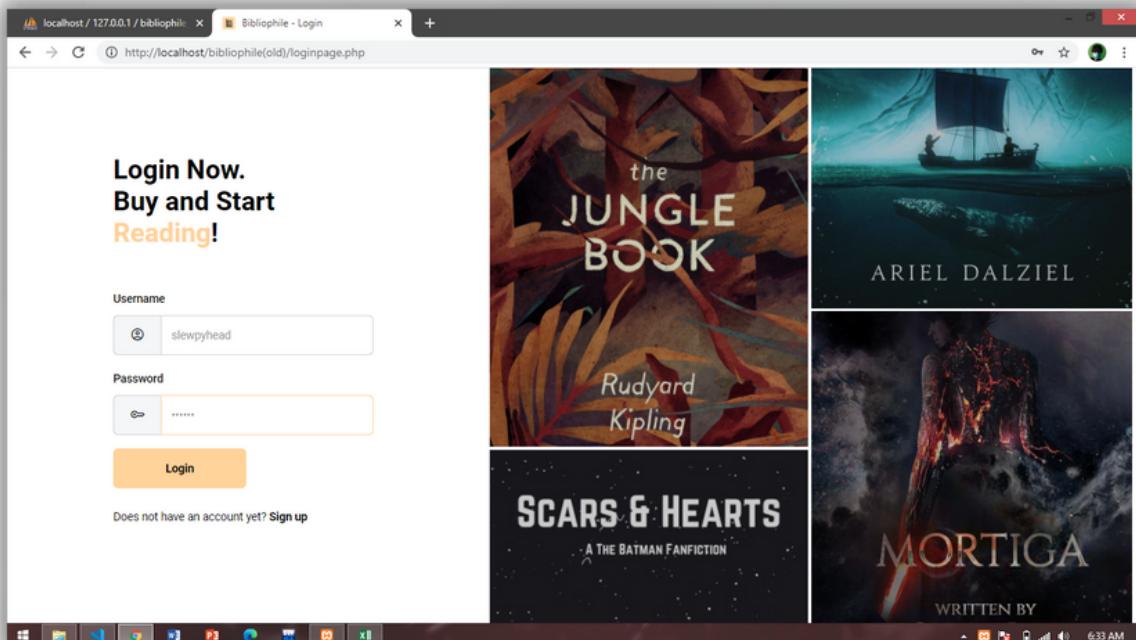
Showing rows 0 - 9 (10 total, Query took 0.0024 seconds.)								
<input type="checkbox"/> Profiling <input type="checkbox"/> Edit inline <input type="checkbox"/> Edit <input type="checkbox"/> Explain SQL <input type="checkbox"/> Create PHP code <input type="checkbox"/> Refresh								
<input type="checkbox"/> Show all Number of rows: 25 Filter rows: Search this table Sort by key: None								
<input type="checkbox"/> Options								
	<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	1	Reggie Reblando	Male	98765467
bibliophile_db						reggiereblando@gmail.com	slewyhead	ziezie
					2	Maricar Romero	Female	94653274
					3	Alyssa Ragos	Female	95736781
					4	Jayzielyn Oguira	Female	98765434
					5	Juan dela Cruz	Male	98765436
					6	Jie Reblando	Male	98645456
					7	Mac Reblando	Male	98765432
					8	Mon Reblando	Male	98765423
					16	Yve Akira	Female	925473923
					17	Lelouch Britania	Male	2147483647

SUPPLEMENTARY COMMAND: SELECTING DATA FROM DATABASE (LOG-IN)

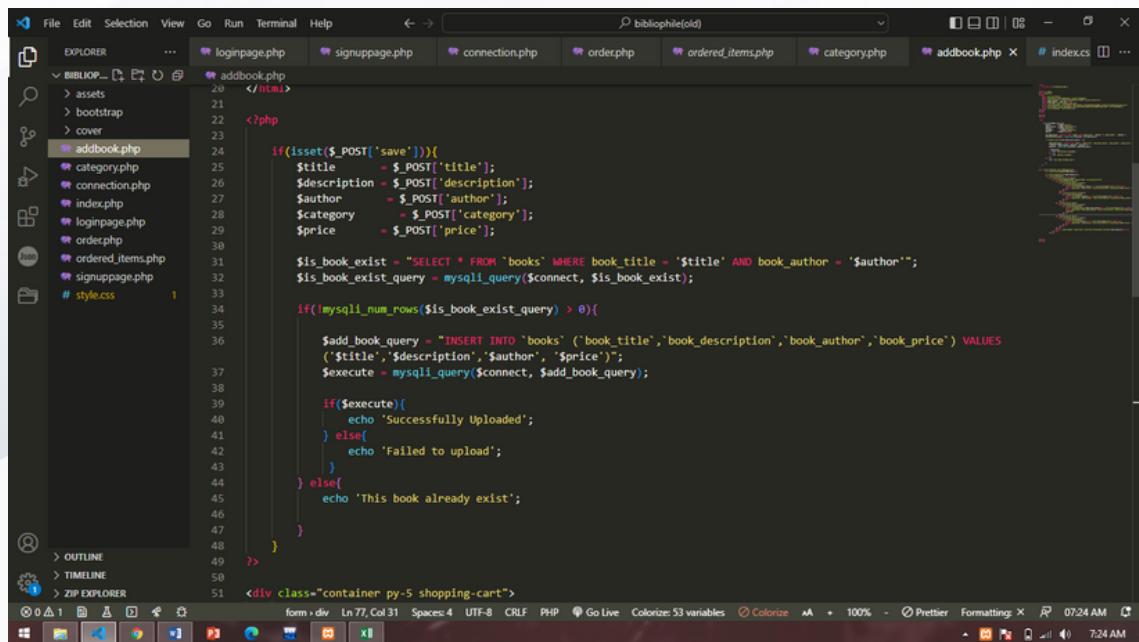
Just like on the sign up page, we also used the `isset()` function in the login form to determine if the login button was clicked by the user. However, unlike on the sign up page where an `INSERT` query is used, in the login form we used a `SELECT` query and an `if-else` condition in order to search if the information that has been input from the form (username and password) has a record in the database. If the condition is met, the user will be redirected to the books section where he/she can order.



A screenshot of a code editor showing the `loginpage.php` file. The code contains PHP logic to handle a login attempt. It checks if the `'login'` button was submitted via POST. If so, it retrieves the `username` and `password` from the POST data. It then runs a `SELECT` query to find a user with that specific combination. If no user is found (`mysql_num_rows > 0`), it outputs an alert message indicating the account does not exist. Otherwise, it redirects the user to the `order.php` page. The code editor interface includes an Explorer sidebar with other files like `index.php`, `connection.php`, and `style.css`, and various toolbars at the bottom.



INSERTING BOOKS TO DATABASE (BOOKS)



The screenshot shows a code editor interface with multiple tabs open. The active tab is 'addbook.php'. The code in the editor is as follows:

```
</html>
<?php
if(isset($_POST['save'])){
    $title = $_POST['title'];
    $description = $_POST['description'];
    $author = $_POST['author'];
    $category = $_POST['category'];
    $price = $_POST['price'];

    $is_book_exist = "SELECT * FROM `books` WHERE book_title = '$title' AND book_author = '$author'";
    $is_book_exist_query = mysqli_query($connect, $is_book_exist);

    if(!$is_book_exist_query) {
        echo "Failed to connect to the database";
    } else {
        if(mysqli_num_rows($is_book_exist_query) > 0) {
            $add_book_query = "INSERT INTO `books` ('book_title', 'book_description', 'book_author', 'book_price') VALUES
            ('$title', '$description', '$author', '$price')";
            $execute = mysqli_query($connect, $add_book_query);

            if($execute) {
                echo 'Successfully Uploaded';
            } else {
                echo 'Failed to upload';
            }
        } else {
            echo 'This book already exist';
        }
    }
}
?>

<div class="container py-5 shopping-cart">
```

To insert a book into the book table in the database (this can only be accessed by admin), we created five variables called \$title, \$description, \$author, \$category, and \$price, where the data inputted from the add book form will be saved and passed. Moreover, we used an INSERT query and isset() function to directly save the information to the database after the user sets the save details button.

DATABASE WITHOUT RECORDS:

The screenshot shows the phpMyAdmin interface for a database named 'bibliophile'. The left sidebar lists databases like 'bibliophile_db', 'information_schema', and 'mysql'. The main area shows a table named 'books' with columns: book_id, book_title, book_description, book_author, book_category, book_price, and date_created. A message at the top states: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0022 seconds)'.

The screenshot shows a web browser window titled 'Bibliophile - Sign Up' with the URL 'http://localhost/bibliophile/old/addbook.php'. The page is titled 'Add Book' and contains a form for adding a book. It includes fields for 'Book Title*', 'Author*', 'Book Description*', and 'Price *'. The 'Book Title' field contains 'The Scars Inside Me', 'Author' field contains 'Bianca Sparacino', 'Book Description' field contains placeholder text, and 'Price' field contains '1000'. A 'Save Details' button is at the bottom.

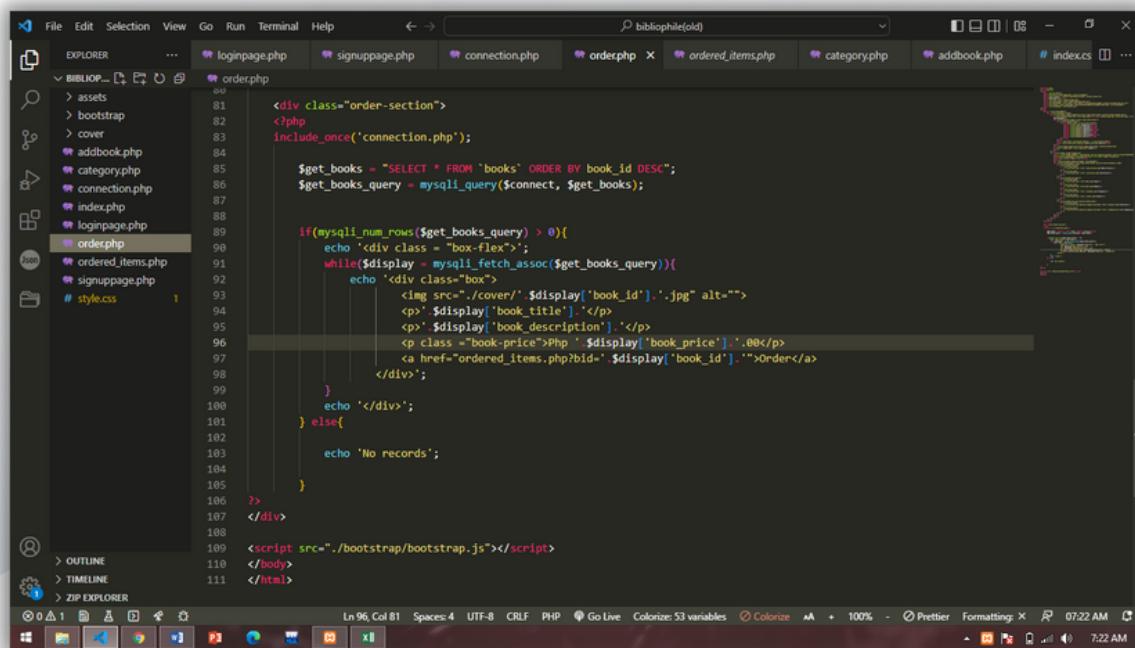
ADDBOOKS FORM:

DATABASE WITH RECORDS:

The screenshot shows the phpMyAdmin interface for the same 'bibliophile' database. The 'books' table now contains 8 records. The data is as follows:

book_id	book_title	book_description	book_author	book_category	book_price	book_cover
1	The Scars Inside Me	Lorem ipsum dolor sit amet consectetur, adipisicing elit. Totar...	Bianca Sparacino	fic	1000	fic.jpg
2	Mortiga	Lorem ipsum dolor sit amet consectetur, adipisicing elit. Mortiga...	Rupi Kaur	fic	1900	fic.jpg
3	Four Walls	Lorem ipsum dolor sit amet consectetur, adipisicing elit. Four Wa...	Ichiro Kishimi & Fumitake Koga	fic	800	fic.jpg
4	The White Raven	Lorem ipsum dolor sit amet consectetur, adipisicing elit. The Whi...	Frances Cha	fic	500	fic.jpg
5	Fantasy of Flights	Lorem ipsum dolor sit amet consectetur, adipisicing elit. Fantasy...	Patty Smith	fic	1010	fic.jpg
6	Scars & Hearts	Lorem ipsum dolor sit amet consectetur, adipisicing elit. Scars &...	Shirley Jackson	fic	600	fic.jpg
7	After Midnight	Lorem ipsum dolor sit amet consectetur, adipisicing elit. After Mi...	Nicholas Spark	fic	600	fic.jpg
8	The Little Mermaid	Lorem ipsum dolor sit amet consectetur, adipisicing elit. The Littl...	Anthony Horowitz	fic	1300	fic.jpg

DISPLAYING DATA FROM DATABASE (BOOKS)



The screenshot shows a code editor interface with multiple tabs open. The active tab is 'order.php'. The code in the editor is as follows:

```
<div class="order-section">
</php>
include_once('connection.php');

$get_books = "SELECT * FROM `books` ORDER BY book_id DESC";
$get_books_query = mysqli_query($connect, $get_books);

if(mysqli_num_rows($get_books_query) > 0){
    echo '<div class = "box-flex">';
    while($display = mysqli_fetch_assoc($get_books_query)){
        echo '<div class="box">
            
            <p>' . $display['book_title'] . '</p>
            <p>' . $display['book_description'] . '</p>
            <p class = "book-price">Php ' . $display['book_price'] . '.00</p>
            <a href="ordered_items.php?bid=' . $display['book_id'] . '">' . $display['book_id'] . 'Order</a>
        </div>';
    }
    echo '</div>';
} else{
    echo 'No records';
}
?>
</div>
<script src=".//bootstrap/bootstrap.js"></script>
</body>
</html>
```

The code uses PHP to query a MySQL database for books and then displays them in a grid format using HTML and CSS. It includes a while loop to iterate through the results and echo the book details.

Lastly, in order to DISPLAY the list of books stored in the database, we used a SELECT * query to get the books from the database. We also set ORDER BY to DESCENDING order to organize the data from the database. In order to display all the books from the list of books on the page, we used a WHILE loop and a function called mysqli_fetch_assoc() under the if condition to SELECT/DISPLAY everything from the books table.

PAGE:

localhost / 127.0.0.1 / bibliophile

Bibliophile

http://localhost/bibliophile(old)/order.php

All Categories Search... Logout

New Arrivals Collection Home Books Category Orders


The Little Mermaid

PHP 1300.00 Order


After Midnight

PHP 600.00 Order


Scars & Hearts

PHP 600.00 Order


Fantasy of Flights

PHP 1010.00 Order

6:33 AM

localhost / 127.0.0.1 / bibliophile

Bibliophile

http://localhost/bibliophile(old)/order.php

Order


The White Raven

PHP 500.00 Order


Four Walls

PHP 800.00 Order


Mortiga

PHP 1900.00 Order


The Scars Inside Me

PHP 1000.00 Order

Lightshot
Screenshot is saved to Screenshot_12.png. Click here to open in the folder.

6:33 AM