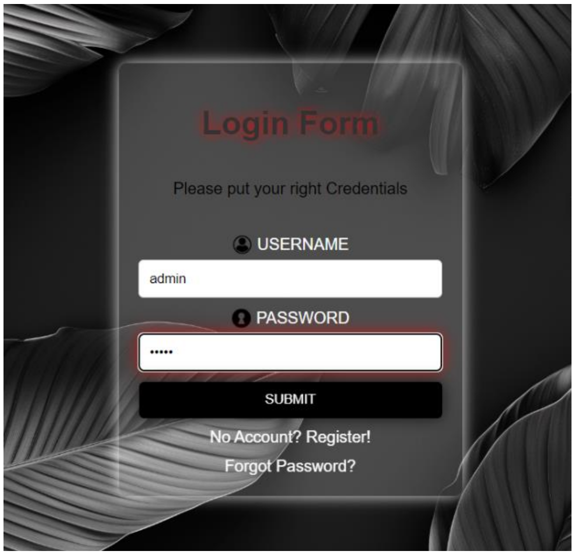
Final Project – Library System

Programming Choice – Web Development – Using HTML, CSS and PHP

Step 1: Login Form

username: admin password: admin



Login Form

Please put your right Credentials

USERNAME

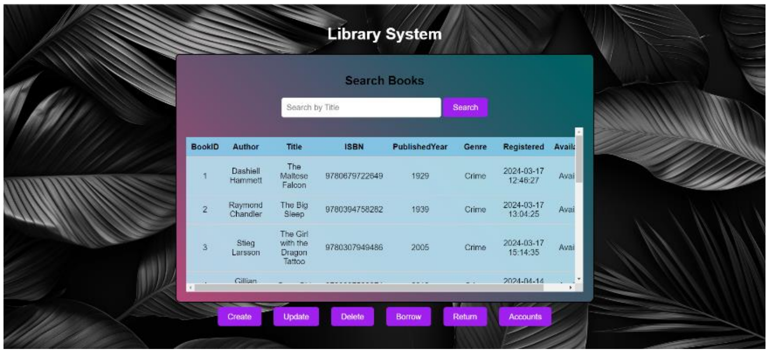
admin

PASSWORD

•••••

Step 2: Search for a Specific books you want to borrow, you can use a keyword to find your ideal book.

Example: In the Search bar type 'Promised'.

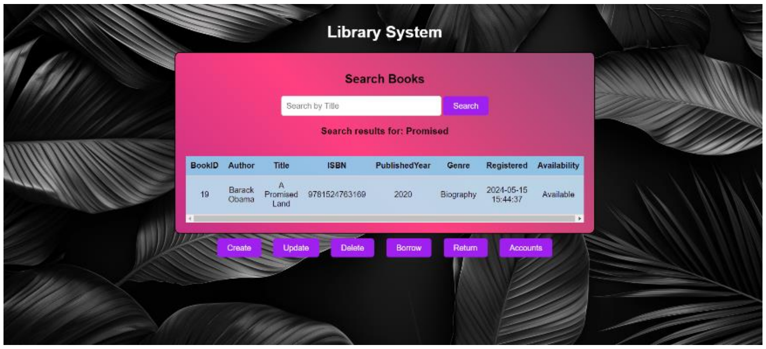


Library System

Search Books

Search by Title

BookID	Author	Title	ISBN	PublishedYear	Genre	Registered	Availa
1	Dashiell Hammett	The Maltese Falcon	9780679722649	1929	Crime	2024-03-17 12:46:27	Avai
2	Raymond Chandler	The Big Sleep	9780394758282	1939	Crime	2024-03-17 13:04:25	Avai
3	Stieg Larsson	The Girl with the Dragon Tattoo	9780307949486	2005	Crime	2024-03-17 15:14:35	Avai
	Gillian					2024-04-14	



Library System

Search Books

Search by Title

Search results for: Promised

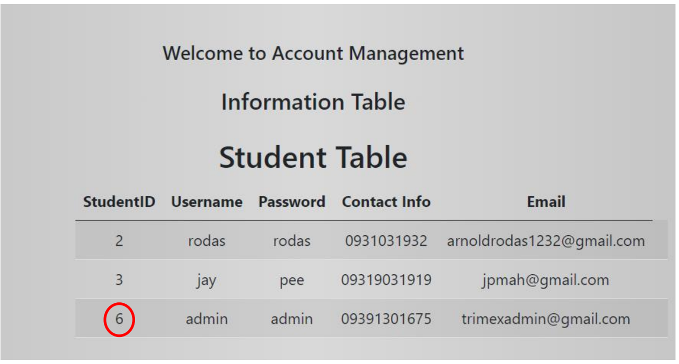
BookID	Author	Title	ISBN	PublishedYear	Genre	Registered	Availability
19	Barack Obama	A Promised Land	9781524763169	2020	Biography	2024-05-15 15:44:37	Available

Result: A Promised Land with a BookID of 19.

Step 3: Borrow Book

Note: You should know the BookID of the books you want to borrow and also your StudentID, you can see your student id in Accounts Management (Go to Status and click the Accounts Button and you can see your StudentID their)

We used the account of admin so our StudentID is 6, and the book id we want to borrow is 19.

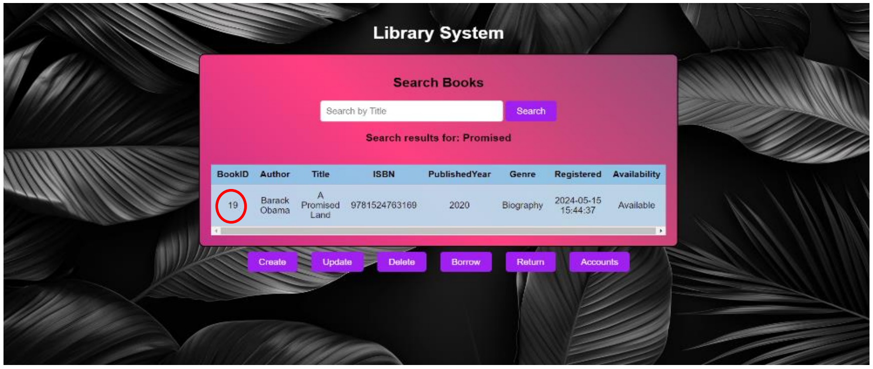


Welcome to Account Management

Information Table

Student Table

StudentID	Username	Password	Contact Info	Email
2	rodas	rodas	0931031932	arnoldrodas1232@gmail.com
3	jay	pee	09319031919	jpmah@gmail.com
6	admin	admin	09391301675	trimexadmin@gmail.com

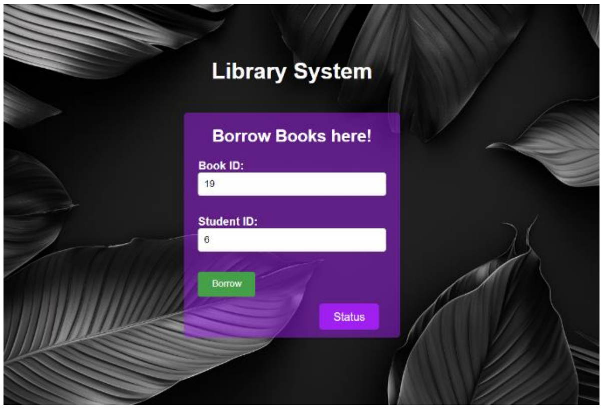


Library System

Search Books

Search results for: Promised

BookID	Author	Title	ISBN	PublishedYear	Genre	Registered	Availability
19	Barack Obama	A Promised Land	9781524763169	2020	Biography	2024-05-15 15:44:37	Available

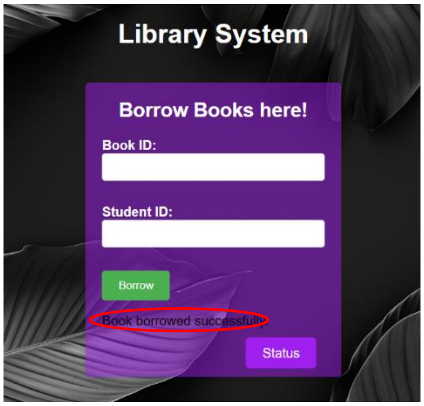


Library System

Borrow Books here!

Book ID: 19

Student ID: 6



Library System

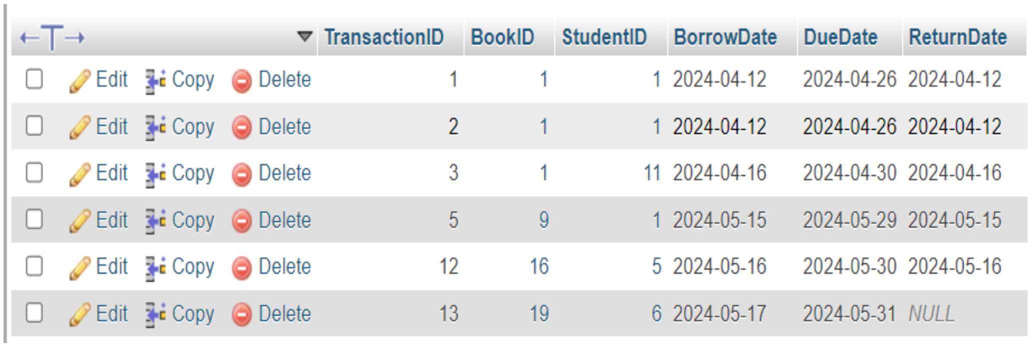
Borrow Books here!

Book ID:

Student ID:

Book borrowed successfully

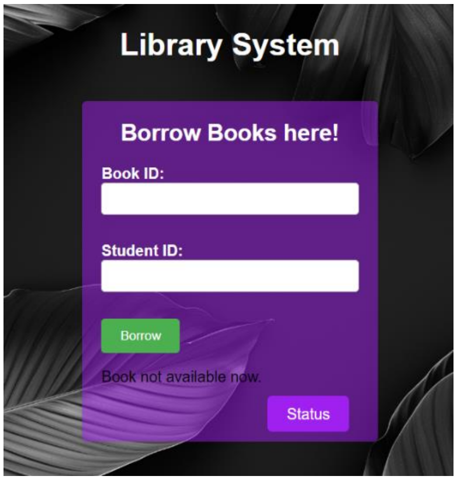
So, after filling up the requirement to borrow books, you can see the message" Book borrowed successfully". And to make sure that you can check it on your localhost.



	TransactionID	BookID	StudentID	BorrowDate	DueDate	ReturnDate
□ Edit Copy Delete	1	1	1	2024-04-12	2024-04-26	2024-04-12
□ Edit Copy Delete	2	1	1	2024-04-12	2024-04-26	2024-04-12
□ Edit Copy Delete	3	1	11	2024-04-16	2024-04-30	2024-04-16
□ Edit Copy Delete	5	9	1	2024-05-15	2024-05-29	2024-05-15
□ Edit Copy Delete	12	16	5	2024-05-16	2024-05-30	2024-05-16
□ Edit Copy Delete	13	19	6	2024-05-17	2024-05-31	NULL

We can here, the BookID we Borrow a while and we can also see the BorrowDate, DueDate and the ReturnDate, we can see the ReturnDate is Null because the borrowed book is not yet return.

Note for Borrowing books, you can't borrow the books if is not return yet.



Library System

Borrow Books here!

Book ID:

Student ID:

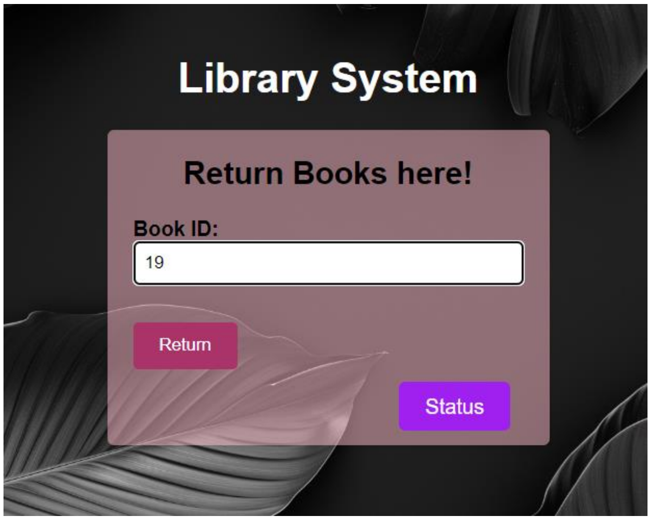
Book not available now.

You can see here the message" Book not available now" because its already borrow by someone.

Step 4: Return Books

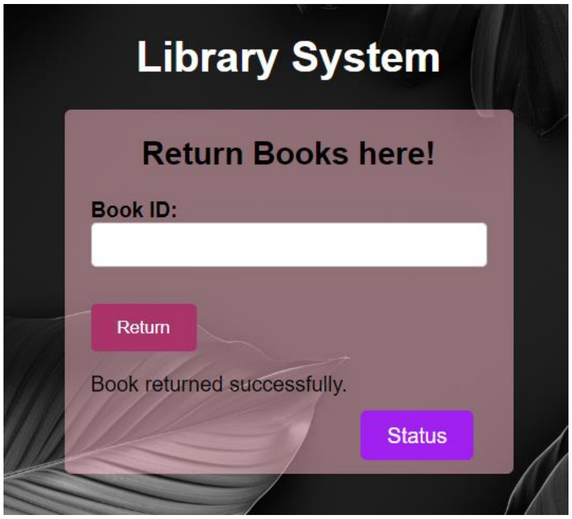
In returning books, you only need to type the BookID.

We will return the book we borrowed earlier. Which the" A Promised Land", with a bookID of 19.


Library System

Return Books here!

Book ID: 19

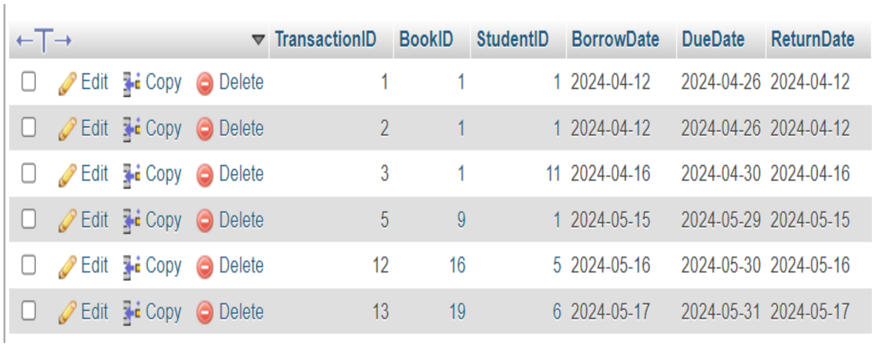

Library System

Return Books here!

Book ID:

Book returned successfully.

Lets Check the Localhost if we successfully returned the book.

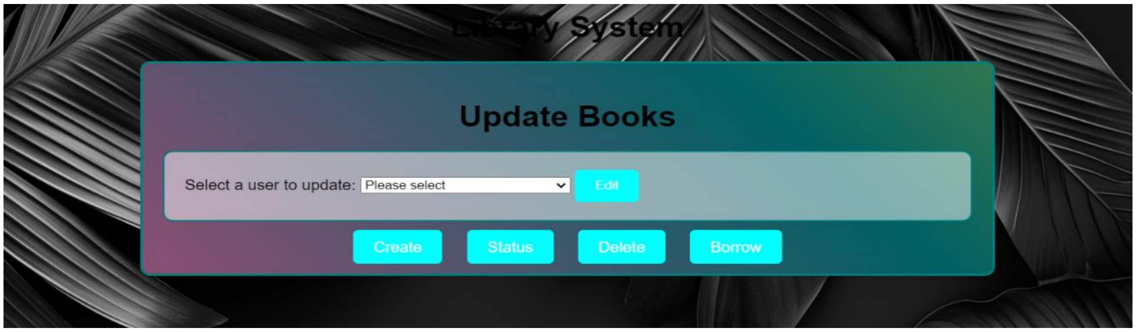


←T→			▼	TransactionID	BookID	StudentID	BorrowDate	DueDate	ReturnDate
□	Edit	Copy	Delete	1	1	1	2024-04-12	2024-04-26	2024-04-12
□	Edit	Copy	Delete	2	1	1	2024-04-12	2024-04-26	2024-04-12
□	Edit	Copy	Delete	3	1	11	2024-04-16	2024-04-30	2024-04-16
□	Edit	Copy	Delete	5	9	1	2024-05-15	2024-05-29	2024-05-15
□	Edit	Copy	Delete	12	16	5	2024-05-16	2024-05-30	2024-05-16
□	Edit	Copy	Delete	13	19	6	2024-05-17	2024-05-31	2024-05-17

We can see the ReturnDate is not Null anymore because we already return the book we borrowed earlier.

Step 5: Update or Edit Books

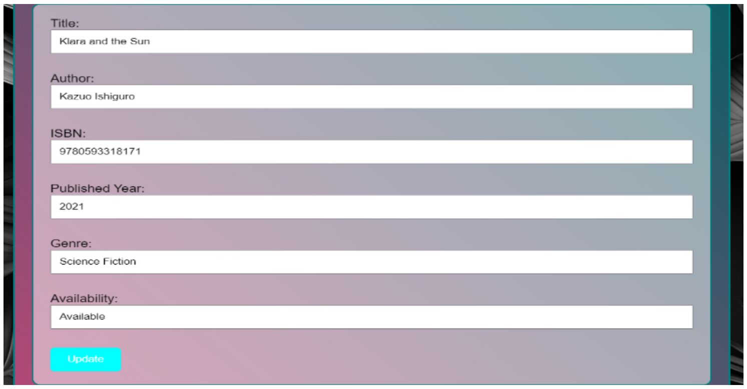
In Editing Books Detail, we go to Update Area which we can see in Status


Library System

Update Books

Select a user to update: Please select

If you want to edit the details of books you can choose the book you want to edit in the dropdown list.


Title: Klara and the Sun

Author: Kazuo Ishiguro

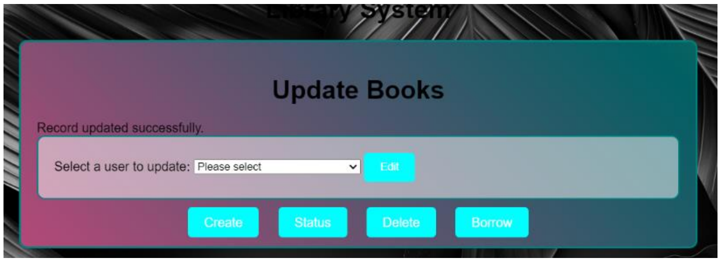
ISBN: 9780593318171

Published Year: 2021

Genre: Science Fiction

Availability: Available

Like this I will edit the Availability to Borrowed, to see if the update setting is working well.

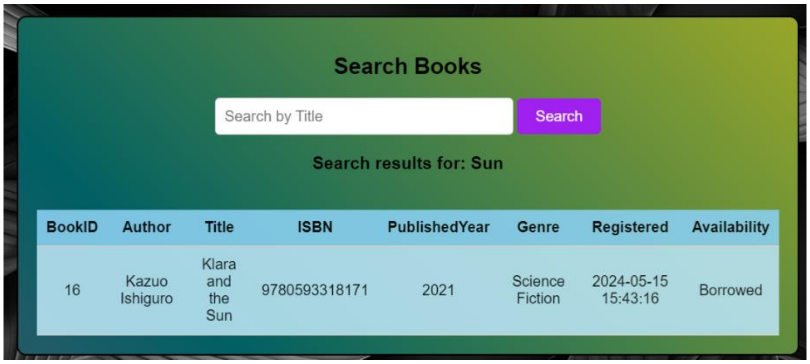


Update Books

Record updated successfully.

Select a user to update: Please select

We can see the message here after I click the update button, it says “Record updated successfully



Search Books

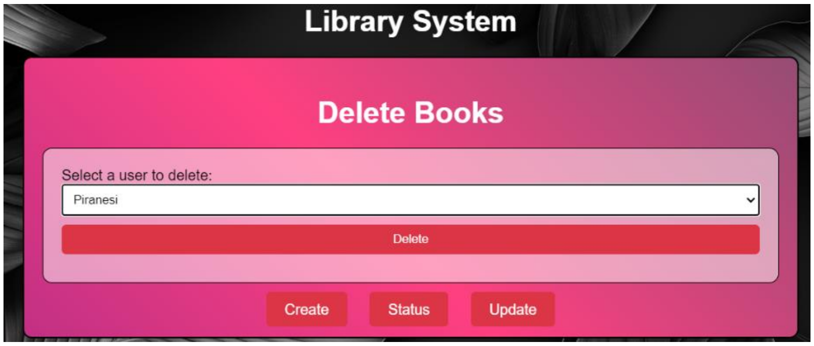
Search results for: Sun

BookID	Author	Title	ISBN	PublishedYear	Genre	Registered	Availability
16	Kazuo Ishiguro	Klara and the Sun	9780593318171	2021	Science Fiction	2024-05-15 15:43:16	Borrowed

We can see here the Availability is now on Borrowed.

Step 6: Delete Books

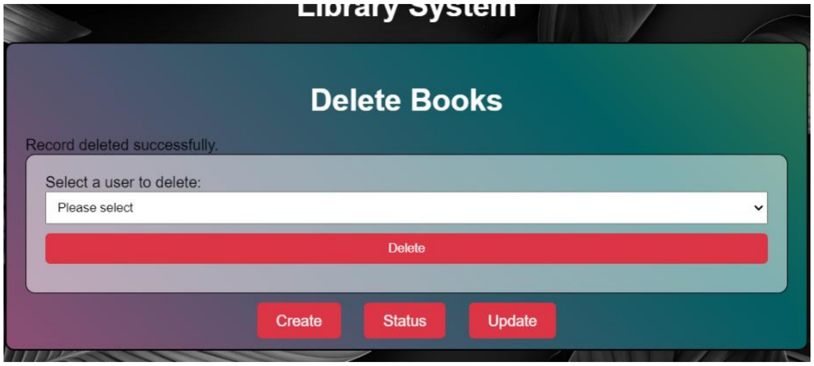
In Deleting Books, you will Select only the Title of books you want to delete.



Delete Books

Select a user to delete: Piranesi

We can see here I selected the ‘Piranesi’, and will delete that now.



Delete Books

Record deleted successfully.

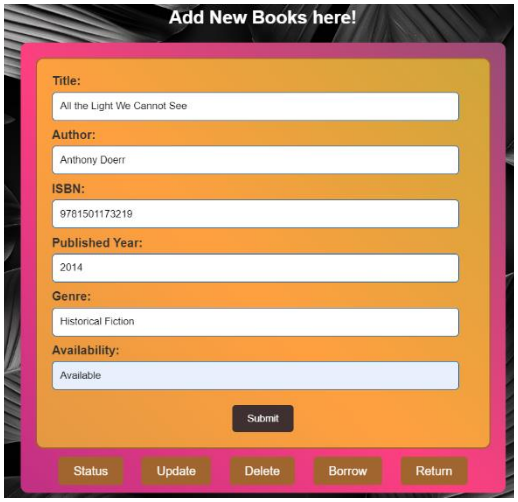
Select a user to delete: Please select

We successfully deleted the books Piranesi.

Note you can’t delete a book with a history of borrow even if is already returned.

Step 7: Create Books or Add Books

To add more books, you go to Status and click the button of Create.



Add New Books here!

Title: All the Light We Cannot See

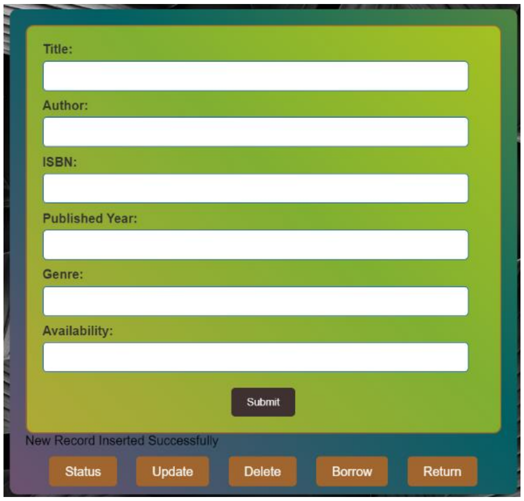
Author: Anthony Doerr

ISBN: 9781501173219

Published Year: 2014

Genre: Historical Fiction

Availability: Available



Title:

Author:

ISBN:

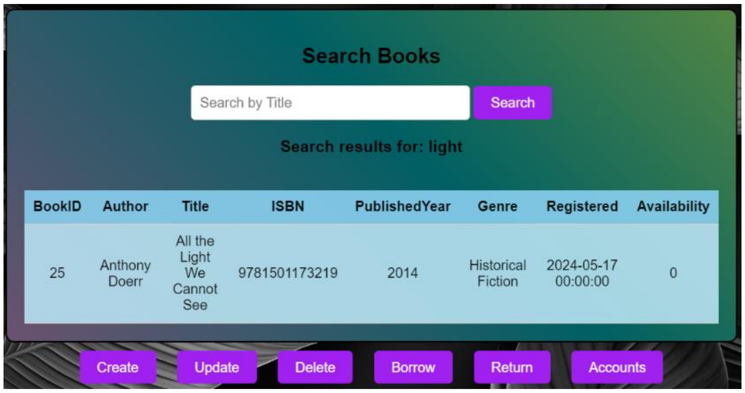
Published Year:

Genre:

Availability:

New Record Inserted Successfully

We Successfully add books and to make sure that, let's check in status.



Search Books

Search by Title

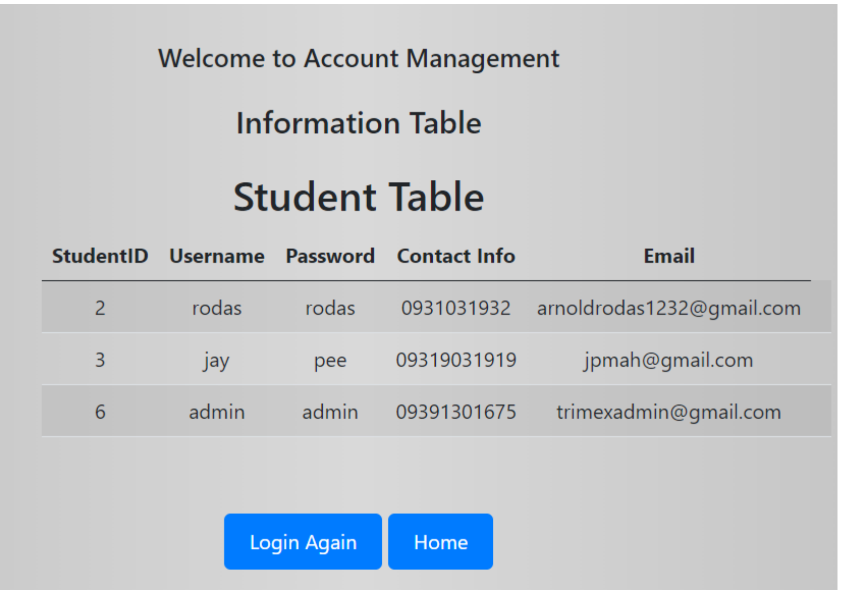
Search results for: light

BookID	Author	Title	ISBN	PublishedYear	Genre	Registered	Availability
25	Anthony Doerr	All the Light We Cannot See	9781501173219	2014	Historical Fiction	2024-05-17 00:00:00	0

We can see here in Registered rows, the time and date that we registered the books.

Step 8: Account Management

To see all the accounts you can see it in Status, just click the Button for Accounts.



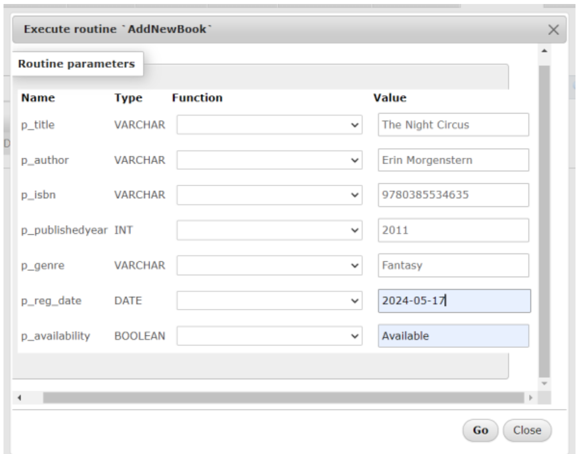
Welcome to Account Management

Information Table

Student Table

StudentID	Username	Password	Contact Info	Email
2	rodas	rodas	0931031932	arnoldrodas1232@gmail.com
3	jay	pee	09319031919	jpmah@gmail.com
6	admin	admin	09391301675	trimexadmin@gmail.com

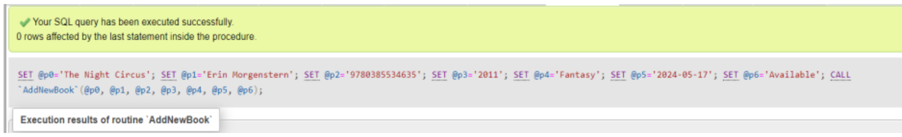
For Technical Requirements **like CTE and Store Procedure**, you can see that in localhost.
example



Execute routine `AddNewBook`

Routine parameters

Name	Type	Function	Value
p_title	VARCHAR		The Night Circus
p_author	VARCHAR		Erin Morgenstern
p_isbn	VARCHAR		9780385534635
p_publishedyear	INT		2011
p_genre	VARCHAR		Fantasy
p_reg_date	DATE		2024-05-17
p_availability	BOOLEAN		Available



✓ Your SQL query has been executed successfully.
0 rows affected by the last statement inside the procedure.

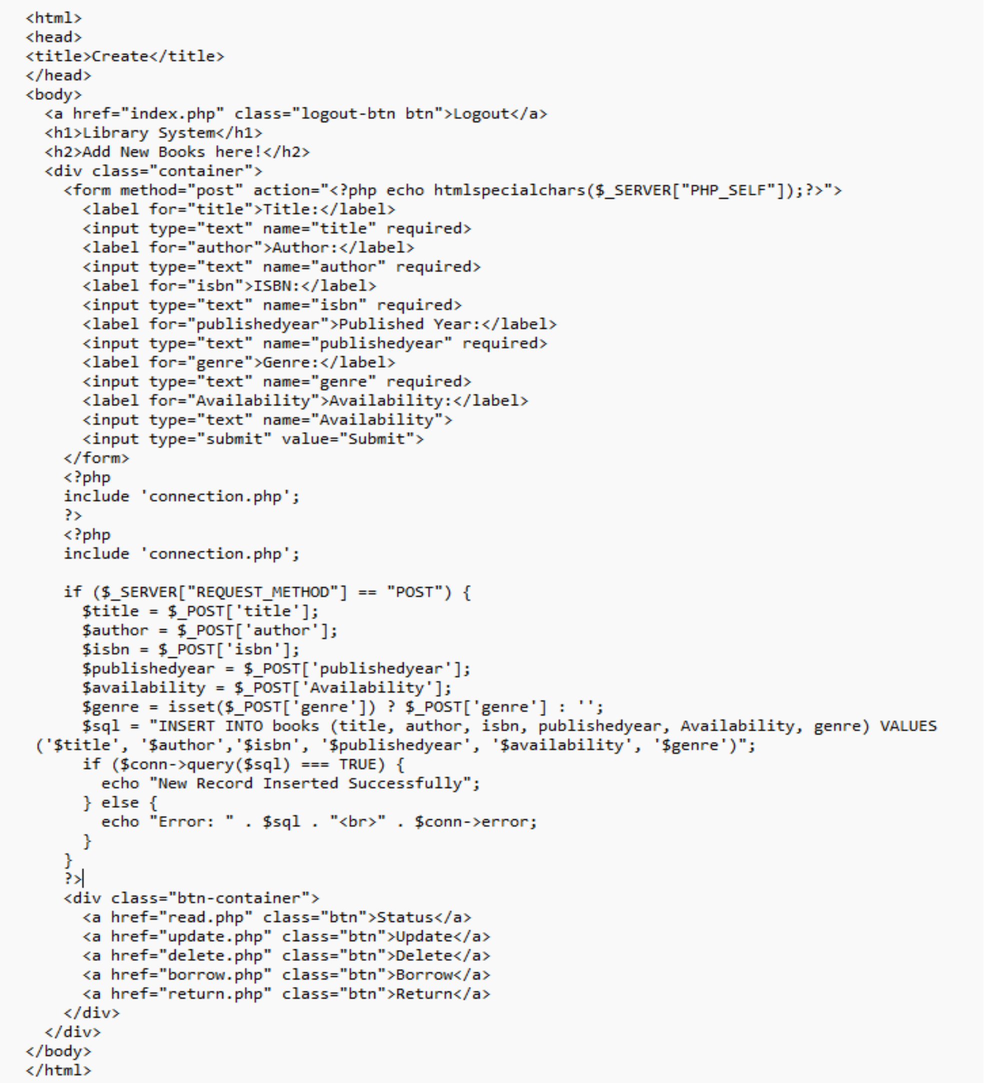
```
SET @p0='The Night Circus'; SET @p1='Erin Morgenstern'; SET @p2='9780385534635'; SET @p3='2011'; SET @p4='Fantasy'; SET @p5='2024-05-17'; SET @p6='Available'; CALL
`AddNewBook`(@p0, @p1, @p2, @p3, @p4, @p5, @p6);
```

Execution results of routine `AddNewBook`

We can see here I just filled up the value for addnewbooks and it successfully add new books with a registered date.
this only an alternative of the library system for adding some books.

Code of the System

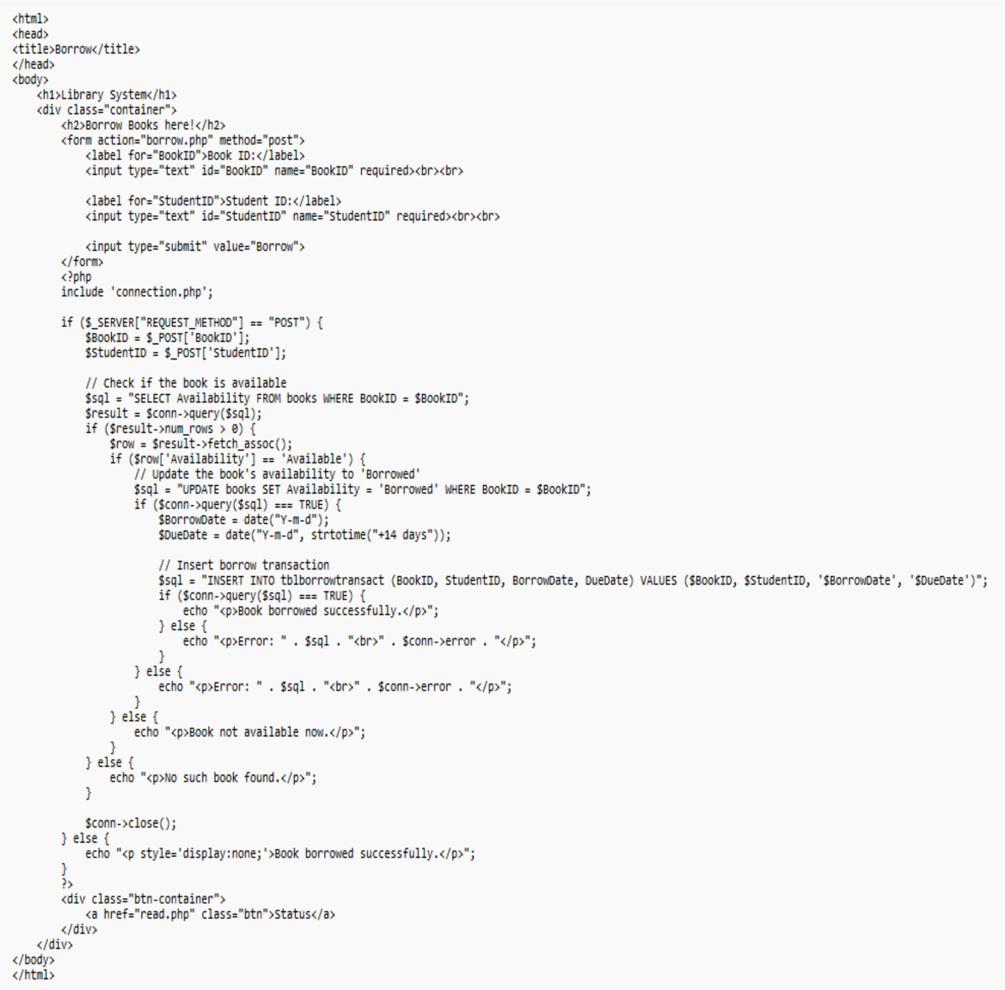
CREATE



```
<html>
<head>
<title>Create</title>
</head>
<body>
  <a href="index.php" class="logout-btn btn">Logout</a>
  <h1>Library System</h1>
  <h2>Add New Books here!</h2>
  <div class="container">
    <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
      <label for="title">Title:</label>
      <input type="text" name="title" required>
      <label for="author">Author:</label>
      <input type="text" name="author" required>
      <label for="isbn">ISBN:</label>
      <input type="text" name="isbn" required>
      <label for="publishedyear">Published Year:</label>
      <input type="text" name="publishedyear" required>
      <label for="genre">Genre:</label>
      <input type="text" name="genre" required>
      <label for="Availability">Availability:</label>
      <input type="text" name="Availability">
      <input type="submit" value="Submit">
    </form>
    <?php
    include 'connection.php';
    ?>
    <?php
    include 'connection.php';

    if ($_SERVER["REQUEST_METHOD"] == "POST") {
      $title = $_POST['title'];
      $author = $_POST['author'];
      $isbn = $_POST['isbn'];
      $publishedyear = $_POST['publishedyear'];
      $availability = $_POST['Availability'];
      $genre = isset($_POST['genre']) ? $_POST['genre'] : '';
      $sql = "INSERT INTO books (title, author, isbn, publishedyear, Availability, genre) VALUES
 ('$title', '$author','$isbn', '$publishedyear', '$availability', '$genre')";
      if ($conn->query($sql) === TRUE) {
        echo "New Record Inserted Successfully";
      } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
      }
    }
    ?>
    <div class="btn-container">
      <a href="read.php" class="btn">Status</a>
      <a href="update.php" class="btn">Update</a>
      <a href="delete.php" class="btn">Delete</a>
      <a href="borrow.php" class="btn">Borrow</a>
      <a href="return.php" class="btn">Return</a>
    </div>
  </div>
</body>
</html>
```

I only show the main code for Create; I didn't include the style.
This code checks if a form has been submitted. If it has, it takes the book details entered in the form (like title, author, and ISBN), and tries to add these details to the database. If it succeeds, it shows a message saying "New Record Inserted Successfully"; if it fails, it shows an error message.

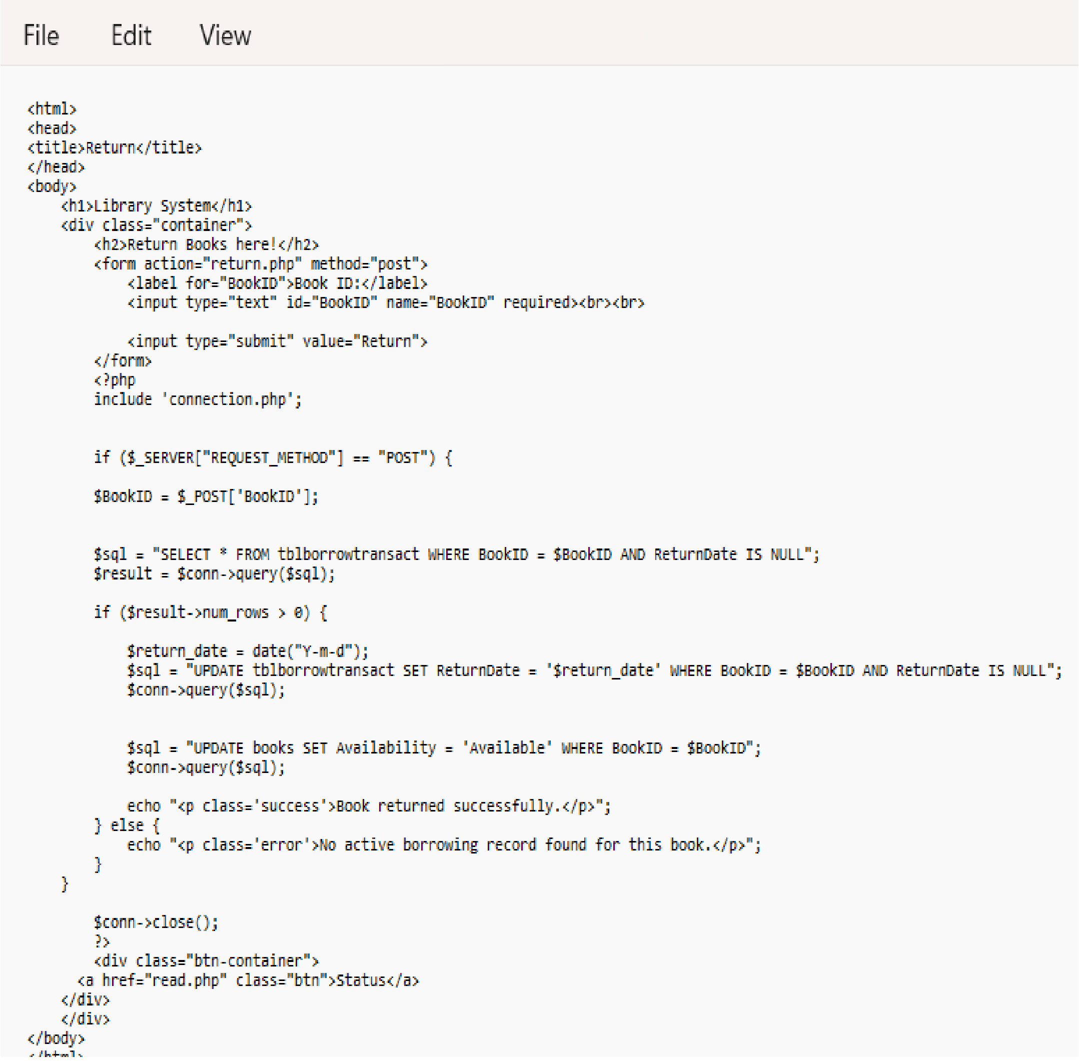
Borrow



```
<html>
<head>
<title>Borrow</title>
</head>
<body>
    <h1>Library System</h1>
    <div class="container">
        <h2>Borrow Books here!</h2>
        <form action="borrow.php" method="post">
            <label for="BookID">Book ID:</label>
            <input type="text" id="BookID" name="BookID" required><br><br>

            <label for="StudentID">Student ID:</label>
            <input type="text" id="StudentID" name="StudentID" required><br><br>

            <input type="submit" value="Borrow">
        </form>
        <?php
        include 'connection.php';

        if ($_SERVER["REQUEST_METHOD"] == "POST") {
            $BookID = $_POST['BookID'];
            $StudentID = $_POST['StudentID'];

            // Check if the book is available
            $sql = "SELECT Availability FROM books WHERE BookID = $BookID";
            $result = $conn->query($sql);
            if ($result->num_rows > 0) {
                $row = $result->fetch_assoc();
                if ($row['Availability'] == 'Available') {
                    // Update the book's availability to 'Borrowed'
                    $sql = "UPDATE books SET Availability = 'Borrowed' WHERE BookID = $BookID";
                    if ($conn->query($sql) === TRUE) {
                        $BorrowDate = date("Y-m-d");
                        $DueDate = date("Y-m-d", strtotime("+14 days"));

                        // Insert borrow transaction
                        $sql = "INSERT INTO tblborrowtransact (BookID, StudentID, BorrowDate, DueDate) VALUES ($BookID, $StudentID, '$BorrowDate', '$DueDate')";
                        if ($conn->query($sql) === TRUE) {
                            echo "<p>Book borrowed successfully.</p>";
                        } else {
                            echo "<p>Error: " . $sql . "<br>" . $conn->error . "</p>";
                        }
                    } else {
                        echo "<p>Error: " . $sql . "<br>" . $conn->error . "</p>";
                    }
                } else {
                    echo "<p>Book not available now.</p>";
                }
            } else {
                echo "<p>No such book found.</p>";
            }

            $conn->close();
        } else {
            echo "<p style='display:none;'>Book borrowed successfully.</p>";
        }
        ?>
        <div class="btn-container">
            <a href="read.php" class="btn">Status</a>
        </div>
    </div>
</body>
</html>
```

This code creates a web page where you can borrow books. You enter the Book ID and your Student ID, and the system checks if the book is available. If it is, the book is marked as borrowed, and the system records your borrow details. If the book isn't available or if there are any issues, it shows a message explaining the problem.

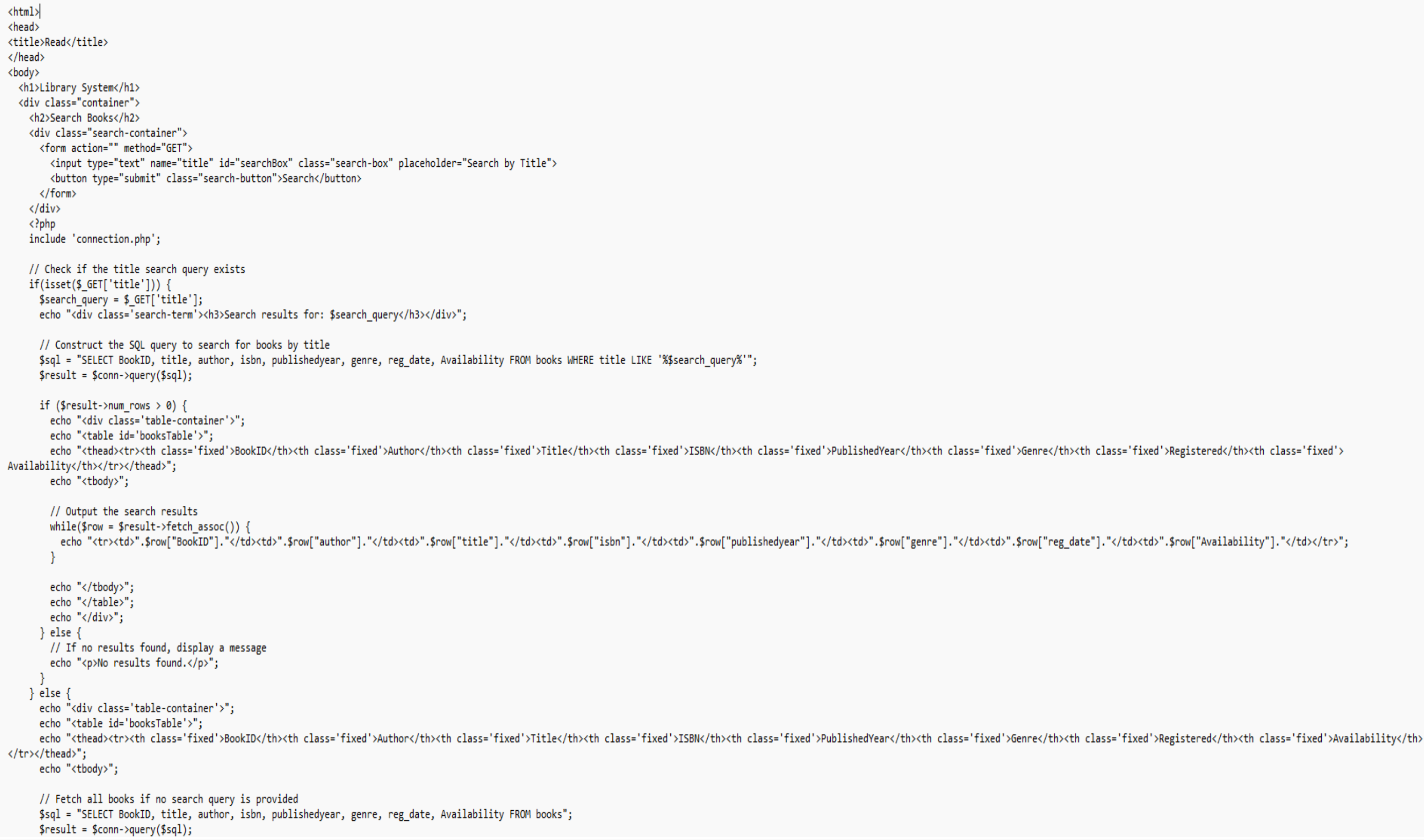
Return



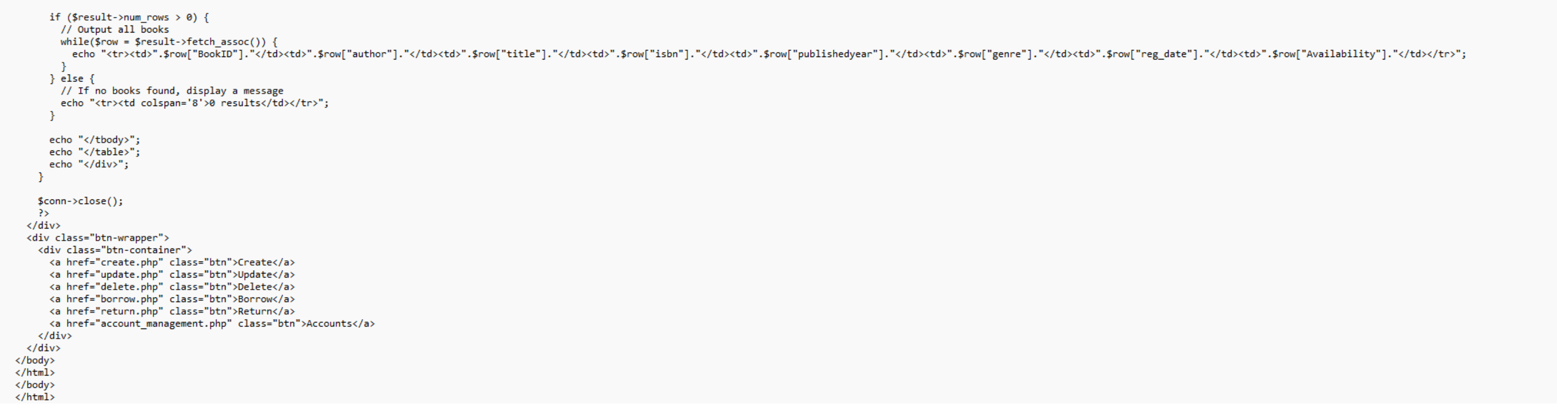
```
<html>
<head>
<title>Return</title>
</head>
<body>
    <h1>Library System</h1>
    <div class="container">
        <h2>Return Books here!</h2>
        <form action="return.php" method="post">
            <label for="BookID">Book ID:</label>
            <input type="text" id="BookID" name="BookID" required><br><br>

            <input type="submit" value="Return">
        </form>
        <?php
        include 'connection.php';


        if ($_SERVER["REQUEST_METHOD"] == "POST") {

        $BookID = $_POST['BookID'];


        $sql = "SELECT * FROM tblborrowtransact WHERE BookID = $BookID AND ReturnDate IS NULL";
        $result = $conn->query($sql);

        if ($result->num_rows > 0) {

            $return_date = date("Y-m-d");
            $sql = "UPDATE tblborrowtransact SET ReturnDate = '$return_date' WHERE BookID = $BookID AND ReturnDate IS NULL";
            $conn->query($sql);


            $sql = "UPDATE books SET Availability = 'Available' WHERE BookID = $BookID";
            $conn->query($sql);

            echo "<p class='success'>Book returned successfully.</p>";
        } else {
            echo "<p class='error'>No active borrowing record found for this book.</p>";
        }
    }

        $conn->close();
        ?>
        <div class="btn-container">
      <a href="read.php" class="btn">Status</a>
    </div>
    </div>
</body>
</html>
```

This code creates a web page where you can return borrowed books. You enter the Book ID, and the system checks if there's an active borrowing record for that book. If there is, it marks the book as returned and updates its status to 'Available'. If there's no active borrowing record or if there are any issues, it shows a message explaining the problem.

Read



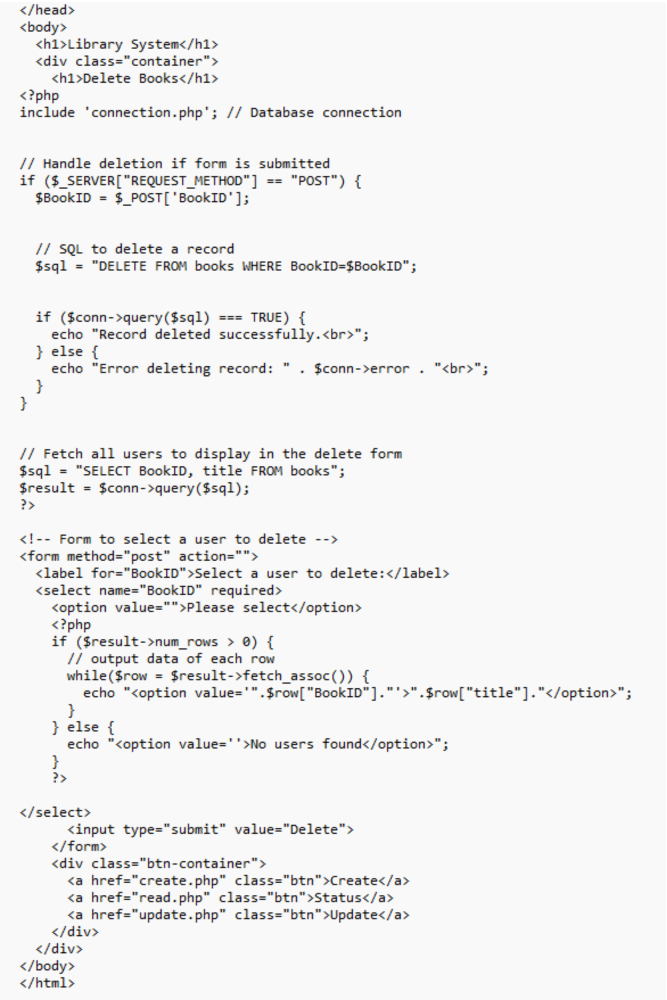
```
<html>
<head>
<title>Read</title>
</head>
<body>
  <h1>Library System</h1>
  <div class="container">
    <h2>Search Books</h2>
    <div class="search-container">
      <form action="" method="GET">
        <input type="text" name="title" id="searchBox" class="search-box" placeholder="Search by Title">
        <button type="submit" class="search-button">Search</button>
      </form>
    </div>
    <?php
    include 'connection.php';

    // Check if the title search query exists
    if(isset($_GET['title'])) {
      $search_query = $_GET['title'];
      echo "<div class='search-term'><h3>Search results for: $search_query</h3></div>";

      // Construct the SQL query to search for books by title
      $sql = "SELECT BookID, title, author, isbn, publishedyear, genre, reg_date, Availability FROM books WHERE title LIKE '%$search_query%'";
      $result = $conn->query($sql);

      if ($result->num_rows > 0) {
        echo "<div class='table-container'>";
        echo "<table id='booksTable'>";
        echo "<thead><tr><th class='fixed'>BookID</th><th class='fixed'>Author</th><th class='fixed'>Title</th><th class='fixed'>ISBN</th><th class='fixed'>PublishedYear</th><th class='fixed'>Genre</th><th class='fixed'>Registered</th><th class='fixed'>
Availability</th></tr></thead>";
        echo "<tbody>";

        // Output the search results
        while($row = $result->fetch_assoc()) {
          echo "<tr><td>".$row["BookID"]."</td><td>".$row["author"]."</td><td>".$row["title"]."</td><td>".$row["isbn"]."</td><td>".$row["publishedyear"]."</td><td>".$row["genre"]."</td><td>".$row["reg_date"]."</td><td>".$row["Availability"]."</td></tr>";
        }

        echo "</tbody>";
        echo "</table>";
        echo "</div>";
      } else {
        // If no results found, display a message
        echo "<p>No results found.</p>";
      }
    } else {
      echo "<div class='table-container'>";
      echo "<table id='booksTable'>";
      echo "<thead><tr><th class='fixed'>BookID</th><th class='fixed'>Author</th><th class='fixed'>Title</th><th class='fixed'>ISBN</th><th class='fixed'>PublishedYear</th><th class='fixed'>Genre</th><th class='fixed'>Registered</th><th class='fixed'>Availability</th>
</tr></thead>";
      echo "<tbody>";

      // Fetch all books if no search query is provided
      $sql = "SELECT BookID, title, author, isbn, publishedyear, genre, reg_date, Availability FROM books";
      $result = $conn->query($sql);
```



```
      if ($result->num_rows > 0) {
        // Output all books
        while($row = $result->fetch_assoc()) {
          echo "<tr><td>".$row["BookID"]."</td><td>".$row["author"]."</td><td>".$row["title"]."</td><td>".$row["isbn"]."</td><td>".$row["publishedyear"]."</td><td>".$row["genre"]."</td><td>".$row["reg_date"]."</td><td>".$row["Availability"]."</td></tr>";
        }
      } else {
        // If no books found, display a message
        echo "<tr><td colspan='8'>0 results</td></tr>";
      }

      echo "</tbody>";
      echo "</table>";
      echo "</div>";
    }

    $conn->close();
    ?>
  </div>
  <div class="btn-wrapper">
    <div class="btn-container">
      <a href="create.php" class="btn">Create</a>
      <a href="update.php" class="btn">Update</a>
      <a href="delete.php" class="btn">Delete</a>
      <a href="borrow.php" class="btn">Borrow</a>
      <a href="return.php" class="btn">Return</a>
      <a href="account_management.php" class="btn">Accounts</a>
    </div>
  </div>
</body>
</html>
</body>
</html>
```

The provided code offers a library system where users can search for books by title using a search box, with results displayed in a table format. If no search query is entered, all books in the library are shown. The displayed table includes columns for BookID, Author, Title, ISBN, Published Year, Genre, Registration Date, and Availability. Additionally, the page includes navigation buttons that allow users to perform other operations such as creating, updating, deleting books, borrowing, returning books, and managing accounts, facilitating comprehensive library management.

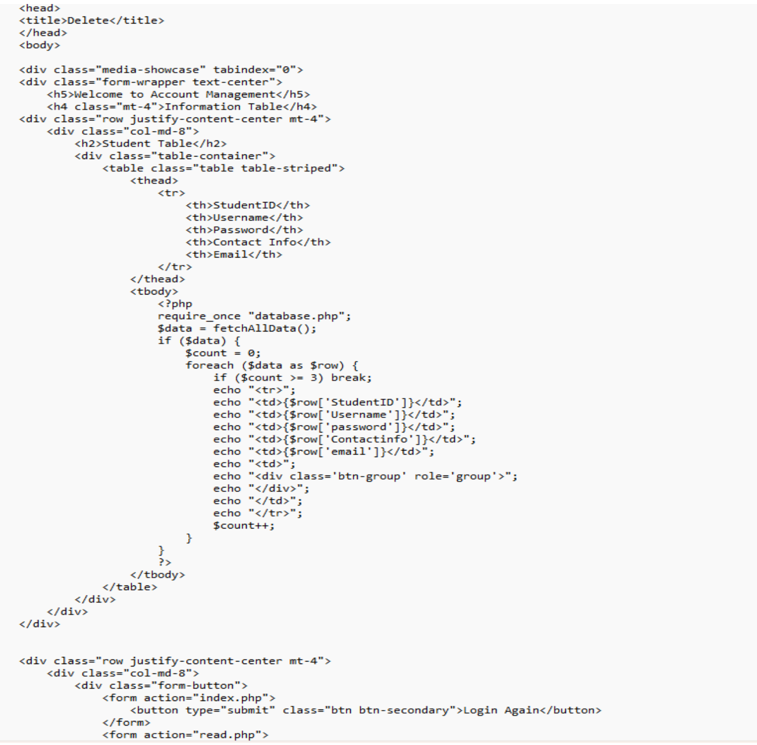
Delete



```
</head>
<body>
  <h1>Library System</h1>
  <div class="container">
    <h1>Delete Books</h1>
<?php
include 'connection.php'; // Database connection


// Handle deletion if form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  $BookID = $_POST['BookID'];


  // SQL to delete a record
  $sql = "DELETE FROM books WHERE BookID=$BookID";


  if ($conn->query($sql) === TRUE) {
    echo "Record deleted successfully.<br>";
  } else {
    echo "Error deleting record: " . $conn->error . "<br>";
  }
}


// Fetch all users to display in the delete form
$sql = "SELECT BookID, title FROM books";
$result = $conn->query($sql);
?>

<!-- Form to select a user to delete -->
<form method="post" action="">
  <label for="BookID">Select a user to delete:</label>
  <select name="BookID" required>
    <option value="">Please select</option>
    <?php
    if ($result->num_rows > 0) {
      // output data of each row
      while($row = $result->fetch_assoc()) {
        echo "<option value='".$row["BookID"]."'>".$row["title"]."</option>";
      }
    } else {
      echo "<option value=''>No users found</option>";
    }
    ?>

</select>
      <input type="submit" value="Delete">
    </form>
    <div class="btn-container">
      <a href="create.php" class="btn">Create</a>
      <a href="read.php" class="btn">Status</a>
      <a href="update.php" class="btn">Update</a>
    </div>
  </div>
</body>
</html>
```

This code allows users to delete books from the library system. It provides a form where users can select a book from a dropdown list and delete it. The PHP script executes an SQL query to delete the selected book from the database, displaying feedback messages about the success or failure of the deletion process. Navigation buttons are also included for other library management functionalities like creating new books, viewing book status, and updating records, enhancing user interaction and system efficiency.

Account Management



```
<head>
<title>Delete</title>
</head>
<body>

<div class="media-showcase" tabindex="0">
<div class="form-wrapper text-center">
    <h5>Welcome to Account Management</h5>
    <h4 class="mt-4">Information Table</h4>
<div class="row justify-content-center mt-4">
    <div class="col-md-8">
        <h2>Student Table</h2>
        <div class="table-container">
            <table class="table table-striped">
                <thead>
                    <tr>
                        <th>StudentID</th>
                        <th>Username</th>
                        <th>Password</th>
                        <th>Contact Info</th>
                        <th>Email</th>
                    </tr>
                </thead>
                <tbody>
                    <?php
                    require_once "database.php";
                    $data = fetchAllData();
                    if ($data) {
                        $count = 0;
                        foreach ($data as $row) {
                            if ($count >= 3) break;
                            echo "<tr>";
                            echo "<td>{$row['StudentID']}</td>";
                            echo "<td>{$row['Username']}</td>";
                            echo "<td>{$row['password']}</td>";
                            echo "<td>{$row['Contactinfo']}</td>";
                            echo "<td>{$row['email']}</td>";
                            echo "<td>";
                            echo "<div class='btn-group' role='group'>";
                            echo "</div>";
                            echo "</td>";
                            echo "</tr>";
                            $count++;
                        }
                    }
                    ?>
                </tbody>
            </table>
        </div>
    </div>
</div>


<div class="row justify-content-center mt-4">
    <div class="col-md-8">
        <div class="form-button">
            <form action="index.php">
                <button type="submit" class="btn btn-secondary">Login Again</button>
            </form>
            <form action="read.php">
```

This code provides an account management interface for the library system. It displays student account information in a table, including StudentID, Username, Password, Contact Info, and Email. Users can view this information and perform actions like logging in again or returning to the home page using the provided buttons.

Link for file of Library System: https://drive.google.com/file/d/158kqjB7QH6AQYjkXSVCG0fdHjoV-CfrD/view?usp=sharing