Worksheet 08

**Book Rental**

# This worksheet covers a sample display page (exampleViewMovies.php) and set of a sample insertion pages (exampleInsertMovies.php, doExampleInsertMovies.php) to scaffold the concepts required to build the library books solution.

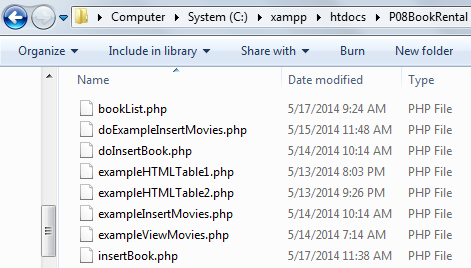
In this worksheet, you will learn how to:

# [Use HTML Table to Structure a Webpage](#_Using_HTML_Tables)

* [Retrieve Records From Database Tables and Display Them](#_Retrieve_Records_From)
* [Retrieve Records From Database Tables and Display Them in an HTML Table](#_Retrieve_Records_From_1)
* [Retrieve Records From Database Tables and Use Them to Populate a Dropdown List](#_Retrieve_Records_From_2)
* [Retrieve HTML Form Values and Insert Them to a Database Table (Recap from P04)](#_Retrieve_HTML_Form)

In your student package, P08BookRental. Save it to C:\xampp\htdocs.

**Make sure that the php files are under P08BookRental folder as shown below**

****

**TODO:** Start NetBeans. Create a PHP Project with Existing Source that points to above folder.

**TODO:** Start the XAMPP Control Panel, and **start both the Apache and MySQL component**.

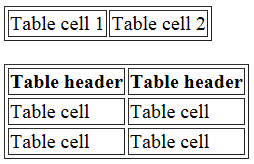
# Using HTML Table to Structure Data

*This section covers the HTML tags commonly used to build an HTML Table. Start with the following resource to see how an HTML Table is created:*

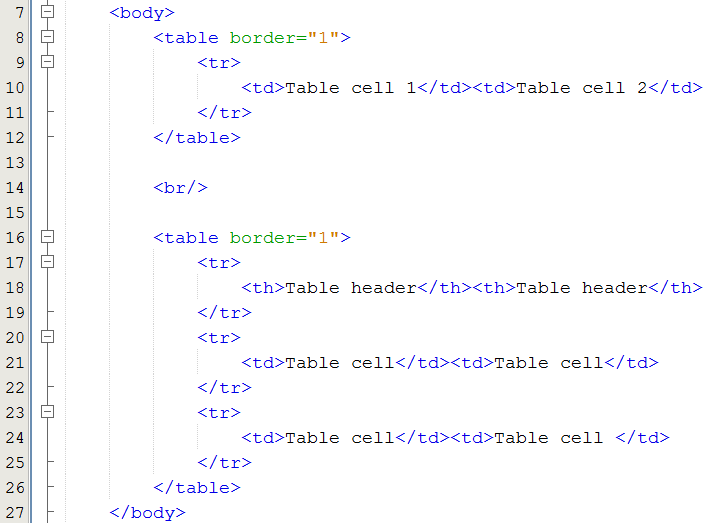
<http://www.w3schools.com/html/tryit.asp?filename=tryhtml_tables>

1. In NetBeans, open the file **exampleHTMLTable1.php.**

Run this page on the browser. You should see the following display:



A portion of the code inside the page is as follows:



Line 8, 12 – HTML Table opening and closing tags: <table> and </table>

Line 9, 11 – Table row opening and closing tags: <tr> and </tr>

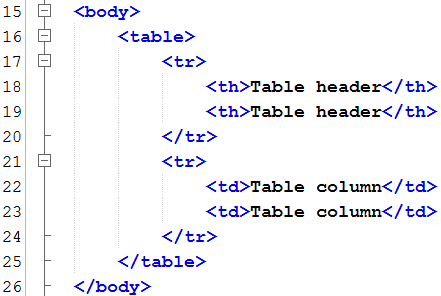
Line 10 – Two table columns, created using the <td> tag. **Note that it is possible to put the <td> section "Table cell 1" and "Table cell 2" on two separate lines**

Line 18 – Two table headers. A table header is created using the <th> tag.

1. The next part is on how to dynamically generate the rows and columns in an HTML Table using PHP.

In NetBeans, open the file **exampleHTMLTable2.php.**

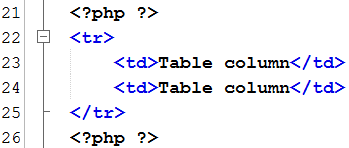
**TODO:** Add the code to create an HTML Table as follows:



Line 16 – 25 – the code for an HTML Table

Next, prepare a pair of PHP opening and closing tags, to enclose the HTML portion that is going to be repeated. **Note that the HTML table itself is not repeated. The table headers are also not repeated. Only the rows below them will be repeated.**

**TODO:** Modify the page as follows:

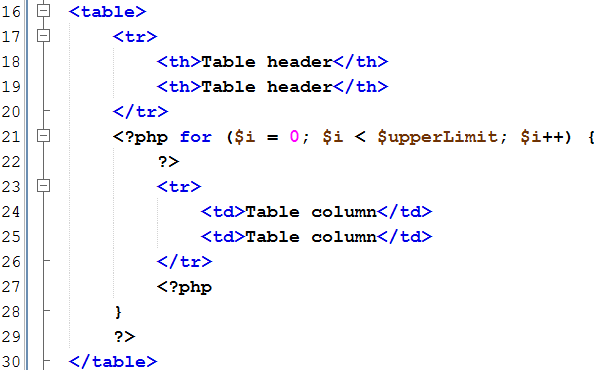


Line 21 & 26 – PHP opening and closing tags.

Next, include the PHP FOR loop to generate the table rows.

Value of variable **$upperLimit** is used as the upper limit.

**TODO:** Modify the page as follows:



Line 21 – open a For Loop :

* The control variable is **$i**.
* The starting value / initialization given to $i is **0**.
* The condition (upper limit in this case) is **$i < $upperLimit** (less than $upperLimit).
* The step is **$i++** (add 1 to $i for every for loop iteration).
* For every loop iteration, a table row with two columns is created.

Line 27 - 29 – close the For Loop.

Run the page **exampleHTMLTable2.php**, and test that it generates 4 table rows.

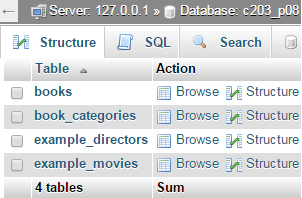
# Retrieve Records from Database Tables and Display Them

*This section covers how to retrieve and display multiple database records using PHP*

1. **TODO:** Open PHPMyAdmin by typing <http://localhost/phpmyadmin> on your browser.

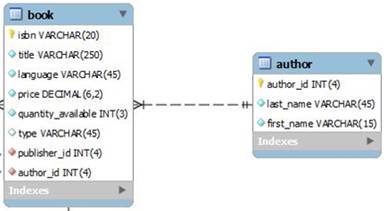
**TODO:** Create database **c203\_p08** and import the file **c203\_p08.sql** into the above database. Refer to **PHPMyAdmin DB Import and Export Guide.docx.**

Once you have done all of the above, you should have the following 4 tables under database **c203\_p08**.

****

1. Recall that in the database module, you have learnt about joined tables.

The following is an example of "one to many" relationship between table author and table book.



**This is how you would create the sql query for the two tables above:**

**Select** title, first\_name, last\_name

**From** book, author

**Where** book.author\_id = author.author\_id

**OR use aliases for table names (note: good to use if the table name is very long like school\_book\_order)**

**Select** title, first\_name, last\_name

**From** book b, author a

**Where** b.author\_id = a.author\_id

**You can also use aliases for field names:**

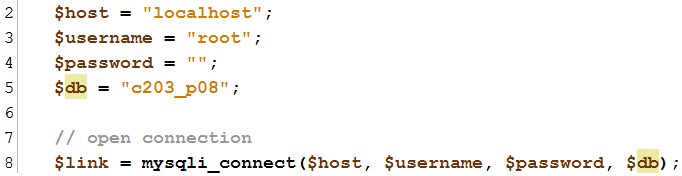
**Select** title, first\_name as fname, last\_name as lname

**From** book b, author a

**Where** b.author\_id = a.author\_id

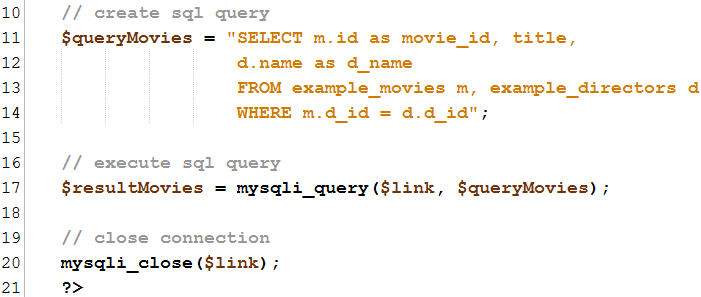
1. In NetBeans, open the file **exampleViewMovies.php**. This file is created to display all records from the table **example\_movies**.

**TODO:** Add the following database related code to the page:



Line 2, 3, 4, 5 – assign values for database connection

Line 8 - open database connection on $link using **mysqli\_connect()** function

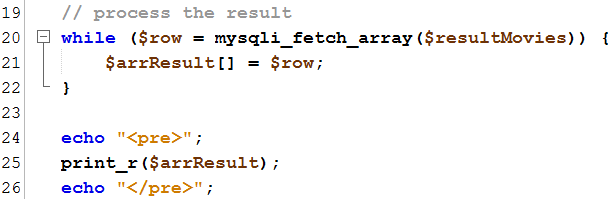


Line 11 - 14 – create / build the sql query using the SELECT keyword

Line 17 – execute the sql query $queryMovies on the database connection $link, by using the **mysqli\_query()** function.

Line 20 – close database connection using **mysqli\_close()** function

Next modification is on moving records data into an array. **TODO:** Modify the page as follows:

****

Line 20 - 22 – processing of the sql query execution result.

The <pre> and </pre> tags enable more readable format of print\_r output.

**Important Note:**

The while loop in **line 20 - 22** will be executed x times, where x is the number of records retrieved from the database.

The code **$row = mysqli\_fetch\_array($resultMovies)** will retrieve a single record and store all fields from that record into $row. In this case, the fields will be movie\_id, title, and d\_name

The code $arrResult[] = $row adds the content of $row to the array $arrResult.

The following shows how you would access $arrResult to get the respective field values. It is showing the first record since we are indicating the first index as 0

$arrResult[0]['movie\_id']

$arrResult[0]['title']

$arrResult[0]['d\_name']

Line 24, 26 – use of <pre> tag to preformat the output for better clarity

Run the page **exampleViewMovies.php** again on a browser, and examine the output.

You will see a printout of the array **$arrResult**.

How many dimensions of arrays are there? How do you access "Dancing with the Cats"?

*Your answer:*

*There are 4 dimensions of array. In the query you need to state that title = “Dancing with the Cats” to filter out all the other results.*

As you can see, there are two dimensions:

* First dimension corresponds to the records from the database table
* Second dimension corresponds to the fields within each record. Since we are using **mysqli\_fetch\_array()**, these fields are represented two times: using numerical index (0, 1, 2, 3, etc) and associative index (movie\_id, title).

1. We will examine another two fetch methods: **mysqli\_fetch\_row()** and **mysqli\_fetch\_assoc()**

**TODO:** Modify the page as follows:



Line 20 – use mysqli\_fetch\_row()

Run the page again on a browser, and examine the output.

How many dimensions are there in the returned array? For each dimension, is it using numerical or associate index? For first record data, indicate how you will print the movie\_id field from $arrResult

*Your answer:*

*4 dimensions. Its using numerical index.*

**TODO:** Modify the page as follows:



Line 20 – use mysqli\_fetch\_assoc()

Run the page again on a browser, and examine the output.

How many arrays are returned? For each array, is it using numerical or associative index?

For first record data, indicate how you will print the movie\_id field from $arrResult.

*Your answer:*

*4. its using associative index. You can*

For further reading on the three fetch methods, you can use the following resources:

<http://sg.php.net/manual/en/mysqli-result.fetch-array.php>

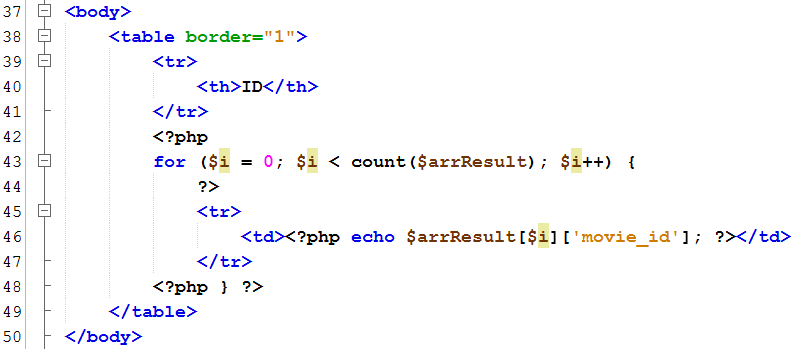
<http://sg.php.net/manual/en/mysqli-result.fetch-assoc.php>

<http://sg.php.net/manual/en/mysqli-result.fetch-row.php>

# Retrieve Records From Database Tables and Display Them in an HTML Table

*This section covers how to retrieve and display multiple database records using PHP and format them using HTML Table*

1. **TODO:** Modify the page **exampleViewMovies.php** to format the output in an HTML table.



Line 38, 49 – HTML Table opening and closing tags

Line 39 – 41 – HTML table row containing the headers

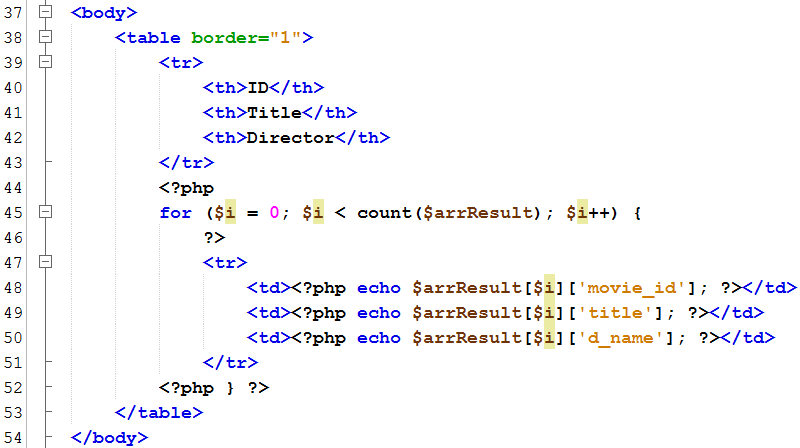
Line 43, 48 – For Loop opening, For Loop closing

Line 45 – 47 – HTML table row that is being **repeated under the For Loop**

Line 46 – HTML table column where field **movie\_id** is displayed using the echo command

Run the page again on the browser. Note that only one field is displayed (ID).

**TODO:** Modify the page further to display all database table columns:



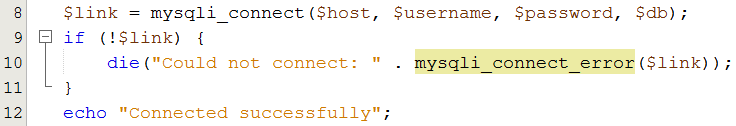
Line 41, 42 – Two additional headers

Line 49 – HTML Table column where field **title** is displayed

Line 50 – HTML Table column where field **d\_name** is displayed

1. Add the *die()* function. It exits the script if there is an error with the connection code.

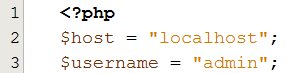
**TODO:** Modify the database connection code in **exampleViewMovies.php**

**

Line 9 – check whether there is an error with $link

Line 10 – if there is an error, execute the die() function and display the exact error

**TODO:** Change the user from "root" to "admin"

****

Run the page again on the browser. What is displayed?

*Your answer*

**Warning**: mysqli\_connect(): (HY000/1045): Access denied for user 'admin'@'localhost' (using password: NO) in **D:\xampp\htdocs\P08\index.php** on line **9**  
Could not connect: Access denied for user 'admin'@'localhost' (using password: NO)

What does the function *die()* do? When should you use it?

*Your answer*

*Die shows when there is a fail to connect and the reason for why it could not connect.*

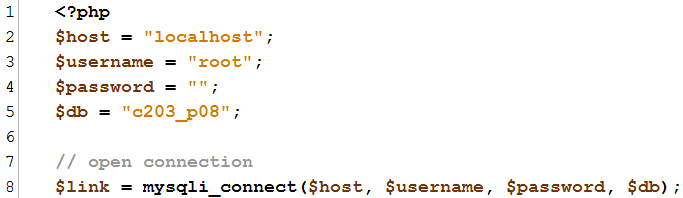
**Remember to change the $username value back to "root"**

# Retrieve Records From Database Tables and Use Them to Populate a Dropdown List

*This section covers how to retrieve and display multiple database records using PHP and insert them into a dropdown list*

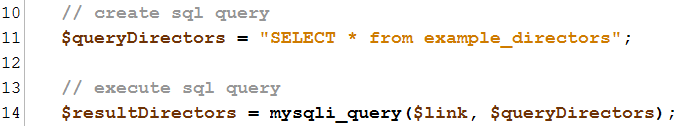
1. We will modify the page **exampleInsertMovies.php**, so that the contents of dropdown list on that page is populated from the database table **directors**

**TODO**: Add the database connection code first:



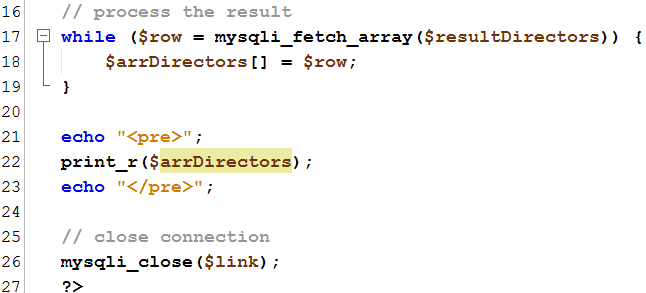
Line 2, 3, 4, 5 – assign values for database connection

Line 8 - open database connection on $link using **mysqli\_connect()** function



Line 11 – create / build the sql query using the SELECT keyword

Line 14 – execute the sql query $queryDirectors on the database connection $link, by using the **mysqli\_query()** function.



Line 26 – close database connection using **mysqli\_close()** function

The while loop in **line 17 - 19** will be executed x times, where x is the number of records retrieved from the database.

The code **$row = mysqli\_fetch\_array($resultDirectors)** will retrieve a single record and store all fields from that record into $row. In this case, the fields will be d\_id and name

The code $arrDirectors[] = $row adds the content of $row to the array $arrDirectors.

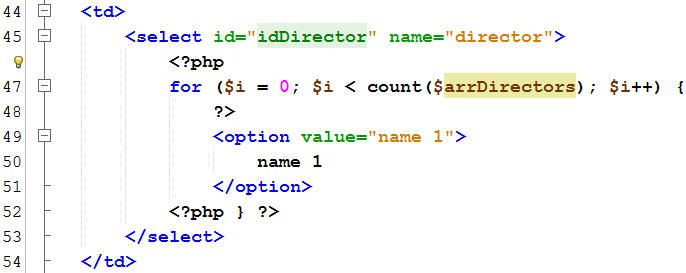
The following shows how you would access $arrDirectors to get the respective field values. It is showing the first record since we are indicating the first index as 0

$arrDirectors[0]['d\_id']

$arrDirectors[0]['name']

1. The next part is on generating the dropdown list <option> tags dynamically

**TODO**: Modify the page as follows:



Line 46, 48 – PHP opening and closing tags

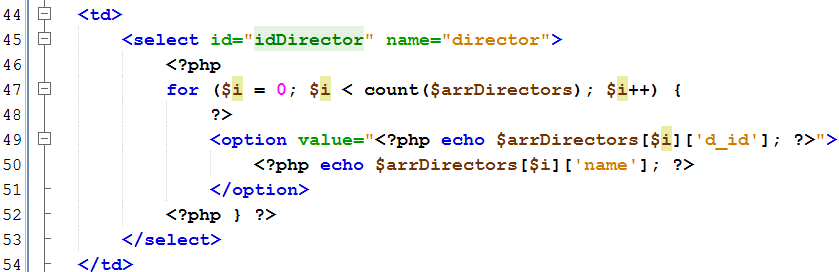
Line 47 – For Loop that goes through the returned records

Line 49 - 51 - <option>, </option> tags, and the content are repeated in the For Loop

Line 52 – For Loop closing bracket

The next part is on generating the **value and contents** of the <option> tags dynamically

**TODO**: Modify the page as follows:



Line 49 – value attribute of <option> tag is assigned with the value of the field **d\_id**

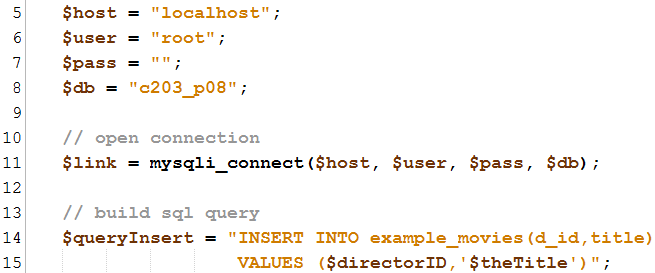
Line 50 – content of the <option> tag is assigned with the value of the field **name**

# Retrieve HTML Form Values and Insert Them to a Database Table (Recap from P05)

*This section is a recap from P05, where form input values are retrieved and then inserted into a database*

1. Here we complete the insertion code for the **example\_movies** table.

**TODO:** Modify the code in **doExampleInsertMovies.php** as follows:

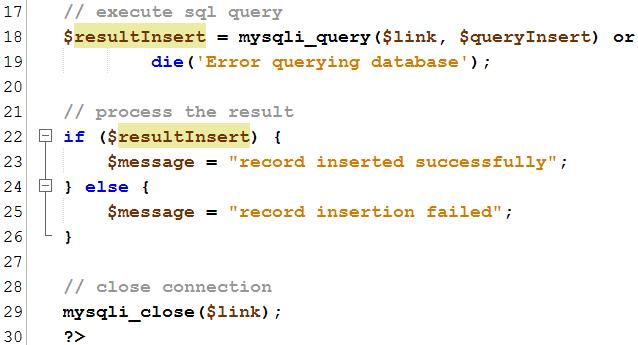


Line 5, 6, 7, 8 – assign values for database connection

Line 11 - open database connection on $link using **mysqli\_connect()** function

Line 14, 15 – build / create an sql query using the INSERT keyword

**TODO:** Further modify to include the codes into **doExampleInsertMovies.php**.

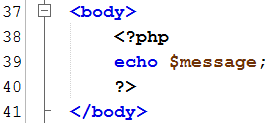


Line 18 – execute the sql query $queryInsert on the database connection $link, by using the **mysqli\_query()** function.

Line 22 - 26 - process the result: check whether the INSERT query is executed successfully. Assign success or error message to $message.

Line 29 – close database connection using **mysqli\_close()** function

**TODO:** Include the echo statement to display the insert status



Line 39 – use echo to print out $message (enclosed within php tags)

**TODO:** Test your pages. Run **exampleInsertMovies.php**, enter the values, and click the submit button. Using PHPMyAdmin, open the database table movies, and check that values are entered correctly.

# Build Your Solution

Complete the pages as indicated below. Refer to the example pages covered in previous sections as a guide.

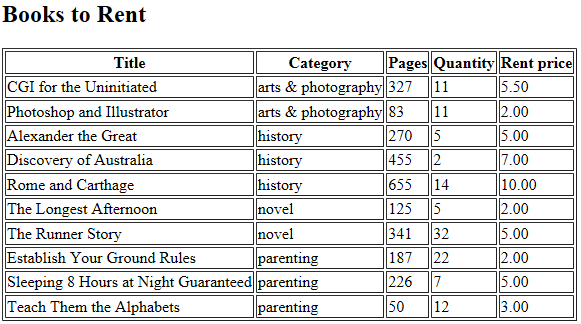
1. **bookList.php**: display all records from the database table **books**
2. **insertBook.php**: modify the dropdown so that it pulls information from the database table **book\_categories**
3. **doInsertBook.php**: complete the database insertion code to table **books**
4. Upon completing all the pages above, export the database c203\_p08 in an sql format. Follow **PHPMyAdmin DB Import and Export Guide.docx**

# Challenge Yourself

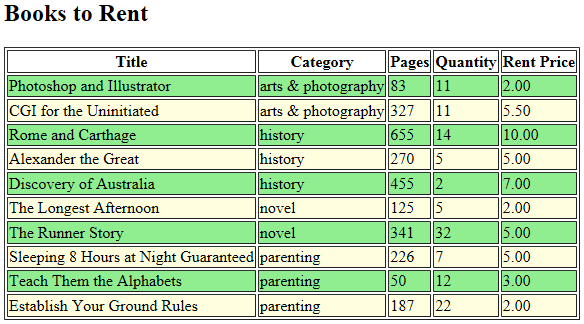
**Resource:**

How to use header function: <http://php.net/manual/en/function.header.php>

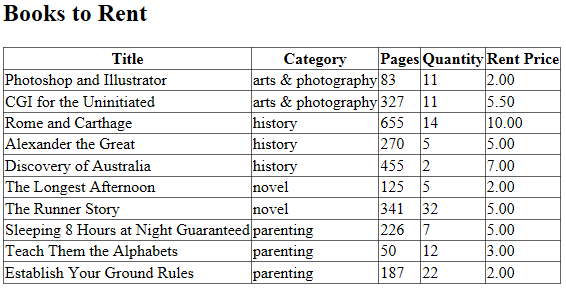
1. On the page **doInsertBook.php**, redirect user automatically to **insertBook.php** upon successful record insertion. Use the **header()** function.
2. On the page **bookList.php**, display book title in ascending order within their category.



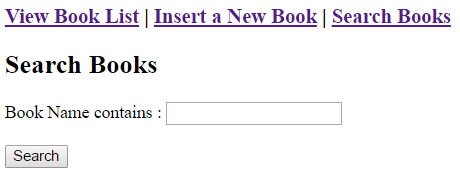
1. On the page **bookList.php**, display the HTML table rows in alternate background colors: lightgreen, lightyellow, and so on. **You will need to use CSS to achieve this.**



1. On the page **bookList.php**, display the HTML table with single border. **You will need to use CSS to achieve this.**



1. Create a “Search Books” page to search for books with titles containing the search criteria.



You need to use the SQL command “like” to perform a wild card search. <http://www.w3schools.com/sql/sql_like.asp>

The search result should return the following columns. If no records are found, the message “No records are found” is displayed.

