CreateElement/AJAX/REST/JSON Lab

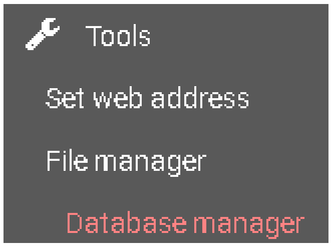
Objectives

In this lab, we will learn how to:

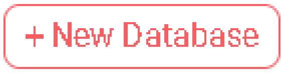
- Send information from a web page to PHP using REST
- Use PHP to query a database
- Retrieve data from PHP into a web form using AJAX and JSON
- Dynamically change a web page using createElement.

Exercise 1: Using REST to Query a Database

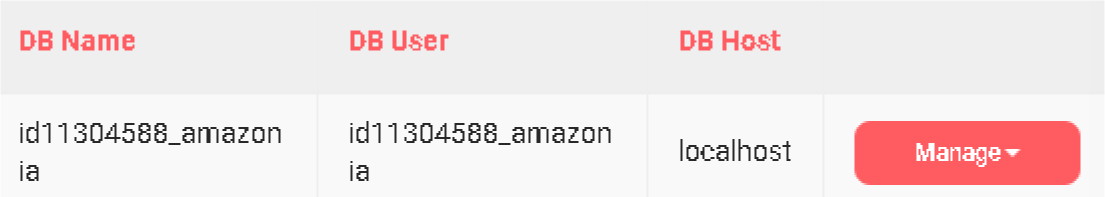
Log into webhost. Select Tools, Database Manager.



Select New Database



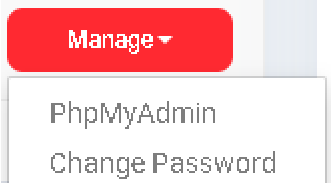
When you create the database, it will not actually give it the name you define. Instead, that name will be appended to a string of digits. In my case, the database is id11304588_amazonia. However, since I created my account, webhost has updated its security. Call your database myamazonia, your username <loginid>amazonia and set your password to Stupidsecurity0!.



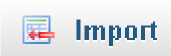
DB Name	DB User	DB Host	
id11304588_amazonia	id11304588_amazonia	localhost	Manage ▾

Regardless, make sure that whatever you put as the database name, username and password, you know what it is. We will need this information later.

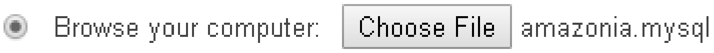
Now click Manage\PhpMyAdmin



In PhpMyAdmin, click on the database. In my case, it is id11304588_amazonia. Click Import.

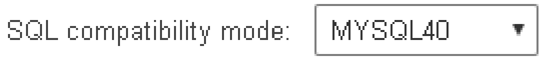


Choose File Amazonia.mysql.



● Browse your computer: Choose File amazonia.mysql

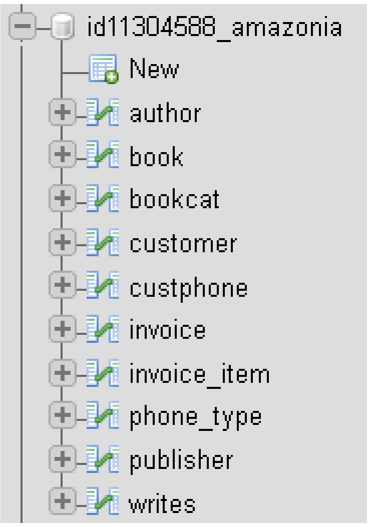
Format should be SQL (default). SQL compatibility mode: MySQL 40



SQL compatibility mode: MYSQL40

Click Go.

If successful, your database should be populated by 10 tables.



id11304588_amazonia
New
author
book
bookcat
customer
custphone
invoice
invoice_item
phone_type
publisher
writes

Exit out of phpMyAdmin and go back to WebHost. You should now have your Amazonia database installed.

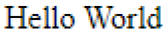
Create a file called books.php

A PHP REST module begins with <?php and ends with ?>.

In books.php, type the following:

```
<?php
  echo 'Hello World';
?>
```

If you are successful, your screen should look like this:


Hello World

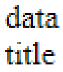
A PHP module receives information from the web via the $_REQUEST[] container. All PHP variables are prefaced by a $ sign. We are going to receive two variables from the web- title, and sort. Change your code now as follows:

```
<?php
  $title=$_REQUEST['title'];
  $sort=$_REQUEST['sort'];
  echo $title;
  echo '<br />';
  echo $sort;
?>
```

You pass information to PHP by putting a question mark at the end of the file name and then putting a sequence of variable value pairs separated by an ampersand (&). Thus, you would say:

http://<your domain>/<your directory>/books.php?title=data&sort=title

Try this now:


data
title

A database query in PHP has three parts. First, you create a PHP Data Object (PDO). This is a connection to the database. Your PDO must include the database driver, the URL to the database, the database name, the username and password. Next, you prepare the SQL query. Finally, you execute it.

Typically, the results of your query are to be sent out in JavaScript Object Notation (JSON) format.

Change your PHP code so it looks like this:

```
1.  <?php
2.   $title=strtoupper($_REQUEST['title']);
3.   if (strlen($title)==0)
4.     $title='%';
5.   else
6.     $title='%'.$title.'%';
7.   $sort=$_REQUEST['sort'];
8.   $mydb = new PDO('mysql:host=localhost;dbname=<database name>', ‘<username>’,
    ‘<password>’);
9.   $stmt=$mydb->prepare("Select book.isbn, title, author_first, author_last, ".
10.                      "listprice, discount, book.category_code, category_name ".
11.                      "from book ".
12.                      "inner join writes ".
13.                      "on book.isbn=writes.isbn ".
14.                      "inner join author ".
15.                      "on writes.authorid=author.author_id ".
16.                      "inner join bookcat on book.category_code=bookcat.category_code ".
17.                      "where upper(title) like :thetitle ".
18.                      "order by ".$sort.", isbn, authororder"
19.                     );
20.  $stmt->execute([":thetitle"=>$title]);
21.  $result=$stmt->fetchAll();
22.  echo json_encode($result);
23. ?>
```

Here’s what the code does.

Line 2: Read the title from the web and capitalize it.

Line 3-6: If the title is empty, change it to %. Otherwise put % in the front and end of the title. % is the wildcard character in SQL. By putting % in front and back it means we are looking for any occurrence of that word in the title.

Line 8: Connect to the database

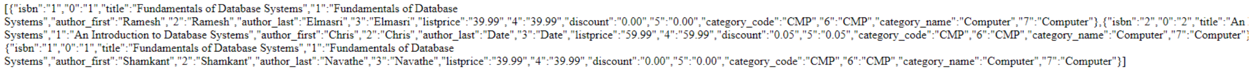
Line 9-19: The SQL statement. NOTE THE EMPTY SPACE BEFORE THE QUOTES AND THE PERIOD AT THE END OF EACH LINE. They are important. Note the sort variable embedded in the query. This is actually a security risk. Unfortunately, at this time, PDO does not easily allow parameters to be passed to an order by clause.

Line 20: Run the SQL query using title in place of the parameter thetitle. By rights, we should do this with sort, but PDO does not allow this.

Line 21: Retrieve the SQL results into a PHP variable called result

Line 22: Convert result to JSON and send it to whoever requested it.

If you did this correctly, you should see:



```
[{"isbn":"1","0":"1","title":"Fundamentals of Database Systems","1":"Fundamentals of Database
Systems","author_first":"Ramesh","2":"Ramesh","author_last":"Elmasri","3":"Elmasri","listprice":"39.99","4":"39.99","discount":"0.00","5":"0.00","category_code":"CMP","6":"CMP","category_name":"Computer","7":"Computer"},{"isbn":"2","0":"2","title":"An
Systems","1":"An Introduction to Database Systems","author_first":"Chris","2":"Chris","author_last":"Date","3":"Date","listprice":"59.99","4":"59.99","discount":"0.05","5":"0.05","category_code":"CMP","6":"CMP","category_name":"Computer","7":"Computer"}
{"isbn":"1","0":"1","title":"Fundamentals of Database Systems","1":"Fundamentals of Database
Systems","author_first":"Shamkant","2":"Shamkant","author_last":"Navathe","3":"Navathe","listprice":"39.99","4":"39.99","discount":"0.00","5":"0.00","category_code":"CMP","6":"CMP","category_name":"Computer","7":"Computer"}]
```

Try this with a number of different titles and sort orders to see what happens.

Exercise 2: Retrieve REST results into JavaScript using AJAX.

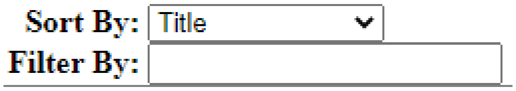
Create a webpage in HTML called viewbooks.htm with the following code:

```
<!doctype html>
<html>
  <head>
    <style>
      .leftcol {
        display: inline-block;
        font-weight: bold;
        width: 80px;
        text-align: right;
      }
      .rightcol{
        display: inline-block;
      }
    </style>
    <script>

    </script>
  </head>
  <body>
    <form>
      <div class="leftcol">
        <label for="sortby">Sort By:</label>
      </div>
      <div class="rightcol">
        <select id="sortby">
          <option value="isbn">ISBN</option>
          <option value="title" selected>Title</option>
          <option value="category_name" >Category Name</option>
          <option value="listprice" >List Price</option>
          <option value="discount" >Discount</option>
        </select>
      </div><br />
      <div class="leftcol">
        <label for="filt">Filter By:</label>
      </div>
      <div class="rightcol">
        <input type="text" id="filt" />
      </div><br />

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

If you did this right, your screen should look like this:



Sort By: Title

Filter By:

The main thing we want to observe is the pulldown, created by the select tag. This is the code for the pulldown

```
1.  <select id="sortby">
2.    <option value="isbn">ISBN</option>
3.    <option value="title" selected>Title</option>
4.    <option value="category_name" >Category Name</option>
5.    <option value="listprice" >List Price</option>
6.    <option value="discount" >Discount</option>
7.  </select>
```

Line 1: The pulldown is created by the select tag

Line 2-6: The options of the pulldown are created by the option tag. Also notice the value attribute. This is the information the option tag actually holds. The innerHTML is what is displayed to the user.

Line 3: The selected attribute identifies which option is currently selected.

Now, we are going to add a table. Add the following code just before the end form tag.

```
<table border="1">
  <thead>
    <tr>
      <th>ISBN</th>
      <th>Title</th>
      <th>Authors</th>
      <th>Category Name</th>
      <th>List Price</th>
      <th>Discount</th>
    </tr>
  </thead>
  <tbody id="changehere">


  </tbody>
</table>
```

The ID changehere is especially important because we are going to work with this.

It is now time for us to do our AJAX magic. Create the following function:

```
1.  function displayBooks(thetitle,thesort) {
2.  const xhttp = new XMLHttpRequest();
3.  xhttp.onload = function() {
4.    var bookarray = JSON.parse(this.responseText);
5.    //more Javascript code will go here
6.  }
7.  if (thetitle.length==0)
8.    xhttp.open("GET", "books.php?sort="+thesort, true);
9.  else
10.   xhttp.open("GET", "books.php?title="+thetitle+"&sort="+thesort, true);
11. xhttp.send();
12. }
```

This is fundamentally the structure of AJAX code, which we explain below:

Line 2: Create an AJAX request.

Line 3-6: When the AJAX request receives something from PHP, do what is in the function. The returned value from PHP is found in this.responseText

Line 4: Convert what PHP returned from JSON to a variable

Line 8/10: Create the connection to PHP

Line 11: Actually perform the connection

The actual function we will apply is as follows. Replace the code that says //more Javascript code will go here with:

```
var x=0;
var rowobj=null;
var cellobj=null;
var theisbn="";
changehere.innerHTML="";
for (x=0; x<bookarray.length; x++){
  if (bookarray[x].isbn==theisbn){
    document.getElementById("author"+theisbn).innerHTML+="<br />"+
            bookarray[x].author_first+" "+bookarray[x].author_last;
  }
  else {
   theisbn=bookarray[x].isbn;
   rowobj=document.createElement("tr");
   cellobj=document.createElement("td");
   cellobj.style="vertical-align: top";
   cellobj.innerHTML=bookarray[x].isbn;
   rowobj.appendChild(cellobj);
```

```
    cellobj=document.createElement("td");
    cellobj.style="vertical-align: top";
    cellobj.innerHTML=bookarray[x].title;
    rowobj.appendChild(cellobj);
    cellobj=document.createElement("td");
    cellobj.style="vertical-align: top";
    cellobj.id="author"+theisbn;
    cellobj.innerHTML=bookarray[x].author_first+" "+bookarray[x].author_last;
    rowobj.appendChild(cellobj);
    cellobj=document.createElement("td");
    cellobj.style="vertical-align: top";
    cellobj.innerHTML=bookarray[x].category_name;
    rowobj.appendChild(cellobj);
    cellobj=document.createElement("td");
    cellobj.style="vertical-align: top";
    cellobj.innerHTML=bookarray[x].listprice;
    rowobj.appendChild(cellobj);
    cellobj=document.createElement("td");
    cellobj.style="vertical-align: top";
    cellobj.innerHTML=bookarray[x].discount;
    rowobj.appendChild(cellobj);
    changehere.appendChild(rowobj);
   }
 }
```

This is a lot of code, but it is quite repetitive. What we are doing is looping through the array of results returned from JSON.

We first blank changehere (changehere.innerHTML=””).

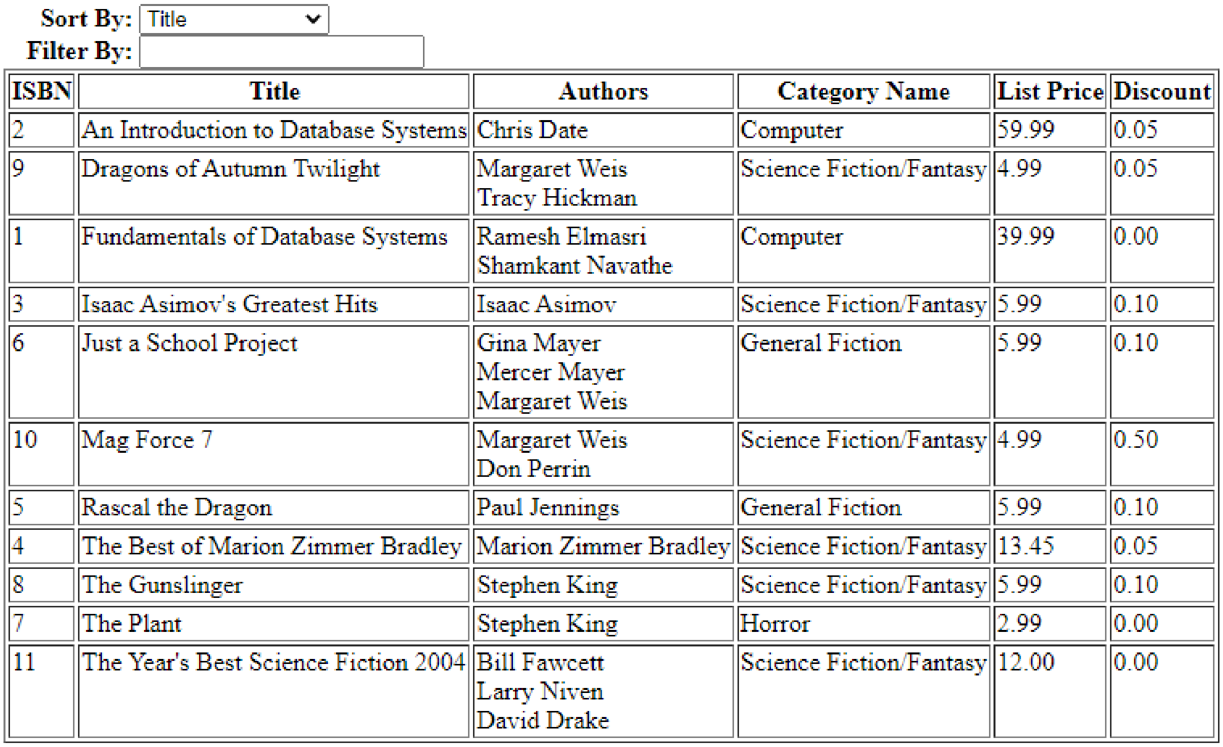
If the row has the same ISBN as the previous row, this means it is the same book. If so, just add the author to the correct column.

Otherwise, we use createElement to create a new row (tr). We add the ISBN, title, author, etc. cells to the row. We do this by usingcreateElement to create the cells (td). We then use appendChild to add it to the row. We then use appendChild to add the row to changehere.

Link the code to the onload event of the body tag:

<body **onload="displayBooks('','title');"**>

Run your code. If you were successful, your screen should look like this:



Sort By: Title

Filter By:

ISBN	Title	Authors	Category Name	List Price	Discount
2	An Introduction to Database Systems	Chris Date	Computer	59.99	0.05
9	Dragons of Autumn Twilight	Margaret Weis Tracy Hickman	Science Fiction/Fantasy	4.99	0.05
1	Fundamentals of Database Systems	Ramesh Elmasri Shamkant Navathe	Computer	39.99	0.00
3	Isaac Asimov's Greatest Hits	Isaac Asimov	Science Fiction/Fantasy	5.99	0.10
6	Just a School Project	Gina Mayer Mercer Mayer Margaret Weis	General Fiction	5.99	0.10
10	Mag Force 7	Margaret Weis Don Perrin	Science Fiction/Fantasy	4.99	0.50
5	Rascal the Dragon	Paul Jennings	General Fiction	5.99	0.10
4	The Best of Marion Zimmer Bradley	Marion Zimmer Bradley	Science Fiction/Fantasy	13.45	0.05
8	The Gunslinger	Stephen King	Science Fiction/Fantasy	5.99	0.10
7	The Plant	Stephen King	Horror	2.99	0.00
11	The Year's Best Science Fiction 2004	Bill Fawcett Larry Niven David Drake	Science Fiction/Fantasy	12.00	0.00

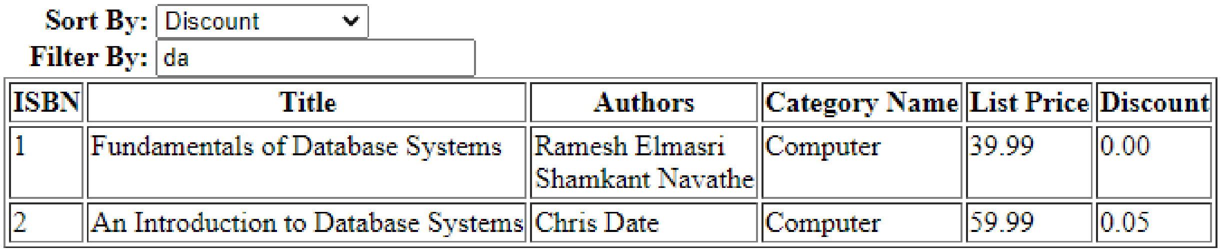
Now we have to link the code to the pulldown and the textbox. The event for a pulldown is onchange.

<select id="sortby" **onchange="displayBooks(filt.value,sortby.value);"**>

Set the onkeyup event of the textbox to do AJAX.

<input type="text" id="filt" **onkeyup="displayBooks(filt.value,sortby.value);"**/>

If you did everything right, you should be able to do this:



Sort By: Discount

Filter By: da

ISBN	Title	Authors	Category Name	List Price	Discount
1	Fundamentals of Database Systems	Ramesh Elmasri Shamkant Navathe	Computer	39.99	0.00
2	An Introduction to Database Systems	Chris Date	Computer	59.99	0.05