Jack Raney

4/21/2016

In this assignment, I chose to demo the retrieval and insertion of database values through my previously used League of Legends Champions database.

I created a simple html page that allows for this:

<!DOCTYPE html>

<html>

<head>

<title>Champions Database Page</title>

<link type="text/css" rel="stylesheet" href="ajaxchampionsstyle.css"/>

<script src="ajaxchampionsscript.js"></script>

<script src="jquery.js"></script>

</head>

<body onload = "onLoad()">

<div id="left">

<p>

Champion Name:<br>

Primary Lane:<br>

Primary Role:<br>

<button type="button" onclick="insertChampion()">Insert</button><br><br>

<button type="button" onclick="showChampions()">Show Champions</button>

</p>

</div>

<div id="right"><br>

<input id="ChampionName" type=text value=""><br>

<select id="Lane">

<option value=1>Top</option>

<option value=2>Jungle</option>

<option value=3>Middle</option>

<option value=4>Bottom</option>

</select><br>

<select id="Role">

<option value=1>Assassin</option>

<option value=2>Fighter</option>

<option value=3>Mage</option>

<option value=4>Marksman</option>

<option value=5>Support</option>

<option value=6>Tank</option>

</select><br>

</div>

<div id ="below">

<p id="results"></p>

<p id="champions"></p>

</div>

</body>

</html>

The results with accompanying css appears as such:



Using the ajax, php, and jquery we were taught in class and through the videos, I created this javascript file to accompany the page:

var champions;

function onLoad()

{

getChampions(false);

}

function insertChampion()

{

var Champion, LaneID, RoleID;

Champion = JSON.stringify($("#ChampionName").val());

LaneID = $("#Lane option:selected").val();

RoleID = $("#Role option:selected").val();

ajax = ajaxInsertChampion("insertChampion", Champion, LaneID, RoleID);

ajax.done(insertChampionCallback);

ajax.fail(function(){

alert("Failure");

});

}

function ajaxInsertChampion(method, Champion, LaneID, RoleID)

{

return $.ajax({

url: 'ShadyAPI.php',

type: 'POST',

data: {method: method,

Champion: Champion,

LaneID: LaneID,

RoleID: RoleID

}

});

}

function insertChampionCallback(response\_in)

{

response = JSON.parse(response\_in);

if (!response["success"])

{

$("#results").html("");

alert("Insert failed on query:" + "\n" + response['querystring']);

}

else

{

$("#results").html(

response['querystring'] + '<br>' +

response['success'] + '<br>');

getChampions(false);

}

}

function showChampions()

{

var championList = "";

{

$.each(champions, function (key, value)

{

var itemString = "";

$.each(value, function (key, item)

{

itemString += "<td>&nbsp" + item + "&nbsp</td>";

});

championList += "<tr>&nbsp" + itemString + "&nbsp</tr>";

});

$("#champions").html("<table><tr><td>&nbspChampion&nbsp</td><td>&nbspLane&nbsp</td><td>&nbspRole&nbsp</td>" + championList + "</table>");

}

}

function getChampions(async)

{

ajax = ajaxGetChampions("getChampions", async);

ajax.done(getChampionsCallback);

ajax.fail(function (){

alert("Failure");

});

}

function ajaxGetChampions(method, async)

{

return $.ajax({

url: "ShadyAPI.php",

type: 'POST',

async: async,

data: {method: method}

});

}

function getChampionsCallback(response\_in)

{

var response = JSON.parse(response\_in);

champions = response["champions"];

if(!response["champions"])

{

$("#results").html("getChampions failed");

}

}

This JavaScript references a PHP file, which I have copied and pasted below:

<?php

echo $\_POST["method"]();

function sanitize($str, $quotes = ENT\_NOQUOTES)

{

$str = htmlspecialchars($str, $quotes);

return $str;

}

function getChampions()

{

$dbConn = mysqli\_connect("23.253.61.96", "ncJonathanR",

";;ncJonathanR;;", "ncJonathanR");

$query = "SELECT Champion, Lane, Role

FROM ncJonathanR.Champions, ncJonathanR.Lanes, ncJonathanR.Roles

WHERE Champions.LaneID = Lanes.LaneID and Champions.RoleID = Roles.RoleID;";

$result = $dbConn->query($query);

if($dbConn->connect\_error){

$return->connect\_error = "Connection failed: " . $dbConn->connect\_error;

$return->success = false;

return json\_encode($return);

}

$champions = array();

if($result){

while($row = $result->fetch\_array()){

$allColumns = array();

for ($i = 0; $i < 3; $i++){

array\_push($allColumns, $row[$i]);

}

array\_push($champions, $allColumns);

}

}

$return = new StdClass();

$return->success = true;

$return->champions = $champions;

$return->querystring = $query;

return json\_encode($return);

}

function insertChampion()

{

if(isset($\_POST['Champion'])){

$Champion = json\_decode(sanitize($\_POST['Champion']));

}

if(isset($\_POST['LaneID'])){

$LaneID = json\_decode(sanitize($\_POST['LaneID']));

}

if(isset($\_POST['RoleID'])){

$RoleID = json\_decode(sanitize($\_POST['RoleID']));

}

$dbConn = mysqli\_connect("23.253.61.96", "ncJonathanR",

";;ncJonathanR;;", "ncJonathanR");

if($dbConn->connect\_error){

die("Connection failed: " . $dbConn->connect\_error);

}

$query = "INSERT INTO ncJonathanR.Champions ( Champion, LaneID, RoleID ) " .

"VALUES ( '" . $Champion . "', " . $LaneID . ", " . $RoleID .

" );";

$result = $dbConn->query($query);

$return = new StdClass();

$return->querystring = (string) $query;

if ($result)

{

$return->success = true;

}

else

{

$return->success = false;

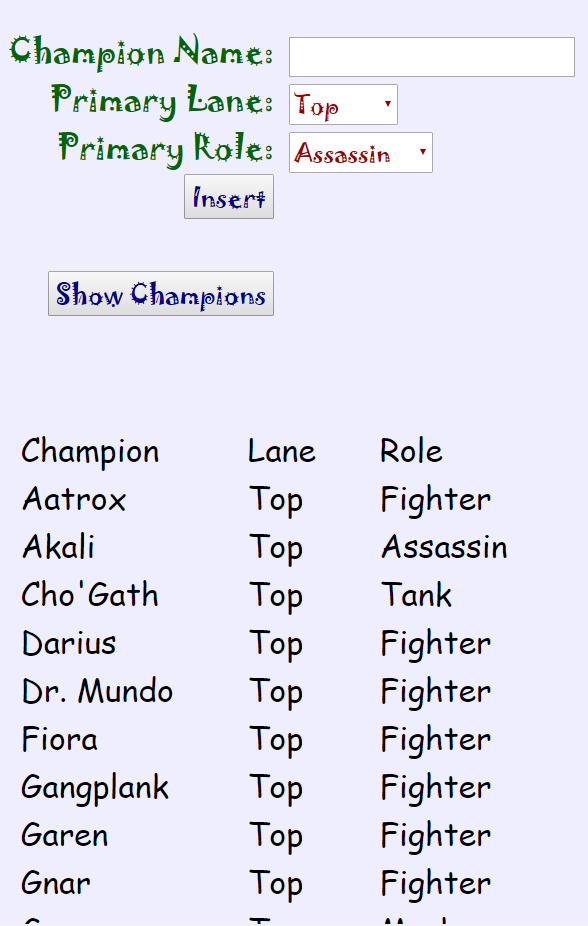
}

return json\_encode($return);

}

That’s enough code. Let’s see an example of the page in action.

When the page loads, it gets the champions from the database, which can be viewed by clicking the “show champions” button:

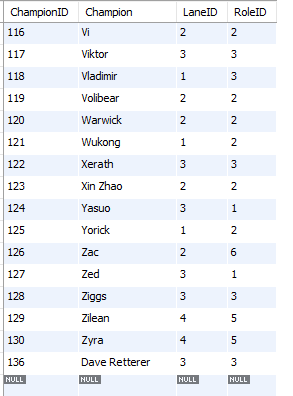


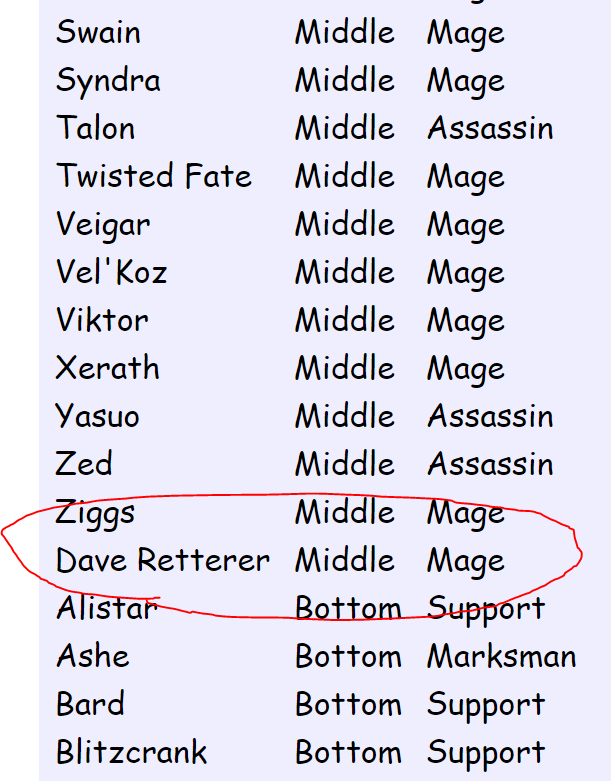
Notice that the champions are put into a table format with column titles for readability.

If you can’t tell, champions can also be inserted into the table using the three parameters at the top of the page. Simply Type a champion name, select the Champ’s lane and role, and press the “insert” button:



This new data can be seen in the database now:



Clicking the “Show Champions” Button will also update the table to display the CSS wizard on the page:  


That’s about it, then.

Special thanks go to Dr. Retterer for his lectures and videos, and to all of the fonts I’ve used in these assignments to date:

Times New Roman

Arial Narrow

Courier New

Comic Sans MS

Calibri

 (Wingdings)

Papyrus

Jokerman

Algerian