Taylor Harris

Database Insert Assignment

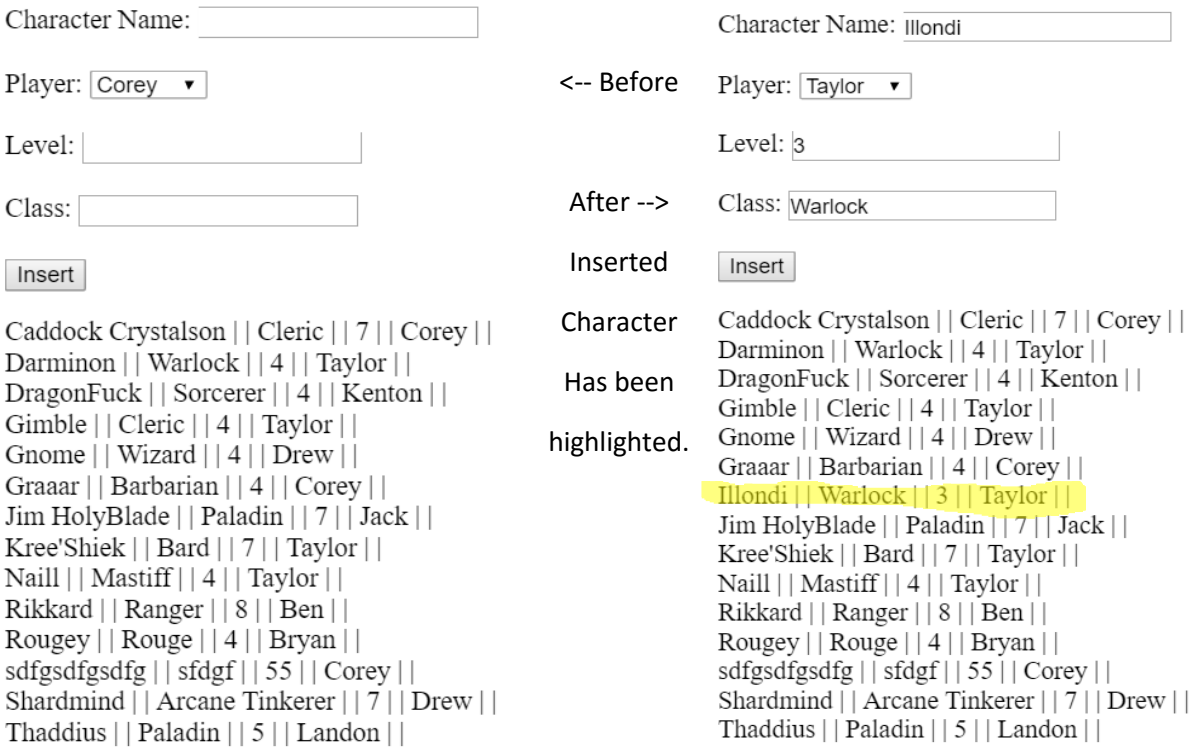
Net-Centric Computing

Prof. Retterer

Database Insert Assignment

This assignment was to create a webpage to display a database and allow the user to insert additional information to the database. My project references two tables, a Player table and a CharacterInfo table. Both Tables are based around current D&D campaigns on campus that I am a part of. The Player table contains a list of player names and an Id number as a foreign key to the CharacterInfo table. The CharacterInfo table contains the Name, Level, and Class of D&D characters as well as an Id reference that functions as a foreign key to the Player Table. I utilized a selection box in html to make the player names as a sort of parent to the characters, in doing so I guaranteed that a user can't add new players only new characters. I did this because a new character wouldn't have an id associated with it which would ruin the references between tables. Below I have screen shots of all my select and insert statements as well as a screen shot of the site and a few screenshots of sample jQuery used to call the Ajax in php files used to select and insert to and from the sql database.

Below are two screenshots of the page from user's perspective one before adding any new characters and one after.



```php
function getCharacterInfo() {
    $dbConn = mysqli_connect(server(), username(), password(), db());
    $query = "select CharacterName, Class, Level, Player_Names from CharacterInfo,
Players where CharacterInfo.Id = Players.Id ";
    $result = $dbConn->query($query);
    if ($dbConn->connect_error) {
        $return->connect_error = "Connection failed: " . $dbConn->connect_error;
        $return->success = false;
        return json_encode($return);
    }
    $Character = array();
    if ($result) {
        while ($row = $result->fetch_array()) {
            $allColumns = array();
            for ($i = 0; $i < 4; $i++) {
                array_push($allColumns, $row[$i]);
            }
            array_push($Character, $allColumns);
        }
    }
    $return = new stdClass();
    $return->success = true;
    $return->Character = $Character;
    $return->querystring = $query;
    return json_encode($return);
}
function getPlayers() {
    $dbConn = mysqli_connect(server(), username(), password(), db());
    $query = "SELECT * FROM Players";
    $result = $dbConn->query($query);
    if ($dbConn->connect_error) {
        $return->connect_error = "Connection failed: " . $dbConn->connect_error;
        $return->success = false;
        return json_encode($return);
    }
    $Players = array();
    if ($result) {
        while ($row = $result->fetch_array()) {
            $allColumns = array();
            for ($i = 0; $i < 2; $i++) {
                array_push($allColumns, $row[$i]);
            }
            array_push($Players, $allColumns);
        }
    }
    $return = new stdClass();
    $return->success = true;
    $return->Players = $Players;
    $return->querystring = $query;
    return json_encode($return);
}
```

This is a screenshot of the two select statements. The first is used to show all the character names, classes, and levels next to the name of their player. The second statement is used to populate the select box used when wanting to insert a character.

```php
function insertCharactersInfo() {
    if (isset($_POST['Level'])) {
        $Level = json_decode(sanitize($_POST['Level']));
    }
    if (isset($_POST['Id'])) {
        $Id = json_decode(sanitize($_POST['Id']));
    }
    if (isset($_POST['CharacterName'])) {
        $CharacterName = json_decode(sanitize($_POST['CharacterName']));
    }
    if (isset($_POST['Class'])) {
        $Class = json_decode(sanitize($_POST['Class']));
    }
    $dbConn = mysqli_connect(server(), username(), password(), db());
    if ($dbConn->connect_error) {
        die("Connection failed: " . $dbConn->connect_error);
    }
    $query = "INSERT INTO CharacterInfo ( CharacterName, Class, Level, Id ) " .
        "VALUES ( " .
        "'" . $CharacterName . "', " .
        "'" . $Class . "', " .
        "'" . $Level . "', " .
        "'" . $Id . "');";
    $result = $dbConn->query($query);
    $return = new stdClass;
    $return->querystring = (string) $query;
    if ($result) {
        $return->success = true;
    } else {
        $return->success = false;
    }
    return json_encode($return);
}
```

This is the insert statement used to add characters to the CharacterInfo table. It uses the id populated from the select box on the html page as Id and user inserted values for the other three variables.

On the next page is a sample of three jQuery statements used to call Ajax statements and used in the process of inserting characters. It also shows the onLoad function used to display the characters and their information as well as populating the select bar.

```
function onLoad() {
    getCharacterInfo(false);
    getPlayers(false);

}

function insertCharacter() {
   // alert("in insertCharacter");
    var CharacterName,
        Id,
        Level,
        Class;
    CharacterName = JSON.stringify($('#CharacterName').val());
    Id = JSON.stringify($('#Id option:selected').val());
    Level = JSON.stringify($('#Level').val());
    Class = JSON.stringify($('#Class').val());
    //alert(CharacterName);
   // alert(Id);
    //alert(Level);
    //alert(Class);
    ajax = ajaxinsertCharactersInfo("insertCharactersInfo", Level, Id, CharacterName, Class);
    ajax.done(insertCharactersCallback);
    ajax.fail(function() {
        alert("Failure");
    });
}

function ajaxinsertCharactersInfo(method, Level, Id, CharacterName, Class) {
    return $.ajax({
        url : 'Database.php',
        type : 'POST',
        data : {
            method : method,
            Level : Level,
            Id : Id,
            CharacterName : CharacterName,
            Class : Class
        }
    });
}

function insertCharactersCallback(response_in) {
    response = JSON.parse(response_in);

    if (!response['success']) {
        $("#results").html("");
        alert("Insert failed on query:" + '\n' + response['querystring']);
        getPlayers(false);
        getCharacterInfo(false);
    } else {
        $("results").html(response['querystring'] + '<br>' + response['success'] + '<br>');
        getPlayers(false);
        getCharacterInfo(false);
    }
}
```