Kyle Cole  
ITS362 Final Project Report

04/19/2020

**Terebigemu.net**

**Grading Rubric:**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Specifics** | **Points** |
| Project Completion | - Do you include your project design?  - Do you implement all the features and functions as designed?  - Do they work as expected?  - Do you implement the app using appropriate Web technologies? | 30 |
| Project Design | - What is the big picture/architecture of your app?  - Is it object-oriented?  - What are the techniques you proposed to use? | 15 |
| User Interface | - Does your application have an easy-to-use, intuitive, and nice-looking user interface?  - How is the usability of your UI?  - Do you choose the right GUI controls for different data types and/or interactions? | 15 |
| Code Style | - Do you use simple and easy-to-follow logic?  - Do you rename all the identifiers? (as opposed to using the default names)  - Do you document your code?  - Do you have any error handling in your code?  - Do you have any hard-code or useless code? | 20 |
| Presentation/Deliverables | - Do you present your completed project effectively?  - Do you answer all audience questions satisfactorily?  - Do you deliver the project documents as required? | 20 |

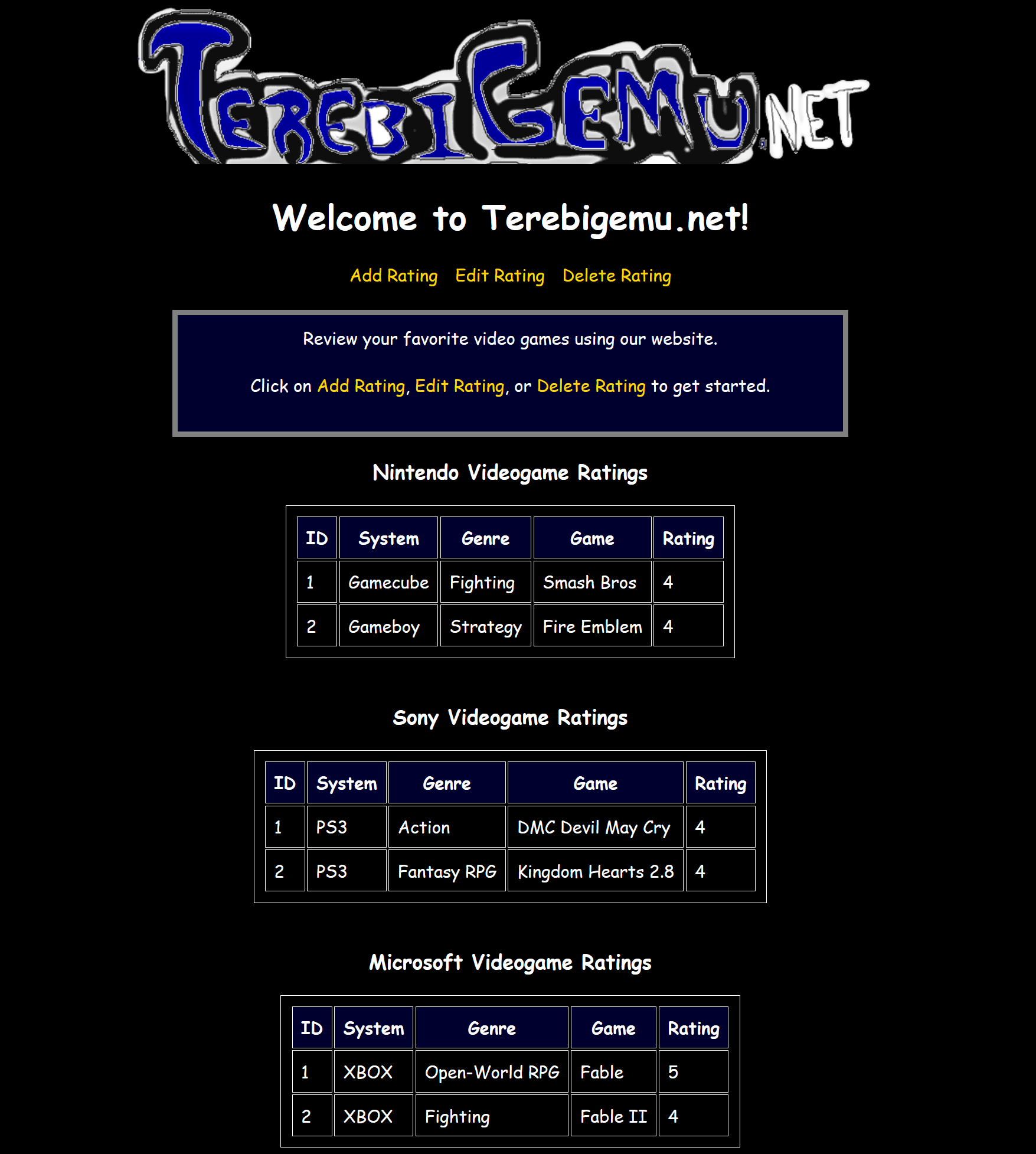
**Background**

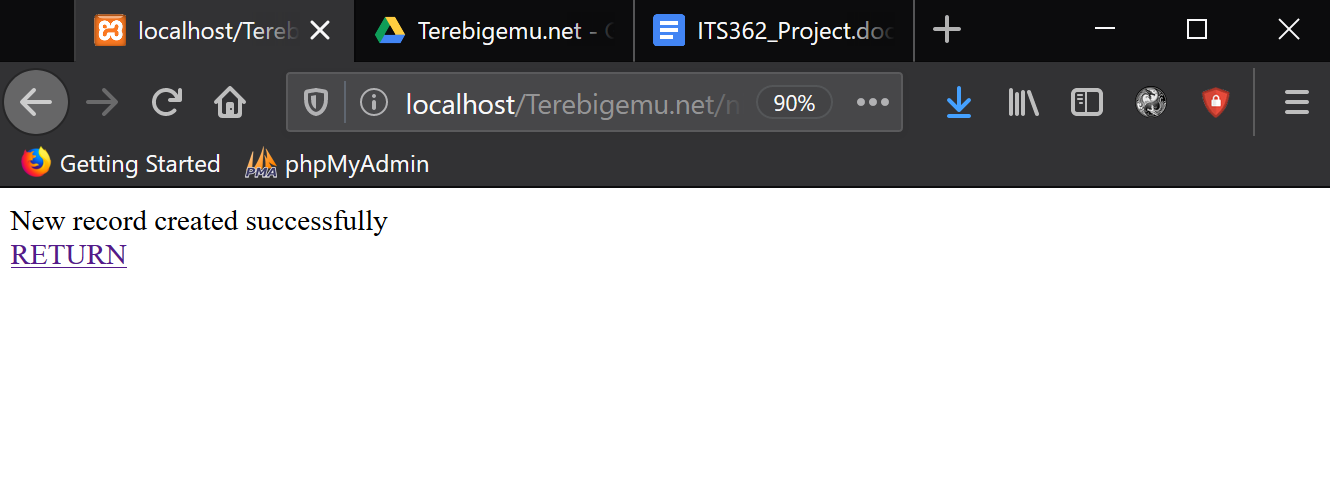
**Terebigemu.net** is a web-based application that allows users to **view**, **insert**, **update**, and **delete** records from a database using the web application. There are a total of three tables in the database: Nintendo, Sony, and Microsoft. Within each of these tables are the following columns: System\_ID, Name, Genre, Game, and Rating.   
  
System\_ID serves as the row number as well as the unique identifier that the database will use to interact with a specific row for a specific table. Name is a placeholder for the name of the videogame platform (system name). XBOX, PS3, and Switch are examples of “Names” (System)s.   
  
Genre is the type of videogame the user is rating. An example of a Genre would be Action RPG. Game is for the name of the videogame that the user is applying a rating to. And Rating is the column where users will place a number from 1 to 5 as their rating.

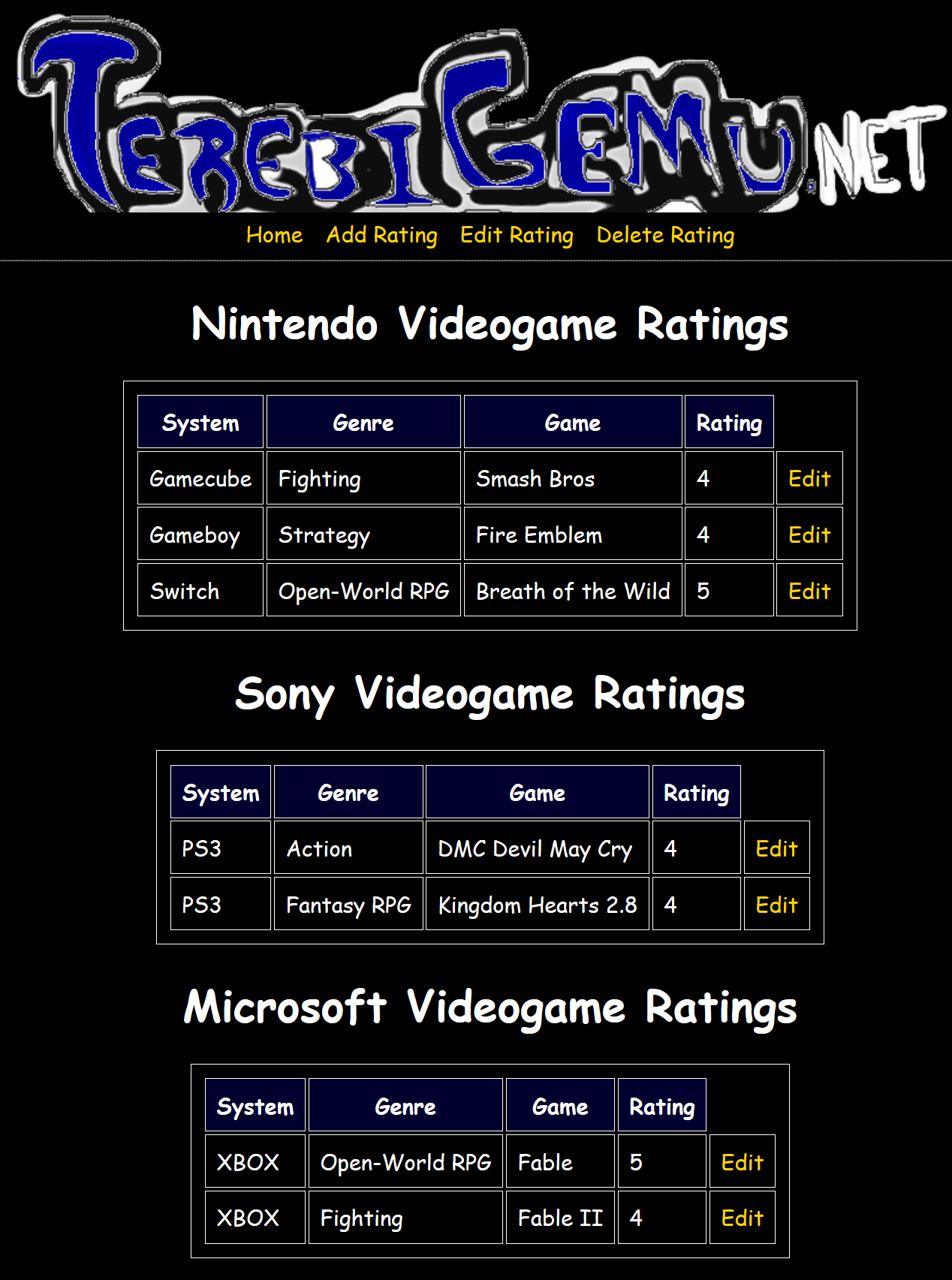
This topic has been approved by the instructor. This project makes use of HTML, PHP, JavaScript, jQuery (standard jQuery and jQuery + UI), and SQL.

**Descriptions**

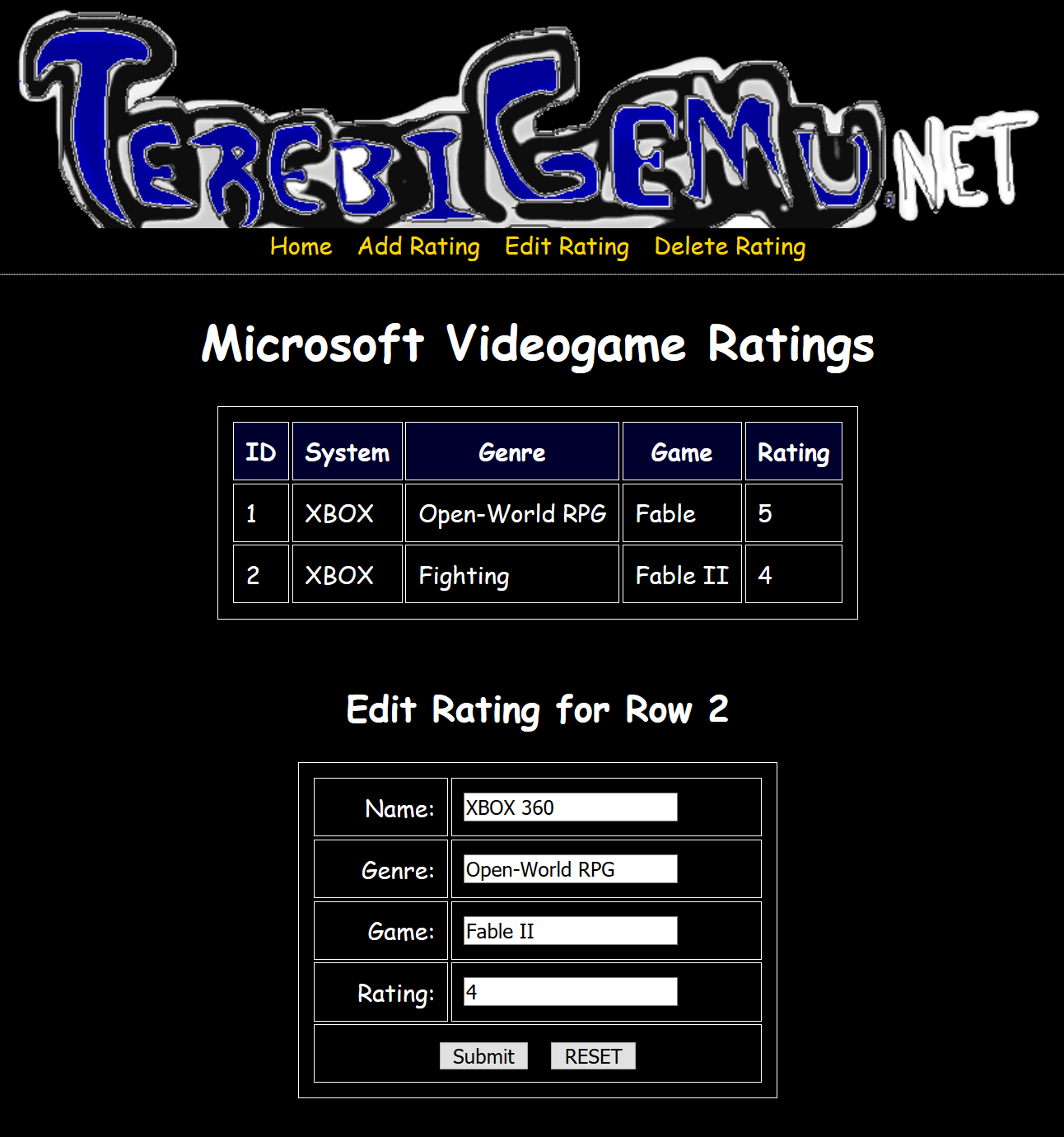
**TGN Home.php**  
  
This page serves as the index.php of this web application. This page displays the current data from all the tables using a SELECT statement, one per table. On this page, the user can click on any of links to other functions of the application (Add Rating, Edit Rating, or Delete Rating). By clicking on these links, the user will be taken to a page that displays information for each of the tables (Nintendo, Sony, and Microsoft) and they will be able to interact with the data they see based on the page they just clicked. For instance, if they clicked Add Rating from the homepage, they would be able to insert data into the database. These links to other functions of the website are within a header file, header.php, which all parts of the application use.

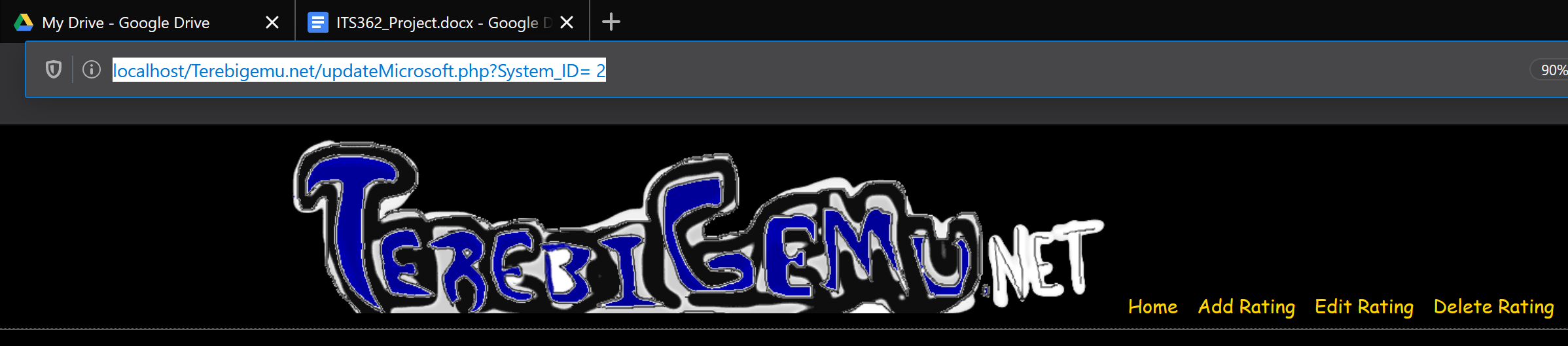
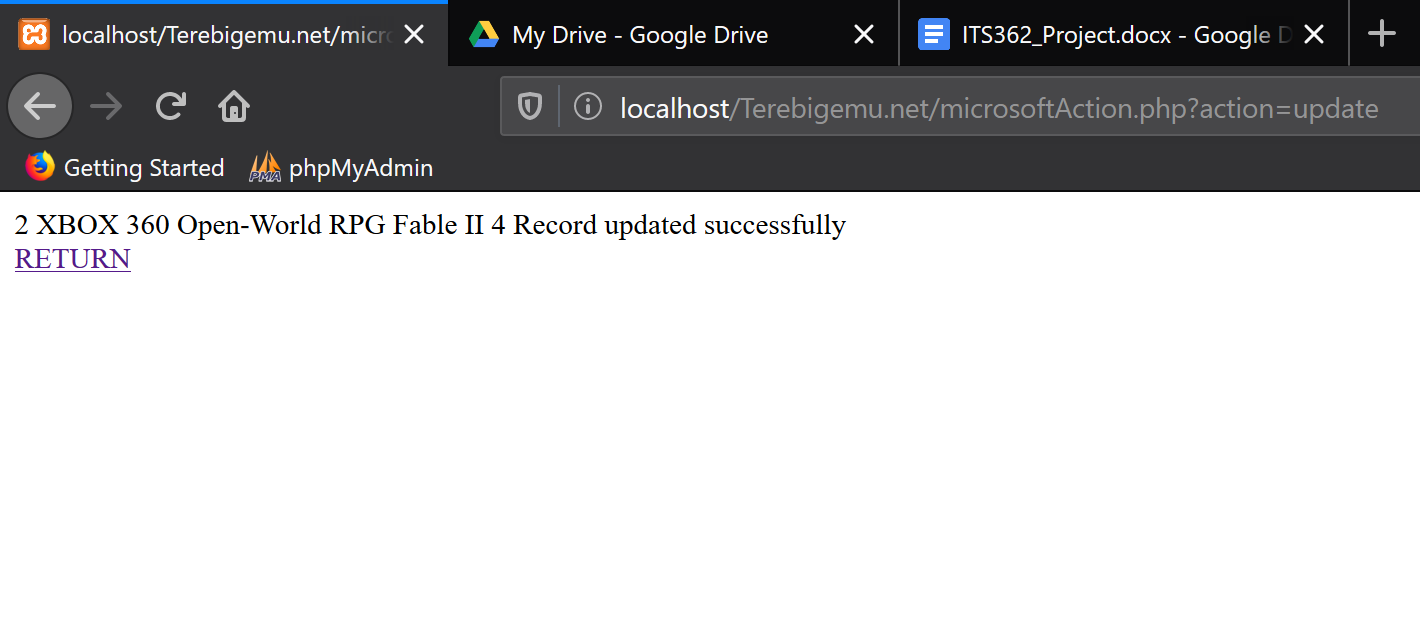
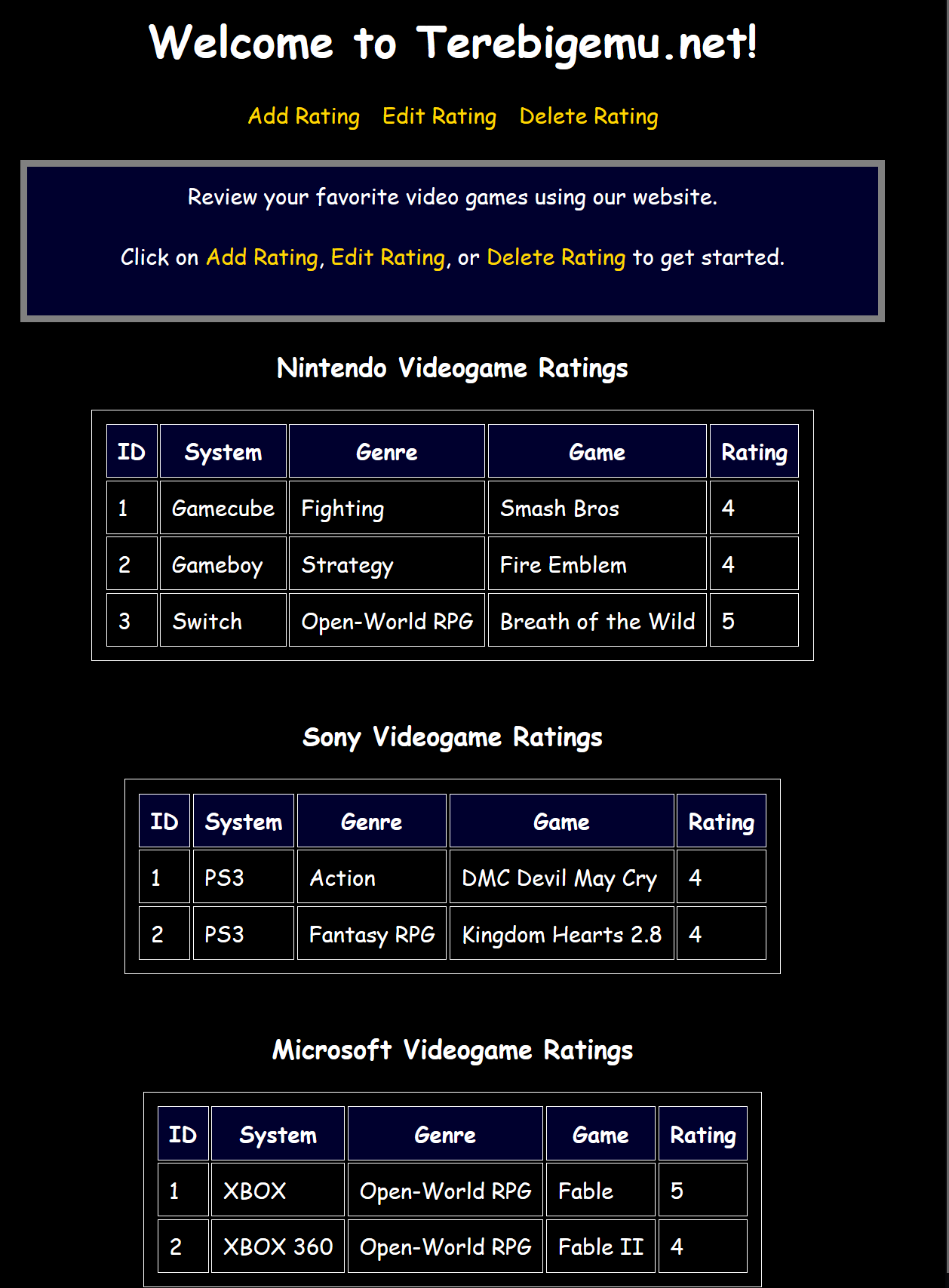
  
TGN Home.php

**addRatings.php**addRatings.php is a page that allows the user to add a single record to any of the 3 tables. This webpage makes use of the tabs UI from jQuery UI.   
  
  
addRatings.php (Nintendo tab)  
  
  
Insert\_Nintendo called and executed successfully   
  
  
Redirected to TGN Home.php to view results of Insert\_Nintendo stored procedure

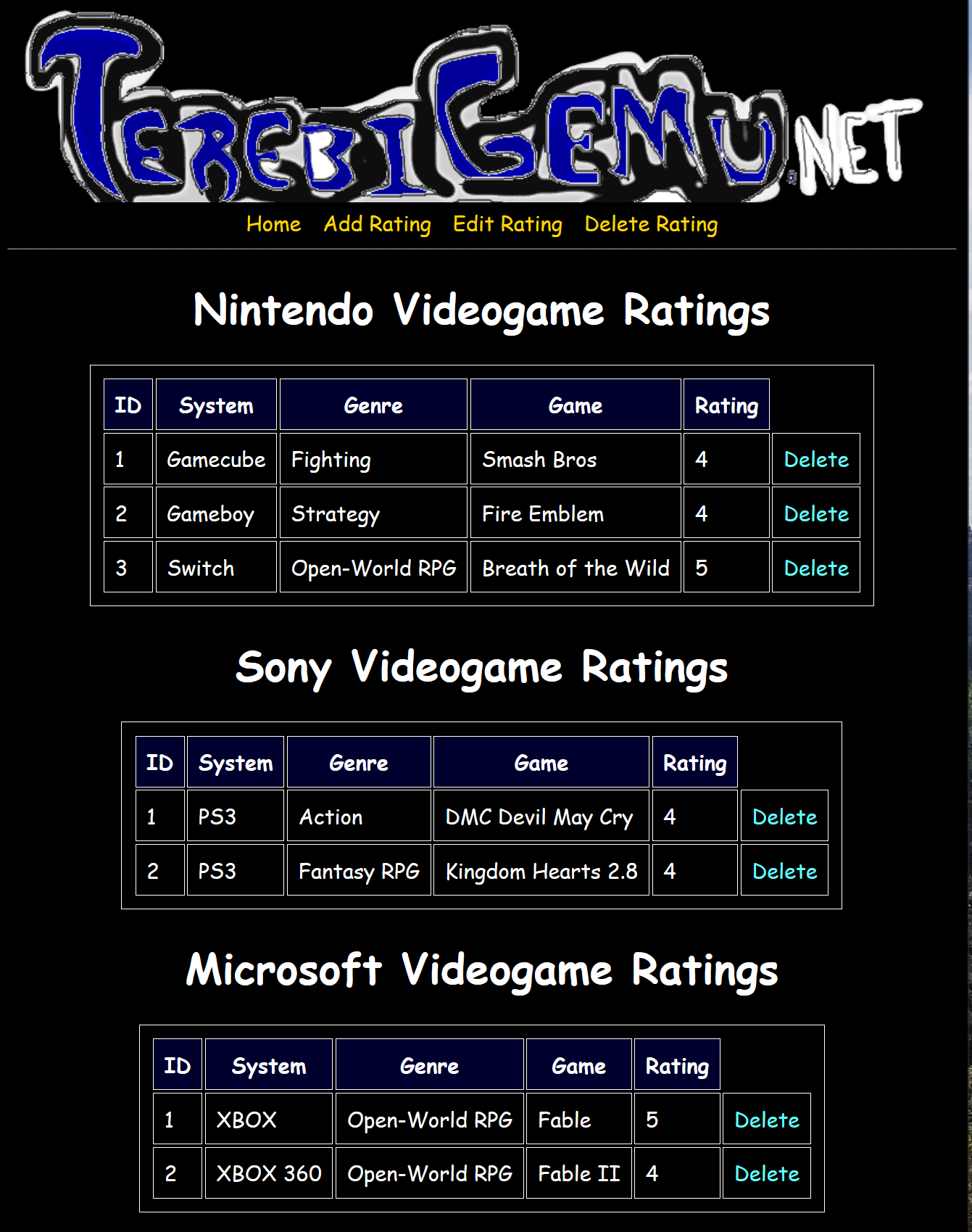
**editRatings.php**editRatings.php is a page that displays the current data in all the tables and allows the user to click an Edit link for the row they would like to edit. The user can only edit one row per table at a time. When the user clicks Edit for any given row of a table, they will be redirected to that table’s update page. For example, if we want to edit the 2nd row of the Microsoft table, we’d be taken to updateMicrosoft.php  
  


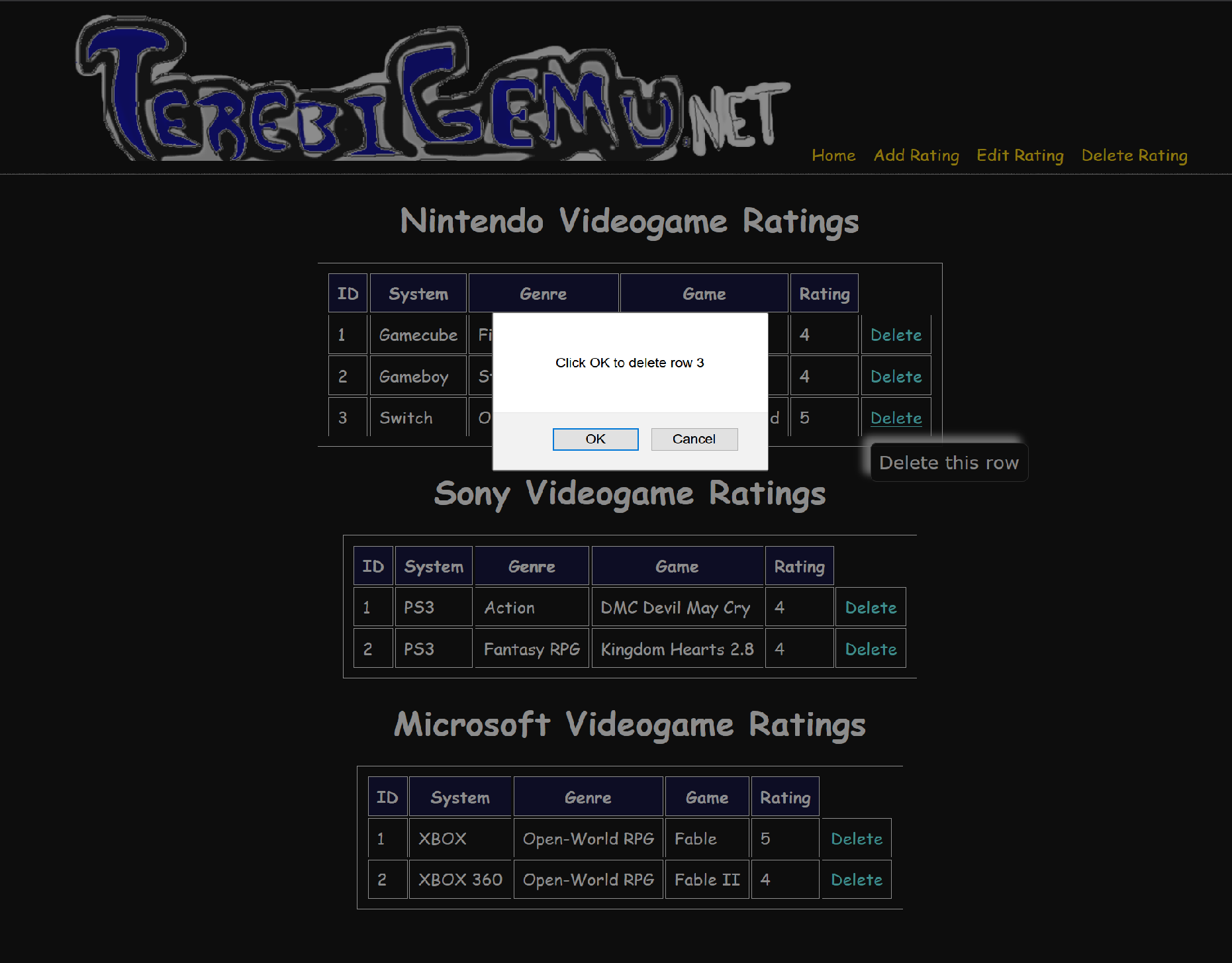
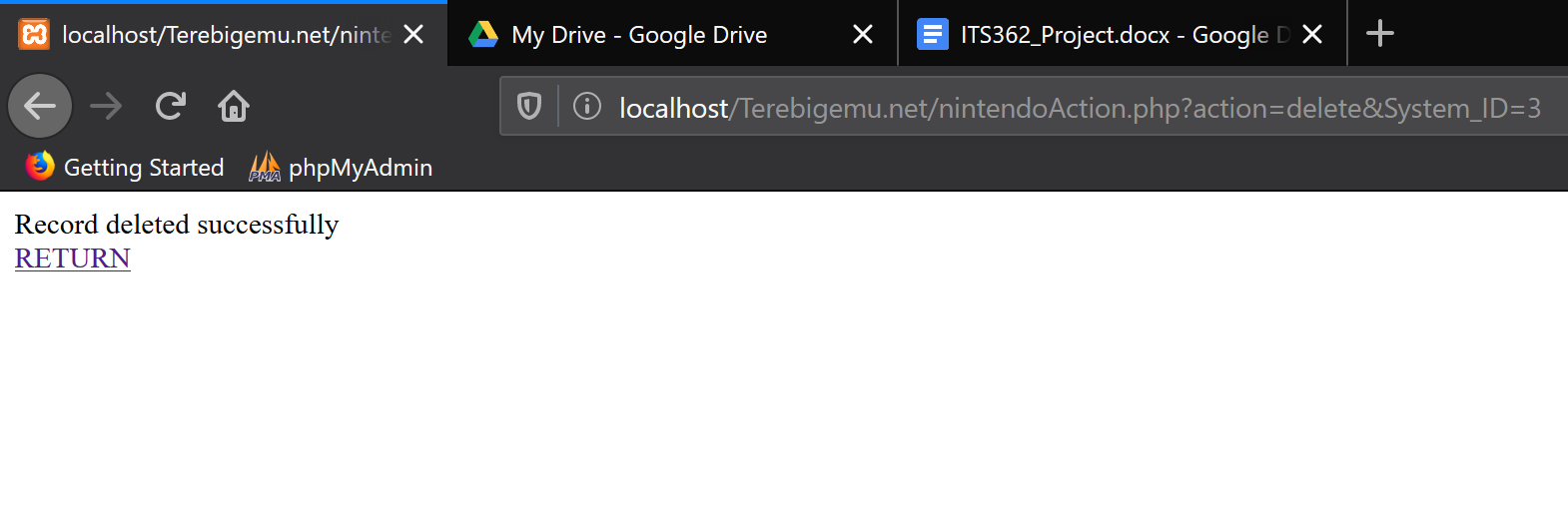
**updateMicrosoft.php**updateMicrosoft.php is the Microsoft table’s update page. Each table has its own update page since you cannot update more than one table at a time using this web application. There is a way to join tables together, however, that is beyond my level of experience and comfort for SQL. All of the other update pages look like this and they function the same way.



  
updateMicrosoft.php has a GET request for the System\_ID of the row the user clicked   
  
It should be noted that all pages that are updating or deleting a certain row must use the ID from the database for the database to change the correct row. To do that, there is a GET request for the System\_ID. You can see the result of those GET requests in the address bar.   
  
  
Update\_Microsoft stored procedure is called and executed successfully   
  
  
Redirected to TGN Home.php after the record was successfully and accurately updated

**deleteRatings.php**

deleteRatings.php is a page where the user can click on any of the Delete links listed on any of the tables to delete that specific row for that specific table. The user cannot erase more than one record at a time.   
  


  
deleteNintendoByID prompts user to confirm the deletion of row 3 (You can also see the tool-tip working here)   
  
  
Delete\_Nintendo stored procedure called and executed successfully   
  
  
Redirected to TGN Home.php after successful deletion of row 3 of the Nintendo table