Jovan Golding

12/11/2024

Prof. Ramos

CIS 344

Final Project Report

This documentation exists to give a written record of the Restaurant Portal project and how it works. This database contains a home, index, customers, view preferences, add preferences, view reservations, and add reservations files. In addition to these files we were provided with a restaurant server and database file with which to join our server to phpMyAdmin and place any functions we may have created. The server is very bare bones in content as I wanted to keep things strictly geared toward the goal of the server, which is managing customers along with their reservations and preferences. In addition to the php files there is also a MySQL server that is made through the use of MySQL Workbench and phpMyAdmin. The MySQL part of the server is arguably the most important aspect as without a database there would be no way to track or store customer related information.

The first thing that was done was the creation of the MySQL database using the tables given. After creating the database it was then named “restaurant_reservations” and was saved locally while phpMyAdmin was then prepared. To fully prepare phpMyAdmin to become a database another database also named “restaurant_reservations” must be made on the server, and then following that the user must import the sql file they made in MySQL Workbench to finish the setup process. Each table was created to the exact specifications listed in the requirements for the project with Customers(CustomerID), Reservation(ReservationID), and

DiningPreferences(PreferenceID) each being the Primary Key of their respective tables with Customers(CustomerID) being a Foreign Key in the Reservation and DiningPreference tables.

I will now talk about the php I used in my project. Starting with “addReservation” this code allows the user to add a reservation for a customer, it comes with options asking for the customer name, reservation time, number of guests, and any special requests. I also added code to bring up a dropdown menu when assigning a reservation to a name entered into the system. Multiple reservations can be made for the same customer, they will be assigned new reservation ID but will keep their same customer ID allowing for repeat visits by customers. The HTML code is what gives the webpage its look and the form with which users input the relevant data.

I also decided to put the ability to add customers on the same page as reservations as that way any new or returning customer can quickly add themselves into the system and then navigate down the page to the reservation section. I used a multiple external css links in order to make the page look slightly better, one being bootstrap.com which provides an amazing collection of basic css, and the other is css that I made myself for fully fleshing out parts I thought looked too plain. The “customers” files displays the Customer ID, Customer Name, and Contact Info for every customer added into the system. In addition to displaying customer information, there is also a delete button that will purge the customer and their data from the system. Instead of creating my own special request file I simply used the one given in the initial files and created a function in the “RestaurantDatabase” file that will post the request to the server.

The “viewPreferences” and “addPreferences” files where the last two files I made as I had trouble deciding where and how I would implement them. I decided on making them each have their own page and kept them separate from the other files in case a customer had no

preferences. I added a dropdown menu to the “addPreferences” file in order to choose a name that was previously added to the system, after choosing you can then assign a favorite table and dietary restrictions for that specific customer. At the bottom of the page if you click the blue button that says “Save Preferences” it will then take you to the “viewPreferences” page automatically, this was added to increase functionality and provide a more fluid experience. The “viewPreferences” file is kept simpler by just displaying all the data from “addPreferences” on a table, I then added a delete button so as to clear up any preferences the customer might not want on a later visit.

I experienced challenges in making some of the queries work as I had forgotten that the column names were case-sensitive and that gave me “undefined array key” errors all over my tables. It took me hours of research and double-checking to find out that the solution only required me to pay more attention. Overall the project was not so much of a nightmare as I had anticipated and I am glad to have done it as my understanding of PHP and MySQL has grown because of it.