Smiller Espinal, CIS 345, Final Project

Restaurant Portal Documentation

The Restaurant Portal is a web application I created to make managing restaurant reservations simple and efficient. Using PHP for server-side scripting, MySQL for storing data, and HTML for the interface, the goal was to build something practical for handling customer reservations. This project focuses on making it easy to add, view, and manage reservations, while keeping everything organized and user-friendly.

The platform is made up of several pages that all work together to get the job done. It starts with the homepage, home.php, which acts like the control center. From here, users can choose to add a new reservation or view the existing ones. The design is straightforward and makes it easy to find what you need. I prioritized functionality over flashy designs to ensure users could navigate easily.

To add reservations, users go to addreservation.php. This page has a form where they can fill in all the details, like the customer's name, phone number, reservation date and time, how many guests are coming, and any dining preferences. Once they submit the form, the data gets sent to processreservation.php, where the inputs are validated and then saved into the database. This ensures smooth operation and prevents any invalid data from being entered.

If users want to see all the reservations, they head over to viewreservation.php. This page retrieves data from the database and displays it in a table format. Each reservation includes details like the customer's name, the date and time of the reservation, the number of guests, and any special requests. If a reservation needs to be canceled, the deletereservation.php page handles this process seamlessly.

The most important part of this project lies in the backend code, specifically in the RestaurantDatabase.php file. This file serves as the connection hub between the web pages and the MySQL database. It includes functions for adding, retrieving, and deleting records, making the system more organized and modular. Keeping all database operations centralized in one file simplifies debugging and future updates.

The database itself, named restaurant_reservations, consists of three tables: customers, reservations, and diningpreferences. The customers table stores essential information like names and phone numbers. The reservations table tracks details like reservation times, the number of guests, and their table preferences. Finally, the diningpreferences table is set up for future use, with plans to expand its functionality to enhance the customer experience.

One of the biggest challenges I faced during this project was working with phpMyAdmin and establishing a proper connection between the PHP code and the MySQL database. Initially, I struggled with configuring the database settings and ensuring the program could correctly insert and retrieve data. Debugging the connection issues required a lot of trial and error, as well as researching error messages and troubleshooting steps. Eventually, I was able to get everything working by carefully checking my connection parameters in the RestaurantDatabase.php file and ensuring that my table structures matched the program's requirements.

Another challenge was ensuring that the reservation system worked dynamically. Passing reservation IDs between pages, especially for the delete functionality, required using PHP’s GET method, which I had to learn and implement correctly. Additionally, setting up foreign keys between tables in phpMyAdmin presented some difficulties, as I had to ensure proper data types and relationships. These challenges helped me better understand how PHP and MySQL interact in a real-world application.

To test and use the Restaurant Portal locally, the project files must be copied to the htdocs folder in a XAMPP installation. After that, the database schema can be imported using phpMyAdmin to set up the necessary tables. By starting Apache and MySQL in XAMPP, the portal becomes accessible through http://localhost/restaurant_portal/home.php.

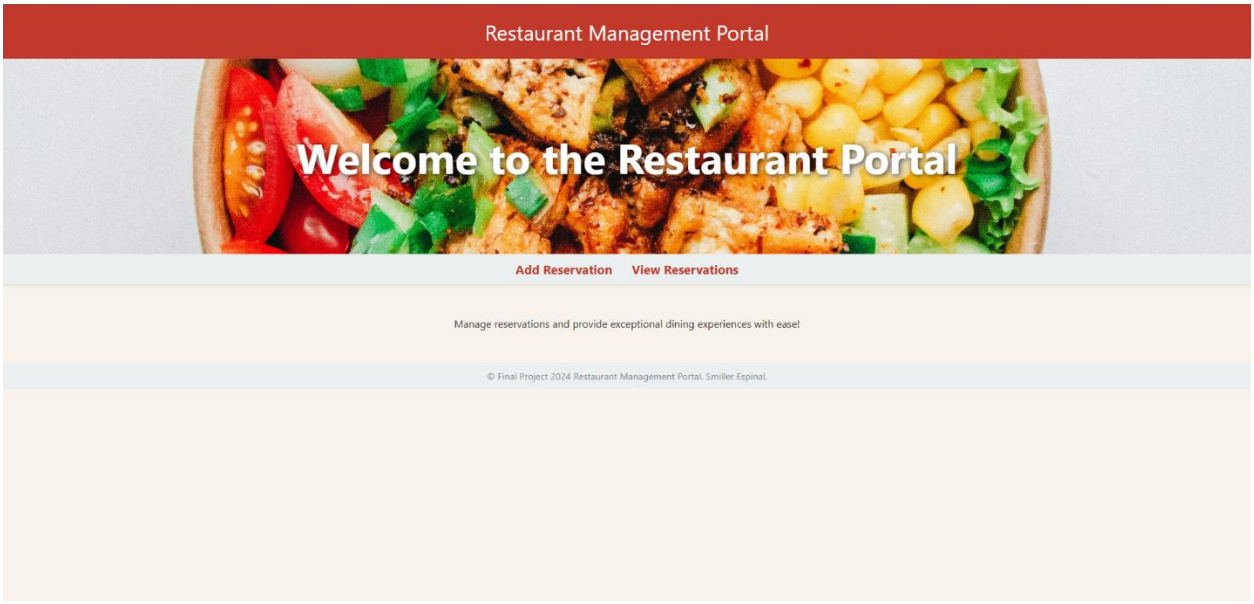
The technologies used in this project include PHP for backend scripting, MySQL for data storage, and HTML for structuring the web pages. While CSS is used sparingly for basic styling, the design is clean and functional. XAMPP served as my local development and testing environment, allowing me to make changes and see results in real-time.

Despite its simplicity, the Restaurant Portal includes features that make it practical. Users can add new reservations through a straightforward form, view all existing reservations in a clean table, and delete reservations when needed. These features ensure that the system is user-friendly and fulfills its purpose.

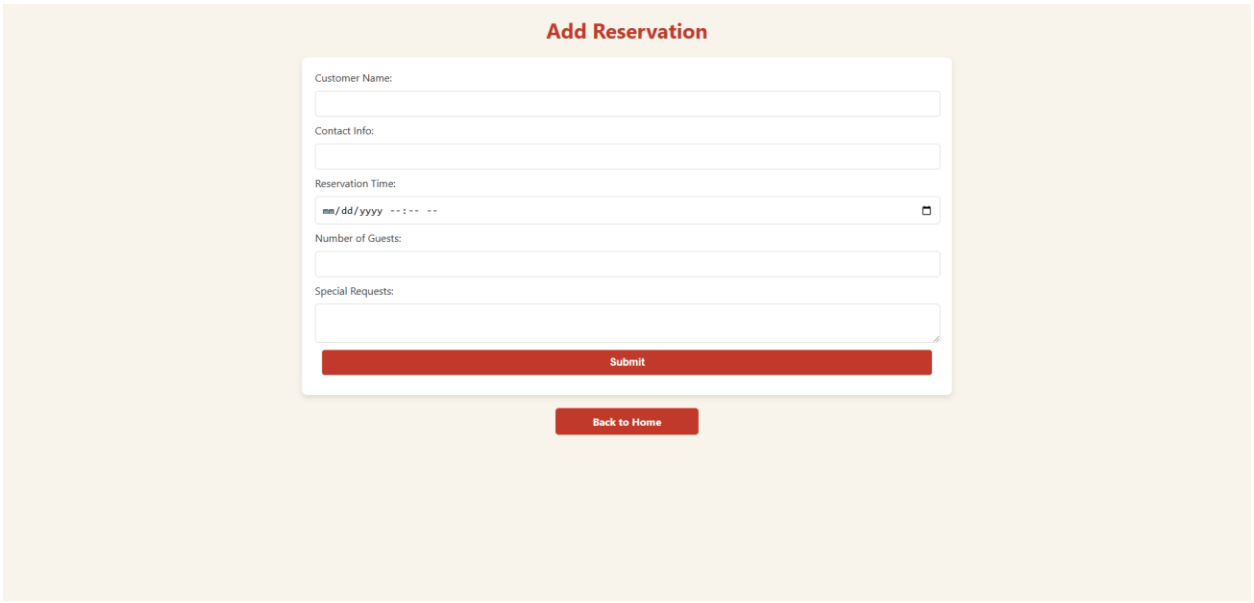
That said, there's still room for improvement. For example, the diningpreferences table hasn't been fully utilized yet, but it's ready for expansion to include more personalized customer options. The design could also benefit from more advanced CSS to make the interface visually appealing. These enhancements are goals I plan to work on if I revisit this project in the future.

Overall, this project was a valuable learning experience. It helped me understand the intricacies of PHP and MySQL and how they can be used to build functional web applications. Although I faced challenges with phpMyAdmin and database connections, I was able to overcome them and create a working system that solves a real-world problem. This project represents a significant step forward in my web development journey, and I'm proud of what I accomplished.

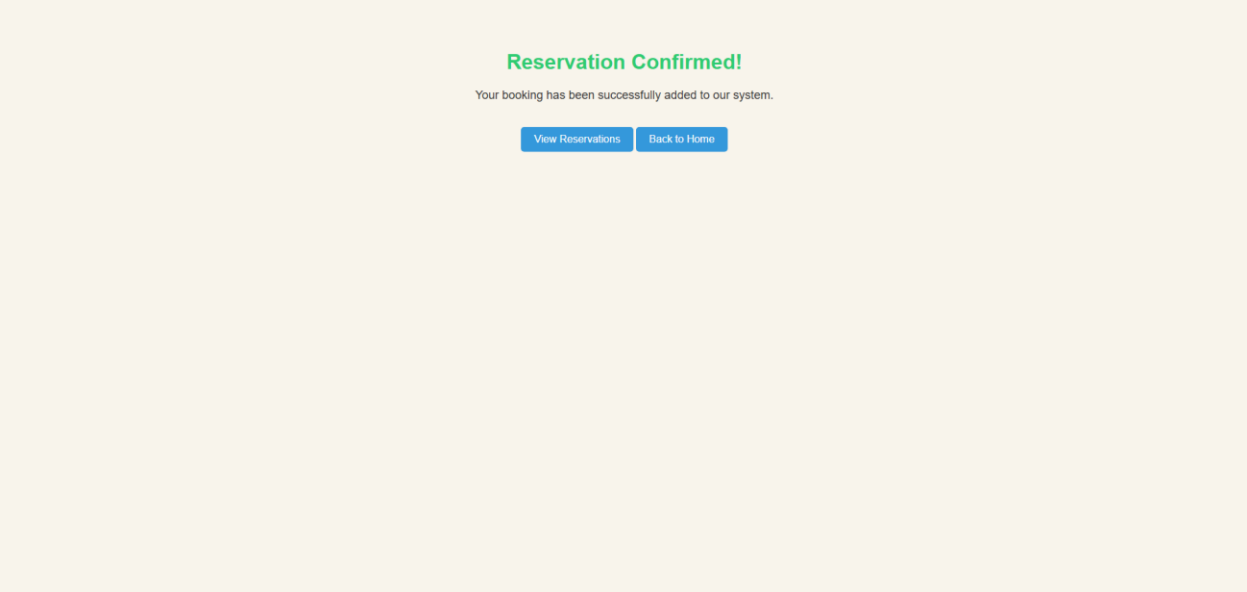
Main Page:



Add Reservation:



Reservation Confirmation:



View Reservations: