CSCI 322 - Client/Server Project - Fall 2022
The Tiny Concert/Dinner/Movie Series

Project Objectives

- Demonstrate proficiency with a server-side scripting language (PHP)
- Demonstrate proficiency with a client-side scripting language (JavaScript)
- Demonstrate appropriate data validation choices (server-side vs. client-side, and if client-side, when?)

Project Description

Imagine that you have been hired to maintain a reservation system for a small company that hosts concerts, dinners or movies (CHOOSE ONE). The company has one small hall that hosts one event daily. The hall has 24 seats, arranged in 4 groups of 6: Tables A, B, C, and D, each with seats numbered 1, 2, 3, 4, 5, and 6. You will build a small reservation system for the events scheduled during a one-week (seven day) schedule.

Your program must provide the following:

- The ability to view events and seat availability as guest.
- The ability for someone to join the series as a user. They must supply their last name, their first name, their email address, and a password.
- A visualization of the seats in the hall, with an indication of which seats are reserved and which are available. A user should never see the name of who is sitting in a reserved seat except perhaps their own seat.
- The ability to see the events and their dates.
- The ability to make a reservation by logged-in users. Users should be able to choose the date of the event and their seat(s) but should not be allowed to choose a seat already reserved.
- The ability to check, alter, or cancel a reservation. Users should be able to change seats for or cancel a reservation they made previously. Make certain that the user cancelling the reservation is the user who made the reservation.
- The ability for a user to view all of their pending reservations.
- Maintain JSON files that contain the following information: (I WILL GET YOU SAMPLES AND CODE THAT READS AND WRITES!)
    - Users (first name, last name, email, password)
    - Events (title, date)
    - Per each event, event title and seating
- You must validate the data that you gather from the user. A user should never be allowed to claim a seat that is already reserved. A user should never cancel a reservation they did not make. Remember, that when a user brings forth the information for a particular event, that another user somewhere may have brought forth the same information. The server is responsible for making certain that reservation information is accurate.

Project Submission

When completed, your project should be located in a subfolder of your public_html folder. When you need to ask me questions about your code, it would be best if you shared the path to this folder with me. This will enable me to help you when needed.

Additionally, each file you create for this project should be in that folder or a subfolder. And, each file you create (other than data files) for this project must have your name near the top of the file.

There will be project demonstrations shortly before the project is due. The time table on this will come out shortly. You will then have a two or three days to complete your project.

A word-processed document explaining the use of your program, the format of data files, your choices for controls and methods of validation is also required. This document should also contain a graph-like schematic of your project describing the flow of data. More details upcoming.

Please be aware of honor code implications. This is your project, your design choices, your interface, your code. Helping each other solve problems is encouraged, but sharing code and/or design is NOT acceptable. Any code taken from an online source or from a text or from lab should be documented as such. Help from another on a particular method/task should be documented as such. The majority of code written for this project must be your own.

Start early and this can be an enjoyable experience! This is not a 2 or 3 day project .. if it was, I would make it due in one week. I am happy to help. Start small and grow the project. Work on one task at a time, making steady progress. Expect to make changes as you go. Look-and-feel, ease-of-use, and functionality are all important aspects to consider.