

Gnouma Sissoko

CIS 344

Project 2

Documentation

The backend of the Gym Membership and Class Booking System is primarily built using PHP, with MySQL for database interactions. This section outlines the core components and functionalities of the PHP backend, including the security features implemented to ensure the system is robust, secure, and user-friendly. Each major part of the application—login, registration, class booking, and payment processing—has its own PHP script that interacts with the database and ensures seamless communication between the front end and back end. Our gym, NGF GYM, has a backend system that was built to give members a safe, user-friendly experience while also managing critical processes like registration, login, class bookings, and payments. Below, I will discuss the significant steps and processes implemented in the backend code, focusing on the logic and functionality behind the PHP and HTML.

When a user registers on the NGF GYM website, their personal information is collected using an HTML form that contains their first and last name, email address, phone number, membership type, and password. Following submission, these details are checked and processed on the server side. The system enters the information into the database to generate a new member record. The user's password is very crucial since it must be safely preserved. Initially, the password was kept in plaintext in my code; however, in a more secure implementation, I would hash the password using industry-standard methods (e.g., `password_hash()`) before storing it to the database to safeguard the user's credentials.

A MySQL database was used to connect with the backend database. The code begins by connecting to the database using the MySQL extension. The connection parameters, including the database host, username, password, and database name, are specified. Once the connection is established, you may run any queries to interact with the data. If the connection fails, the system suspends further execution and produces an error, which is critical for debugging and verifying that the program functions properly. Without a legitimate database connection, no data can be processed or accessed, therefore establishing a successful connection is the first crucial step in the backend.

The session management was used in the login system to keep track of which users were logged in. Upon login, a unique session identification is produced, and user-specific information (such as the user ID) is saved in the session. This lets users to access restricted sites, such as their booking or account page, without having to log in several times. If a user attempts to visit a restricted page without a valid session, they will be promptly routed to the login page. This prevents unwanted access to sensitive data and allows only authenticated users to interact with the system's essential functionalities.

The booking system enables customers to secure spaces in gym courses. When a member picks a class, the system queries the database to determine the number of available spaces. The available slots are determined by subtracting the number of current bookings from the maximum class capacity. If there are available places, the user's reservation is processed, and their information is stored in the database's bookings table. The system guarantees that class capacity is not exceeded. The user is told that the booking was successful, and a booking confirmation is presented on the screen.

When a user successfully books a class, the backend system determines the pricing depending on their membership category (premium or normal). The price is maintained in the database's payments table, along with the payment method (for example, online payment). This guarantees that each class booking is linked to a payment, allowing the gym to manage all revenue from member subscriptions and class bookings. If a user's payment is properly completed, a record of the transaction is kept, serving as an audit trail for both the gym and the individual.

In addition to maintaining payment information in the database, the system records all transactions in a text file. This log contains essential data such as the member's ID, class name, membership type, and payment amount. This log is kept on the server and may be used to audit, troubleshoot, and track previous information. By storing transaction information in a text file, the gym adds an extra degree of backup and accountability to the database.

Throughout the program, good error handling is critical to creating a pleasant user experience. When a user meets a problem, such as attempting to book a full class or entering incorrect login credentials, the system displays clear, useful error messages. For example, if a class is completely booked, the user will get a notification stating that no spaces are available and may be encouraged to choose an alternative class or time. Similarly, if there is a problem with a payment or the user enters incorrect login information, appropriate feedback is offered to help them through the process of solving the issue.

To guarantee a smooth experience, the system manages user redirection following crucial actions such as logging in, registering, booking a class, or making a payment. For example, after a successful registration, the user is led to the login page, and after a successful booking, they are provided a confirmation page including their booking information. This contributes to a

structured flow throughout the website. Redirection guides users to the next logical step in navigating the system, preventing them from getting stuck or confused within the application.

In summary, the NGF GYM backend includes numerous critical features for managing membership, bookings, payments, and user interaction. The usage of a database to record member and transaction information, together with session management for secure user identification, forms a solid basis for the system. The class booking and payment systems allow users to reserve spaces in classes and pay for memberships, while transaction logs add another degree of transparency. Error management and user feedback ensure a pleasant user experience, while redirection helps users navigate the site. This software efficiently meets the essential backend demands of a gym management system while focusing on security, efficiency, and user satisfaction.