Smiller Espinal & Joshua Ford Group Project

Project Documentation for Rental Equipment Booking System

The Rental Equipment Booking System is a web-based application designed to allow users to easily browse, rent, and manage various types of equipment, such as cameras, laptops, sporting gear, and tools. The primary objective of this system is to provide a user-friendly platform for renting equipment, while ensuring that the system operates efficiently and securely. This system includes both customer and admin functionalities. Customers can browse available equipment, rent items, and manage their orders, while admins can oversee the rentals, manage equipment statuses, and view rental histories.

The initial design of the system aimed to create an intuitive user interface with clear categorization of available equipment, seamless rental processing, and a secure login and rental experience. The website's structure includes sections for browsing available equipment, making rental requests, and providing customer and admin access. The project utilizes HTML, CSS, and PHP for the front-end and back-end development, with MySQL serving as the database management system. The design was based on simplicity, ensuring users can navigate the system without confusion while keeping the data interactions efficient.

The development process began with setting up the MySQL database, which included multiple tables: Users, Equipment, Rentals, Transactions, and Maintenance. The database structure was designed to ensure the seamless relationship between these tables, using foreign key constraints to link rentals with users and equipment. The system queries the Equipment table to display available items for rent and updates the Rentals table when a user rents an item. Additionally, the Transactions table tracks payments, and the Maintenance table keeps records of equipment maintenance. Queries were optimized to ensure fast and efficient retrieval of data, especially when dealing with large datasets of available equipment.

PHP was used to handle the back-end logic, such as user authentication, rental processing, and session management. A key feature of the system is its dynamic rental process, where users can select available equipment and fill in a form with their details to confirm the rental. Each piece of equipment is displayed dynamically by querying the database, which allows the system to automatically update based on the current inventory and availability. Admin users have additional functionality, such as managing the status of equipment and viewing rental transactions, which provides them with greater control over the system.

Security was a top priority throughout the development of the system. The application employs session management techniques to ensure that only authenticated users can rent equipment and make transactions. Password security is implemented by hashing passwords using PHP's password_hash() function and verifying them with password_verify() to prevent unauthorized access. Cross-Site Request Forgery (CSRF) protection was added to every form on the website to prevent malicious requests from unauthorized sources. Additionally, input sanitization is used to prevent Cross-Site Scripting (XSS) attacks, and prepared statements ensure that the system is protected from SQL injection vulnerabilities. These security features are designed to protect both user data and the integrity of the system.

One of the major challenges faced during the development was implementing a robust CSRF protection system. This required creating and validating CSRF tokens for every form submission, ensuring that all requests to rent equipment were legitimate and not forged by third-party attackers. Another challenge was ensuring that the password storage system was secure, as it required a detailed understanding of cryptographic hashing algorithms. Ensuring that the system handled multiple transactions concurrently without database inconsistencies was another hurdle, which required careful management of the Rentals and Equipment tables through atomic transactions. Additionally, making sure that the database queries were optimized to handle large volumes of equipment data and rental requests was a significant challenge.

The project also faced difficulties in integrating the database with the website, especially in ensuring that equipment could be dynamically added and removed from the inventory as items were rented or returned. This required constant testing to ensure that updates to the equipment status were properly reflected across all areas of the system, particularly when items were marked as "rented" or "under maintenance." The inclusion of transaction management added an additional layer of complexity, as each rental involved both updating the rental records and processing payments, which required a seamless connection between the system's front end and back end.

The Rental Equipment Booking System offers a secure and efficient platform for renting equipment, with a focus on user-friendly features and strong security practices. The system includes a dynamic rental process, admin dashboard, and transaction management, all while ensuring that data is handled securely. By implementing industry-standard security features, the application minimizes the risk of unauthorized access and ensures that sensitive information such as user credentials and transaction details is protected. The challenges faced during development, particularly with security and database integration, provided valuable insights into building secure and scalable web applications.

The development of the Rental Equipment Booking System was a collaborative effort between two group members, Smiller Espinal and Joshua Ford. Smiller Espinal created the main page and everything PHP related while Joshua Ford created the SQL database for the program.

However, working within a group presented challenges, particularly with communication. With only two out of the three group members actively participating in the project, it became difficult to maintain consistent communication and coordination. This impacted the development process, as tasks had to be distributed between the two active members, leading to a higher workload and occasional delays in completing features. Despite these challenges, the two active members managed to successfully develop and implement the core functionalities of the system, including database integration, equipment rental management, and security measures. The experience highlighted the importance of clear and consistent communication in group projects, as well as the value of teamwork when handling complex tasks.

Database Schema and Data Management for Rental Equipment Booking System

The Rental Equipment Booking System is designed to streamline the management of equipment rentals, track transactions, and ensure proper equipment maintenance. This database schema comprises several integrated tables, including Users, Equipment, Rentals, Transactions, and Maintenance, each serving a specific purpose in the system's functionality. Additionally, triggers are employed to enforce business rules, maintaining data consistency and integrity.

The database consists of five primary tables: Users, Equipment, Rentals, Transactions, and Maintenance. The Users table stores user details such as name, email, role, and contact information. It differentiates users into 'Customer' and 'Admin' roles, facilitating role-based functionalities. The Equipment table manages details of rentable items, including their name, description, category, rental rates, and availability status. The Rentals table records rental transactions, associating users with equipment, tracking rental dates, and managing statuses like 'Pending,' 'Completed,' or 'Canceled.' The Transactions table captures payment details for rentals, specifying amounts and payment methods. Finally, the Maintenance table tracks maintenance activities for equipment, ensuring they remain in optimal condition.

Initial data for the database was populated using SQL INSERT statements, simulating real-world entries. The Users table includes a variety of sample records, such as customers and administrators, each with unique roles and attributes. The Equipment table features items like cameras, drills, and sports gear, each categorized and assigned rental rates. Rentals were created to simulate customer transactions, demonstrating the flow of equipment being rented, returned, and processed for payments. Maintenance records illustrate the upkeep of equipment, adding realism to the dataset.

Joins play a crucial role in retrieving meaningful insights from the database by combining data from multiple tables. For instance, INNER JOINs can link the Rentals and Equipment tables to identify which equipment items have been rented by specific users. Similarly, LEFT JOINs could be used to fetch all rental records, including those without associated transactions, to identify pending payments. RIGHT JOINs could highlight all equipment, showing which items have not been rented, helping administrators manage inventory effectively. These joins ensure comprehensive data analysis and informed decision-making.

Triggers enhance the database's functionality by automating critical updates and enforcing constraints. The BeforeRentalInsert trigger ensures that only available equipment can be rented, signaling an error if an attempt is made to rent unavailable items. Simultaneously, it marks equipment as 'Rented' when a rental is initiated. The AfterRentalUpdate trigger updates the availability status of equipment when rentals are completed or canceled, returning them to the 'Available' state. These triggers maintain data integrity and streamline operational workflows.

The Rental Equipment Booking System's database schema effectively models real-world rental operations, providing a robust foundation for managing users, equipment, transactions, and maintenance activities. By leveraging SQL joins and triggers, the system ensures efficient data retrieval and consistency, supporting seamless operation and informed decision-making.