

EQUIPMENT TRACKING SYSTEM

Lolitha Y – PES1UG22CS308

Manya Gaonkar- PES1UG22CS336

TABLE OF CONTENT:

Sl.NO	TITLE
1	INTRODUCTION
2	PROBLEM STATEMENT AND OBJECTIVES
3	SOFTWARE REQUIREMENT SPECIFICATIONS
4	DATABASE DESIGN
5	IMPLEMENTATION
6	RESULTS
7	CONCLUSION
8	TRIGGERS
9	JOIN

INTRODUCTION

BACKGROUND

With the growing need for equipment rental services in various sectors, there is a demand for an efficient system that manages rentals and inventories. Traditional methods of managing rentals often involve cumbersome manual processes, which can lead to inefficiencies and errors. This project aims to address these challenges through a digital solution.

PROJECT GOALS

- Develop a user-friendly website for equipment rentals.
- Implement an efficient back-end system for managing inventory and user data.
- Provide real-time tracking of equipment availability and rental statuses.

Scope of Work

The project encompasses:

- Front-end Development: Creating a responsive and interactive user interface.
- Back-end Development: Developing server-side logic and database management.

- Database Design: Structuring and managing data related to users, equipment, orders, and Inventory

PROBLEM STATEMENT AND OBJECTIVE

PROBLEM STATEMENT

Manual equipment rental processes are prone to errors, inefficiencies, and difficulties in tracking inventory. There is a need for a streamlined, automated system that can manage rentals and track inventory in real-time.

OBJECTIVES

- To create an intuitive platform for users to browse, rent, and manage equipment.
- To enable administrators to track and manage equipment inventory seamlessly
- To ensure data security and integrity through secure authentication and authorization mechanisms.

SOFTWARE REQUIREMENT SPECIFICATION

HARDWARE REQUIREMENTS

- Server: A server with sufficient processing power and memory to run PHP scripts and host a MySQL database. Example specifications include:
- CPU: Dual-core processor
- RAM: 4GB
- Storage: 20GB SSD
- Client: Standard web browser (e.g., Chrome, Firefox, Safari) on a computer or mobile device.

SOFTWARE REQUIREMENTS

- Server-side Technologies:
 - PHP 7.4 or higher for server-side scripting.
 - MySQL 5.7 or higher for database management.
- Client-side Technologies:
 - HTML5 for structuring web content.
 - CSS3 for styling and layout.
 - JavaScript for interactive elements and dynamic content.

FUNCTIONALS REQUIREMENTS

- User Registration: Allow users to create accounts with personal details.
- User Sign-in: Enable registered users to log in with email and password.

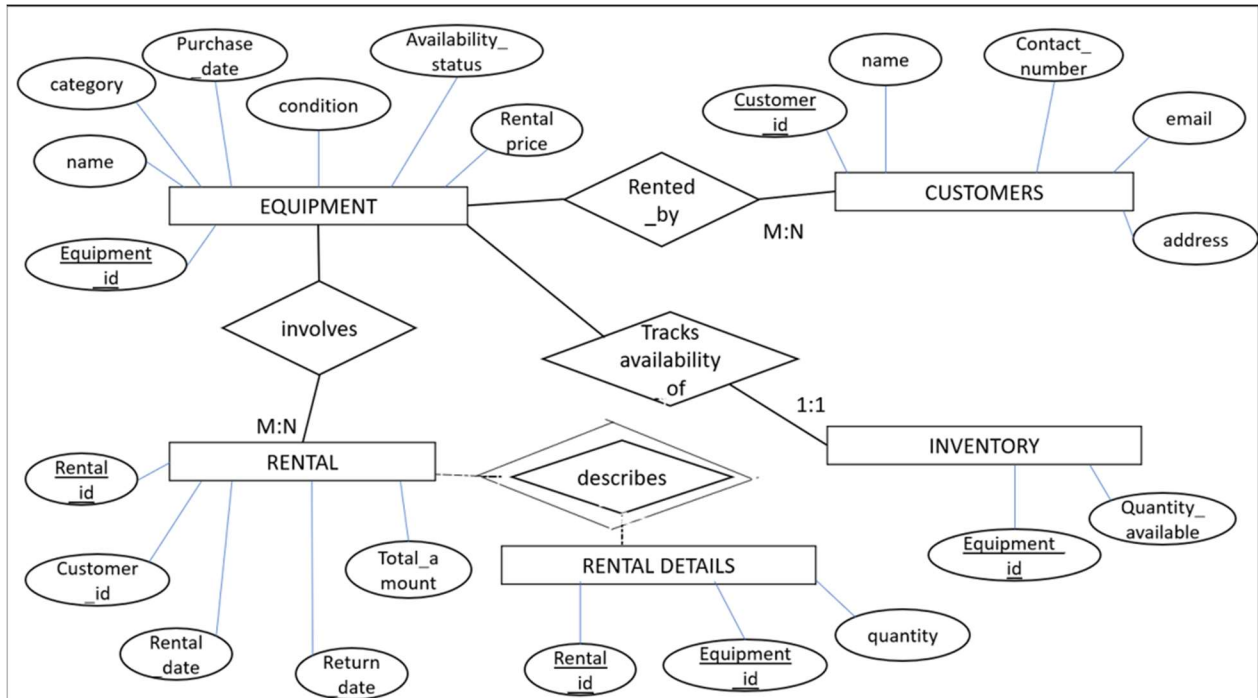
- Equipment Browsing: Display available equipment with details and images.
- Order Placement: Facilitate the process for users to place rental orders and view order history.

NON-FUNCTIONAL REQUIREMENTS

- Performance: The website should load within 3 seconds and handle up to 100 concurrent users.
- Security: Implement secure user authentication and data encryption.

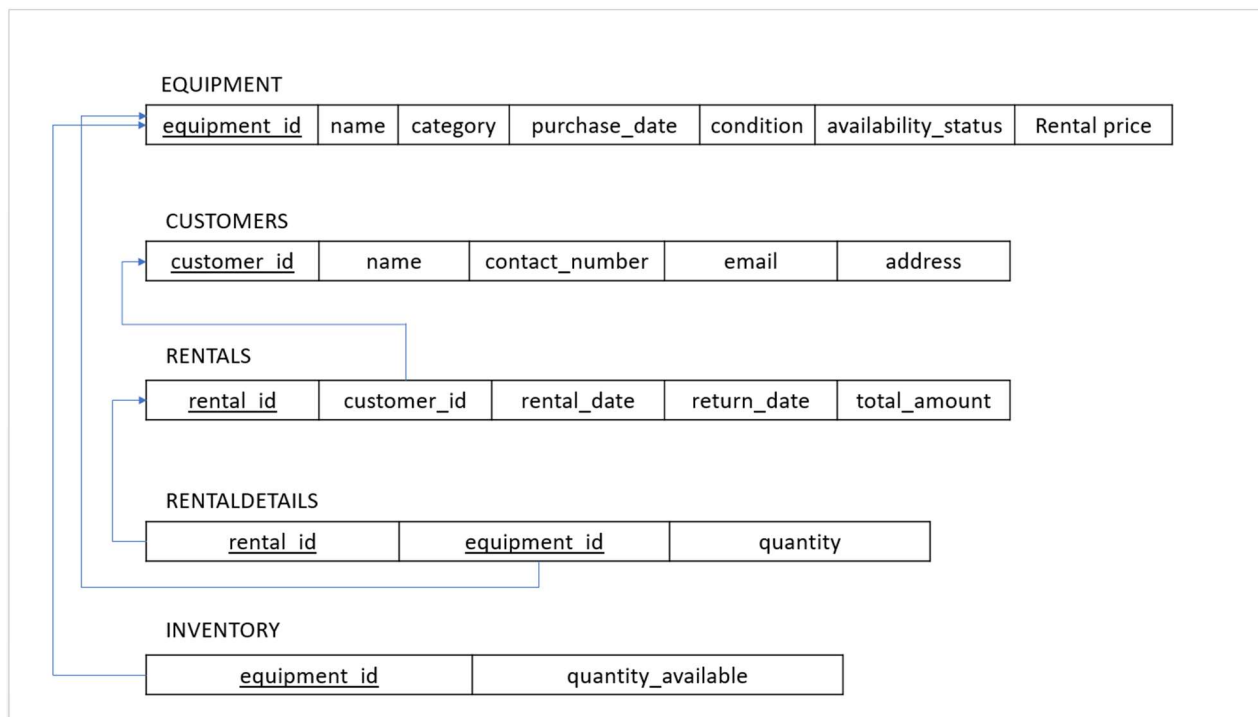
DATABASE DESIGN

ENTITY-RELATIONSHIP DIAGRAM



- User: Has a one-to-many relationship with Orders (one user can place multiple orders).
- Order: Contains a many-to-many relationship with Equipment (an order can include multiple items, and each equipment item can be part of multiple orders).

SCHEMEA DIAGRAM



1. EQUIPMENT

- Attributes:
 - o equipment_id: A unique identifier for each piece of equipment.
 - o name: The name of the equipment.
 - o category: The category to which the equipment belongs.
 - o purchase_date: The date when the equipment was purchased.
 - o condition: The current condition of the equipment.
 - o availability_status: Indicates whether the equipment is available for rent.
 - o rental_price: The price at which the equipment can be rented.

2. REGISTER

- Attributes:

- o firstname: The first name of the user. o lastname: The last name of the user. o gender: The gender of the user. o dob: Date of birth of the user. o email: Email address of the user. o password: Password for the user's account.
- o confirm_password: Confirmation of the password for account security.

3. ORDERS

- Attributes:

- o customer_name: The name of the customer placing the order. o contact_no: Contact number of the customer. o email: Email address associated with the order.
- o house_address: Address of the customer. o equipment_id: ID of the equipment being rented. o order_date: The date when the order was placed.

4. RESET_PASSWORD

- Attributes:

- o new_password: The new password that the user wants to set.
- o confirm_new_password: Confirmation of the new password.

5. SIGNIN

- Attributes:

o email: Email address used to sign in. o password: Password

associated with the email for authentication.

IMPLEMENTATION

FRONT-END DEVELOPMENT

- The front-end implementation involves creating the user interface using HTML, CSS enabling users to interact with the equipment rental system. It includes designing web pages for browsing equipment, user registration, login, and order placement, ensuring a responsive and user-friendly experience across devices.

- Technologies Used-

- HTML: For structuring web pages.

- CSS: For styling and layout.

- JavaScript: For interactive elements and dynamic content.

- Features-

- Responsive Design: Ensures usability on various devices.

- Interactive Elements: Includes forms, buttons, and dynamic content updates.

BACK-END DEVELOPMENT

- The back-end implementation involves setting up a server-side environment using PHP and MySQL to handle data processing, storage, and retrieval. It manages user authentication, equipment inventory, and order processing, ensuring secure and efficient communication between the front-end and the database.

- Technologies Used-

- PHP: For server-side scripting and handling user requests.
- MySQL: For managing database operations and queries.
- Server-side Logic-
 - User Authentication: Scripts for registration, login, and session management.
 - Order Processing: Scripts for handling order placement and updates.
- Database Integration-
 - Data Handling: PHP scripts interact with the MySQL database to manage user data, equipment details, and order information.

LINKING OF FRONTEND AND BACKEND

- In this project, the `config.php` file:
 - Connects to the Database: It stores the database credentials, enabling consistent and secure connections across the application.
 - Centralizes Configuration: By including `config.php` in various PHP scripts, it ensures consistent database access and simplifies updates, improving maintainability.
 - Ensure Consistency: By including config.php in files like register.php, signin.php, and order.php, the project ensures that all parts of the application use the same database connection parameters, avoiding errors and simplifying maintenance.

TESTING

- Testing Strategies-
 - Unit Testing: Testing individual components and functions.
 - Integration Testing: Ensuring different parts of the system work together.
- Test Cases-

- Registration Test: Verify that new users can register successfully.
- Order Test: Ensure that orders are processed and reflected in the database correctly.

RESULTS

HOME PAGE:



ABOUT PAGE:

A wide-angle photograph of a large industrial warehouse filled with tall metal shelving units stacked with cardboard boxes. The perspective is looking down a long aisle, with yellow safety lines painted on the floor. The image is used as a background for the top section of the website.

About EquipTrack

[Home](#) [About](#) [Services](#) [Sign In](#) [Register](#)

At EquipTrack, our mission is to simplify equipment rental and inventory tracking for businesses of all sizes. We strive to provide reliable, efficient, and user-friendly solutions to help you manage your assets effectively and focus on what you do best.

EquipTrack is dedicated to providing top-quality rental equipment and seamless inventory tracking solutions for various needs.

SERVICE PAGE:

Our Services

[Home](#)[About](#)[Services](#)[Sign In](#)[Register](#)

Equipment ID	Name	Category	Purchase Date	Condition	Availability Status	Rental Price	Order
1	Excavator	Construction	2022-01-10	Good	Available	50000.00	Order
2	Tents and canopies	Events	2022-06-11	Good	Available	2000.00	Order
3	Tables and Chairs	Events	2022-03-22	Good	Available	2500.00	Order
4	Projectors	Office	2023-04-15	Good	Available	5000.00	Order
5	Wifi Hotspots	Office	2024-01-01	Good	Available	1000.00	Order
6	Coffee Maker	Party	2023-02-07	Good	Available	700.00	Order
7	Barbeque and Grills	Party	2023-02-05	Good	Available	1500.00	Order
8	Dinnerware and Flatware	Party	2023-12-10	Good	Available	1550.00	Order
9	Sound Systems	Audio and Visual	2023-04-20	Good	Available	3000.00	Order

© 2024 EquipTrack. All rights reserved.

ORDER PAGE :

Order Equipment

[Home](#) [About](#) [Services](#) [Sign In](#) [Register](#)

Name:

Contact No:

Email:

House Address:

© 2024 EquipTrack. All rights reserved.

REGISTER PAGE:

Register

[Home](#) [About](#) [Services](#) [Sign In](#) [Register](#)

First Name:

Please fill out this field.

Last Name:

Gender:

Date of Birth:

Email:

© 2024 EquipTrack. All rights reserved.

SIGNIN PAGE:

Sign In

[Home](#)[About](#)[Services](#)[Sign In](#)[Register](#)

Email:

Password:

Sign In

Forgot Password?

© 2024 EquipTrack. All rights reserved.

RESET_PASSWORD:

Reset Password

[Home](#)[About](#)[Services](#)[Sign In](#)[Register](#)

New Password:

Confirm New Password:

Reset Password

© 2024 EquipTrack. All rights reserved.

CONCLUSION

SUMMARY

The Basic Equipment Rental Website successfully meets its objectives by providing a comprehensive platform for equipment rental. The integration of front-end and back-end technologies ensures a functional and user-friendly system.

LESSON LEARNED

- Development Challenges: Encountered issues related to integrating front-end and back-end components.
- Solutions: Implemented debugging and testing strategies to resolve issues.

CHALLENGES

- Data Security: Ensured secure handling of user information and transactions.
- System Performance: Optimized the system to handle expected traffic and load.

FUTURE SCOPE

- Proposed Enhancements-
 - Advanced Search Filters: Implementing advanced search options for users to find equipment more efficiently.
 - User Reviews and Ratings: Allowing users to rate and review equipment.
- Technology Upgrades-
 - Mobile App Development: Developing a mobile application to enhance accessibility.
 - Improved Performance: Upgrading server infrastructure to support higher loads.

EXPANSION OPPORTUNITIES

- Multi-language Support: Adding support for multiple languages to cater to a broader audience.
- Integration with Payment Gateways: Enabling online payment processing for rentals

TRIGGERS:

DELIMITER \$\$

CREATE TRIGGER log_invalid_email

BEFORE INSERT ON register

FOR EACH ROW

BEGIN

INSERT INTO log_table (log_message)

VALUES (CONCAT('Checking email: ', NEW.email));

IF EXISTS (SELECT 1 FROM register WHERE email = NEW.email) THEN

INSERT INTO log_table (log_message)

VALUES (CONCAT('Duplicate email found: ', NEW.email));

INSERT INTO invalid_login_attempts (email, reason)

VALUES (NEW.email, 'Duplicate email');


```

-- Commented out SIGNAL to test logging behavior without it

-- SIGNAL SQLSTATE '45000'

-- SET MESSAGE_TEXT = 'Duplicate email logged. Insert operation denied.';

END IF;

END$$

DELIMITER ;

DELIMITER $$

CREATE TRIGGER log_incorrect_password

AFTER UPDATE ON register

FOR EACH ROW

BEGIN

    IF (NEW.password != OLD.password) THEN

        INSERT INTO invalid_login_attempts (email, reason)

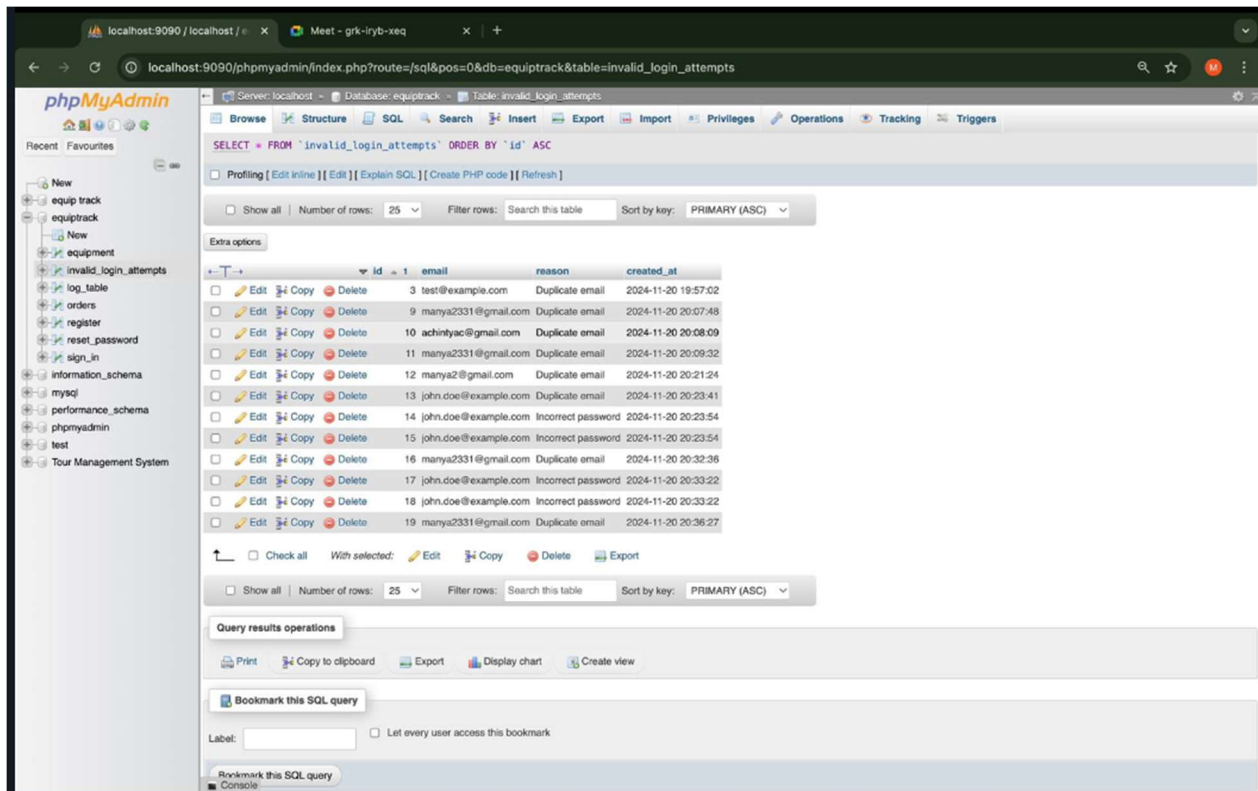
        VALUES (NEW.email, 'Incorrect password');

    END IF;

END$$

```

DELIMITER ;



1. Join to Retrieve All Equipment with Rental Details

Assuming you have a rentals table to track when and by whom equipment is rented, you can join the equipment table with the rentals table to get information about rented items. `SELECT e.equipment_id, e.name, e.category, e.rental_price, r.rental_date, r.rental_duration, r.customer_id`

`FROM equipment e`

`JOIN rentals r ON e.equipment_id = r.equipment_id`

`WHERE e.availability_status = 'Rented';`

2. Join to Get Equipment and Their Rental History

If you want to get the entire rental history for each piece of equipment, including past rentals, you could write a query like this:

```
SELECT e.equipment_id, e.name, e.category, e.purchase_date, r.rental_date,  
r.rental_duration, r.customer_id
```

```
FROM equipment e
```

```
LEFT JOIN rentals r ON e.equipment_id = r.equipment_id
```

```
ORDER BY e.equipment_id, r.rental_date DESC;
```