Asset Management System Report

1. Introduction

The Asset Management System is designed to efficiently track and manage institutional assets. It allows users to register assets, update information, and maintain a record of asset conditions and assignments. The system is developed using PHP, MySQL, HTML, and CSS to ensure a seamless user experience.

2. Database Setup

The system utilizes a MySQL database named **citycot** with a table **asset_reg** structured as follows:

Database Schema (SQL Script)

```
CREATE DATABASE citycot;

CREATE TABLE asset_reg (
   asset_id INT PRIMARY KEY AUTO_INCREMENT,
   asset_name VARCHAR(255) NOT NULL,
   category VARCHAR(255) NOT NULL,
   purchased_date DATE NOT NULL,
   location VARCHAR(100) NOT NULL,
   asset_condition VARCHAR(50) NOT NULL,
   assigned_to VARCHAR(255),
   notes VARCHAR(500)
);
```

3. System Features

3.1 Asset Registration

- Users can add assets by providing relevant details such as asset name, category, purchase date, location, condition, assigned personnel, and notes.
- The form ensures mandatory fields are filled before submission.

3.2 Asset Modification

- Users can update existing records to reflect changes in asset conditions or assignments.
- Data validation is enforced to maintain consistency.

3.3 Asset Deletion

- Users can remove obsolete or disposed assets from the database.
- A confirmation prompt is presented before deletion to prevent accidental removal.

3.4 Data Display

- The system lists all registered assets in a structured table.
- Each record includes an update and delete option.

4. User Interface Design

The interface is designed with simplicity and responsiveness in mind:

- **Forms:** Structured with input fields, dropdowns, and buttons for smooth interaction.
- **Tables:** Display data neatly with alternating row colors for readability.
- **Mobile Responsiveness:** CSS ensures proper layout adjustments on different screen sizes.

5. Implementation Details

- **Backend:** PHP with MySQLi is used for database interactions.
- **Frontend:** HTML and CSS create a user-friendly experience.
- **Security Considerations:**
  - Basic input validation is implemented.
  - SQL Injection is mitigated using prepared statements (optional future improvement).

6. Conclusion

The Asset Management System provides a simple yet effective solution for tracking and managing institutional assets, ensuring data integrity, easy updates, and a user-friendly experience. This Asset Registration System marks the culmination of my training at **CityCotiHub**, where I encountered various challenges, especially with displaying updated and deleted records. Implementing these functionalities required a deep understanding of database interactions, debugging, and troubleshooting. Although these obstacles were initially frustrating, they provided valuable learning experiences that strengthened my technical skills and enhanced my ability to manage complex data processes.

Throughout the development of this system, I focused on creating an interface that was both intuitive and reliable, ensuring that the users could easily access and modify asset information without errors. Going forward, future improvements may include implementing user authentication and role-based access control to ensure data security and provide more personalized access to the system.

This project has not only solidified my understanding of asset management but also prepared me for the challenges I may face in real-world software development. I am confident that the knowledge and skills I’ve gained through this project will be instrumental as I continue my career in software engineering.