# Lab Equipment Booking System

## CPT_S 451 Intro to Database Systems

Parteek Kumar

Aayush Shrestha

Cass Dahle

# Abstract

Managing lab equipment efficiently is a critical challenge in educational institutions and research facilities. Traditional booking methods lead to mismanagement, scheduling conflicts, and limited visibility of equipment availability. This project aims to develop a web-based Lab Equipment Booking system that provides a structured and automated solution. The system enforces role-based access control (RBAC), to streamline lab operations. Key features include user authentication and verification, equipment search and filtering, time-slot validation, booking modifications, and usage analytics.The system ensures data integrity, security and scalability by implementing database constraints and encryption. By optimizing lab resource management, this solution enhances accessibility, reduces administrative overhead, and improves overall efficiency of laboratory operations in academic settings.

# Table of Contents

# Introduction

## Overview of Project

In educational institutes, laboratories serve as crucial environments for research, and experimentation. However, managing lab equipment efficiently presents significant challenges in tracking inventory, and ensuring proper usage. Our project aims to address these issues by developing a Lab Equipment Booking system that allows users to reserve, track, and manage equipment. This system will enforce RBAC, and real-time equipment availability to optimize lab operations.

## Problem Statement

Lack of a centralized system results in - lost or misplaced equipment, double bookings of the same equipment, lack of visibility and availability, and potential for manual errors. These inefficiencies result in wasted time, reduced accessibility to essential lab resources, and disruptions in academic and research activities. Our system provides a centralized, automated, and secure solution to mitigate these issues.

## Objectives and Scope

The primary objective of this project is to build a scalable and Secure web Application that enables role-based management of lab equipment. The system will:

1. Facilitate user registration and verification
2. Provide real time equipment tracking
3. Allow users to filter and search lab equipment
4. Enable a booking system
5. Generate Usage Analytics

## Importance and Potential Impact

Implementing this system will streamline lab operation, reducing equipment conflicts, and enhancing accessibility to lab resources. By integrating real-time monitoring, and analytics it

also allows institutions to make data driven decisions regarding equipment usage policies and procurement. Furthermore, the exportable booking logs will facilitate external reporting and assist in administrative improvements.

# Requirements

## Functional

1. User Registration and Verification
    1. Allow users (students and teachers) to register and verify roles.
        1. Ensures access and roles are controlled
2. User Authentication
    1. Users can log in using verified credentials, access levels are enforced through roles
        1. Only registered and verified users can book equipment.
3. View Equipment and Availability
    1. The system must display a real-time view of available equipment and their status.
        1. Helps users plan reservations, and view availability
4. Search Equipment
    1. Users should be able to search for equipment based on type.
        1. Make it easier to find specific equipment.
5. Filter Equipment
    1. Filter based on type of equipment
        1. View all equipment of a certain type.
6. Equipment Inventory Management
    1. Must maintain details of all available lab equipment, including specifications and status.
        1. Provides a structured way to manage lab resources and track status

7. Equipment Booking System
    1. Users must be able to select equipment and reserve
        1. Core functionality that enables organized equipment usage.
8. System to prevent double-booking
    1. Prevent double booking
        1. Prevent scheduling conflicts, equipment should not be reserved for overlapping time slots.
9. Modify / Cancel Booking
    1. Users should be able to modify and cancel reservations
        1. Provide flexibility, and option to cancel booking.
10. Booking Transactions
    1. Track each booking, ensure foreign key relationship between users and equipment.
        1. Enforces data integrity, and linking bookings
11. Usage Analytics
    1. Display booking patterns / peak usage times / most frequently used equipment
        1. Provides insights, and ideas to optimize equipment availability
12. Export Booking Data
    1. Allow booking and usage logs as CSV or JSON for reporting.
        1. Facilitate external data processing.

# User Stories

## User Registration and Verification

**As a student**, I want to be able to register as a student so that I can access features for lab equipment booking.

**As a teacher**, I want to be able to register as a teacher so that I can manage or book equipment for classes.

**As a lab manager**, I want to be able to register as a lab manager so that I can define what equipment is available.

### User Authentication

**As a student**, I want to log in with my information so that I can access equipment booking for my assigned lab section.

**As a teacher**, I want to log in with my information so that I can access the equipment booking for my assigned lab section.

**As a lab manager**, I want to log in with my information so that I can access equipment inventory for my lab section(s).

**As a system administrator**, I want to enforce role-based access control so that only verified users can book equipment, create equipment, etc.

### View Equipment and Availability

**As a student, teacher, or lab manager**, I want to be able to see a real-time view of of available equipment and their status for my lab section(s) so that I can optimize my bookings

**As a system administrator**, I want to be able view the available equipment for all lab sections so that I can oversee all bookings

### Equipment Inventory Management

**As a lab manager**, I want to maintain an updated inventory of what equipment is available for my lab and add/remove equipment as necessary so that the inventory is kept up-to-date

### Search and Filter Equipment

**As a registered user**, I want to be able to search and filter equipment so that I can find equipment I want to book

## Booking System

**As a student or teacher**, I want to be able to reserve equipment for my lab section(s) so that I can use that equipment

**As a student or teacher**, I want to be able to modify or cancel bookings so that I can have flexibility if schedules change or other events occur

**As a lab manager**, I want to be able to cancel user's bookings/reservations for my lab section(s) so that I can prevent booking of equipment for various reasons (eg equipment out of date, equipment broke between reservation time and the booking time etc)

**As a system administrator**, I want to be able to cancel user's bookings/reservations for any lab section so that I can solve any system-wide booking issues that may occur.

**As a system administrator**, I want to ensure that bookings are linked to users and equipment using foreign keys so that transactional integrity can be maintained

## Usage Analytics

**As a lab manager**, I want to be able to view usage analytics so that I can optimize inventory for people's needs

## Export Booking Data

**As a system administrator**, I want to be able to export bookings logs into CSV/JSON formats so that external data processing can be conducted

## Time-Slot Validation

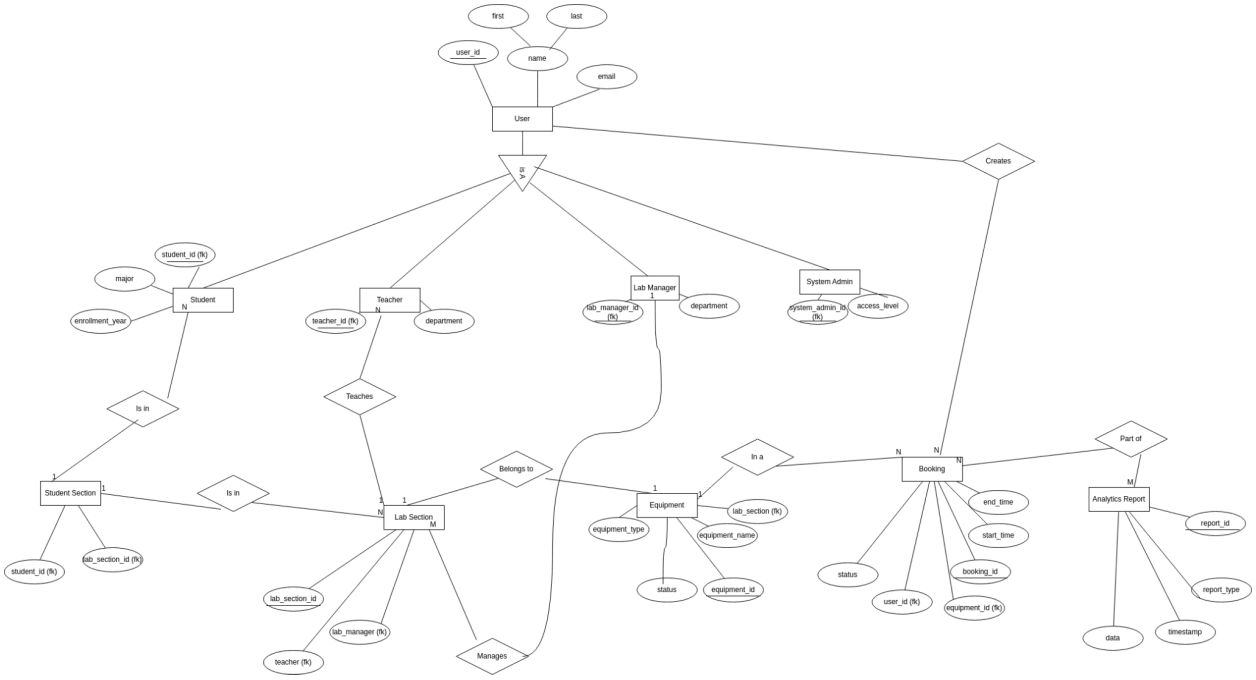
**As a registered user**, I want the system to enforce time-slot validation so that overlapping bookings are auto-rejected.

# Non-Functional Requirements

1. Security and Data Protection
    1. Build infrastructure allowing RBAC and encrypt sensitive user data
        1. Protect user information
2. Concurrency
    1. Prevent race conditions in concurrent bookings
        1. Ensures data consistency and multi-user environments.
3. Error handling and Failover Support
    1. Log failed transactions and handle errors gracefully.
        1. Allows quick transactions and prevents system crashes.
4. Data integrity Constraints
    1. Use Primary Keys, Foreign Keys and check constraints to enforce valid data entry.
        1. Prevent orphan records and inconsistency.
5. Stability
    1. Use indexing, portioning, and optimized query execution plans to support large scale bookings.
        1. Ensures that the system remains performant.
6. Scalability
    1. Horizontal Scaling, and Sharding to support increasing user and equipment data.
        1. Ensure database can handle growth without performance degradation
7. Backups and Recovery
    1. Scheduled backups and recovery schedule / plans
        1. Protect against data loss due to failures
8. Support for different Labs
    1. Design the database to be able to support multiple labs with separate resource pools
        1. Enables flexibility in managing lab-specific booking policies.

# Database Design

## Entity-Relationship Diagram




## Relational Tables

### Handling of Relationships

- 1:1 Relationships:
    - A User can only belong to one specialized role, ensuring clear role-based access control.
    - Equipment belongs to a single Lab Section, preventing duplication across multiple sections.
- 1:M Relationships:

    - A User can make multiple Bookings, but each Booking must belong to a single User to maintain accountability.
    - A Teacher or Lab Manager can be responsible for multiple Lab Sections, but a Lab Section is assigned a single Teacher and a single Lab Manager.
- M:N Relationships:
    - Students are enrolled in multiple Lab Sections, and each Lab Section can have multiple Students. The Student Section table models this relationship.
    - Analytics Reports aggregate data from multiple Bookings, allowing system-wide and lab-specific usage tracking.

## Table Schema

### USER Table

```
CREATE TABLE USER (
   user_id INT PRIMARY KEY,
   name VARCHAR2(100),
   email VARCHAR2(100) UNIQUE
);
```

### STUDENT Table

#### Specialization of USER

```
CREATE TABLE STUDENT (
   student_id INT PRIMARY KEY,  -- also a FK referencing USER(user_id)
   major VARCHAR2(100),
   enrollment_year INT,
   CONSTRAINT FK_Student_User FOREIGN KEY (student_id) REFERENCES USER(user_id)
);
```

### TEACHER Table

#### Specialization of USER

```
CREATE TABLE TEACHER (
   teacher_id INT PRIMARY KEY,  -- also a FK referencing USER(user_id)
   department VARCHAR2(100),
   CONSTRAINT FK_Teacher_User FOREIGN KEY (teacher_id) REFERENCES USER(user_id)
);
```

### LAB_MANAGER Table

### Specialization of USER

```
CREATE TABLE LAB_MANAGER (
   lab_manager_id INT PRIMARY KEY,  -- also a FK referencing USER(user_id)
   department VARCHAR2(100),
   CONSTRAINT FK_LabManager_User FOREIGN KEY (lab_manager_id) REFERENCES USER(user_id)
);
```

## SYSTEM_ADMIN Table

### Specialization of USER

```
CREATE TABLE SYSTEM_ADMIN (
   system_admin_id INT PRIMARY KEY,  -- also a FK referencing USER(user_id)
   access_level VARCHAR2(50),
   CONSTRAINT FK_SystemAdmin_User FOREIGN KEY (system_admin_id) REFERENCES USER(user_id)
);
```

## LAB_SECTION Table

```
CREATE TABLE LAB_SECTION (
   lab_section_id INT PRIMARY KEY,
   teacher_id INT,
   lab_manager_id INT,
   CONSTRAINT FK_LabSection_Teacher FOREIGN KEY (teacher_id) REFERENCES TEACHER(teacher_id),
   CONSTRAINT FK_LabSection_LabManager FOREIGN KEY (lab_manager_id) REFERENCES LAB_MANAGER(lab_manager_id)
);
```

## EQUIPMENT Table

```
CREATE TABLE EQUIPMENT (
   equipment_id INT PRIMARY KEY,
   equipment_name VARCHAR2(100),
   equipment_type VARCHAR2(50),
   status VARCHAR2(50),
   lab_section_id INT,
   CONSTRAINT FK_Equipment_LabSection FOREIGN KEY (lab_section_id) REFERENCES LAB_SECTION(lab_section_id)
);
```

## BOOKING Table

```
CREATE TABLE BOOKING (
   booking_id INT PRIMARY KEY,
   user_id INT,
   equipment_id INT,
   start_time DATETIME,
   end_time DATETIME,
   status VARCHAR2(50),
   CONSTRAINT FK_Booking_User FOREIGN KEY (user_id) REFERENCES USER(user_id),
```

```
    CONSTRAINT FK_Booking_Equipment FOREIGN KEY (equipment_id) REFERENCES EQUIPMENT(equipment_id)
);
```

## STUDENT_SECTION Table

### Junction Table for Students and Lab Sections

```
CREATE TABLE STUDENT_SECTION (
    student_id INT,
    section_id INT,
    PRIMARY KEY (student_id, section_id),
    CONSTRAINT FK_StudentSection_Student FOREIGN KEY (student_id) REFERENCES STUDENT(student_id),
    CONSTRAINT FK_StudentSection_LabSection FOREIGN KEY (section_id) REFERENCES LAB_SECTION(lab_section_id)
);
```

## ANALYTICS_REPORT Table

```
CREATE TABLE ANALYTICS_REPORT (
    report_id INT PRIMARY KEY,
    report_type VARCHAR2(50),
    date DATE,
    data TEXT
);
```

## BOOKING_ANALYTICS_REPORT Table

### Junction Table for Booking and Analytics Report

```
CREATE TABLE BOOKING_ANALYTICS_REPORT (
    booking_id INT,
    report_id INT,
    PRIMARY KEY (booking_id, report_id),
    CONSTRAINT FK_BookingAnalytics_Booking FOREIGN KEY (booking_id) REFERENCES BOOKING(booking_id),
    CONSTRAINT FK_BookingAnalytics_Report FOREIGN KEY (report_id) REFERENCES ANALYTICS_REPORT(report_id)
);
```