**Software Requirements Specification (SRS)**  
**For Computer Lab Management System**

**1. Introduction**

**1.1 Purpose**

The **Computer Lab Management System** aims to provide an efficient and user-friendly solution for managing computer labs, including handling lab resources, tracking inventory, managing user access, logging system usage, and generating reports.

**1.2 Scope**

This system will allow administrators to:

* Manage lab devices and software.
* Track system usage and stock levels.
* Generate receipts and logs for lab activities.
* Resolve lab issues and maintain a system history log.
* Manage user access levels and permissions.
* Maintain lab location details dynamically.

**1.3 Definitions, Acronyms, and Abbreviations**

* **Admin**: A person who has full control over the lab management system.
* **User**: A registered person who can access the system to use lab resources.
* **Lab Inventory**: A record of all devices, systems, and stock in the lab.
* **SRS**: Software Requirements Specification.

**1.4 References**

* IEEE Standard for Software Requirements Specification.
* Relevant database and PHP documentation.

**1.5 Overview**

This document describes the system's overall architecture, functional and non-functional requirements, and constraints.

**2. Overall Description**

**2.1 Product Perspective**

The **Computer Lab Management System** is a web-based application designed for efficient management of multiple computer labs within an institution. It will integrate **PHP**, **MySQL**, and **JavaScript** for an interactive and database-driven experience.

**2.2 Product Features**

* **User Authentication & Authorization**: Secure login and role-based access.
* **Lab Inventory Management**: Track and manage hardware/software in the labs.
* **Issue Logging & Resolution**: Track and resolve technical problems.
* **Stock Management**: Monitor available stock and generate receipts.
* **Lab Logs & Reports**: Maintain system logs for tracking usage history.
* **Lab Address Management**: Store dynamic lab locations and details.

**2.3 User Characteristics**

* **Administrators**: Manage all aspects of the lab system, including user roles, inventory, and reports.
* **Lab Technicians**: Handle issue logging and system maintenance.
* **Students/Users**: Access lab systems and request support.

**2.4 Constraints**

* The system must be built using **PHP, MySQL, JavaScript, and HTML/CSS**.
* Must support **multiple labs with different databases**.
* Should work on **modern browsers**.
* Security measures must be enforced to prevent unauthorized access.

**3. Specific Requirements**

**3.1 Functional Requirements**

* **User Authentication**
  + Users must log in using a valid username and password.
  + Role-based access (Admin, Technician, Student).
* **Lab Management**
  + Ability to add, update, and remove labs.
  + Dynamic database selection based on lab.
* **Inventory Management**
  + List all available lab equipment.
  + Add or remove stock items.
  + Generate and print stock reports.
* **Issue Tracking & Resolution**
  + Users can report system issues.
  + Technicians can update the status of reported issues.
* **System Logs & Reports**
  + Track all lab activities.
  + Generate usage reports by date and lab.
* **Lab Location Management**
  + Store and retrieve lab locations dynamically.
  + Ensure correct mapping of lab names to database records.

**3.2 Non-Functional Requirements**

* **Security**: Role-based access control, encrypted passwords, and SQL injection protection.
* **Performance**: The system should handle multiple concurrent users efficiently.
* **Usability**: A clean and user-friendly interface for ease of use.
* **Scalability**: Should support multiple labs with separate databases.
* **Maintainability**: Code should be modular and well-documented for future improvements.

**3.3 Database Requirements**

* Each lab has its own database (e.g., lab\_a\_db, lab\_b\_db).
* The **main authentication database (main\_auth)** stores user credentials and lab mappings.
* The **stock table (lab\_x\_stocks)** manages inventory for each lab.
* The **lab location table (lab\_x\_locations)** stores building, floor, and room details.

**4. External Interface Requirements**

**4.1 User Interfaces**

* **Login Page**: Secure authentication for users.
* **Dashboard**: Displays system status and quick access to key features.
* **Inventory Management Panel**: CRUD operations for stock and equipment.
* **Issue Reporting Page**: Users can report problems with lab equipment.
* **Reports & Logs Page**: Admins can generate and download reports.

**4.2 Hardware Interfaces**

* Supports **PCs, laptops, and tablets**.
* Requires **internet access** for database connectivity.

**4.3 Software Interfaces**

* Developed in **PHP and MySQL**.
* Uses **HTML, CSS, and JavaScript** for front-end.
* Compatible with **Windows and Linux-based servers**.

**4.4 Communication Interfaces**

* HTTP(S) for web-based access.
* Email notifications for critical issues.

**5. Other Requirements**

**5.1 Security Requirements**

* **Encryption**: Secure sensitive user data.
* **Access Control**: Implement user roles and permissions.
* **Database Security**: Prevent SQL injection and unauthorized access.

**5.2 Performance Requirements**

* System should handle at least **50 concurrent users**.
* Response time for queries should be **under 2 seconds**.

**5.3 Business Rules**

* Only **authorized personnel** can modify lab inventory.
* All **stock changes must be logged** for accountability.

**6. Appendices**

* **Appendix A**: ER Diagram for the system.
* **Appendix B**: List of database tables and their schema.

**Approval & Review**

This document is subject to approval by the **project manager and development team** before implementation.

**Software Requirement Specification (SRS)**

**Computer Lab Management System**

**1. Introduction**

**1.1 Purpose**

The Computer Lab Management System (CLMS) is designed to streamline the management and operation of computer labs by providing functionalities such as device and system tracking, inventory management, issue resolution, and logging system activities.

**1.2 Document Conventions**

* **Bold Text**: Section titles and key terms
* *Italic Text*: Emphasized content
* Code Block: Sample commands or snippets

**1.3 Intended Audience and Reading Suggestions**

This document is intended for:

* System Administrators
* Developers
* End Users
* IT Staff
* Project Managers

**1.4 Product Scope**

CLMS provides a centralized platform for managing multiple labs, ensuring smooth operations by enabling system monitoring, resource allocation, and issue resolution.

**1.5 References**

* IEEE SRS Standards
* Database Management System Concepts
* Web Application Security Guidelines

**2. Overall Description**

**2.1 Product Perspective**

The CLMS is a web-based application that integrates with existing IT infrastructure, providing secure role-based access to lab managers and administrators.

**2.2 Product Functions**

* **User Authentication** (Admin, Lab Manager, Staff)
* **Device and System Management**
* **Stock and Inventory Tracking**
* **Issue Reporting and Resolution**
* **System Logs and Activity Tracking**
* **QR Code-Based Login**
* **Lab Location Information Management**

**2.3 User Characteristics**

* **Admin**: Full access to all features
* **Lab Manager**: Can manage devices, inventory, and logs
* **Staff**: Can report issues and monitor lab systems

**2.4 Constraints**

* Web-based system requiring internet access
* Database should support multiple lab environments
* Secure authentication mechanisms must be in place

**2.5 Assumptions and Dependencies**

* System is hosted on a secured server
* Users have basic knowledge of computers
* Integration with third-party authentication (if required)

**3. Specific Requirements**

**3.1 Functional Requirements**

* **User Authentication System** (Username/Password & QR Code Login)
* **Manage Labs** (Create, Modify, Delete Labs)
* **Manage Devices and Systems**
* **Track Stock and Inventory**
* **Report and Resolve Issues**
* **View System Logs**
* **QR Code Integration for User Authentication**
* **Lab Location Management**

**3.2 External Interface Requirements**

**3.2.1 User Interfaces**

* Responsive Web Interface (Bootstrap)
* Login, Dashboard, Management Panels
* QR Code Scanning Module

**3.2.2 Hardware Interfaces**

* Server with database support
* Barcode/QR Code Scanner (Optional)

**3.2.3 Software Interfaces**

* Database: MySQL
* Backend: PHP
* Frontend: HTML, CSS, JavaScript

**3.3 Performance Requirements**

* Support multiple concurrent users
* Load time should not exceed 3 seconds per request
* Data backup should be available

**3.4 Security Requirements**

* Role-based authentication
* Data encryption
* Secure API endpoints

**3.5 Other Non-functional Requirements**

* Scalability: Should support additional labs and users
* Maintainability: Modular code structure
* Reliability: 99.9% uptime

**4. Appendices**

* Sample database schema
* User role descriptions
* API endpoints

This document serves as a detailed guideline for the development and deployment of the **Computer Lab Management System**.