A
MINI PROJECT-I REPORT
ON

“ICT Lab Management System”

Submitted in Partial Fulfillment of Requirement for the award of Degree of

BACHELOR OF TECHNOLOGY

COMPUTER SCIENCE AND ENGINEERING

of

Dr. Babasaheb Ambedkar Technical University, Lonere Submitted By

Student Name	Examination Number
Mr. Vikas Sadashiv Mali	2167971242038
Mr. Yash Rajaram Mane	2167971242030
Mr. Prathmesh Pradip Shinde	2167971242055
Mr. Sanket Rajendra Jagtap	2167971242060
Mr. Jayant Arvind Wagh	2167971242044

UNDER THE GUIDANCE OF

Prof. N. V. Deshmukh

Raosaheb Wangde Master Charitable Trust’s
Dnyanshree Institute of Engineering and Technology
Sajjangad Road, Tal. Dist. Satara, Maharashtra State, 415 013.
2023-2024

Certificate

This is to certify that the mini project-I report entitled, **"ICT Lab Management System" Submitted by**

Student Name	**Examination Number**
Mr. Vikas Sadashiv Mali	2167971242038
Mr. Yash Rajaram Mane	2167971242030
Mr. Prathmesh Pradip Shinde	2167971242055
Mr. Sanket Rajendra Jagtap	2167971242060
Mr. Jayant Arvind Wagh	2167971242044

It is a bonafide work carried out by these students under guidance of Prof. N. V. Deshmukh. It has been accepted and approved for the partial fulfillment of the requirement of Dr. Babasaheb Ambedkar Technical University, Lonere, for the award of the degree of Bachelor of Technology (Computer Science and Engineering). This Mini Project-I work and project report has not been earlier submitted to any other Institute or University for the award of any degree or diploma.

Prof. N. V. Deshmukh	**Dr. S. P. Kosbatwar**	**Dr. A. D. Jadhav**
(Guide)	(Head of dept.)	(Principal)

Prof.(External Examiner) :
Place : Satara
Date:

ABSTRACT

Information and Communication Technology (ICT) labs play a crucial role in providing students and researchers with access to the resources they need to succeed. However, managing these labs effectively can be a challenge. In this report, we present the development of a web-based ICT lab management system to address this challenge. The ICT Lab Management System provides a comprehensive solution for managing ICT lab resources, including hardware, software, and consumables. It also facilitates lab scheduling and booking, maintains accurate records of lab usage, and generates reports on ICT lab usage. The ICT Lab Management System is designed to be user-friendly and easy to use, and it has been successfully implemented in a pilot study at a local university.

Keywords :

ICT lab management, Scheduling, Usage Tracking, Reporting.

Contents

1. Introduction 5
    1.1 Purpose 5
    1.2 Proposed Solution 5
    1.3 System Overview 5
2. Literature Review 6
3. Diagrams 7
    3.1 Data Flow Diagram 7
    3.2 Sequence Diagram 8
    3.3 Use Case Diagram 8
    3.4 Class Diagram 9
    3.5 Deployment Diagram 9
4. Specification 10
5. System 11
6. Implementation 14
7. Testing And Evaluation 17
    7.1 White Box Testing 17
    7.2 Black Box Testing 19
    7.3 Test Case Generation 21
8. Conclusion And Future Scope 23
    8.1 Conclusion 23
    8.2 Future Scope 23
9. References 25

1. Introduction

1.1 Purpose

ICT labs play a vital role in providing students and researchers with the tools they need to succeed in their studies and research. However, managing these labs effectively can be a challenge. Manually managing lab resources, such as hardware, software, and consumables, can be time-consuming and error-prone. Additionally, scheduling lab time and resources can be difficult, and it can be challenging to track lab usage accurately.

1.2 Proposed Solution

To address these challenges, we propose the development of a web-based ICT lab management system (ICT LMS). The ICT LMS provides a comprehensive solution for managing ICT lab resources, including hardware, software, and consumables. It also facilitates lab scheduling and booking, maintains accurate records of lab usage, and generates reports on ICT lab usage.

1.3 System Overview

an ICT lab management system (ICT LMS) is a software application that helps educational institutions manage their ICT labs effectively. It provides a centralized platform for managing ICT lab resources, scheduling lab time, tracking usage, and generating reports. ICT LMS can provide several benefits to educational institutions, including improved resource utilization, enhanced user experience, reduced operational costs, data-driven decision-making, and enhanced security.

2. Literature Review

ICT lab management systems (ICT LMS) play an important role in improving resource utilization, managing scheduling, tracking usage, and generating reports. They provide a centralized platform that streamlines ICT lab operations. ICT LMS has been shown to enhance user experience for both instructors and students by simplifying lab access, resource booking, and communication. Instructors can schedule lab time for their classes, assign students to lab workstations, monitor student activity in the lab, and submit feedback on lab resources. Students can book lab time, use lab resources to complete coursework, request assistance from lab technicians, and submit feedback on lab resources.

ICT Lab Management System also promotes data-driven decision-making in ICT lab management by providing detailed usage reports and analytics, which enable administrators to identify trends, optimize resource allocation, and evaluate the performance of their labs. This data-driven approach has been shown to improve lab utilization rates, reduce maintenance costs, and enhance overall lab effectiveness. While ICT Lab Management System offers significant benefits, it also presents challenges, such as initial investment in hardware, software, and training required to set up and maintain the system.

To overcome these challenges, institutions should provide comprehensive training and support for users, as well as incentives for using the system effectively. Implementing ICT LMS can improve resource utilization, enhance user experience, reduce operational costs, and promote data-driven decision-making. Institutions should carefully consider their needs and resources before implementing an ICT LMS and should provide comprehensive training and support to ensure user acceptance and effective utilization of the system.

3. Diagrams

3.1 Data Flow Diagram

Data Flow Diagram (DFD) is a visual representation that illustrates the flow of data within a system. It consists of processes, data stores, data flows, and external entities, organized in hierarchical levels. DFDs are used to comprehend the movement of data, identify inputs and outputs, and understand the structure of a system.

1. Level-0 Data flow Diagram

Figure 1: Level 0 Data flow Diagram

2. Level-1 Data flow Diagram

Figure 2: Level 1 Data flow Diagram

3. Level-2 Data flow Diagram

Figure 3: Level 2 Data flow Diagram

3.2 Sequence Diagram

Sequence Diagram showcases the chronological order of interactions between objects in a system. Objects, lifelines, messages, and activation boxes are key elements in this diagram. It aids in understanding the dynamic behavior of a system by illustrating how objects collaborate over time. Sequence diagrams are valuable for visualizing the flow of messages between objects during the execution of a specific scenario. Together, these diagrams play essential roles in designing, documenting, and communicating aspects of software systems throughout the development lifecycle.

Figure 4: Sequence Diagram

3.3 Use Case Diagram

Use Case Diagram focuses on the interactions between a system and external entities, known as actors. Actors can be users or external systems, while use cases represent specific functionalities or tasks the system performs. Use case diagrams are beneficial for visualizing system functionalities, identifying actors, and depicting the relationships between them.

Figure 5: Use Case Diagram

3.4 Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

Figure 6: Sequence Diagram

3.5 Deployment Diagram

The deployment diagram visualizes the physical hardware on which the software will be deployed. It portrays the static deployment view of a system. It involves the nodes and their relationships. It ascertains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node. Since it involves many nodes, the relationship is shown by utilizing communication paths.

Figure 7: Sequence Diagram

4. Specification

Hardware :
Ram- 4GB ,Processor- intel i3/ Ryzen 3, Hard Disk- 256GB

Software :
Operating System- Windows, Linux

Tools Used :

1. **Programming language :**
   Html, Css, Javascript ,Php
2. **Framework :**
   Bootstrap
3. **Database management system :**
   MYSQL, Xampp
4. **Integrated Development Environment :**
   Visual Studio code, Sublime Text
5. **Documentation :**
   Star UML, Latex

5. System

1. Visual studio code :

Figure 8: Vs code

Visual Studio Code (VS Code) is a popular source-code editor developed by Microsoft. It supports various programming languages and features like syntax highlighting, debugging, and Git integration.

2. Xampp :

Figure 9: Xampp

XAMPP is a free, open-source cross-platform web server solution stack. It includes Apache (web server), MySQL (database), PHP, and Perl, facilitating local development environments.

3. MYSQL :

Figure 10: MYSQL

MySQL is a popular open-source relational database management system (RDBMS). It is widely used for storing and managing data in web applications.

4. Html :

Figure 11: HTML

HTML (Hypertext Markup Language) is the standard markup language for creating web pages. It structures content using elements, each represented by tags.

5. CSS :

Figure 12: CSS

CSS (Cascading Style Sheets) is a style sheet language used for describing the look and formatting of a document written in HTML. It enhances the visual presentation of web pages.

6. JavaScript :

Figure 13: JavaScript

JavaScript (JS) is a versatile programming language used for creating dynamic content on websites. It enables interactive features and client-side scripting.

7. PHP :

Figure 14: PHP

PHP is a server-side scripting language designed for web development. It is embedded in HTML code and executed on the server, generating dynamic web pages.

8. BootStrap :

Figure 15: Bootstrap

Bootstrap is a front-end framework that simplifies the development of responsive and mobile-first websites. It provides predesigned components and a grid system.

6. Implementation

1. Login Page for HOD

Figure 16: Login Page

2. Login Page for Admin

Figure 17: Admin Login Page

3. Home Page

Figure 18: Home Page

4. Staff section

Figure 19: Staff section

5. Department Section

Figure 20: Department Section

6. Leave History

Figure 21: Leave History

7. Testing And Evaluation

7.1 White Box Testing

White-box test used to test the inner part of the "box", and it focuses on utilizing inside learning of the product to direct the choice of test information. This tests include: structural test, glass-box and clear-box. White box testing is expensive than black box. It takes the source code, before the tests can be. This is significantly more like in the assurance of exact information and the assurance if the product is or is not right. This testing is connected just with the product item; it can't guarantees that the entire specification has been executed. The clear box or WhiteBox name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "Black Box Testing symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested

Figure 22: White Box Testing

For the project involving staff details management and leave application system, white box testing can be structured to thoroughly examine the internal logic and components of the software. Here's an outline of white box testing scenarios specific to this project:

1. **Input Validation Testing:**
   - Verify that the system properly validates and handles different types of input for user details (e.g., names, IDs)
   - Test how the system responds to invalid inputs, such as special characters or incorrect date formats.

2. Code Coverage Testing:

- Ensure that all code paths are executed, including different branches for user details management and new equipment application processes.
- Test each function and method to guarantee complete coverage of the codebase.

3. Boundary Value Analysis:

- Evaluate the system's behavior at boundary values, such as the minimum and maximum allowed lengths for user details or the maximum requests for new equipment at the same time.
- Test the system's response to values near these boundaries.

4. Error Handling Testing:

- Validate how the system handles errors, such as incorrect login credentials, submission of incomplete forms, or attempting to apply for new equipment on non-working days.
- Confirm that appropriate error messages are displayed, and the system gracefully handles unexpected situations.

5. Security Testing:

- Review the system's code for potential security vulnerabilities, such as SQL injection or cross-site scripting (XSS).
- Ensure that sensitive information, like passwords or personal details, is handled securely.

6. Integration Testing:

- Examine how different components of the system interact with each other, focusing on the integration between user details management and equipment application modules.
- Verify that data is appropriately passed between modules without loss or corruption.

7. Performance Testing:

- Assess the system's performance under different loads, especially during peak times when required many equipment simultaneously access or update their details or submit application requests.
- Identify and address any performance bottlenecks.

8. Concurrency Testing:

- Simulate scenarios update their details or submit new equipment application simultaneously
- Ensure that the system maintains data integrity and consis-

tency under concurrent interactions.

9. Code Optimization Testing:

- Review the code for opportunities to optimize performance, enhance readability, and adhere to best coding practices.
- Identify areas where code refactoring or improvements can be made.

10. Usability Testing:

- While typically associated more with black box testing, elements of usability can be assessed at the code level, ensuring that the code supports a user-friendly interface and adheres to HCI principles.

7.2 Black Box Testing

Black box testing is a technique of software testing which examines the functionality of software without peering. into its internal structure or coding. The primary source of black box testing is a specification of requirements that is stated by the customer. In this method, tester selects a function and gives input value to examine its functionality, and checks whether the function is giving expected output or not. If the function produces correct output, then it is passed in testing, otherwise failed. The test team reports the result to the development team and then tests the next function. After completing testing of all functions if there are severe problems, then it is given back to the development team for

Figure 23: White Box Testing

Certainly, for the user details management and equipment application system project, black box testing focuses on assessing the functionalities and behavior of the application without delving into its internal code. Here are black box testing scenarios tailored for this project:

1. User Authentication Testing:

- Verify that users can successfully log in with valid credentials.
- Test the system's response to incorrect login credentials, ensuring appropriate error messages are displayed.

2. Staff Details Management:

- Test the functionality of adding new user details, ensuring all required fields are processed correctly.
- Validate the system's ability to update existing user details accurately.
- Confirm the proper handling of deleting user records.

3. Leave Application Testing:

- Submit new equipment application requests for various scenarios, including different durations and reasons.
- Test the system's response to conflicting application requests or overlapping dates.
- Ensure that application requests are correctly reflected in the system.

4. User Interface Testing:

- Verify that the user interface is intuitive and adheres to HCI principles.
- Test navigation through the system to ensure users can easily access different sections (e.g., admin details, HOD details, new equipment application).
- Confirm that form fields and buttons are labeled appropriately.

5. application Approval Process:

- Test the system's handling of application requests by lab lab incharge, ensuring they can efficiently review, approve, or decline requests.
- Verify that notifications or status updates are correctly displayed to lab lab incharge after approval or rejection.

6. Boundary Value Testing:

- Submit data near the boundary values for different fields, such as the maximum and minimum lengths for staff details or the maximum duration for leave requests.
- Ensure the system handles these boundary values appropriately.

7. Cross-Browser and Cross-Device Compatibility Testing:

- Test the application on different browsers (e.g., Chrome, Firefox, Safari) and devices (e.g., desktops, tablets, smartphones) to ensure consistent functionality and appearance.

8. Concurrency Testing:
- Simulate scenarios where multiple users access the system simultaneously to assess how it handles concurrent interactions without data corruption or conflicts.

9. Error Handling and Messages:
- Validate error messages and system notifications, ensuring they are clear, concise, and provide guidance on resolving issues.

10. Security Testing:
- Assess the security of user data during interactions with the system.
- Test for unauthorized access attempts and ensure proper encryption of sensitive information.

11. Performance Testing:
- Evaluate the system's response time and performance under normal and peak loads, ensuring it meets acceptable performance standards.

Black box testing, when systematically conducted with these scenarios, helps ensure that the staff details management and leave application system performs as expected from an end-user perspective and meets the specified requirements.

7.3 Test Case Generation

Testing aims at finding errors in a system or program. A set of tests is also called a test suite. Test case generation is the process of generating test suites for a particular system. Model-based Testing (MBT) is a technique to generate test suites for a system from a model describing the system. One usually tries to generate test suite which satisfies a given coverage criterion.

1. Positive Test Cases

Positive testing is the type of testing that can be performed on the system by providing the valid data as input. It checks whether system behaves as expected with positive inputs. This test is done to check the system that does what it is supposed to do.

2. Negative Test Cases

Negative Testing is a variant of testing that can be performed on the system by providing invalid data as input. It checks whether an system behaves as expected with the negative inputs. This is to test the system does not do anything that it is not supposed to do so.

8. Conclusion And Future Scope

8.1 Conclusion

ICT lab management systems play a vital role in optimizing ICT lab operations, enhancing user experience, and promoting data-driven decision-making in educational institutions. By addressing the challenges of implementation and adopting effective strategies, institutions can leverage ICT LMS to create a more efficient, effective, and user-friendly ICT learning environment.

However, implementing ICT LMS also presents challenges, such as initial investment, integration with existing systems, and user adoption. To overcome these challenges, institutions should carefully consider their needs and resources before implementing an ICT LMS and should provide comprehensive training and support to ensure user acceptance and effective utilization of the system.

An ICT lab management system is an essential tool for educational institutions that want to manage their ICT labs effectively and provide a positive learning environment for students. By streamlining resource management, scheduling, and usage tracking, ICT LMS can help to improve the efficiency and effectiveness of ICT labs, supporting the educational goals of the institution.

8.2 Future Scope

The future scope of the project extends beyond its current functionalities, opening avenues for further enhancements and adaptations:

Sure, here are 10 points on the future scope of ICT lab management systems:

1. Integration with Artificial Intelligence (AI): ICT LMS can be integrated with AI to provide intelligent resource allocation, personalized learning recommendations, and predictive maintenance suggestions.

2. Mobile Access and Automation: ICT lab management sys-

tems can incorporate mobile apps to allow users to access and manage lab resources remotely, and automate routine tasks such as booking lab time and requesting assistance.

3. Real-time Monitoring and Analytics: ICT LMS can provide real-time monitoring of lab usage and performance, enabling proactive identification of issues and optimization of resource allocation.

4. Enhanced User Experience and Personalization: ICT lab management systems can personalize the user experience by tailoring lab recommendations and resource allocation based on individual needs and preferences.

5. Integration with Learning Management Systems (LMS): ICT LMS can be integrated with lab management systems to provide a seamless learning experience by synchronizing lab access with course requirements and assignments.

6. Support for Emerging Technologies: ICT LMS can adapt to support emerging technologies, such as virtual reality (VR) and augmented reality (AR), by providing specialized resource management and usage tracking capabilities.

7. Data-driven Decision Making and Optimization: ICT LMS can provide advanced analytics and reporting tools to enable data-driven decision making for lab resource allocation, scheduling, and maintenance.

8. Integration with Building Management Systems (BMS): ICT lab management systems can integrate with BMS to optimize energy consumption and environmental impact of lab operations.

9. Support for Collaborative Learning and Research: ICT lab management systems can facilitate collaborative learning and research by providing tools for group scheduling, resource sharing, and project management.

10. Adaptability to Changing Educational Needs: ICT lab management systems should be adaptable to changing educational needs and pedagogical approaches to remain relevant and effective in the future.

9. References

1. W3Schools. (2023). HTML Tutorial. Retrieved from [https://www.w3schools.com/html/] (https://www.w3schools.com/html/)

2. PHP: Hypertext Preprocessor. (2023). PHP Manual. Retrieved from[https://www.php.net/manual/en/] (https://www.php.net/manual/en/)

3. JavaScript MDN. (2023). JavaScript. Retrieved from [https://developer.mozilla.org/en-US/docs/Web/JavaScript] (https://developer.mozilla.org/en-US/docs/Web/JavaScript)