RESEARCH ARTICLE OPEN ACCESS

# Remote Lab Assistance System

Angel Donny
(Assistant Professor, City Engineering College, and Bangalore
Email: angel_d@cityengineeringcollege.ac.in)
Shisher J
(Computer Science, City Engineering College, and Bangalore
Email: shisher51@gmail.com)
G Suhas Raj
(Computer Science, City Engineering College, and Bangalore
Email: gsuhasraj11@gmail.com)
Shekhar C M
(Computer Science, City Engineering College, and Bangalore
Email: shekharyadavcm22@gmail.com)

----------------------------------------************************----------------------------------

## Abstract:
The Remote Lab Assistance System is a web-based platform developed to provide remote access to laboratory resources and expert assistance. The system allows users to interact with lab setups, request guidance, and monitor experiments in real time without the need for physical presence. It aims to improve accessibility and efficient utilization of laboratory infrastructure.
The application is developed using the **MERN stack (MongoDB, Express.js, React.js, and Node.js)**, following a client–server architecture for secure data handling and smooth communication. This system supports flexible and scalable learning, making it suitable for educational institutions to enhance practical learning beyond traditional lab environments.

**Keywords ---** Real time lab monitoring, Remote access, Anti-cheating system, Real-Time Interaction, Accessibility, MERN stack

----------------------------------------************************----------------------------------

## I. INTRODUCTION

Practical laboratory sessions play a crucial role in technical education by helping students apply theoretical knowledge to real-world scenarios. However, traditional laboratory systems often face limitations such as restricted access, limited time slots, lack of resources, and the requirement for physical presence. These challenges can reduce the effectiveness of practical learning, especially for students who are unable to access labs regularly. The **Remote Lab Assistance System** is proposed to overcome these limitations by enabling users to access laboratory assistance and resources remotely through a web-based platform. The system allows students to interact with lab environments, request guidance, and receive support from experts without being physically present in the laboratory. This approach improves accessibility, flexibility, and efficient utilization of lab infrastructure.
The system is developed using the **MERN stack (MongoDB, Express.js, React.js, and Node.js)**, which provides a scalable, secure, and responsive client–server architecture. By integrating modern web technologies, the Remote Lab Assistance System enhances practical learning experiences and supports digital transformation in educational institutions.

## II. PROBLEM STATEMENT & OBJECTIVES

**Problem Statement:** Traditional laboratory systems require physical presence, fixed schedules, and adequate infrastructure, which limits accessibility for many students. Factors such as limited lab availability, time constraints, and lack of expert guidance reduce the effectiveness of practical learning. There is a need for a system that allows students to access laboratory assistance remotely while ensuring efficient resource utilization and continuous support.

**Objectives:**

1. To design and develop a **Remote Lab Assistance System** that enables users to access lab support from any location. Enable hands-free addition, removal, and confirmation of orders.
2. To provide real-time interaction between students and lab experts through a web-based platform..
3. To ensure secure user authentication and data management using modern web technologies..
4. To implement a scalable system using the **MERN stack** for efficient performance and future expansion.
5. To enhance practical learning by reducing dependency on physical laboratory presence.

## III. PROPOSED SYSTEM & ARCHITECTURE

The system follows a **client–server architecture** implemented using the **MERN stack**. The **React.js** front end provides a responsive and user-friendly interface for users to interact with the system. The **Node.js and Express.js** backend handles application logic, user authentication, and communication between components. **MongoDB** is used as the database to store user information, requests, and system data securely. The front end communicates with the backend through RESTful APIs, ensuring smooth data exchange and system reliability.

**System Architecture Overview**

The proposed architecture consists of the following components:

**Frontend Web Application**

- Displays active protocols, timer status, and system feedback while capturing voice commands.

**Communication Layer**

- Uses the Web Speech API to provide a seamless flow of recognized commands to the server.

**Backend Server**

- Processes requests, validates experimental steps, and manages the database.

**Database (Menu + Orders)**

- Stores, step-by-step instructions, safety guidelines, and user-recorded data..

**Workflow of the Proposed System**

1. The user accesses the Remote Lab Assistance System through a web browser.
2. New users register and existing users log in using secure authentication.
3. After successful login, the user submits a request for lab assistance or selects the required laboratory service.
4. The request is sent to the server, where it is processed and stored in the database.
5. The lab administrator or expert receives the request and provides guidance or assistance remotely.
6. Real-time interaction and updates are managed through the backend server.
7. All user activities and responses are securely saved for future reference.
8. The user logs out after completing the session.

**Key Features of the Proposed System**

- **Remote Access** – Users can get lab help from anywhere using the internet.
- **Live Support** – Students can communicate with lab experts in real time.
- **Secure Login** – User accounts are protected through secure authentication.
- **MERN Technology** – The system is developed using MongoDB, Express.js, React.js, and Node.js.
- **Data Storage** – User details and lab requests are stored safely in the database.
- **Easy to Expand** – The system can handle more users and new features in the future.

## IV. METHODOLOGY

The Remote Lab Assistance System is developed using the MERN stack following a client–server approach. The system handles user requests through a web interface and processes them using backend services. Secure data storage and communication are ensured throughout the system.
The major steps are:

1. **Requirement Analysis**
   The system requirements are collected by analyzing the needs of students and lab administrators.
   Functional and non-functional requirements are identified clearly.
   This helps in defining the overall scope of the system.
2. **System Design**
   A client–server architecture is designed for smooth communication.
   The system structure is planned using the MERN stack.
   Data flow and module interactions are defined in this stage.
3. **Frontend Development**
   The user interface is developed using React.js.
   It provides easy navigation and user interaction.
   Forms and pages are created for login and lab requests.
4. **Backend Development**
   Node.js and Express.js are used to handle server-side logic.
   APIs are created to process user requests securely.
   Authentication and validation are managed in this layer.
5. **Database Management**
   MongoDB is used to store user details and lab assistance data.
   Data is stored in collections for easy access and management.
   Security measures are applied to protect stored information.
6. **Testing and Deployment**
   The system is tested to ensure proper functionality and performance.
   Errors are identified and fixed during testing.
   Finally, the application is deployed for user access.

## V. ALGORITHM / PSEUDOCODE

```
BEGIN

1.  START the Remote Lab Assistance System.
2.  DISPLAY the home page to the user.
3.  IF user is new THEN register by collecting user details.
4.  ELSE prompt the user to enter login credentials.
5.  VALIDATE the entered credentials from the database.
6.  IF credentials are invalid THEN display error message and return to login page.
7.  ELSE grant access and DISPLAY the user dashboard.
8.  SELECT the required lab assistance option.
9.  SUBMIT the lab assistance request.
10. STORE the request details in the database.
11. ADMIN/EXPERT views the request and sends a response.
12. LOGOUT the user and STOP the system.
```

## VI. IMPLEMENTATION

The Remote Lab Assistance System is implemented using the **MERN stack**. The frontend is developed with React.js, while Node.js and Express.js handle backend operations.

1. **Frontend (React JS):** The frontend is developed using **React.js** to create a responsive user interface. It allows users to register, log in, and submit lab assistance requests. The frontend communicates with the backend using RESTful APIs.
2. **Backend (Node.js & Express):** The backend is implemented using **Node.js and Express.js**.It handles user authentication, request processing, and business logic.

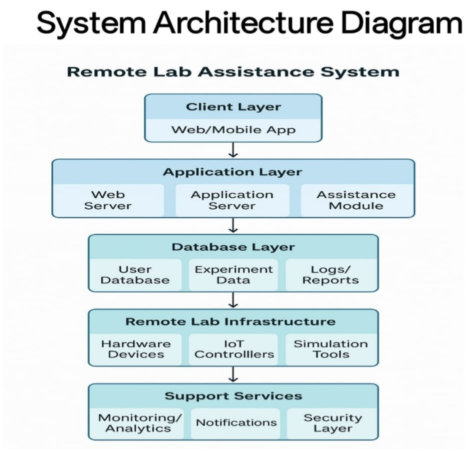
APIs are used to manage communication between frontend and database.

3. **Database (MongoDB):**The database is developed using MongoDB. It stores user information, lab requests, and response details securely. Collections are used for efficient data storage and retrieval.

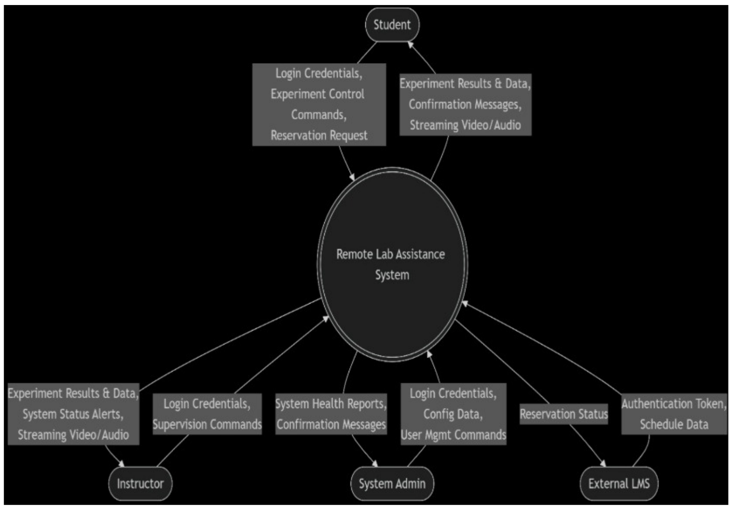
## VII. SYSTEM ARCHITECTURE DIAGRAMS

The system architecture diagram represents the overall structure of the Remote Lab Assistance System. It shows how users interact with the frontend developed using React.js. The frontend communicates with the backend built using Node.js and Express.js through RESTful APIs. The backend processes requests and stores data securely in the MongoDB database.

The backend, implemented using Node.js and Express.js, handles application logic, user authentication, and request processing. It acts as an intermediate layer between the frontend and the database, validating user actions and managing system workflows. The database layer uses MongoDB to store user information, lab assistance requests, and response details securely. This layered architecture ensures scalability, security.



Data Flow Diagram


Student
Login Credentials, Experiment Control Commands, Reservation Request
Experiment Results & Data, Confirmation Messages, Streaming Video/Audio
Remote Lab Assistance System
Experiment Results & Data, System Status Alerts, Streaming Video/Audio
Login Credentials, Supervision Commands
System Health Reports, Confirmation Messages
Login Credentials, Config Data, User Mgmt Commands
Reservation Status
Authentication Token, Schedule Data
Instructor
System Admin
External LMS

## VIII. RESULTS

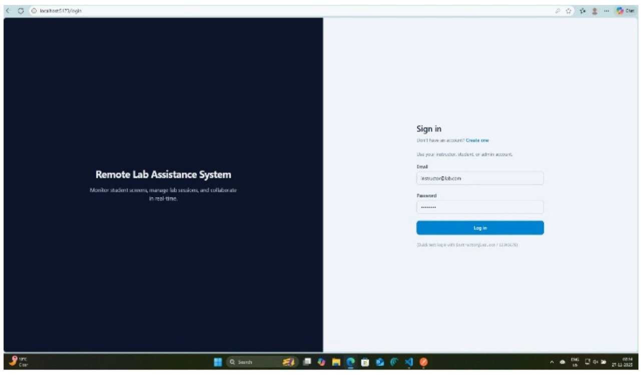
The Remote Lab Assistance System was successfully implemented and tested using the MERN stack. The system allowed users to register, log in securely, and request lab assistance through a user-friendly interface. Smooth communication between the frontend, backend, and database was achieved, ensuring reliable performance, secure data storage, and effective interaction between students and lab experts.

**Implemented Features:**

- User registration and secure login functionality.
- Dashboard for submitting lab assistance requests.
- Real-time interaction between users and lab experts.
- Backend API handling using Node.js and Express.js.
- Secure data storage and management using MongoDB.

**Outcome:**

The implemented system effectively addresses the limitations of traditional laboratory access by enabling remote assistance through a web-based platform. It improves accessibility, flexibility, and efficient utilization of lab resources while ensuring data security. The use of the MERN stack makes the system scalable and suitable for future enhancements, supporting modern digital learning environments.



Remote Lab Assistance System

Monitor student screens, manage lab sessions, and collaborate in real-time.

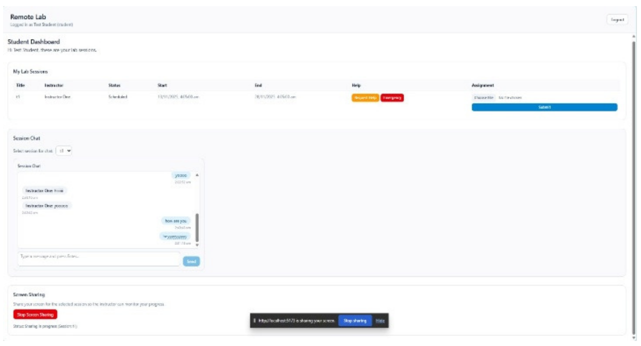
Sign in

Don't have an account? Create one

Use your instructor, student, or admin account.

Email: instructor@lab.com

(Quick test login with instructor@lab.com / 12345678)



Remote Lab

Logged in as Test Student (student)

Student Dashboard

Hi Test Student, these are your lab sessions.

My Lab Sessions

| Title | Instructor | Status | Start | End | Help | Assignment |
|---|---|---|---|---|---|---|
| t1 | Instructor One | Scheduled | 13/11/2025, 4:05:00 am | 28/11/2025, 4:05:00 am | Request Help Emergency | Choose file No file chosen Submit |

Session Chat

Select session for chat: t1

Session Chat

yooo

Instructor One: hiiii

Instructor One: yooooo

how are you

Type a message and press Enter...

Screen Sharing

Share your screen for the selected session so the instructor can monitor your progress.

Screen Sharing in progress (Session: t1)

http://localhost:5173 is sharing your screen.

## IX. CONCLUSION

**Conclusion:**

The Remote Lab Assistance System provides an effective solution to overcome the limitations of traditional laboratory access. By enabling remote interaction between students and lab experts, the system improves accessibility and flexibility in practical learning. The application is developed using the MERN stack, which ensures a scalable and efficient client–server architecture. Secure authentication and reliable data storage enhance The encouragement from the department greatly contributed to this work. I truly appreciate their support. I would also like to thank my friends and classmates for their cooperation, discussions, and assistance during the project development. Their support and teamwork helped overcome challenges effectively.

system safety and performance. The system reduces dependency on physical lab presence and optimizes the use of laboratory resources. It supports continuous learning beyond fixed lab schedules. The user-friendly interface makes the system easy to use for both students and administrators.

**Future Scope:**

- Integration of **live video streaming** for real-time lab demonstrations.
- Support for **IoT-based lab equipment control** from remote locations.
- Addition of **AI-based assistance** for automated guidance and query handling.
- Development of a **mobile application** for easier access.
- Enhanced **security features** such as multi-factor authentication.
- Support for **multiple labs and institutions** on a single platform.

## X. ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my project guide for their continuous guidance, valuable suggestions, and encouragement throughout the course of this project. Their technical insights and constant support played a crucial role in the successful completion of this work. I am thankful for the time and effort they dedicated to reviewing and improving the project. Their motivation helped me stay focused and confident. I sincerely appreciate their mentorship. I extend my heartfelt thanks to the Head of the Department and all the faculty members of the Computer Science and Engineering department for providing the necessary facilities and academic support.

## REFERENCES

- MongoDB Inc., *MongoDB Documentation*, Available: https://www.mongodb.com/docs
- Node.js Foundation, *Node.js Official Documentation*, Available: https://nodejs.org/en/docs
- Express.js, *Express – Web Framework for Node.js*, Available: https://expressjs.com

- Facebook Open Source, *React.js Documentation*, Available: https://react.dev
- R. Buyya, C. Vecchiola, and S. T. Selvi, *Mastering Cloud Computing*, McGraw Hill, 2013.
- A. S. Tanenbaum, *Computer Networks*, 5th Edition, Pearson Education, 2011.
- M. Fowler, *Patterns of Enterprise Application Architecture*, Addison-Wesley, 2002.
- IEEE, "Remote Laboratories in Engineering Education," *IEEE Transactions on Education*.
- REST API Tutorial, *RESTful Web Services*, Available: https://restfulapi.net
- W3C, *Web Technologies and Standards*, Available: https://www.w3.org
- I. Sommerville, *Software Engineering*, 10th Edition, Pearson Education, 2016.
- W. Stallings, *Data and Computer Communications*, 10th Edition, Pearson Education, 2014.
- E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1994.