CS331 - Computer Networks

Project Proposal

Smart Guard System

February 7, 2025

Indian Institute of Technology
Gandhinagar

TEAM:

- Aryan Sahu: 22110038
- Bhavik Patel: 22110047
- Hitesh Kumar: 22110098
- Jinil Patel: 22110184

Introduction

Maintaining an optimal learning environment is essential for student performance, instructor comfort, and overall operational efficiency in educational institutions. This project proposes developing a networked IoT-based monitoring system to track environmental conditions such as temperature, humidity, light levels, and air quality across classrooms and laboratories. A centralized dashboard will provide real-time visual insights into the environmental parameters of each monitored space, enabling proactive decision-making and improved resource utilization.

Objectives

The primary objectives of this project include:

1. **Data Transmission:** Implement a system to send sensor data from a microcontroller to a server endpoint over the internet using HTTP POST requests.
2. **Database Design:** Create a scalable database schema to store data from various sensors, accommodating additional classrooms or new sensors in the future.
3. **Data Insertion:** Develop a server-side script to receive sensor data at the endpoint and insert it into the database.
4. **Data Retrieval:** Build a server-side script or API to query the database and provide real-time data for the dashboard.
5. **Centralised Monitoring System**: Develop a user-friendly dashboard to display real-time sensor readings and historical trends.
6. **Alert Configuration:** Configure the system to generate automated notifications for threshold breaches of key environmental parameters.
7. **Scalability and Modularity**: Design an architecture that allows seamless integration of additional sensors and expansion to new rooms or buildings.

Hardware Components

To ensure accurate environmental monitoring, the system will incorporate the following sensors and hardware:

- **Temperature and Humidity Sensor:** DHT22/DHT11
- **Light Sensor:** BH1750
- **Air Quality Sensor:** MQ 135 sensor
- **Microcontroller:** ESP 32 module

Software Components

The project will utilise the following software technologies:

- **Programming Language**: Python/C/C++ for microcontroller interfacing.
- **Backend Framework**: Node.js with Express.js (to handle API requests and data storage)
- **Frontend Framework**: React.js (for an interactive and real-time dashboard)
- **Database**: MySQL/PostgreSQL (for time-series data storage and retrieval)
- **Communication Protocol**: MQTT/HTTP (for lightweight IoT data transmission)

System Workflow

1. Sensors in each room collect environmental data at regular intervals.
2. The microcontroller processes and transmits this data via Wi-Fi using the MQTT/HTTP protocol and stores the data in the database after processing through the backend.
3. The React-based frontend dashboard retrieves and visualizes this data in real-time.
4. Automated alerts are sent to the administrators if predefined thresholds (e.g., high CO2 levels) are exceeded.

Timeline and Milestones

The project will be completed over nine weeks as follows:

1. **Requirement Analysis (Week 1)**
   - Define sensor specifications, dashboard features, and user roles.
   - Determine database structure, communication protocols, and alert configurations.
   - Assign roles and tasks to team members.

2. **Hardware Setup (Week 2)**
   - Configure and test sensors for data collection.

- Set up the microcontroller and ESP32 module for data transmission.
- Conduct initial trials for reliable Wi-Fi connectivity.

3. **Backend Development (Weeks 3-4)**
   - Design and implement the database schema for storing sensor data.
   - Build backend API endpoints to receive, process, and store data.
   - Set up MQTT/HTTP protocol for lightweight data communication.
   - Ensure proper authentication for data insertion and retrieval.

4. **Frontend Development (Weeks 4-5)**
   - Develop an interactive dashboard using React.js.
   - Create visualizations (graphs, charts) to display real-time and historical data.
   - Implement alert messages and notifications for threshold breaches.

5. **System Integration (Weeks 6-7)**
   - Integrate hardware, backend, and frontend to validate end-to-end data flow.
   - Test data transmission reliability and consistency under various conditions.
   - Implement error handling and failover mechanisms.

6. **User Testing (Week 8)**
   - Gather feedback from potential users (instructors, administrators).
   - Improve UI/UX based on feedback.
   - Conduct performance testing to identify and fix bottlenecks.

7. **Final Optimization & Deployment (Week 9)**
   - Optimize database queries and API performance.
   - Finalize alert configurations.
   - Deploy the system for real-world use.

8. **Project Completion (Week 10)**
   - Document all technical aspects of the project.
   - Prepare and deliver the final project presentation and handover.

Expected Outcomes

Upon successful implementation, this system is expected to have the following functionalities:

- **Efficient Data Flow:** Smooth and real-time transmission of sensor data from microcontrollers to the server without data loss.
- **Flexible Database Structure:** A well-designed schema capable of handling diverse sensor data types and supporting future expansion needs.
- **Robust Data Handling:** Seamless server-side processing and accurate storage of incoming sensor information.
- **Dynamic Data Access:** Quick and secure access to current and historical sensor data through optimized API endpoints.
- **Comprehensive Dashboard:** Intuitive interface providing actionable insights through graphs, charts, and status indicators.
- **Future-Proof System:** A modular and scalable design ready to integrate new devices, rooms, or functionalities easily.

Expected Individual Contribution

- **Aryan Sahu:**
  - Leads backend development (API design).
  - Ensures proper data insertion, retrieval, and database optimization.
- **Bhavik Patel:**
  - Manages frontend development using React.js.
  - Develops the dashboard with real-time data visualizations and alerts.
- **Hitesh Kumar:**
  - Responsible for hardware setup and sensor interfacing with the microcontroller.
  - Ensures accurate sensor data collection and Wi-Fi transmission.
- **Jinil Patel:**
  - Oversees system integration and testing.
  - Ensures smooth data flow from sensors to the dashboard and handles performance optimization.

Conclusion

This IoT-based classroom and lab monitoring system will simplify environmental management by automating data collection and providing real-time insights. The system will help efficiently identify and address issues like poor air quality or temperature fluctuations. With its scalable and modular design, the project lays a strong foundation for adding new features or sensors in the future, making it a practical solution for educational institutions.