iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: IV Month of publication: April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80373

Smart IoT-Based Electric Vehicle Monitoring System

Chodavarapu Venkata Sai Jaganadha Swatesh[1], Angara Murali[2], Chelluboina Shalini[3], Kamana Dileep Sai Teja[4], Naga Sai Vallika[5], Pindi Aswini[6], Yatham S. S. Vara Prasad[7]
[1, 2, 3, 4, 5]*Department of Electrical and Electronics Engineering, Bonam Venkata Chalamayya Engineering College, Affiliated to JNTUK, Andhra Pradesh, India*
[6]*Project Guide Dr. S. Srikanth, Department of Electrical and Electronics Engineering, Bonam Venkata Chalamayya Engineering College, Affiliated to JNTUK, Andhra Pradesh, India*

Abstract: The rapid advancement of the Internet of Things (IoT) and web technologies has enabled the development of intelligent systems for real-time monitoring and data management. This project presents a Smart IoT-Based Electric Vehicle Monitoring System that integrates embedded hardware components with a robust backend software architecture for efficient system operation and remote accessibility. The hardware implementation consists of a microcontroller-based platform (NodeMCU/ESP8266 or Arduino UNO) interfaced with components such as a DHT11 temperature and humidity sensor, relay modules, DC motor, cooling fan, and GPS module. These components work together to continuously collect real-time environmental and battery-related parameters, ensuring accurate monitoring of the system without involving any control mechanisms.
Keywords: Internet of Things (IoT), Arduino UNO, NodeMCU (ESP8266), Embedded Systems, DHT11 Sensor, Relay Module, Automation System, Environmental Monitoring, Smart Control System, Spring Boot, REST API, Web Application.

I. INTRODUCTION

The rapid advancement of Internet of Things (IoT) technology has enabled the development of smart systems for real-time monitoring data management. This project presents a Smart IoT-Based Electric Vehicle Monitoring System that integrates hardware components such as Arduino UNO or NodeMCU (ESP8266), DHT11 temperature and humidity sensor, voltage sensor module, and LCD display to continuously monitor critical vehicle parameters. The system collects real-time data from sensors and processes it using embedded programming, while a backend developed using Java Spring Boot enables efficient data handling and communication through REST APIs. By combining IoT hardware with modern software technologies, the system reduces manual intervention, improves accuracy, and provides a scalable solution for real-time electric vehicle monitoring without manual intervention. On the software side, the system is developed using Java Spring Boot framework, which provides a scalable and efficient backend for data processing and communication. The backend follows a structured architecture consisting of controller, service, and model layers to handle data flow, business logic, and API interactions. RESTful APIs are used to enable communication between the hardware system and the backend server, allowing real-time data transmission, storage, and monitoring. The integration of IoT hardware with a Spring Boot-based backend ensures improved reliability, data accuracy, and system scalability. The proposed system eliminates the need for traditional manual monitoring and provides a centralized platform for automated control and data management. This system is cost-effective, flexible, and suitable for applications such as smart home automation, environmental monitoring. Overall, the project demonstrates an efficient combination of embedded systems and modern web technologies to achieve intelligent automation.

A. Problem Statement

The existing monitoring systems face several challenges due to lack of real-time data access and efficient monitoring mechanisms.
The major problems are:

1) Lack of real-time monitoring of battery parameters such as temperature, humidity, and voltage
2) Inefficient use of energy and resources
3) High dependency on manual checking and traditional monitoring methods
4) Delayed identification of abnormal battery conditions
5) No centralized system for data management.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

These limitations create the need for an automated IoT-based system that can monitor and control devices efficiently.

II. RELATED WORK

A. Existing Approaches

Earlier electric vehicle monitoring systems were primarily based on standalone embedded platforms without internet connectivity. These systems supported only local monitoring of basic parameters and required manual observation, making them inefficient and unsuitable for real-time applications. Most of these approaches utilized simple microcontrollers integrated with sensors to measure parameters such as temperature and voltage, but lacked remote accessibility and continuous data tracking.

Some systems introduced limited improvements by incorporating sensors and basic data logging mechanisms, but they did not provide proper communication frameworks or centralized data management. As a result, these systems were restricted in functionality and were unable to support advanced features such as real-time monitoring, remote data visualization, and scalable IoT-based integration.

III. PROPOSEDSYSTEM

A. System Architecture

The proposed system follows **a** three-tier architecture consisting of hardware, communication, and application layers.

1) Hardware Layer: It includes components such as Arduino UNO or NodeMCU (ESP8266), DHT11 sensor, relay modules, DC motor, cooling fan, and GPS module. It is responsible for collecting environmental data and performing control operations based on the processed results.
2) Communication Layer: This layer enables data transfer between the hardware and the backend system. The NodeMCU sends real-time sensor data to the server using internet connectivity and REST API communication.
3) Application Layer: Developed using Java Spring Boot, this layer handles data processing, business logic, and system control. It manages REST APIs, processes incoming data, and ensures proper communication between the hardware and the database.

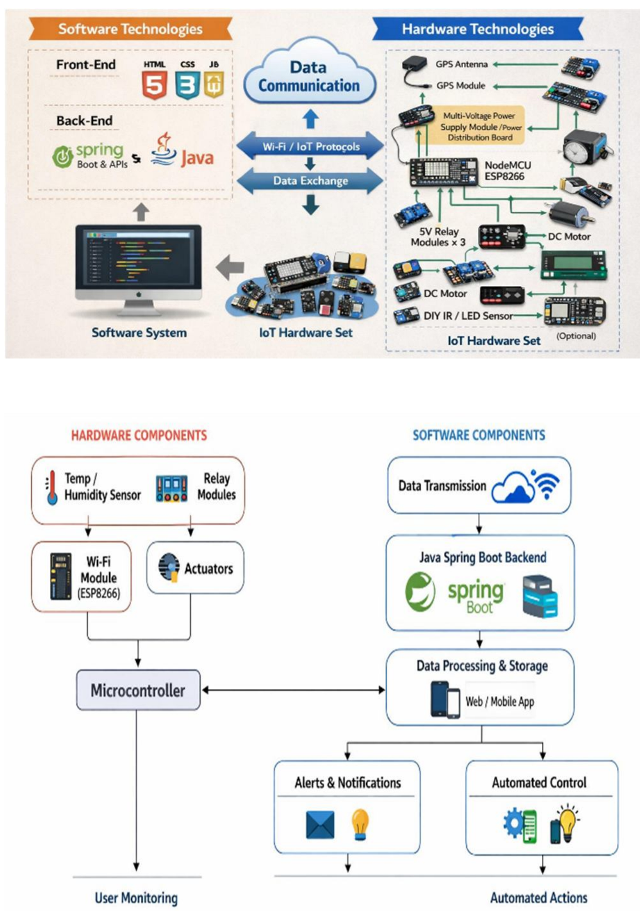


Fig 1 : Unified Architecture of Smart IoT-Based Electric Vechile Monitoring System.

B. Module D

Modules of the System

1) Sensor Module

This module is responsible for collecting environmental data using sensors.

- Measures temperature and humidity using DHT11
- Provides real-time data to the system
- Ensures continuous monitoring

2) Processing and Control Module

This module handles data processing and device control.

- The microcontroller (Arduino UNO or NodeMCU ESP8266) processes the sensor data
- Applies predefined conditions
- Controls devices like DC motor and cooling fan using relay modules

3) Communication Module

This module enables interaction between hardware and backend system.

- Transfers data from NodeMCU to server
- Uses internet connectivity and REST APIs
- Supports real-time data communication

4) User and Admin Module

This module provides system access and monitoring capabilities to:

- Users can monitor environmental data and device status
- Admin can manage system operations
- Ensures proper control and supervision

5) Backend and Database Management Module

This module manages software processing and data storage.

- Developed using Java Spring Boot
- Handles APIs and business logic
- Stores sensor data and system information in the database

C. System Implementation

(IoT with Java Spring Boot)

The proposed system is implemented by integrating IoT hardware components with a backend developed using Java Spring Boot. The hardware setup includes NodeMCU (ESP8266), DHT11 sensor, relay modules, DC motor, cooling fan, and GPS module. The NodeMCU is programmed using embedded C through Arduino IDE to read sensor data and control devices based on predefined conditions.

The NodeMCU collects real-time environmental data and sends it to the backend server through HTTP requests using REST APIs. On the software side, the system is developed using the Spring Boot framework, which provides a structured architecture consisting of controller, service, and model layers.

These components handle data processing, API communication, and system logic efficiently. The backend interacts with the database to store sensor data and device status for monitoring and analysis. The integration of hardware and software ensures real-time operation, reliable communication, and efficient system performance.

IV. RESULT AND DISCUSSION

The proposed system was tested under various conditions and performed effectively. The DHT11 sensor accurately measured environmental data, and the Arduino UNO or NodeMCU processed it to control devices like the fan and motor using relay modules. The data was successfully transmitted to the backend using REST APIs, and the Spring Boot application handled it efficiently. The system showed reliable performance with real-time monitoring and reduced manual effort.

The system proved to be reliable, scalable, and suitable for real-time deployment in smart automation environments.

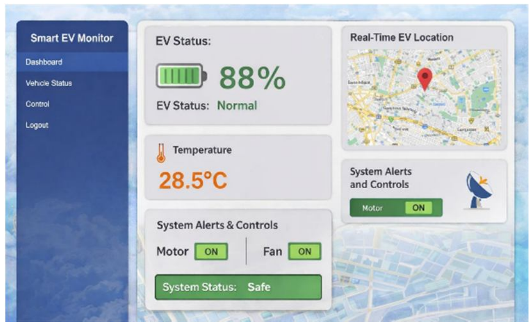


Fig.3. Result

V. ACKNOWLEDGEMENT

We sincerely express our gratitude to our project guide Dr. S. Srikanth for their valuable guidance, support, and encouragement throughout the development of this project. We also thank the faculty members of our department for their continuous assistance and suggestions. We are thankful to our institution and management for providing the necessary facilities and resources to successfully complete this project. We also extend our appreciation to our friends for their support and cooperation during the project work. Finally, we express our heartfelt thanks to our family members for their constant encouragement and motivation.

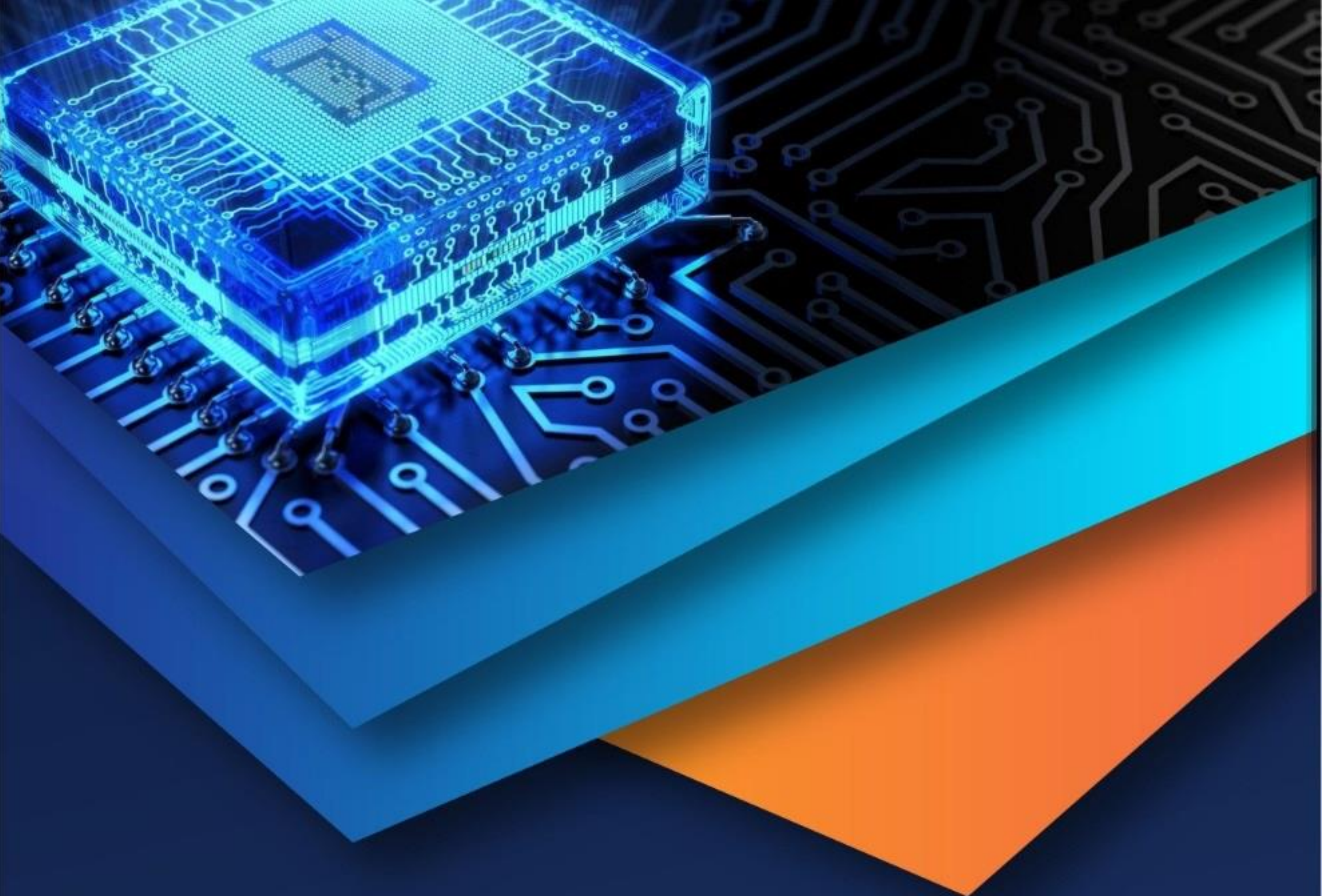
VI. CONCLUSION

The Smart IoT-Based Electric Vechile Monitoring System provides an efficient solution for real-time monitoring and automatic device control using IoT technology. By integrating hardware components with a Java Spring Boot backend, the system ensures reliable communication, accurate data processing, and reduced manual effort. The system reduces manual effort and enhances battery safety by enabling continuous monitoring and early detection of abnormal conditions. It is cost-effective, scalable, and suitable for modern electric vehicle applications, with future scope for enhancements such as cloud integration, mobile application support, and advanced data analytics.

REFERENCES

[1] A. Bhaga and V. Madisetti, Internet of Things: A Hands-On Approach, Universities Press, 2015.
[2] D. Hanes, G. Salgueiro, P. Grossetete, R. Barton, and J. Henry, IoT Fundamentals: Networking Technologies, Protocols, and Use Cases, Cisco Press, 2017.
[3] O. Hersent, D. Boswarthick, and O. Elloumi, The Internet of Things: Key Applications and Protocols, Wiley, 2012.
[4] A. S. Tanenbaum and D. J. Wetherall, Computer Networks, 5th ed., Prentice Hall, 2011.
[5] R. T. Fielding, "Architectural Styles and the Design of Network-Based Software Architectures," Ph.D. dissertation, University of California, Irvine, 2000.
[6] Spring, "Spring Boot Reference Documentation," Available: https://spring.io/projects/spring-boot
[7] Espressif Systems, "ESP8266EX Datasheet," 2020.
[8] Arduino, "Arduino Documentation," Available: https://www.arduino.cc
[9] Aosong Electronics, "DHT11 Temperature and Humidity Sensor Datasheet," 2018.
[10] R. S. Pressman and B. R. Maxim, Software Engineering: A Practitioner's Approach, 8th ed., McGraw-Hill, 2015.
[11] I. Sommerville, Software Engineering, 10th ed., Pearson, 2016.
[12] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994.
[13] M. Fowler, Patterns of Enterprise Application Architecture, Addison-Wesley, 2002.
[14] G. Booch, J. Rumbaugh, and I. Jacobson, The Unified Modeling Language User Guide, Addison-Wesley, 1999.
[15] H. F. Korth, A. Silberschatz, and S. Sudarshan, Database System Concepts, 6th ed., McGraw-Hill, 2011.
[16] R. Elmasri and S. B. Navathe, Fundamentals of Database Systems, 7th ed., Pearson, 2016.
[17] D. Comer, Internetworking with TCP/IP, Prentice Hall, 2006.
[18] J. Postel, "Transmission Control Protocol," RFC 793, 1981.
[19] Fielding et al., "Hypertext Transfer Protocol – HTTP/1.1," RFC 2616, 1999.
[20] K. Ashton, "That 'Internet of Things' Thing," RFID Journal, 2009.

[21] V. K. Garg, Wireless Communications and Networking, Morgan Kaufmann, 2007.
[22] S. Monk, Programming Arduino: Getting Started with Sketches, McGraw-Hill, 2011.
[23] P. P. Ray, "A Survey on Internet of Things Architectures," Journal of King Saud University, 2018.
[24] L. Bass, P. Clements, and R. Kazman, Software Architecture *in Practice*, Addison-Wesley, 2012.
[25] IEEE, "IEEE Standard for Software Requirements Specifications," IEEE Std 830-1998.



10.22214/IJRASET

INDEX COPERNICUS 45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY