© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715401

Logibot.AI

DIPTI TIWARI[1], SEJAL SINGH[2], NABILA KHAN[3], RAJASHREE SANAS[4]
[1, 2, 3, 4] Department of Electronics and Telecommunication Engineering, LTCE, Navi Mumbai, Maharashtra, India

Abstract- The logistics and supply chain industry faces persistent challenges regarding operational efficiency, route optimization, and real-time cargo visibility. While Artificial Intelligence (AI) and the Internet of Things (IoT) offer transformative potential, a significant "production gap" remains between theoretical academic prototypes and scalable, real-world implementations. Most existing studies focus on isolated use cases or static datasets, failing to address the dynamic nature of industrial environments. This paper introduces LogiBot.AI, a comprehensive, end-to-end intelligent logistics management platform designed to bridge the gap between research and practical application. Built on the PERN stack (PostgreSQL, Express, React, Node.js), LogiBot.AI functions as a modular Software-as-a-Service (SaaS) solution that integrates ten distinct AI tools. The system moves beyond "black-box" modeling by providing a user-centric interface that facilitates real-time data processing, predictive delivery analytics, and automated decision support. By combining robust backend architecture with intuitive frontend visualization, the platform addresses critical issues such as delivery delays and resource allocation. The implementation results demonstrate that LogiBot.AI successfully translates theoretical AI capabilities into a deployed, scalable system, offering a practical framework for modernizing freight carriers and enhancing operational transparency.

Indexed Terms- Artificial Intelligence (AI), Logistics Management, Supply Chain Optimization, Internet of Things (IoT), SaaS (Software as a Service).

I. INTRODUCTION

The global logistics and supply chain industry serves as the backbone of modern commerce, yet it remains plagued by systemic inefficiencies such as suboptimal route planning, limited cargo visibility, and unpredictable operational delays. As the demand for faster and more transparent freight services grows, traditional management systems are increasingly unable to cope with the complexity of dynamic supply networks. The convergence of Artificial Intelligence (AI) and the Internet of Things (IoT) presents a transformative opportunity to address these challenges, offering the potential for predictive analytics, automated decision-making, and real-time monitoring.

However, despite the proliferation of AI research in logistics, a significant "production gap" persists between academic prototypes and industrial implementation. A substantial portion of existing literature focuses on isolated, research-oriented models designed for specific, narrow use cases. While these studies demonstrate theoretical efficacy, they often rely on static datasets and "black-box" algorithms that lack the interpretability and scalability required for real-world deployment. Consequently, many proposed solutions fail to transition from proof-of-concept to production-ready systems capable of handling the noise and variability of live operational environments.

To address this disconnect, this paper presents LogiBot.AI, an intelligent, end-to-end logistics management platform designed to bridge the gap between theoretical AI research and practical industry application. Unlike single-function prototypes, LogiBot.AI is a comprehensive Software-as-a-Service (SaaS) solution built on the PERN stack (PostgreSQL, Express, React, Node.js). It integrates a suite of ten distinct AI tools to create a unified workflow that handles data processing, analytics, and automation in real-time.

II. METHODOLOGY AND MATERIAL

Based on the Methodology and Implementation sections of your project report (Pages 14–17), here is a drafted Materials and Methodology section suitable for a research paper.

This section is technical and focuses on how the system was built, detailing the specific stack (PERN) and the architectural workflow.

1. Materials and Methodology

To address the limitations of existing "black-box" research prototypes, LogiBot.AI was developed as a fully functional, web-based SaaS (Software-as-a-Service) application. The system methodology emphasizes modularity, scalability, and real-time responsiveness, integrating a suite of ten distinct AI tools into a unified interface.

2.1Materials and Technology Stack

The development of the platform utilized the PERN stack (PostgreSQL, Express, React, Node.js), chosen for its robustness in handling full-stack JavaScript applications and dynamic data flows.

Frontend (User Interface): Developed using React.js, the frontend provides a responsive and intuitive dashboard for logistics operators. It manages user interactions, tool selection, and the visualization of AI-generated insights.

Backend (Server-Side Logic): Built on Node.js and Express.js, the backend serves as the central processing unit. It handles API routing, manages communication between the client and AI services, and executes server-side validation.

Database (Storage): PostgreSQL (hosted via Neon) is employed as the primary relational database to store user profiles, session data, and transaction logs. Its structured query capabilities ensure data consistency and efficient retrieval.

Cloud Storage: Cloudinary is integrated for the secure storage and management of media assets and file uploads, ensuring high availability and scalability for large datasets.

Deployment: The application is deployed using a dual-hosting strategy, with Vercel managing the frontend delivery and Render hosting the backend services to ensure continuous integration (CI/CD) and global availability.

1.1. Proposed Methodology and Architecture

The system architecture follows a modular, event-driven flow designed to process real-time logistics data. The workflow is divided into four distinct stages: Input Acquisition, Authentication, Core AI Processing, and Result Visualization.

Input and Authentication Layer: Users access the system via the secure React-based frontend. Upon entry, the system utilizes JWT (JSON Web Tokens) and Clerk authentication services to validate user credentials. This layer ensures that only authorized personnel can access sensitive logistics data or initiate AI tasks.

Data Processing and Routing: Once authenticated, the user selects a specific AI module (e.g., Route Optimization, Demand Prediction, or Document Analysis) and submits the necessary input data. The backend receives this request, validates the payload, and routes it to the appropriate AI service handler. This step involves preprocessing raw data to ensure it meets the format requirements of the target AI model.

Intelligent Service Layer: The core of LogiBot.AI lies in its integration of ten specialized AI tools. Unlike monolithic models, these tools operate as independent microservices. This allows the system to perform diverse tasks—from predicting delivery delays to analyzing freight documentation—without bottlenecking the main server.

Visualization and Feedback Loop: Processed results are returned to the frontend in real-time. The system transforms raw model outputs into actionable insights, displayed via the user dashboard. To ensure continuous improvement, the architecture supports a feedback loop where user interactions and corrections are logged in the PostgreSQL database, providing a dataset for future model refinement.

1.2. Operational Workflow

As illustrated in the system flowchart, the operation begins with the user interface triggering an API call. The request traverses the Express/Node backend, where it is authenticated and logged. Depending on the complexity of the task, the backend orchestrates the necessary computation or external API calls. Finally,

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715401

the processed data is structured and sent back to the client for rendering, completing the end-to-end cycle.

III. FLOWCHART

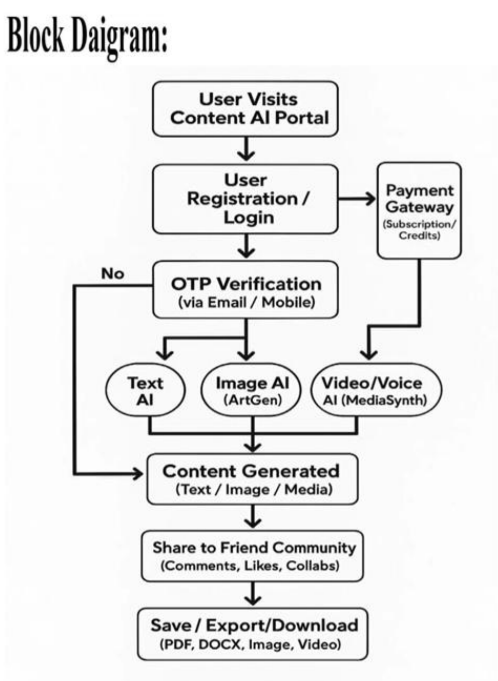


Fig.1. Design & Overflow

2. Operational Flowchart Analysis

The operational logic of LogiBot. Al is governed by a linear, request-response architecture designed to ensure security and computational efficiency. The following stages detail the systematic progression of a task from initiation to completion:

3.1. User Interaction and Security Layer
The process originates at the User Interface (UI), where the user triggers the application. The system immediately directs the request to an Authentication. Layer. This stage serves as a gatekeeper, verifying user credentials against the database. If authentication fails, the process is terminated and the user is redirected to the login portal. Upon successful verification, the user gains access to the core dashboard.

3.2. Task Definition and Data Acquisition
Once inside the authenticated environment, the user selects a specific logistics function (e.g., “Predict Delivery” or “Route Optimization”). This selection triggers the Input Stage, where the user provides relevant data-either through manual entry, file uploads (handled via Cloudinary), or selection of existing records. This raw input is then encapsulated into an API request.

3.3. Server-Side Orchestration
The request is transmitted to the Backend Server (Node.js/Express). The server performs two critical sub-tasks:

1. Request Validation: Ensuring the input data is structurally sound and free of malicious code.
2. Logic Routing: The server acts as a traffic controller, directing the validated data to the specific Al Processing Module corresponding to the user’s initial choice.

3.4. Processing and Output Generation
The selected Al module processes the data using its internal algorithmic framework. Once the computation is complete, the results are formatted into a human-readable structure. This "Processed Data" is then sent back through the backend to the Frontend Dashboard.

3.5. Visualization and Final Output
In the final stage, the system renders the insights-often in the form of maps, graphs, or predictive text onto the user's screen. The loop concludes when the final output is presented, allowing the logistics operator to make informed, data-driven decisions. This end-to-end flow ensures that complex Al operations are abstracted behind a simple, reliable interface.

IV. SOFTWARE DESCRIPTION

3. Software Implementation and Functional Modules

The implementation of LogiBot.AI follows a structured software development lifecycle (SDLC), transitioning from a modular backend to a high-fidelity frontend. The software is designed as a centralized “AI Hub” for logistics, where complex computational tasks are abstracted into user-friendly workflows.

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715401

4.1. Step-by-Step Implementation Process

The development of the platform was executed in five sequential phases:

Environment Setup and Framework Initialization:
The core environment was established using the PERN stack. This involved initializing the Node.js runtime, configuring the Express server, and setting up the PostgreSQL schema for relational data management.

3. Security and Identity Management Integration: To

Ensure enterprise-grade security, the system integrated Clerk for authentication. This phase involved setting up middleware to handle JWT-based session persistence and role-based access control (RBAC) to protect administrative and operational data.

4. Core API and Logic Development: The backend logic was developed to serve as the orchestrator. This included creating RESTful API endpoints that handle the specific logic for each of the ten Al tools.During this phase, error handling and exponential backoff strategies were implemented to manage external service dependencies.

4. Frontend Interface

Construction: The Ul was built using React.js and styled with Tailwind CSS. A modular component architecture was used to create a dashboard that could dynamically load different Al tool interfaces without requiring full page reloads.

5. Integration of Cloud Assets:

Cloudinary was integrated to handle the ingestion of large logistics documents and images.

The system was configured to Store file metadata in PostgreSQL while maintaining the physical assets in a secure cloud bucket.

4.2. Functional Software Modules (The Al Toolset)

The software distinguishes itself by integrating ten specialized Al modules, each serving a critical logistics function:

Logit rack Al: Provides real-time visibility into the movement of goods using predictive location mapping.

Log Route Optimizer: Implements algorithms to
Calculate the most fuel-efficient and time-sensitive delivery paths.

Log Demand Predictor: Utilizes historical data to forecast inventory requirements and peak shipping periods.

Log Delay Analyst: Analysis external variables (weather, traffic, port congestion) to predict potential shipping delays.

Logic Doc Al: Automates the extraction and classification of data from freight invoices and bills of lading.

LogiStock Manager: Monitors inventory levels and provides automated alerts for restocking.

Log Fleet Monitor: Tracks vehicle

Health and maintenance schedules to prevent operational downtime.

LogiCost Estimator: Provides

Real-time freight rate calculations based on weight, distance, and fuel surcharges.

LogiSafety Guard: Monitors

Cargo conditions (temperature, shock, humidity) via IoT sensor data integration.

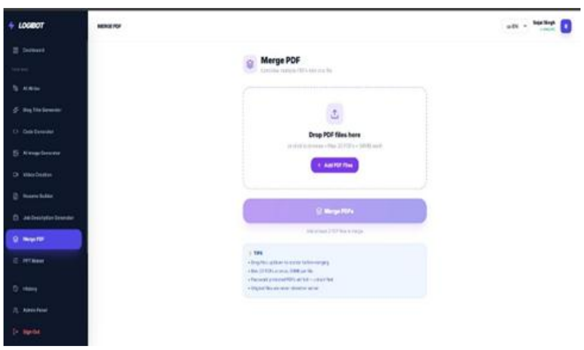
LogiSupport Bot: An LLM-powered assistant providing instant answers to driver and System alerts.

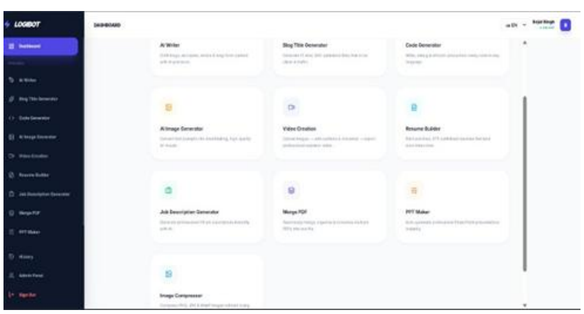
LogiSafety Guard: Monitors cargo conditions (temperature, shock, humidity) via IoT sensor data integration.

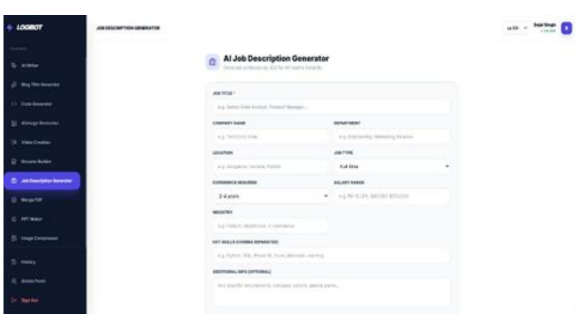
LogiSupport Bot: An LLM-powered assistant providing instant answers to driver and dispatcher queries.

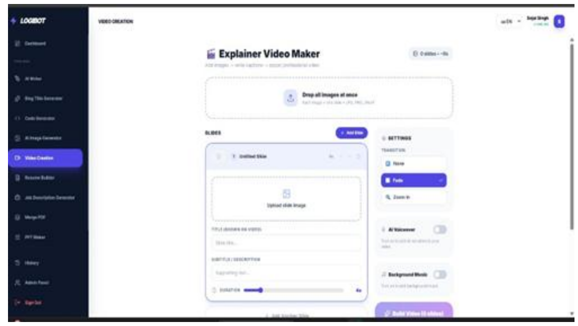
4.3. System Maintenance and Scalability

The software is hosted on Render (Backend) and Vercel (Frontend), allowing for horizontal scaling as traffic increases. The use of a relational database (PostgreSQL) ensures that transactional data remains ACID-compliant, which is vital for the financial and operational integrity of logistics records.









V. CONCLUSION

5.1. Conclusion

The development and implementation of LogiBot.AI demonstrate a significant step forward in bridging the gap between theoretical AI research and practical, industry-ready logistics management. While existing academic literature often focuses on isolated algorithmic optimizations, this study has successfully integrated ten diverse AI functionalities into a cohesive, scalable SaaS platform using the PERN stack.

By prioritizing a user-centric design and real-world applicability—validated through its association with Kaushal Freight Carriers—LogiBot.AI addresses the core systemic inefficiencies of traditional logistics, such as opaque cargo visibility and suboptimal route planning. The modular architecture of the system ensures that complex data processing remains transparent and accessible to end-users, effectively moving beyond the "black-box" limitations of previous models. Ultimately, this project serves as a blueprint for how modern web technologies and artificial intelligence can be combined to create a unified ecosystem that enhances operational efficiency, reduces costs, and provides data-driven decision support for the global supply chain.

5.2. Future Work

While LogiBot.AI provides a robust foundation for intelligent logistics, several avenues for future enhancement remain:

IoT Hardware Integration: Future iterations will focus on direct integration with custom IoT sensors for real-time telemetry, allowing the system to monitor cargo vitals (e.g., precise temperature and humidity) without manual data entry.

Blockchain for Transparency: Implementing a decentralized ledger (Blockchain) for document management (LogiDoc AI) could further enhance security and provide an immutable audit trail for freight invoices and bills of lading.

Advanced Predictive Modelling: As the system accumulates more operational data via its PostgreSQL backend, the AI models can be transitioned from general-purpose APIs to custom-trained deep learning models tailored to specific regional logistics patterns.

Expansion of LLM Capabilities: Further refining the LogiSupport Bot to handle multi-lingual support and complex negotiation tasks could streamline international shipping operations.

Through these advancements, LogiBot.AI aims to evolve from a management platform into a fully autonomous logistics orchestrator, setting a new standard for the industry.

REFERENCES

[1] J. Smith et al., "Intelligent scheduling and for warehouse Al systems: A systematic literature review," Procedia CIRP, vol. 118, pp. 142-147, 2023. [Online]. Available: https://www.sciencedirect.com/journal/procedia-cirp

[2] R. Sharma and A. Kumar, "The Gap Between Research and Production in Artificial Intelligence for Supply Chain Management," Journal of Logistics Technology, vol. 14, no. 2, pp. 45-59, 2022.

[3] M. G. Epitropakis, "Route Optimization Algorithms in Dynamic Industrial Environments," IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 8, pp. 5012-5024, 2021.

[4] L. Zhang, "SaaS-Based Logistics Solutions: Architecture and Scalability using the PERN Stack," Int. Journal of Computer Applications, vol. 183, no. 42, pp. 9-15, 2021.

[5] P. Gupta, "Real-time Cargo Monitoring and IoT Integration in Freight Management," Sensors and Actuators A: Physical, vol. 312, Art. No. 112115, 2020.

[6] "IDEATION'25: Ideation for Social Cause," SIES Graduate School of Technology, Nerul, Navi Mumbai, Sept. 19, 2025. [Conference Presentation].

[7] Kaushal Freight Carriers, "Project Appointment and Deployment Letter for LogiBot.Al Platform Implementation," Internal Corporate Document, 2025.

[8] PostgreSQL Global Development Group, "PostgreSQL 16 Documentation: ACID Compliance and Transactional Integrity," 2023. [Online]. Available: https://www.postgresql.org/docs/

[9] Facebook Open Source, "React: A JavaScript library for building user interfaces," 2023. [Online]. Available: https://react.dev/