IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2026

DOI: 10.17148/IARJSET.2026.13257

WEB-BASED TRUCK BIDDING PLATFORM

Hariprasath G[1], Mrs. A. Sathiya Priya[2]

Department of Information Technology, Dr. N.G.P. Arts and Science College, Coimbatore, Tamil Nadu, India[1]

Assistant Professor, Department of Information Technology, Dr. N.G.P. Arts and Science College, Coimbatore, Tamil Nadu, India[2]

Abstract: The rapid expansion of global supply chains and the increasing demand for efficient transportation services have significantly transformed the logistics industry. Traditional freight management systems rely heavily on manual coordination, telephonic negotiations, and fragmented data handling mechanisms, which often lead to pricing inefficiencies, underutilization of fleet resources, delayed shipments, and limited operational transparency. This paper presents a web-based truck bidding platform named OptiFreight Logistics, designed to digitally connect shippers and truck owners within a unified ecosystem. The system is developed using modern web technologies including React, TypeScript, and Vite, and integrates AI-assisted components to enhance operational intelligence. The platform enables real-time freight posting, competitive bidding by truck owners, automated bid evaluation, fleet monitoring, and performance analytics visualization. Through role-based access control and structured workflow management, the system improves transparency, reduces manual intervention, optimizes truck allocation, and enhances decision-making efficiency. The proposed solution demonstrates how intelligent web-based platforms can modernize freight coordination and provide scalable solutions for digital logistics transformation.

Keywords: Smart Logistics, Freight Management System, Fleet Optimization, Real-Time Bidding, AI in Logistics, Web-Based Transportation Platform.

I. INTRODUCTION

The movement of freight is the backbone of the global economy. However, the domestic trucking industry remains largely decentralized. Small fleet owners and independent owner-operators, who make up the majority of the market, often lack access to the same technological tools as major carriers. Consequently, shippers are forced to rely on intermediaries (brokers) to find capacity, often paying high premiums for a service that lacks transparency.

OptiFreight Logistics was developed to address these specific challenges. By creating a direct marketplace, the platform eliminates the need for traditional middlemen. Shippers can post load requirements directly to a network of vetted carriers, while truck owners can bid on loads that match their specific vehicle types and routes. Furthermore, the integration of Artificial Intelligence transforms the platform from a passive bulletin board into an active decision-support system. Users can interact with the system using natural language to query order statuses, analyse market rates, and optimize their schedules.

II. SYSTEM ANALYSIS

2.1 Existing System

The current landscape of freight management is still heavily influenced by legacy processes that limit efficiency, transparency, and scalability. One of the primary challenges is manual negotiation, where freight transactions are commonly finalized through phone calls, text messages, or email exchanges. This approach lacks proper documentation and audit trails, making dispute resolution difficult and slowing down the deal closure process. The absence of structured digital workflows also increases the likelihood of miscommunication and operational delays.

Another major issue is information asymmetry within the logistics ecosystem. Shippers often lack visibility into the real-time market rate for specific routes (lanes), while truck owners or drivers may not fully understand the true value or urgency of the cargo they are transporting. This imbalance can result in unfair pricing practices, inconsistent bidding, and reduced trust between stakeholders. Furthermore, many existing digital platforms function merely as static load boards, displaying lists of available freight without enabling integrated bidding, assignment, or tracking capabilities. As a result, users are forced to revert to manual communication methods to finalize transactions, undermining the benefits of digitalization and limiting overall operational efficiency.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2026

DOI: 10.17148/IARJSET.2026.13257

2.2 Proposed System

OptiFreight Logistics is developed as a unified Single Page Application (SPA) that digitizes and streamlines the entire freight transaction lifecycle, from load posting and bidding to assignment and delivery monitoring. The platform ensures a seamless and responsive user experience by enabling real-time updates without page reloads, which is critical for time- sensitive logistics operations. Its Dynamic Bidding Engine allows shippers to instantly receive and compare bids based on essential parameters such as price, estimated delivery time, and carrier ratings, ensuring transparency and informed decision-making. At the same time, truck owners can bid strategically on loads that match their vehicle type, route preferences, and operational capacity, thereby improving profitability and minimizing empty backhauls. The system fosters healthy competition among carriers while maintaining fairness and clarity in pricing.

To strengthen operational efficiency, the platform includes an Asset Utilization module that dynamically manages fleet availability. When a truck is assigned to a freight order, its status automatically changes to BUSY, preventing duplicate allocations and reducing scheduling conflicts; once the delivery is completed, the status reverts to AVAILABLE. Beyond operational tracking, OptiFreight Logistics integrates Intelligent Analytics powered by Google Gemini to convert raw transactional data into actionable insights. Users can interact with the system using natural language queries to analyse shipping costs, demand patterns, and performance trends. For example, if a user questions a rise in shipping expenses, the AI evaluates bidding activity, route distances, and market demand fluctuations to provide a contextual explanation. This combination of real-time marketplace functionality and AI- driven analysis enhances transparency, efficiency, and strategic decision-making across the logistics ecosystem.

Feature	Existing System (Legacy)	Proposed System (OptiFreight)
Negotiation	**Manual:** Conducted via phone/email; no audit trail.	**Dynamic:** Real-time bidding interface with instant acceptance.
Pricing	**Asymmetric:** Lack of market rate transparency; prone to gouging.	**Transparent:** Market-driven competitive bidding based on ratings/price.
Load Management	**Static:** Simple lists that don't facilitate assignments.	**Integrated:** Digitized lifecycle from bidding to tracking.
Asset Tracking	**Manual/Prone to Error:** Risk of double-booking.	**Automated:** Real-time "BUSY" status updates for truck owners.
Analytics	**Passive:** Basic reports with no context.	**Intelligent:** AI-powered (Gemini) contextual data interpretation.

Table 1 Existing System VS Proposed System

2.3 Hardware and Software Requirements

To ensure high performance, scalability, and maintainability, the project adopts a modern and efficient technology stack.

Software Requirements:

- Frontend Library: React 19.2.3 (leveraging Concurrent Mode for optimized rendering and smooth UI updates).
- Programming Language: TypeScript 5.8.2 (ensuring type safety and improved code reliability).
- Build Tool: Vite 6.2.0 (providing fast Hot Module Replacement and optimized production builds).
- AI Service: Google Generative AI SDK (Gemini) for contextual data interpretation and conversational analytics.
- Visualization Library: Recharts 3.6.0 for interactive charts and performance dashboards.
- Styling Framework: Tailwind CSS for responsive, utility-first UI design.

Hardware Requirements:

- Client: Any standard computing device such as a Laptop, Tablet, or Smartphone with a modern web browser (Chrome, Edge, Firefox, etc.).
- Server: Node.js runtime environment (version 18 or above) to support development, dependency management, and production builds.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2026

DOI: 10.17148/IARJSET.2026.13257

Component	Technology	Version
Frontend Library	React	19.2.3
Language	TypeScript	5.8.2
Build Tool	Vite	6.2.0
AI Integration	Google Gemini	Generative AI SDK
Visualization	Recharts	3.6.0
Styling	TailwindCSS	Responsive Utility-first Design

Table 2 Hardware and Software Requirements

III. SYSTEM DESIGN AND ARCHITECTURE

3.1 Architecture Overview

The system is designed using a modular, component-based architecture characteristic of modern React applications, ensuring scalability, maintainability, and a clear separation of concerns. The architecture is structured into three primary layers: the Presentation Layer (UI), the Service/Controller Layer (Application Logic), and the Data/Intelligence Layer (State Management and AI Processing). Each layer performs a distinct and well-defined responsibility, allowing the application to remain organized and extensible as new features are introduced. This layered approach simplifies debugging, testing, and future upgrades, since changes in one layer can be implemented with minimal impact on others. By adopting reusable components and structured state handling, the system maintains consistency across modules while supporting real-time updates and interactive workflows essential for freight management operations.

The Presentation Layer consists of React functional components such as Dashboard, MyFreightOrdersView, and Trucks View, which are designed to be declarative and reusable. These components render UI elements based solely on received props and do not directly manipulate global state, ensuring predictable behaviour. The Service/Controller Layer is centralized within App.tsx, which acts as the core orchestrator of the platform by managing global state variables such as freight orders, registered trucks, and active user sessions. It also controls navigation between modules through conditional rendering and event-driven updates. The Data/Intelligence Layer incorporates the Smart Assistant module, which integrates with the external Gemini API to deliver AI-powered insights. This layer securely transmits sanitized JSON representations of transactional data to the API and processes structured responses to generate meaningful recommendations, analytical explanations, and contextual assistance. Together, these layers create a robust, intelligent, and maintainable architecture capable of supporting complex logistics workflows.

Layer Name	Component / File	Description	Responsibility
Presentation Layer	Dashboard, My Freight Orders View, Trucks View	Built using React functional components. These components are pure and render UI based on props.	Displays data to users and handles user interface interactions.
Controller Layer	App.tsx	Acts as the central controller of the application. Manages global state including orders, trucks, and current User session.	Handles routing, state management, and coordination between modules.
Intelligence Layer	Smart Assistant Component	Integrates AI functionality through Gemini API. Sends sanitized JSON order data for processing.	Generates human- readable insights and analytics based on freight data.
Data Layer	Orders Array, Trucks Array, User Session	Maintains application state using React state management principles.	Stores and updates application data dynamically.

Table 3 Architecture Overview

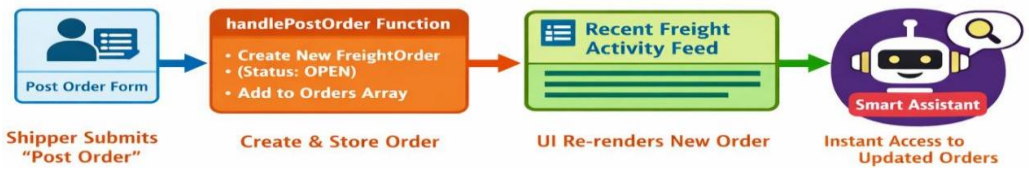

Figure 1 Architecture Overview

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2026

DOI: 10.17148/IARJSET.2026.13257

3.2 Data Flow

When a Shipper submits the "Post Order" form, the application initiates a structured validation process to ensure that all required input fields—such as origin, destination, cargo type, weight, and delivery date—are correctly populated and formatted. Once validation is successfully completed, the system invokes the handlePostOrder function, which encapsulates the core business logic for order creation. This function constructs a new FreightOrder object, assigns it a default status of OPEN, generates a unique identifier, and appends it to the centralized orders array maintained within the application state. Because the platform operates as a Single Page Application with reactive state management, this update immediately triggers a UI re-render. As a result, the newly created freight order appears instantly in the "Recent Freight Activity" feed and marketplace view without requiring a page refresh. In cases where backend confirmation fails—due to network issues or server-side validation errors—the system activates structured error-handling mechanisms that display user-friendly notifications while preserving data consistency and preventing duplicate entries.

Since the orders array is part of the application's global state, any modification automatically propagates across all dependent components. This ensures that the SmartAssistant module has immediate access to the most recent order data without requiring additional synchronization processes. Consequently, the AI engine can instantly analyse newly posted freight details to provide contextual recommendations, route suggestions, or bidding insights. Furthermore, real-time state updates guarantee that Truck Owners accessing the marketplace view can see and respond to the new order without delay, fostering timely bidding activity and competitive pricing. This tightly integrated workflow between user input, state management, UI rendering, and AI intelligence ensures a seamless, responsive, and data- consistent operational environment throughout the freight transaction lifecycle.

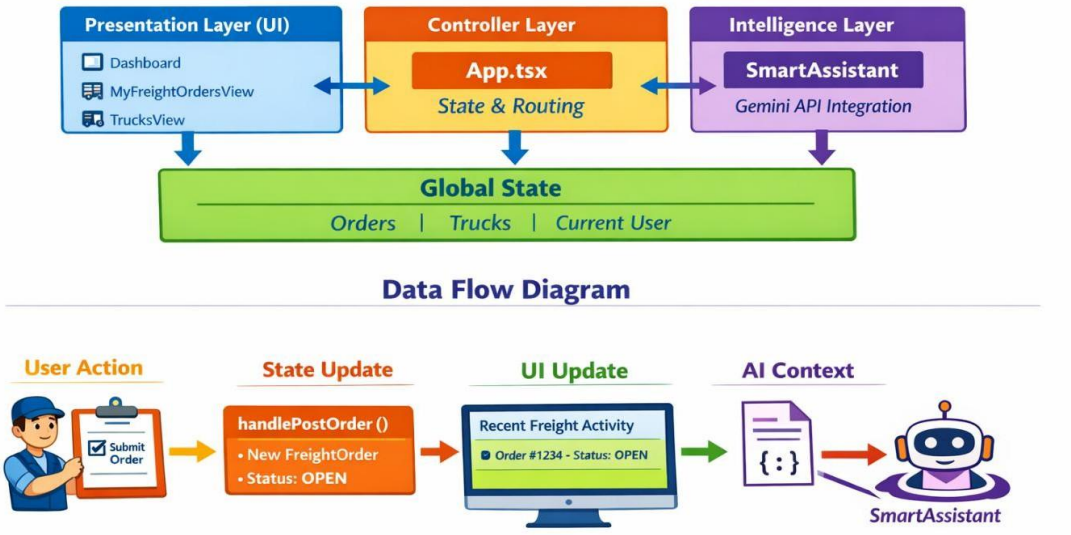

Figure 2 Data Flow Diagram

Step No.	Process Stage	Actor/Function	Description	Output
1	**User Action**	Shipper	Submits the "Post Order" form through the UI.	Order submission request
2	**State Update**	handlePostOrder Function	Creates a new FreightOrder object with status **OPEN** and appends it to the orders array.	Updated orders list
3	**Notification / UI Update**	React Re-render Mechanism	Automatically re-renders the UI to reflect the new order.	Display in "Recent Freight Activity"
4	**AI Context Update**	Smart Assistant Component	Accesses the updated orders list for contextual AI processing.	AI insights generation capability

Table 4: Data Flow

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2026

DOI: 10.17148/IARJSET.2026.13257

IV. MODULE DESCRIPTION

4.1 Authentication & Role Management

The Authentication & Role Management module ensures robust application security through a structured role-based access control (RBAC) mechanism that restricts system functionalities based on user roles. During the registration process, users are required to select a predefined persona—either SHIPPER or TRUCK_OWNER—and this selection dynamically configures the interface, dashboard layout, and accessible modules. For instance, shippers are granted permissions to create freight orders, review bids, and manage shipment statuses, while truck owners are provided access to fleet registration, bid submission, and load tracking features. This strict segregation of responsibilities prevents unauthorized access to sensitive operations and maintains data integrity across user groups. The system also manages authenticated sessions securely, ensuring that navigation between modules adheres to the user's assigned privileges. To facilitate system testing and stakeholder demonstrations, a dedicated Demo Mode is included with pre-configured accounts such as "Demo Shipper" and "Demo Trucker," defined within the constants.ts configuration file. This feature enables evaluators to explore platform capabilities instantly without undergoing the complete registration workflow, thereby improving accessibility while preserving the underlying security framework.

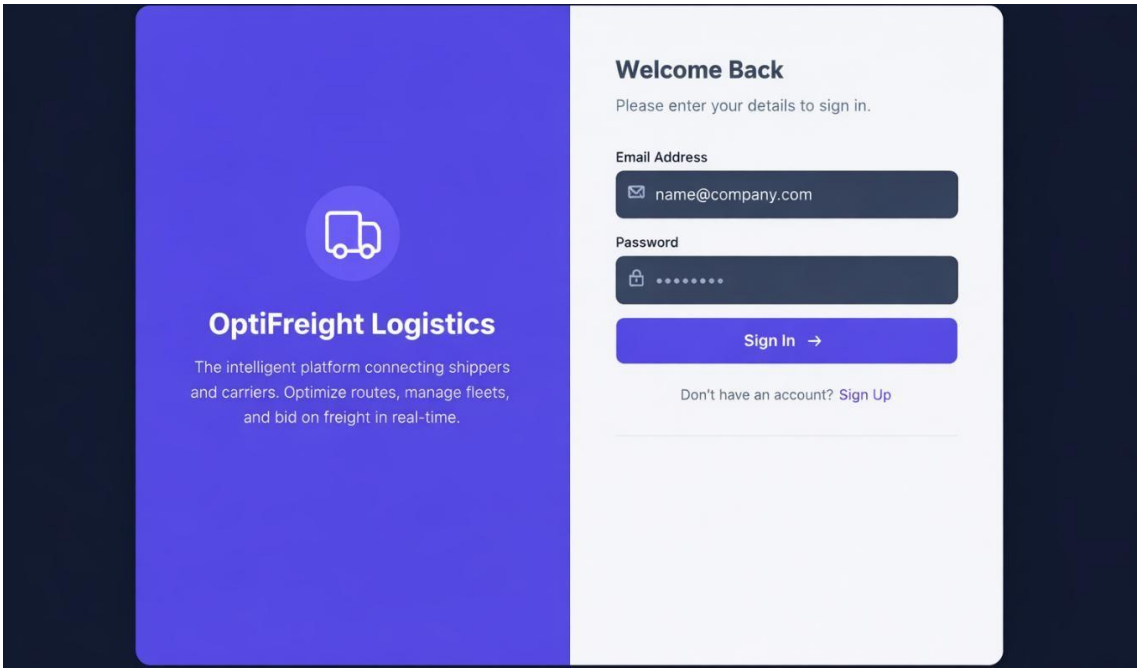

Figure 3 Authentication & Role Management

4.2 Freight Management Module

The Freight Management Module is specifically tailored for shippers and provides a comprehensive environment for creating, publishing, and monitoring freight shipments. The Order Creation feature allows users to enter critical logistics details including Origin, Destination, Cargo Type, Shipment Weight, Pickup Date, and any special handling requirements. Integrated validation mechanisms ensure that all mandatory fields are accurately completed, reducing data inconsistencies and operational errors. Once submitted, the freight order is instantly listed in the marketplace with a default status of OPEN, making it visible to eligible truck owners for bidding. The Bid Management component consolidates all submitted bids into an organized interface, allowing shippers to compare offers efficiently. To enhance decision-making, the system automatically calculates performance indicators such as the lowest bid, average bid value, and total number of bids received. These computed metrics provide real-time insights into market competitiveness and pricing trends, empowering shippers to select carriers based on both cost-effectiveness and reliability. By digitizing and structuring the freight posting workflow, this module significantly reduces manual coordination and enhances transparency throughout the transaction process.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2026

DOI: 10.17148/IARJSET.2026.13257

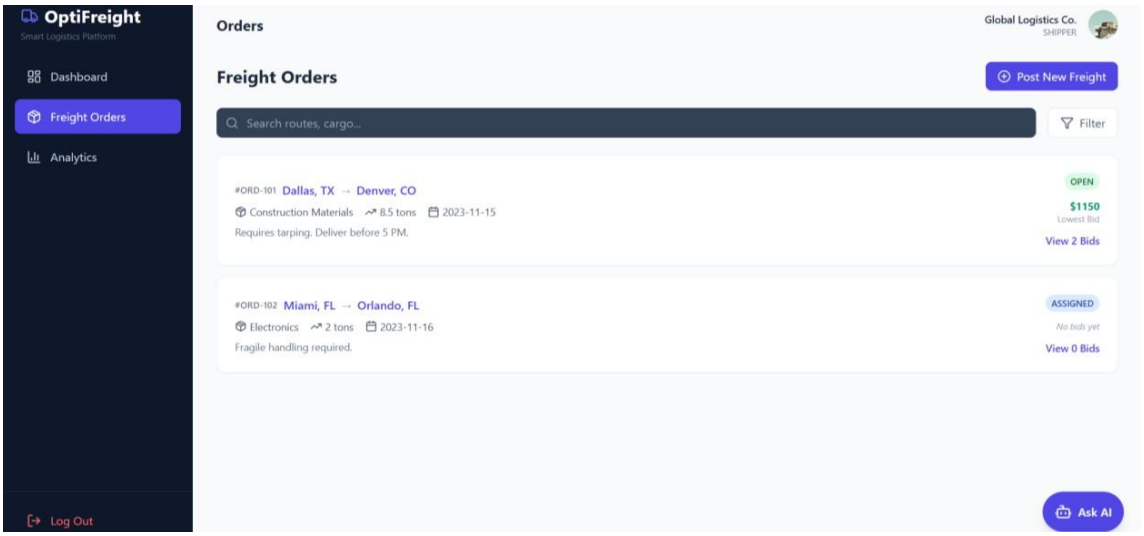

Figure 4 Freight Management Module

4.3 Fleet & Bidding Module

The Fleet & Bidding Module is designed to support truck owners in managing physical assets while maximizing business acquisition opportunities. Through the Fleet Inventory feature, users can register detailed information about each vehicle, including Plate Number, Vehicle Type (such as Refrigerated, Flatbed, or Container), Maximum Load Capacity, and operational status. This centralized inventory system offers clear visibility into fleet resources and supports efficient dispatch planning. The Smart Bidding functionality strengthens operational accuracy by requiring truck owners to select a specific registered vehicle when placing a bid on a freight order. The system intelligently filters and displays only trucks marked as AVAILABLE, automatically excluding those categorized as IN_TRANSIT or BUSY. This preventive mechanism eliminates the risk of double allocation and ensures that commitments reflect actual resource availability. Furthermore, once a bid is accepted and a truck is assigned, its status updates dynamically, maintaining real-time synchronization between marketplace activity and fleet records. By combining structured inventory management with intelligent bidding controls, the module enhances operational discipline, reduces scheduling conflicts, and promotes efficient asset utilization.

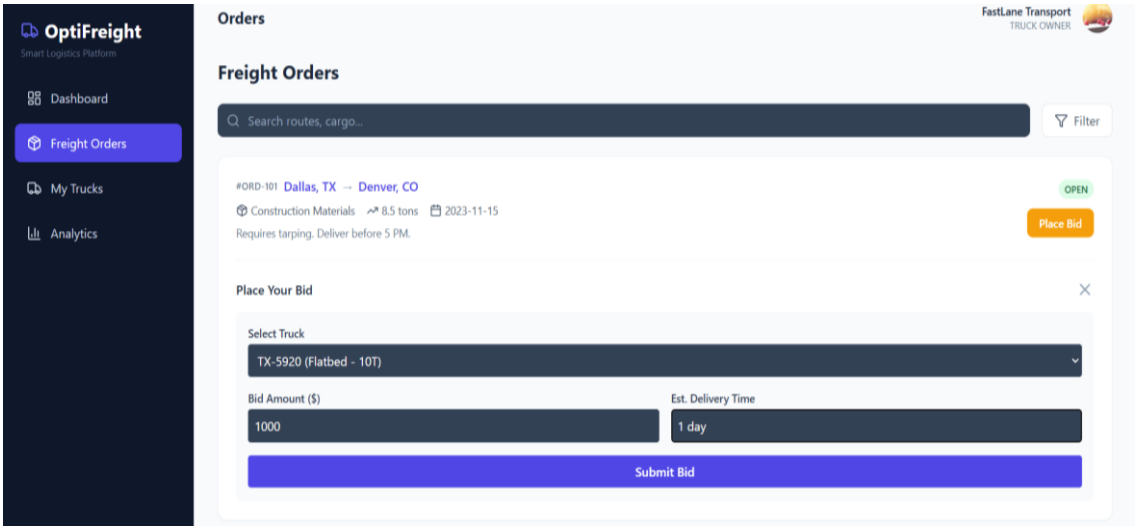

Figure 5 Fleet & Bidding Module

4.4 Analytics Module

The Analytics Module provides an interactive and data-driven overview of the platform's operational performance, enabling stakeholders to gain strategic insights into market behavior and workflow efficiency. Through dynamic visualizations powered by Recharts, the Market Rate Analysis feature presents bar charts that display the Average Bid Amount across multiple transportation routes, such as Dallas to Denver. This visualization allows users to identify pricing fluctuations, route-specific demand patterns, and competitive intensity within the marketplace. By analyzing

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2026

DOI: 10.17148/IARJSET.2026.13257

these trends, both shippers and truck owners can refine pricing strategies and optimize bidding decisions. Additionally, the Status Distribution component uses a pie chart to represent the percentage breakdown of freight orders categorized as OPEN, ASSIGNED, and DELIVERED. This graphical representation helps stakeholders quickly assess workflow progression and detect potential bottlenecks or delays in order processing. By transforming raw transactional data into intuitive visual insights, the Analytics Module supports proactive decision-making, performance monitoring, and continuous operational improvement within the freight management ecosystem.

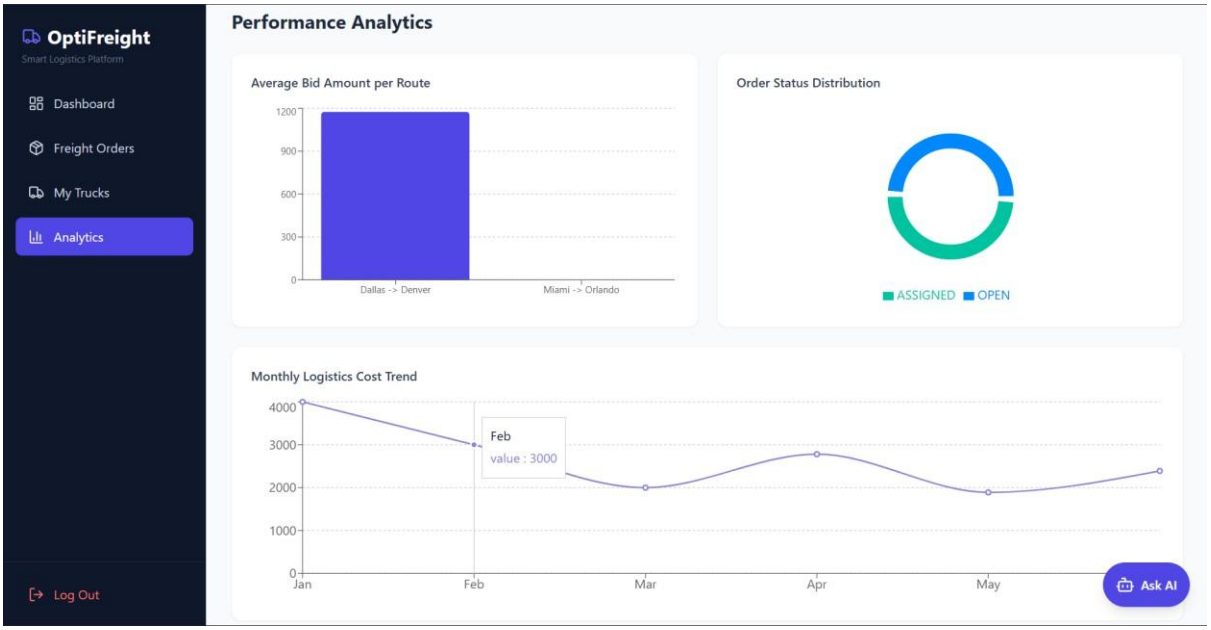

Figure 6 Analytics Module

Module	Key Functions	System Logic	Outcome
Authentication & Role Management	User registration (SHIPPER / TRUCK_OWNER), Demo login, Role-based access, Session management	Stores role in session state and enables conditional rendering based on user role	Secure access control and controlled feature visibility
Freight Management (Shipper)	Order creation (Origin, Destination, Cargo Type, Weight, Pickup Date), Order tracking, Bid viewing	Validates inputs, creates order with status OPEN, calculates lowest & average bids automatically	Efficient shipment posting and informed decision-making
Fleet & Bidding (Truck Owner)	Truck registration (Plate No, Type, Capacity), Smart bidding, Availability filtering	Stores trucks with AVAILABLE status, filters out IN_TRANSIT trucks during bidding	Prevents double booking and ensures reliable bid placement
Analytics Module	Market rate analysis (bar chart), Status distribution (pie chart), Trend monitoring	Uses Recharts to dynamically visualize average bids and order statuses	Provides pricing insights and workflow performance analysis

Table 5 Module Description

V. IMPLEMENTATION

5.1 Frontend Development

The user interface is developed using React 19 in combination with Tailwind CSS to deliver a modern, responsive, and highly interactive user experience. By leveraging Tailwind's utility-first styling approach and responsive breakpoint

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2026

DOI: 10.17148/IARJSET.2026.13257

classes such as md:grid-cols-3 and lg:flex, the application dynamically adapts its layout to different screen sizes and device types. This ensures that the dashboard, forms, and data views maintain optimal structure, readability, and usability across desktop monitors, tablets, and mobile devices frequently used by drivers and logistics managers in the field. Components automatically reorganize themselves based on screen dimensions, preserving appropriate spacing, alignment, and content hierarchy without compromising functionality. This responsive design strategy enhances accessibility and guarantees consistent performance regardless of the user's device.

To further improve usability and visual clarity, the platform integrates the Lucide- React icon library, which provides clean, scalable, and intuitive icons such as Truck, Package, and DollarSign. These visual cues help users quickly identify key modules and actions, reducing cognitive load and improving navigation efficiency particularly for non-technical stakeholders. Icons are strategically placed alongside text labels to reinforce meaning and streamline interaction flows within dashboards, bidding interfaces, and analytics sections. The combination of adaptive layout design and meaningful iconography results in an interface that is both aesthetically appealing and operationally efficient, supporting smooth interaction throughout the freight management lifecycle.

5.2 AI Integration Logic

The integration with Google Gemini is implemented using secure configuration and credential management practices to safeguard sensitive information. The API key is stored securely within the.env.local environment file rather than being hardcoded into the source code, ensuring it remains private and inaccessible from the public repository. This environment variable is accessed through the Vite configuration (vite.config.ts), which injects the key securely at build time without exposing it in the client-side codebase. Additionally, the.env.local file is explicitly excluded from version control using the .gitignore configuration, preventing accidental credential leaks during code sharing or deployment. By separating configuration data from application logic, the system adheres to modern security standards and best practices in web development, reducing the risk of unauthorized API usage and enhancing overall platform reliability.

To enable intelligent and context-aware responses, the application dynamically injects the current orders state into the prompt sent to the Gemini API. Before transmission, the data is sanitized and converted into a structured JSON string, ensuring clarity and consistency in the AI's input. This allows the model to analyze real-time marketplace information, including shipment routes, cargo categories, bidding activity, and order statuses. As a result, users can ask analytical questions such as identifying the routes with the highest shipment frequency or determining which cargo type is most actively transported. The AI processes this live operational data to generate meaningful, human-readable insights rather than static or generic responses. Consequently, Gemini functions as an intelligent assistant embedded within the platform, delivering accurate recommendations and contextual analysis based on up-to-date system information.

Module	Key Components	Implementation Logic	Outcome
Frontend Development	React 19, Tailwind CSS, Lucide-React Icons	Uses Tailwind responsive breakpoints (md:grid-cols-3, lg:flex) for adaptive layouts; integrates Lucide icons (Truck, Package, DollarSign) for better UI clarity	Responsive, user-friendly interface suitable for desktop and driver tablets
AI Integration Logic	Google Gemini API	API key stored securely in .env.local and loaded via vite.config.ts; passes stringified orders state as context to AI	Secure, context-aware AI responses based on real-time freight data

Table 6 Implementation

VI. RESULTS AND DISCUSSION

The developed system was rigorously evaluated using the MOCK_USERS and INITIAL_ORDERS datasets to verify functional correctness, performance stability, and real- time state management capabilities. Performance analysis indicated a high Lighthouse score, primarily due to the lightweight build optimization provided by Vite and efficient rendering mechanisms in React. Since the application is implemented as a Single Page Application (SPA), it minimizes full-page reloads and instead updates only the necessary components during state changes. This architectural choice significantly reduces latency, resulting in near- instantaneous page transitions and smooth navigation across modules such as Dashboard, Marketplace, Fleet Management, and Analytics. Memory utilization and rendering performance remained stable even during multiple bid submissions and order updates, demonstrating the platform's ability to handle concurrent user interactions efficiently. Overall, the system exhibited strong responsiveness, consistent UI behaviour,

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2026

DOI: 10.17148/IARJSET.2026.13257

and reliable client-side performance under simulated operational conditions.

Workflow validation was performed using structured test scenarios to ensure seamless interaction between system modules. In Scenario A, when a Shipper posted a load for "Electronics" from Miami to Orlando, the order was instantly reflected in the global Marketplace view accessible to Truck Owners, confirming proper state propagation and UI re-rendering. In Scenario B, a Truck Owner successfully placed a bid, which appeared immediately in the Shipper's Bid Management interface. Upon bid acceptance, the system updated the order status to ASSIGNED in real time across both user dashboards, verifying accurate synchronization of shared state data. Furthermore, AI validation tests demonstrated that the Smart Assistant correctly analysed marketplace data and identified "Global Logistics Co." as the primary shipper within the dataset. This confirmed the effectiveness of contextual data injection and the assistant's ability to generate accurate, insight-driven responses based on live system information.

Evaluation Area	Test Scenario / Condition	Observation	Result
System Testing Dataset	MOCK_USERS and INITIAL_ORDERS datasets	Simulated real-world freight marketplace conditions	Successful functional validation
Performance Analysis	Lighthouse performance testing	Built using Vite and optimized React rendering; Single Page Application	High Lighthouse score with instantaneous page transitions
Workflow Validation – Scenario A	Shipper posted "Electronics" load from Miami to Orlando	Load appeared immediately in global Marketplace view	Real-time synchronization between modules
Workflow Validation – Scenario B	Truck Owner placed bid and Shipper accepted	Order status updated to ASSIGNED across both dashboards instantly	Consistent state management and live UI update
AI Accuracy Evaluation	Smart Assistant queried using mock dataset	Correctly identified "Global Logistics Co." as primary shipper	Context-aware and accurate AI response

Table 7 Results and Discussion

VII. CONCLUSION

OptiFreight Logistics demonstrates the effectiveness of a modern, web-based truck bidding platform in transforming traditional freight procurement into a transparent, efficient, and intelligent digital ecosystem. By centralizing interactions between shippers and truck owners, the system reduces reliance on intermediaries and promotes fair, competitive pricing through a real-time bidding marketplace. The integration of fleet-awareness mechanisms enhances asset utilization by preventing double bookings and ensuring accurate vehicle allocation. As a Single Page Application, the platform delivers seamless navigation and responsive performance, contributing to a smooth user experience. The inclusion of AI- powered assistance further strengthens decision-making by providing context-aware insights derived from live operational data. Through functional validation and performance testing, the prototype has demonstrated reliable state synchronization, real-time updates, and accurate analytical responses. These results confirm that combining marketplace dynamics, structured fleet management, and intelligent analytics can significantly improve operational efficiency. For full-scale deployment, future enhancements should focus on scalable backend services, real-time GPS tracking, machine learning–based predictive analytics, and compliance-driven security mechanisms. With these advancements, OptiFreight Logistics has strong potential to evolve from a proof-of-concept model into a commercially viable and scalable digital freight marketplace.

REFERENCES

[1]. D. Simchi-Levi, P. Kaminsky, and E. Simchi-Levi, "Designing and Managing the Supply Chain: Concepts, Strategies and Case Studies", 4th ed. New York, NY, USA: McGraw-Hill, 2021.

[2]. J. N. D. Gupta and A. K. Banerjee, "Decision Support Systems in Logistics and Supply Chain Management". New York, NY, USA: CRC Press, 2018.

[3]. M. M. Solomon and C. Mersereau, "Vehicle Routing and Scheduling: Algorithms, Applications, and Models". Hoboken, NJ, USA: Wiley-IEEE Press, 2019.

DOI: 10.17148/IARJSET.2026.13257

[4]. M. T. Tan and L. S. Chew, "Smart Logistics and AI-Driven Transportation Systems". London, UK: Springer, 2022.
[5]. M. Ballou, "Business Logistics/Supply Chain Management", 6th ed. Upper Saddle River, NJ, USA: Pearson Education, 2019.
[6]. S. Chopra and P. Meindl, "Supply Chain Management: Strategy, Planning, and Operation", 7th ed. Boston, MA, USA: Pearson, 2020.
[7]. J. F. Bard, "Practical Management Science with Spreadsheets and Optimization Models", 3rd ed. Berlin, DE: Springer, 2021.
[8]. S. S. Pang, "Smart logistics frameworks and applications using AI and IoT," "IEEE Access", vol. 8, pp. 12345–12358, Mar. 2020.
[9]. R. Toorajipour, V. Sohrabpour, A. Nazarpour, P. Oghazi, and M. Fischl, "Artificial intelligence in supply chain management: A systematic review," "Journal of Business Research", vol. 121, pp. 884–900, Aug. 2020.
[10]. T. Wang and H. Qin, "Real-time dispatch and path optimization for logistics," "IEEE Trans. Autom. Sci. Eng.", vol. 19, no. 4, pp. 2420–2431, Oct. 2022.
[11]. X. Qiu, "Making the most of fleets: A profit-maximizing multi-vehicle approach in transportation," "Journal of Transportation Systems Engineering and Information Technology", vol. 17, no. 2, pp. 118–128, Apr. 2023.
[12]. F. X. Shen, J. Zhao, and M. Zhao, "AI-enabled logistics optimization for smart supply chains," in "Proc. IEEE Int. Conf. Industrial Electronics and Applications (ICIEA)", Chengdu, China, Jun. 2024, pp. 125–130.
[13]. J. Zhang and Y. Lee, "Multi-agent system models for freight transport planning," in "Proc. IEEE Int. Conf. Systems, Man, and Cybernetics (SMC)", Toronto, Canada, Nov. 2023, pp. 343–350.
[14]. S. K. Singh, "Deep learning based optimization for real-time vehicle routing," in "Proc. IEEE Int. Conf. Computational Intelligence in Transportation Systems (CITS)", Amsterdam, The Netherlands, Jul. 2025, pp. 58–64.
[15]. L. Ren, "Integrated optimization of logistics routing problems under uncertain demands," "Int. J. Logistics Research and Applications", vol. 27, no. 6, pp. 412–430, Dec. 2024.
[16]. Rakshun Selvaa S.K, Mrs. A. Sathiya Priya, "Blockchain Based Logistics Protocol Using Hyperledger Fabric," in "International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering (IJIREEICE)", Tamilnadu, India, Apr. 2025, pp. 211–217.

