AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3069-0102
VOL.6, NO. 2(2) (2026)

DEBEATS: Full Stack Food Delivery Application

Mr. Ayushman Biswal
Student, Dept. of CSE(AI),
GIFT Autonomous, Bhubaneswar

Mr. Debasis Sahoo
Student, Dept. of CSE(AI),
GIFT Autonomous, Bhubaneswar

Asst. Prof. Mr. Debaprasad Nanda
Assistant Professor, Dept. of CSE(IT),
GIFT Autonomous, Bhubaneswar

***Abstract*— DEBEATS is a real-time food delivery application designed to improve the speed and efficiency of online food ordering. The system provides live order tracking using GPS and map integration technologies, helping customers, restaurants, and platform authorities monitor rider locations in real time. It also includes review management, secure payment processing, restaurant onboarding, and delivery handling features. Riders can update delivery details and live locations, while administrators can manage restaurants and orders through a centralized dashboard. The system is developed using React.js, Node.js, Express.js, and MongoDB. Overall, DEBEATS improves communication, reduces manual work, and provides a reliable solution for modern online food delivery management.**

Keywords*— *Food Delivery, Real-Time Tracking, GPS Monitoring, Order Management, Web Application, Delivery Tracking System, Secure Payment System.

I. INTRODUCTION

In modern urban environments, food delivery systems play an important role in ensuring safe and timely meals for customers. Traditional food ordering systems mainly depend on phone calls and manual communication methods, which often create confusion and uncertainty among customers, restaurants, and platform authorities. Due to increasing traffic congestion and changing delivery conditions, it becomes difficult to monitor the exact location and arrival time of food deliveries using conventional methods. The lack of proper real-time communication and monitoring facilities also affects delivery quality and operational efficiency. Recent advancements in GPS tracking, web technologies, and real-time data communication have enabled the development of intelligent delivery management systems. Real-time tracking systems allow users to monitor rider movement continuously and improve coordination between riders, administrators, and users. These systems provide better transparency, reduce unnecessary waiting time, and improve overall delivery management. However, many existing delivery systems mainly focus on order tracking and do not provide complete operational management functionalities such as review handling, payment dispute management, restaurant recording, and delivery tracking. The proposed DEBEATS system is designed as a centralized real-time food delivery monitoring platform that integrates live tracking, communication, and order management into one system. It also includes review management, issue alerts, restaurant request handling, payment recording, and centralized administration facilities. Riders can manage deliveries and update live location data, while administrators can monitor riders, routes, and operational activities through a centralized dashboard.

II. PROBLEM STATEMENT AND EXISTING CHALLENGES

Traditional food ordering systems face several operational and communication-related challenges due to the absence of centralized digital monitoring platforms. Most local restaurants still depend on manual communication methods and third-party phone schedules for managing deliveries and order activities. These methods are often inefficient in handling dynamic delivery conditions such as traffic congestion, route changes, rider breakdowns, and unexpected delays.

A. Inadequacies in Real-Time Spatial Tracking

One of the major problems in traditional ordering systems is the absence of real-time tracking facilities. Customers and restaurants generally depend on estimated delivery timings without knowing the exact location of riders. During traffic congestion or route diversions, users remain uncertain about food arrival times, leading to unnecessary waiting and confusion.

B. Latency in Stakeholder Communication

Effective communication is essential for delivery management, especially during order issues or unexpected situations. In traditional ordering systems, communication between riders, customers, and restaurant authorities mainly depends on phone calls or manual messaging systems. These methods are often slow and unreliable during critical situations such as vehicle breakdowns, accidents, delays, or order changes.

C. Inefficiencies in Analog Data Management

Another major challenge in existing ordering systems is manual record management. Delivery-related records such as rider details, route information, payment reports, restaurant records, and reviews are often maintained using paper registers or spreadsheets. Manual record handling increases paperwork and makes delivery management more time-consuming and less efficient. Maintaining records manually also increases the possibility of data inconsistency, record duplication, and information loss. Administrators face difficulties in managing and accessing delivery records quickly when required. Review management and payment tracking become inefficient due to the absence of centralized digital storage systems.

D. Vulnerabilities in Safety and Monitoring

Delivery safety is one of the most important concerns in online food delivery systems. Traditional systems do not provide centralized monitoring or issue alert mechanisms for

quickly responding to critical situations. In cases of accidents, vehicle breakdowns, or unexpected delays, administrators may not receive immediate information regarding the situation. The absence of continuous delivery monitoring reduces operational visibility and makes it difficult for platform authorities to ensure delivery safety effectively. Since there is no centralized monitoring platform, administrators cannot easily track rider locations, route performance, or delivery activities in real time.

E. Imperative for a Smart Delivery System

The limitations of traditional ordering systems highlight the need for a modern and intelligent delivery management platform capable of integrating real-time tracking, centralized communication, operational monitoring, review handling, and issue management into one system. A smart delivery system can improve communication efficiency, delivery safety, and operational transparency while reducing manual work and management complexity. The integration of GPS technologies, real-time communication systems, and centralized administration platforms can significantly improve the overall efficiency of food delivery management.

III. PROPOSED SYSTEM ARCHITECTURE: DEBEATS

The proposed DEBEATS system is designed to provide a smart and centralized solution for managing food delivery activities in real time. The system mainly focuses on improving delivery speed, communication, operational transparency, and monitoring efficiency. Traditional food ordering systems mostly depend on manual communication and fixed scheduling methods, which are not suitable for handling dynamic delivery conditions such as traffic congestion, route delays, and emergency situations. To overcome these limitations, DEBEATS integrates real-time GPS tracking, communication facilities, review management, payment handling, restaurant management, and centralized monitoring into a single web-based platform. The system allows customers and restaurants to monitor the live location of riders through a map-based interface. Riders can manage deliveries, update order information, and send issue alerts when required. Platform administrators can continuously monitor delivery activities through a centralized dashboard and manage riders, routes, reviews, restaurant requests, and operational records efficiently. The integration of multiple delivery functionalities into one platform improves coordination among all users and reduces manual management efforts.

A. Architectural Overview

DEBEATS follows a client-server architecture where the frontend, backend, database, and GPS tracking modules work together to provide real-time delivery management. The frontend interface is developed using React.js and communicates with the backend server through RESTful APIs. The backend server handles user requests, processes delivery data, and manages communication with the MongoDB database. The system continuously receives GPS coordinates from rider devices and updates the latest location information in the database. Whenever a user opens the tracking interface, the frontend fetches updated rider location data from the server and displays it on the digital map. This allows users to monitor the current location and movement of riders in real time.

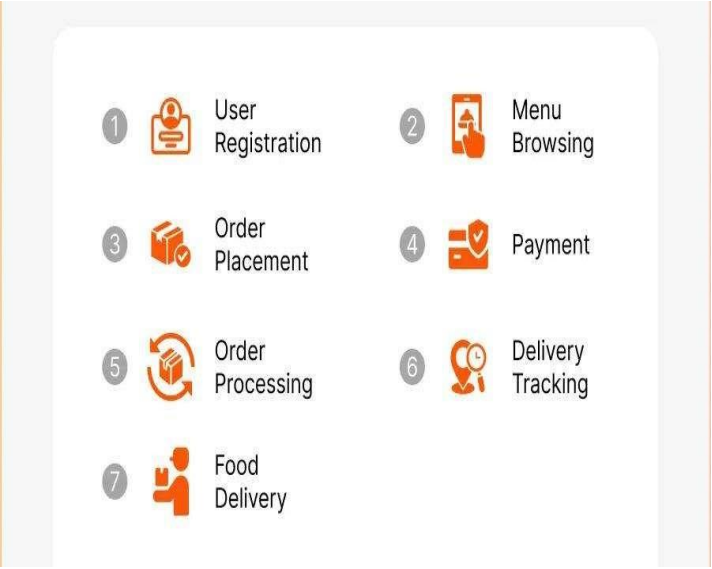


Fig. 1. High-Level Workflow of the DEBEATS Application

Apart from tracking facilities, the system also provides operational management functionalities such as review handling, payment recording, restaurant request generation, and issue alert handling. The centralized design of the platform simplifies delivery management and improves transparency between customers, restaurants, riders, and administrators. The proposed system is designed to support scalability and responsiveness so that it can efficiently manage delivery activities even when the number of users and orders increases. The use of modern web technologies and real-time communication mechanisms helps in improving overall system performance and user experience.

B. Client-Side Interface (User Module)

The user module is designed for customers and restaurants to monitor delivery activities. Users can track the live location of riders through a map-based interface and receive updates regarding rider movement and estimated arrival time. The module also allows users to submit reviews related to delays, food issues, or delivery problems directly to the administrator dashboard for review and resolution.

C. Courier Interface (Rider Module)

The rider module plays an important role in maintaining real-time delivery updates and operational activities. Riders can start and end deliveries, update live location data, and manage order records through the system interface. The GPS-enabled rider device continuously sends location coordinates to the backend server. The module also supports payment recording and issue request generation. Riders can record fuel expenses, tips, or other delivery-related data digitally instead of maintaining manual records. In case of vehicle issues or routing requirements, riders can directly submit issue requests to the administrator. Another important functionality of the rider module is the issue alert system. During emergencies such as vehicle breakdowns, accidents, or unexpected delays, riders can immediately send alerts through the system

.

D. Centralized Administrative Control

The admin module acts as the central control unit of the DEBEATS platform. Administrators can monitor all delivery activities through a centralized dashboard that displays restaurants, routes, riders, reviews, restaurant requests, and operational records in one interface. Administrators can manage rider details, assign orders to routes, review delivery records, and monitor real-time rider movement continuously. The dashboard also allows administrators to handle user reviews and issue requests efficiently. Since all delivery records are stored digitally in MongoDB, administrators can access and manage operational information more effectively compared to traditional manual systems. It also allows administrators to monitor delivery safety and identify operational issues quickly.

E. Operational Workflow

The working process of DEBEATS begins when the rider device continuously sends GPS coordinates to the backend server through internet connectivity. The backend processes the incoming location information and stores the updated coordinates in the MongoDB database. The frontend periodically fetches the latest delivery data through REST APIs and updates the live tracking interface dynamically. Whenever users access the tracking page, the system displays the current location of riders along with route information and estimated arrival details. If any review or issue request is generated, the information is directly stored in the database and forwarded to the administrator dashboard for further action. During emergencies, riders can generate issue alerts that are instantly reflected in the admin panel. This improves communication speed and allows administrators to respond quickly during critical situations. The complete workflow of the system helps improve delivery safety, communication efficiency, and operational coordination among all stakeholders involved in food delivery management.

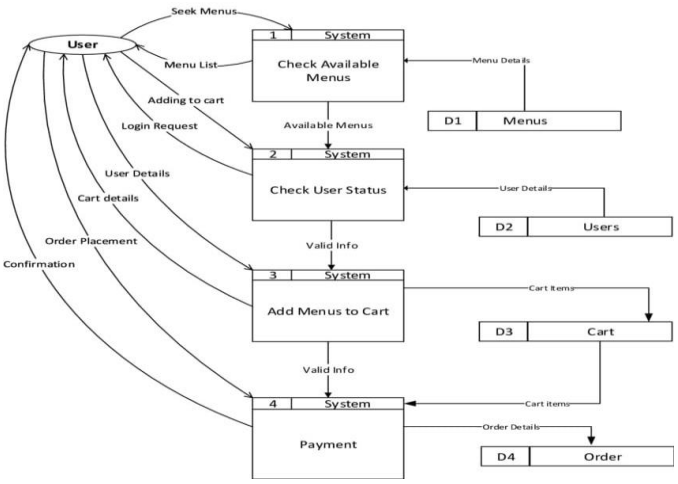


Fig. 2. Data Flow Diagram of the DEBEATS System

IV. IMPLEMENTATION METHODOLOGY AND DEVELOPMENT STACK

The development of the DEBEATS system follows a structured and modular approach to ensure efficient implementation and smooth delivery management. The methodology mainly focuses on requirement analysis, system planning, frontend and backend development, database integration, and real-time communication. The complete development process is carried out using modern web technologies and follows the Agile software development methodology for continuous improvement and testing. The Agile methodology is selected because it supports iterative development, faster testing, flexibility, and continuous feedback during the implementation process. Instead of developing the complete system at once, the project is divided into smaller modules such as tracking management, review handling, rider management, admin dashboard, and issue alert handling. Each module is developed and tested separately before final integration into the complete system.

A. Systems Requirement Analysis

Requirement analysis is one of the most important stages in system development. In this phase, the major problems in traditional ordering systems are studied carefully to identify the functional and non-functional requirements of the DEBEATS platform. The functional requirements of the system include real-time order tracking, review management, rider management, issue alert handling, payment recording, restaurant request management, and centralized administrative monitoring. The non-functional requirements mainly focus on system scalability, user-friendly interface design, real-time responsiveness, reliability, data security, and operational efficiency. The requirement analysis phase helps in understanding how different users such as customers, restaurants, riders, and administrators will interact with the system. It also helps in identifying the necessary technologies and system architecture required for implementation.

B. Client-Side Engineering

The frontend of the DEBEATS system is developed using React.js and Tailwind CSS to provide a responsive and user-friendly interface. React.js is selected because of its component-based architecture, reusable UI components, and efficient rendering capabilities. The user interface is designed to be simple and interactive so that users can easily access delivery information without technical difficulties. Tailwind CSS is used for responsive styling and improved UI design consistency across different devices. The frontend continuously communicates with the backend through REST APIs to fetch updated delivery and tracking information dynamically.

C. Server-Side Infrastructure

The backend of the system is developed using Node.js and Express.js. The backend acts as the core processing unit of the DEBEATS platform and is responsible for handling user requests, authentication, database operations, and API communication. Express.js is mainly used for route handling, API development, middleware integration, and request-response management within the backend system. The backend continuously processes delivery data and maintains communication between frontend modules and the MongoDB database. MongoDB is used for storing and managing delivery-related data efficiently. The backend continuously receives GPS location data from rider devices and processes delivery information in real time. It also manages review handling, issue requests, order alerts, and operational records efficiently. The

backend architecture is designed to support scalability and high performance so that the system can handle multiple delivery activities simultaneously without affecting responsiveness.

D. API Protocol and Telemetry

RESTful APIs are implemented to support communication between the frontend and backend components. APIs are responsible for sending and receiving delivery data, user requests, review details, and live tracking information. The system continuously updates rider locations using real-time GPS coordinate transmission from rider devices. Whenever the frontend requests updated location information, the backend retrieves the latest GPS data from MongoDB and sends it to the frontend interface dynamically. Postman is used for API testing and validation during development. API testing helps verify request-response handling, data consistency, authentication functionality, and system reliability. The implementation of real-time communication mechanisms significantly improves tracking accuracy and delivery monitoring efficiency.

E. Quality Assurance and Testing

System testing is performed to ensure proper functionality and performance of all modules in the DEBEATS platform. Different testing techniques are used during development to verify tracking accuracy, API functionality, database connectivity, user authentication, review handling, dashboard operations, and issue alert response. The system is tested under different delivery scenarios to ensure stable performance and reliable communication between modules. Testing results show that the DEBEATS system can efficiently handle real-time delivery management and monitoring operations.

V. SYSTEM DESIGN AND COMPONENT INTEGRATION

The DEBEATS system is designed using a centralized client-server architecture that supports real-time delivery monitoring and operational management. The complete system consists of frontend modules, backend services, database integration, GPS tracking components, and communication APIs. The architecture is designed to ensure scalability, responsiveness, and efficient communication between different system modules. The implementation process mainly focuses on integrating real-time tracking, review management, issue communication, payment handling, restaurant management, and centralized monitoring into a single web-based platform. Each module is developed and integrated carefully to provide smooth delivery management and improved user experience.

A. Distributed Network Architecture

The overall architecture of DEBEATS consists of frontend, backend, database, and GPS tracking layers that work together to support real-time delivery monitoring and operational management. Each layer performs specific functionalities while maintaining continuous communication with other system components. The frontend layer is responsible for displaying delivery information and user interfaces. It allows customers, restaurants, riders, and administrators to interact with the system through web pages and dashboards. The backend layer handles business logic, API processing, user authentication, and communication between modules. MongoDB acts as the database layer and stores operational records and delivery data efficiently. The GPS tracking layer continuously sends live rider coordinates to the backend server. The backend processes the incoming data and updates the frontend dynamically through APIs. This architecture allows users to monitor rider movement in real time and improves delivery transparency. The modular architecture also improves maintainability and scalability because new features can be added without affecting the existing system structure.

B. Geospatial Tracking Engine

The live tracking module is one of the core components of the DEBEATS platform. This module allows customers, restaurants, and administrators to monitor the real-time location of riders through map integration technologies. The GPS-enabled rider device continuously transmits location coordinates to the backend server using internet connectivity. The backend processes the received coordinates and stores them in MongoDB. Whenever users access the tracking interface, the frontend requests updated location information through APIs and displays the current rider position dynamically on the map. The live tracking module reduces uncertainty regarding order arrival timings and helps users estimate delivery delays more accurately. It also improves operational visibility for administrators by allowing continuous monitoring of route movement and delivery activities. The implementation of real-time tracking significantly improves delivery communication and coordination compared to traditional manual ordering systems.

C. Dispute and Review Resolution System

The review management system is designed to improve communication between users and platform administrators. Customers and restaurants can submit reviews related to delivery delays, rider behavior, route issues, or food problems directly through the platform. Whenever a review is submitted, the information is stored in the database and immediately reflected in the administrator dashboard. Administrators can review complaint details, monitor review status, and take appropriate actions efficiently. Digital review handling improves transparency and reduces manual paperwork. It also allows platform authorities to maintain proper review records and improve service quality through centralized monitoring. The review module helps establish better communication between delivery users and platform authorities while simplifying dispute resolution processes.

D. Critical Alert Infrastructure

Delivery emergencies such as vehicle breakdowns, accidents, route diversions, or delays require immediate communication between riders and administrators. The DEBEATS platform includes an issue alert system that allows riders to instantly notify administrators during critical situations. The issue alert module is integrated directly into the rider interface. Whenever an emergency occurs, riders can generate alerts through the system. The alert information is immediately transmitted to the backend server and displayed on the administrator dashboard. The implementation of issue communication facilities improves delivery safety and reduces response time during unexpected situations. Administrators can quickly monitor the situation and take necessary actions to manage delivery operations effectively. The notification system also helps improve user confidence and delivery reliability.

E. Command Center Dashboard

The admin dashboard acts as the centralized monitoring and management interface for the complete DEBEATS platform. Administrators can monitor restaurants, routes, riders, reviews, issue alerts, restaurant requests, and payment records through a single dashboard interface. The dashboard provides real-time operational visibility and allows administrators to manage delivery activities efficiently. Administrators can assign riders to routes, monitor delivery performance, review feedback, and track live rider movement continuously through the platform. Since all delivery data is stored digitally, administrators can easily access operational records and monitor delivery activities without depending on manual documentation. The centralized dashboard improves communication and coordination between platform staff, riders, customers, and restaurants. The implementation of centralized monitoring significantly reduces manual work and improves delivery management efficiency.

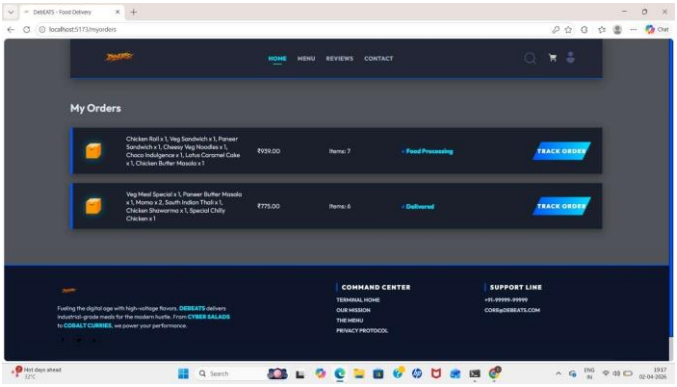


Figure -3. Order Processing and Real-Time Tracking Interface

F. Strategic Advantages

The DEBEATS platform provides several advantages compared to traditional ordering management systems. The integration of real-time tracking, centralized communication, and digital management functionalities improves delivery safety, transparency, and operational efficiency. The DEBEATS platform provides several advantages including real-time order tracking, improved communication between users and administrators, faster issue response, centralized delivery monitoring, digital review handling, reduced paperwork, and efficient restaurant and payment management. The system improves operational transparency and simplifies delivery coordination for platform authorities. The proposed system also improves coordination between customers, restaurants, riders, and platform authorities while reducing operational complexity. The use of modern web technologies and centralized management mechanisms provides a scalable and reliable solution for modern online food delivery systems.

Traditional Ordering System	DEBEATS System
Manual communication methods	Centralized communication platform
No real-time tracking facility	Real-time GPS bus tracking
Paper-based record management	Digital data management
No issue alert mechanism	Integrated issue alert system
Manual review handling	Centralized review management
Difficult delivery monitoring	Centralized admin dashboard
Delayed delivery updates	Real-time delivery updates
Higher paperwork and management effort	Reduced manual work and improved efficiency

Table 1. Comparison of Traditional System and DEBEATS

VI. RESULTS AND DISCUSSION

The DEBEATS system was successfully implemented and tested for real-time online delivery monitoring and management. Different modules of the system including live tracking, review management, issue alerts, payment handling, restaurant management, and centralized administration were tested under various delivery scenarios to evaluate system performance and operational efficiency. The implementation results show that the proposed system provides better delivery transparency, improved communication, and more efficient operational management compared to traditional ordering systems. The integration of GPS technologies, real-time communication, and centralized monitoring significantly improves delivery coordination between customers, restaurants, riders, and administrators.

A. System Performance Analysis

The overall performance of the DEBEATS platform was analysed based on tracking responsiveness, API response time, database communication, and operational efficiency. The system was able to continuously process GPS location updates and display rider movement dynamically without significant delay. The frontend interface remained responsive during continuous tracking operations and handled multiple delivery activities efficiently. MongoDB provided fast storage and retrieval of delivery records, reviews, and tracking information. Backend APIs also maintained stable communication between different system modules during testing. The use of React.js and Node.js improved frontend responsiveness and backend processing efficiency respectively.

B. Tracking Accuracy and Monitoring

The live tracking module demonstrated accurate and continuous delivery monitoring during implementation testing. The GPS tracking system successfully updated rider locations in real time and displayed current movement on the map interface dynamically. Customers and restaurants were able to monitor the exact location of riders and estimate arrival times more effectively compared to traditional ordering systems. Administrators could also monitor route movement continuously through the centralized dashboard. The implementation of real-time tracking significantly reduced uncertainty regarding delivery delays and improved communication between users and platform authorities. The

testing results showed that the system maintained stable tracking performance even during continuous location updates and multiple delivery activities

.

C. Communication Efficiency

The implementation of centralized communication modules improved coordination between riders, administrators, customers, and restaurants. The review management system allowed users to submit delivery-related reviews digitally without depending on manual communication methods. The issue alert module also improved communication speed during critical situations such as vehicle breakdowns, route changes, or delivery delays. Administrators received issue notifications instantly through the dashboard, allowing faster response and operational management. Compared to traditional ordering systems, the DEBEATS platform reduced communication delays significantly and improved information sharing among delivery stakeholders.

D. Operational Improvements

The digital implementation of delivery management functionalities reduced manual paperwork and simplified operational activities significantly. Payment recording, restaurant request handling, review management, and route monitoring were all managed digitally through the centralized system. Administrators could access delivery records more efficiently compared to manual documentation systems. The centralized dashboard also simplified delivery monitoring by integrating all operational information into a single platform. Digital storage of delivery data improved record organization and reduced the possibility of data inconsistency or information loss. Restaurant monitoring and payment management also became more systematic and transparent. The implementation results clearly showed that centralized digital management improves operational efficiency and delivery coordination compared to conventional ordering systems.

E. User Experience and System Reliability

The user interface of the DEBEATS platform was designed to provide a simple and responsive experience for different users including customers, restaurants, riders, and administrators. During testing, users were able to access delivery information and system functionalities without technical difficulties. The frontend interface remained responsive across different devices and supported smooth navigation between tracking, review handling, and dashboard modules. The backend system also maintained stable API communication and database connectivity during continuous delivery operations. The implementation of real-time communication and centralized monitoring improved overall system reliability and delivery management efficiency. Users experienced better delivery transparency and improved coordination compared to traditional manual ordering systems. The testing results demonstrate that the DEBEATS platform provides a reliable, scalable, and user-friendly solution for modern online food delivery management.

VII. FUTURE ENHANCEMENTS

The The DEBEATS platform provides a strong foundation for developing more advanced and intelligent food delivery management systems in the future. Although the current implementation successfully supports real-time tracking, centralized monitoring, review handling, issue communication, and operational management, several additional features can further improve system efficiency, scalability, and user experience.

A. Native Mobile Ecosystem Integration

Currently, the DEBEATS platform is implemented as a web-based application. In future development, dedicated mobile applications for Android and iOS platforms can be developed to provide easier accessibility for customers, restaurants, riders, and administrators. Mobile application integration will allow users to receive delivery updates more conveniently through smartphones. Push notifications can also be implemented for instant communication regarding food preparation timings, route changes, delivery delays, and promotional alerts. The mobile application can further improve user engagement and delivery monitoring efficiency.

B. Algorithmic Route Optimization

Artificial Intelligence and Machine Learning techniques can be integrated into the DEBEATS platform to improve delivery optimization and route prediction. AI-based systems can analyse traffic conditions, historical delivery data, and route patterns to predict delays and suggest optimized delivery routes automatically. The implementation of intelligent route prediction can help reduce delivery delays and improve fuel efficiency. AI integration can significantly improve operational efficiency and make the delivery system smarter and more adaptive to changing traffic conditions.

C. Contactless Handoff Authentication

A QR or OTP-based verification system can also be integrated with the DEBEATS platform in future development. Verification codes can be assigned to customers, and order completion can be recorded automatically whenever customers receive the food. The delivery information can be stored directly in the centralized database and made accessible to restaurants and administrators through the platform. This feature can improve food security and provide better monitoring of delivery activities. The integration of automated verification systems can also reduce manual dispute handling and improve operational transparency.

D. Cloud Deployment and Scalability

Future implementation of the DEBEATS system can include cloud deployment to improve scalability, data accessibility, and system performance. Cloud-based infrastructure can support large-scale delivery operations involving multiple cities, restaurants, and users simultaneously. It can also support better system maintenance and performance optimization. The use of cloud technologies will allow the platform to handle large amounts of delivery data more efficiently while maintaining stable system performance and reliability.

VIII. CONCLUSION

The DEBEATS system is a smart and centralized real-time food delivery monitoring platform developed to improve delivery safety, communication, and operational efficiency. The system integrates live order tracking, review management, issue alerts, restaurant handling, and centralized administration into a single web-based platform. The implementation of GPS tracking and modern web technologies such as React.js, Node.js, Express.js, and MongoDB improves delivery transparency and reduces manual work. The system helps customers, restaurants, riders, and administrators monitor delivery activities more efficiently and improves overall coordination and communication. Overall, DEBEATS provides a reliable and user-friendly solution for modern online food delivery management, with future scope for mobile applications, AI-based route optimization, and cloud integration.

REFERENCES

[1] S. Sharma and R. Gupta, "Smart Transportation Management Systems for Educational Institutions," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 4, pp. 210–216, 2021.

[2] A. Kumar and P. Singh, "Real-Time Vehicle Tracking Using GPS and Web Technologies," *IEEE International Conference on Intelligent Transportation Systems*, pp. 145–150, 2020.

[3] M. Patel and K. Verma, "Communication Challenges in School Transportation Systems," *International Journal of Engineering Research & Technology (IJERT)*, vol. 10, no. 6, pp. 325–330, 2021.

[4] React.js Documentation, Meta Platforms Inc. [Online]. Available: https://react.dev/

[5] T. Brown and S. Wilson, "Limitations of Traditional Transportation Monitoring Systems," *International Journal of Transportation Engineering*, vol. 8, no. 3, pp. 98–104, 2020.

[6] P. Sharma and A. Das, "GPS-Based School Bus Monitoring and Tracking System," *International Conference on Smart Computing and Electronics Systems*, pp. 412–417, 2021.

[7] R. Mehta and V. Shah, "Improving Transportation Communication Using Real-Time Monitoring Systems," *IEEE Access*, vol. 9, pp. 55672–55680, 2021.

[8] K. Rao and S. Mishra, "Digital Record Management for Smart Transportation Systems," *International Journal of Computer Applications*, vol. 176, no. 21, pp. 15–20, 2020.

[9] D. Singh and P. Roy, "Emergency Alert Mechanisms in Intelligent Transportation Systems," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 5, pp. 2271–2276, 2020.

[10] J. Lee and H. Kim, "Centralized Monitoring Platforms for Smart Transportation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5643–5652, 2021.

[11] K. Beck et al., "Manifesto for Agile Software Development," Agile Alliance, 2001. [Online]. Available: https://agilemanifesto.org/

[12] React.js Documentation, "Building User Interfaces with Components," Meta Platforms Inc. [Online]. Available: https://react.dev/

[13] Node.js Documentation, OpenJS Foundation. [Online]. Available: https://nodejs.org/

[14] MongoDB Documentation, MongoDB Inc. [Online]. Available: https://www.mongodb.com/docs/

[15] Express.js Documentation, Express Foundation. [Online]. Available: https://expressjs.com/