Smart Logistics Routing with Real Time Mapping

Project report submitted to

Visvesvaraya National Institute Of Technology, Nagpur

In partial fulfilment of the requirements for the award of the degree.

Bachelor of Technology

In

Computer Science and Engineering

by

Sahil Baori	**BT20CSE022**
Harsh Shaktawat	**BT20CSE046**
Navin Choudhary	**BT20CSE089**
Aditya Wagh	**BT20CSE123**

Under the Guidance of

Dr. M.P. Kurhekar

Department of Computer Science and Engineering

Visvesvaraya National Institute Of Technology

Nagpur 440010 (India)

2024

DECLARATION

We hereby declare that this project work titled **SMART LOGISTICS ROUTING WITH REAL TIME MAPPING** is carried out by us in the Department of Computer Science and Engineering of Visvesvaraya National Institute of Technology, Nagpur under the guidance of Dr. M.P. Kurhekar. The work is original and has not been submitted earlier whole or in part for the award of any degree/diploma at this or any other Institution/University.

Sr. No.	Names	Enrollment No.	Signature
1	Sahil Baori	BT20CSE022	
2	Harsh Shaktawat	BT20CSE046	
3	Navin Choudhary	BT20CSE089	
4	Aditya Wagh	BT20CSE123	

Department of Computer Science and Engineering
Visvesvaraya National Institute of Technology, Nagpur

CERTIFICATE

This is to certify that the project titled "**Smart Logistics Routing With Real Time Mapping**", submitted by Sahil Baori, Harsh Shaktawat, Navin Choudhary and Aditya Wagh in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering, VNIT Nagpur. The work is comprehensive, complete and fit for final evaluation.

Dr. M. P. Kurhekar

(Project Guide)

Associate Professor,

Computer Science and Engineering,

VNIT, Nagpur

Dr. M. P. Kurhekar

Head of Department,

Department of Computer Science and Engineering,

VNIT, Nagpur

Date:

ACKNOWLEDGMENTS

We would like to extend our heartfelt gratitude to Dr. M.P. Kurhekar for his invaluable guidance, support, and mentorship throughout the development of the project. His mentorship has been invaluable in our academic and professional growth, and we are immensely thankful for his contributions to our development.

We extend our heartfelt gratitude to all individuals who have provided unwavering support and encouragement, both directly and indirectly. Your steadfast belief in our endeavours has been instrumental in driving the successful completion of this project.

Thanking you,

Sahil Baori

Harsh Shaktawat

Navin Choudhary

Aditya wagh

ABSTRACT

The logistics sector plays a crucial role in facilitating the movement of goods across regions, impacting economic growth and sustainability. Inspired by the government's Unified Logistics Interface Platform, this thesis addresses the challenge of optimising truck selection within the logistics ecosystem. Leveraging MapMyIndia APIs, our research focuses on developing a user-friendly shipping website with a smart backend.

Our main goal is to create a system that intelligently matches trucks with user specifications like location, product type, and weight. By incorporating these factors into our decision-making process, we aim to optimise routes and resource allocation, ultimately enhancing efficiency, cutting costs, and reducing environmental impact.

Through a comprehensive analysis and implementation of algorithms, our study seeks to contribute to the government's initiative of streamlining logistics operations. By making shipping processes more efficient, cost-effective, and environmentally friendly, we aim to support broader national objectives of economic development and sustainability. This thesis not only presents a theoretical framework but also provides practical insights into the application of advanced technologies in the logistics domain.

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER 1

INTRODUCTION

The logistics sector in India stands as a linchpin in the nation's economic landscape, serving as the backbone of trade and commerce. With a vast and diverse geography coupled with a burgeoning economy, the efficient movement of goods across regions is paramount for sustained growth and development. However, despite its pivotal role, the logistics industry in India has long grappled with challenges ranging from infrastructure deficiencies to operational inefficiencies.

At the heart of these challenges lies the need for a cohesive and integrated logistics ecosystem that can streamline processes, optimise resources, and enhance overall efficiency. Recognizing this imperative, the concept of a logistics interface platform has emerged as a transformative solution to address the complexities of the logistics landscape.

Before embarking on our project journey, a foundational step was taken in the form of constructing a causal loop diagram encompassing the myriad factors influencing the logistics ecosystem in India. This approach allowed for a holistic understanding of the interrelationships and feedback loops within the system, illuminating the root causes of inefficiencies and bottlenecks.

The utilisation of a causal loop diagram provided invaluable insights into the dynamics of the logistics sector, guiding the development of our project framework and strategy. By visualising the intricate web of cause-and-effect relationships, we were able to identify leverage points where interventions could yield the most significant impact.

Moreover, the causal loop diagram served as a roadmap for designing the logistics interface platform, informing the selection of features and functionalities aimed at addressing systemic challenges and enhancing operational efficiency. Through iterative refinement and validation, the project was tailored to align with the emergent needs and dynamics of the logistics ecosystem.

In parallel, the government's initiative towards establishing a Unified Logistics Interface Platform (ULIP) represents a significant step forward in transforming India's logistics landscape. By providing a unified platform for stakeholders to seamlessly integrate and collaborate, ULIP aims to enhance transparency, reduce operational bottlenecks, and drive overall efficiency in the logistics sector.

Through the integration of advanced technologies such as artificial intelligence, data analytics, and IoT (Internet of Things), ULIP holds the promise of optimizing the entire logistics value chain, from warehousing and inventory management to transportation and last-mile delivery.

Furthermore, ULIP fosters collaboration and synergy among stakeholders by providing a common platform for communication and coordination. Through features such as integrated messaging systems, document management, and performance tracking, it enables seamless collaboration across organisational boundaries, thereby reducing delays, errors, and misunderstandings.

In conclusion, the integration of a causal loop diagram into the project development process not only provided a robust analytical foundation but also facilitated a systematic approach towards addressing the multifaceted challenges plaguing the logistics sector in India. By leveraging the power of technology and fostering greater collaboration among stakeholders, India can unlock the full potential of its logistics sector, driving economic growth, enhancing competitiveness, and fostering sustainability in the years to come, supported by initiatives such as ULIP.

1.1 Background Challenges

The logistics and shipping departments in India face several challenges that hinder their efficiency and effectiveness. Some of these challenges are:

1. *High Logistics Costs* - India faces some of the highest logistics costs in the world, primarily due to inefficiencies in transportation, inventory management, and warehousing, which adversely impact the competitiveness of businesses and hinder economic growth.

2. *Poor Last-Mile Connectivity* - Last-mile connectivity remains a significant challenge in India, especially in rural and remote areas, resulting in delayed deliveries, increased costs, and customer dissatisfaction.

3. *Limited Technology Adoption* - The adoption of advanced technologies such as GPS tracking, route optimization, and warehouse management systems is relatively low in the Indian logistics sector, hindering operational efficiency and productivity and also there are more paper-based processes resulting in limited visibility across the supply chain, and inefficiencies in operations.

4. *Inefficient Warehousing Practices* - Inadequate warehousing infrastructure, lack of standardised practices, and manual handling processes contribute to inefficiencies in inventory management, stockouts, and pilferage, affecting supply chain performance.

5. *Poor Supply Chain Visibility* - Limited end-to-end visibility across the supply chain, including tracking of raw materials, inventory in transit, and finished goods, makes it difficult for businesses to anticipate demand fluctuations, optimise inventory levels, and respond effectively to market dynamics.

6. *Inadequate Intermodal Connectivity* - The lack of seamless integration between different modes of transportation, such as road, rail, air, and waterways, hinders multi-modal logistics operations and limits the efficiency gains that could be achieved through intermodal transportation.

7. *Seasonal Variability and Weather Challenges* - India's diverse climate and seasonal variations pose challenges for logistics operations, especially during monsoon seasons, leading to disruptions in transportation networks, delays in deliveries, and increased risk of damage to goods.

8. *Infrastructure Congestion and Traffic Congestion* - Congestion on key transportation routes, including highways, ports, and airports, leads to delays in

transit times, increased fuel consumption, and higher operating costs for logistics providers, impacting overall efficiency and service levels.

9. *Limited Access to Finance and Capital* - Small and medium-sized logistics enterprises often face challenges in accessing finance and capital for investment in technology, infrastructure upgrades, and expansion initiatives, constraining their ability to compete and innovate in the market.

10. *High Logistics Costs* - The logistics costs in India are among the highest globally, primarily due to inefficiencies in transportation, warehousing, and inventory management, as well as high fuel prices and toll charges, which impact the competitiveness of businesses and hinder economic growth.

11. *Fragmented Supply Chain* - The logistics and shipping industry in India is highly fragmented, with multiple intermediaries involved in the supply chain, including freight forwarders, customs brokers, and transporters, leading to coordination challenges and communication gaps.

12. *Environmental Impact* - The logistics and shipping industry in India contributes to environmental pollution and carbon emissions due to inefficient transportation practices, reliance on fossil fuels, and inadequate sustainability measures, necessitating the adoption of eco-friendly practices and technologies.

1.2 Motivation

The motivations behind the project outlined are:

1. *Relevance to Real-World Challenges* - We are drawn to projects that address real-world challenges, and the logistics sector presents numerous inefficiencies that impact businesses and the environment. By working on this project, we have the opportunity to tackle these challenges head-on and make a tangible difference in the industry.

2. *Opportunity for Innovation* -The logistics sector is ripe for innovation, especially with the advancements in technology such as MapMyIndia APIs. We are excited about the prospect of applying these technologies to develop novel solutions that optimise truck selection and route planning, ultimately improving efficiency and reducing costs.

3. *Supporting Government Initiatives* - Inspired by the government's Unified Logistics Interface Platform, the research aims to contribute to the broader initiative of streamlining logistics operations. By aligning with government priorities, the project aims to support national objectives of economic development and sustainability.

4. *Learning and Skill Development* - Working on a project of this nature allows us to gain valuable hands-on experience and develop practical skills in areas such as data analysis, algorithm development, and software engineering. This project offers a platform for us to apply theoretical knowledge gained in the classroom to real-world problems, enhancing our academic and professional development.

5. *Impact and Contribution* - We are motivated by the opportunity to make a meaningful impact through our work. By developing a system that improves logistics efficiency and reduces environmental impact, we can contribute to broader sustainability goals and support the needs of stakeholders within the logistics ecosystem.

1.3 Objectives

The objectives behind the project outlined in the abstract are as follows:

1. *Designing an Intelligent System* - The primary objective is to design a system capable of intelligently selecting trucks within the logistics ecosystem. This involves developing algorithms and integrating various parameters to optimise truck selection based on user inputs such as location, product type, and weight.

2. *Enhancing Logistics Experience* - The project aims to enhance the overall logistics experience by considering factors such as fuel consumption, driver charges, and load balancing. By integrating these parameters into the decision-making process, the platform seeks to improve efficiency and reduce costs for logistics operators.

3. *Optimising Routes and Resource Allocation* - By optimising routes and resource allocation, the project aims to improve efficiency and minimise environmental impact within the logistics ecosystem. This involves leveraging advanced technologies and algorithmic analysis to optimise truck selection and route planning.

4. *Supporting Government Initiatives* - Inspired by the government's Unified Logistics Interface Platform, the project aims to support broader government initiatives aimed at streamlining logistics operations. By aligning with government priorities, the platform seeks to contribute to national objectives of economic development and sustainability.

5. *Providing Practical Insights* - The project aims to provide practical insights into the application of advanced technologies in the logistics domain. By bridging the gap between theoretical frameworks and practical implementation, the thesis seeks to demonstrate the feasibility and effectiveness of these technologies for industry practitioners and policymakers.

Overall, the objectives behind the project focus on designing an intelligent system, enhancing the logistics experience, optimising routes and resource allocation, supporting government initiatives, and providing practical insights for industry stakeholders.

CHAPTER 2

LITERATURE SURVEY

The literature survey constitutes a comprehensive exploration of the theoretical foundations, methodologies, and technological tools that underpin our project on system dynamics and causal diagrams, as well as key concepts in logistics management, challenges in logistics management, and the evolution from traditional to modern logistics applications. In this section, we delve into the fundamental principles of logistics management, examining the intricacies of supply chain optimization, inventory management, and transportation logistics. We explore the challenges faced by logistics practitioners, including issues related to infrastructure, cost efficiency, and sustainability. Additionally, we examine the transition from traditional logistics practices to modern, technology-driven approaches, highlighting the role of advanced technologies such as artificial intelligence, data analytics, and IoT in revolutionising the logistics landscape. Finally, we discuss the tech stack employed in our project, encompassing the integration of various software tools and platforms to facilitate the development and implementation of our logistics interface platform.

2.1 System Dynamics

System dynamics, pioneered by Jay W. Forrester in the 1950s, is a methodology for understanding the behaviour of complex systems over time. At its core, system dynamics involves the creation of computer-based models that simulate the dynamic interactions between various components of a system. These models facilitate the exploration of feedback loops, nonlinear relationships, and time delays, offering insights into system behaviour that traditional linear models may overlook.

In the realm of system dynamics, scholars and practitioners have applied this methodology across diverse domains, including economics, business management, environmental science, public policy, and healthcare. Notable works by Forrester, such as "Industrial Dynamics" and "Principles of Systems," laid the groundwork for understanding dynamic systems and feedback mechanisms.

Furthermore, the widespread adoption of system dynamics has led to the development of specialised software tools for modelling and simulation. Systems thinking, a core tenet of system dynamics, emphasises the interconnectedness of system components and the importance of understanding systemic behaviour holistically.

2.2 Causal Diagrams

Causal diagrams serve as graphical representations of causal relationships between variables within a system. These diagrams, which include causal loop diagrams (CLDs), directed acyclic graphs (DAGs), and Ishikawa diagrams, provide a visual framework for understanding how changes in one variable influence others. By delineating causal pathways and feedback loops, causal diagrams aid in identifying key drivers of system behaviour and predicting the consequences of interventions.

In particular, causal loop diagrams (CLDs) are widely used in system dynamics modelling. CLDs consist of interconnected nodes representing variables and arrows denoting causal relationships between them. Closed loops within CLDs indicate feedback mechanisms, which can be reinforcing (positive feedback) or balancing (negative feedback). Understanding these feedback loops is essential for comprehending the dynamic behaviour of complex systems. Moreover, CLDs can be converted into stocks and flows, allowing for a more detailed representation of the accumulation and flow of resources within the system.

2.3 Logistics System

A logistics system encompasses a network of interconnected processes, resources, and stakeholders involved in the planning, execution, and control of the flow of goods and services from point of origin to point of consumption. At its core, a logistics system aims to efficiently manage the movement and storage of products, ensuring timely delivery while minimising costs and maximising customer satisfaction. This involves the coordination of various activities, including transportation, warehousing, inventory management, packaging, and information exchange, to streamline operations and optimise resource utilisation.

Advanced technologies, including Geographic Information Systems (GIS), Internet of Things (IoT), and Artificial Intelligence (AI), have revolutionised logistics systems, enabling real-time tracking, predictive analytics, and automated decision-making. These technologies facilitate the optimization of routes, the selection of transportation modes, and the allocation of resources, leading to improved operational efficiency and cost savings. Moreover, they enable logistics operators to respond swiftly to dynamic market conditions, mitigate risks, and enhance the overall resilience of the supply chain.

2.4 Key Concepts in Logistics Management

Fig 2.1: Concepts in Logistics Management

Logistics management encompasses a multitude of interconnected processes and activities aimed at ensuring the efficient flow of goods and services from point of origin to point of consumption. At its core, logistics management involves the coordination and optimization of various functions such as transportation, inventory management, warehousing, and order fulfilment. One of the key concepts in logistics management is inbound and outbound logistics. Inbound logistics involves the management of materials and resources flowing into the organisation, including procurement, receiving, and storage. On the other hand, outbound logistics focuses on the movement of finished products from the organisation to the end customers, encompassing activities such as order processing, packaging, and distribution.

Another fundamental concept in logistics management is supply chain integration. In today's interconnected business environment, effective coordination and collaboration among supply chain partners are essential for achieving seamless operations and superior customer service. Supply chain integration involves aligning the activities of suppliers, manufacturers, distributors, and retailers to create a unified and responsive supply chain network. This integration enables real-time visibility into inventory levels, demand fluctuations, and production schedules, allowing organisations to make informed decisions and respond quickly to changing market conditions.

Transportation management is also a critical aspect of logistics management. It involves the planning, execution, and optimization of transportation activities to ensure the timely and cost-effective movement of goods. Transportation management encompasses various modes of transportation, including road, rail, air, and sea, each offering unique advantages and challenges. Effective transportation management involves route optimization, carrier selection, freight consolidation, and shipment tracking, all aimed at minimising transportation costs while maximising service levels.

Inventory management is yet another key concept in logistics management. Effective inventory management involves balancing the costs associated with holding inventory against the risks of stockouts and lost sales. Inventory management techniques such as just-in-time (JIT) inventory, economic order quantity (EOQ) models, and ABC analysis help organisations optimise inventory levels and improve operational efficiency. By maintaining optimal inventory levels, organisations can reduce carrying costs, minimise stockouts, and improve overall supply chain performance.

In summary, logistics management encompasses a range of key concepts aimed at ensuring the efficient and cost-effective flow of goods and services throughout the supply chain. By understanding and effectively managing these concepts, organisations can achieve greater operational efficiency, enhance customer satisfaction, and gain a competitive edge in the marketplace.

2.5 Challenges in Logistics Management

Logistics management, despite its crucial role in supply chain operations, faces a multitude of challenges that can hinder the efficiency and effectiveness of logistics processes. One significant challenge is the complexity of last-mile delivery. The last-mile delivery refers to the final leg of the delivery process, from a distribution centre to the end customer's doorstep. This stage often involves navigating through dense urban areas, dealing with traffic congestion, and coordinating deliveries to individual households. Additionally, the increasing demand for same-day and next-day delivery services has put additional pressure on logistics providers to streamline last-mile operations while maintaining cost-effectiveness.

Transportation capacity constraints pose another challenge in logistics management. Fluctuations in demand, seasonal peaks, and disruptions such as natural disasters or labour strikes can strain transportation networks and lead to capacity shortages. As a result, logistics providers may face difficulties in securing adequate transportation capacity to meet customer demand, leading to delays, increased costs, and service disruptions. Managing transportation capacity effectively requires proactive planning, collaboration with carriers, and leveraging technology to optimise routing and scheduling.

Supply chain disruptions represent a significant challenge for logistics management. Disruptions can arise from various sources, including natural disasters, geopolitical events, supplier failures, or unexpected changes in demand. These disruptions can disrupt the flow of goods, cause delays in deliveries, and impact customer service levels. Managing supply chain disruptions requires resilience and agility, with organisations implementing strategies such as risk mitigation measures, alternative sourcing options, and contingency plans to minimise the impact of disruptions and maintain business continuity.

Inventory management challenges also contribute to the complexity of logistics management. Balancing inventory levels to meet customer demand while minimising carrying costs and stockouts requires careful planning and forecasting. Inaccurate demand forecasts, volatile demand patterns, and SKU proliferation can lead to inventory imbalances, excess inventory, or stockouts. Effective inventory management strategies, such as demand forecasting, safety stock optimization, and inventory visibility technologies, are essential for addressing these challenges and optimising inventory levels throughout the supply chain.

In conclusion, logistics management faces numerous challenges that require proactive planning, collaboration, and innovation to overcome. By addressing challenges such as last-mile delivery complexity, transportation capacity constraints, supply chain disruptions, and inventory management issues, organisations can enhance the efficiency, resilience, and competitiveness of their logistics operations.

2.6 Traditional Logistics Applications vs Modern Logistics Application

Traditional logistics applications are often characterised by manual processes and paper-based documentation. In these systems, tasks such as order processing, inventory management, and transportation scheduling are typically performed manually, leading to inefficiencies and delays. For example, inventory tracking may involve manual counting and recording of goods, which can be time-consuming and prone to errors. Similarly, transportation planning may rely on manual route optimization, without considering real-time traffic conditions or dynamic factors.

Moreover, traditional logistics applications often lack real-time visibility into the supply chain. Information silos and fragmented systems make it challenging to track the movement of goods and monitor inventory levels accurately. This lack of visibility can lead to stockouts, overstocking, and disruptions in the supply chain, impacting customer service and profitability. On the other hand, modern logistics applications leverage advanced technologies to overcome these challenges and optimise supply chain operations. For instance, Geographic Information Systems (GIS) enable real-time tracking of shipments and provide insights into optimal routing based on factors such as traffic conditions, weather, and road closures. Internet of Things (IoT) devices, such as sensors and RFID tags, allow for continuous monitoring of inventory levels and conditions, enabling proactive replenishment and reducing stockouts.

Artificial Intelligence (AI) and machine learning algorithms play a crucial role in modern logistics applications by automating decision-making processes and optimising resource allocation. For example, AI-powered demand forecasting models analyse historical sales data, market trends, and external factors to predict future demand accurately. Similarly, AI-driven routing algorithms dynamically adjust transportation routes based on real-time data, minimising fuel consumption and delivery times. Cloud computing technology enables seamless integration and collaboration across the supply chain, enabling stakeholders to access real-time information from any location. This improves communication, coordination, and visibility, leading to greater efficiency and responsiveness in logistics operations.

Overall, modern logistics applications transform supply chain management by harnessing the power of advanced technologies to optimise processes, enhance visibility, and improve decision-making. By embracing innovation and digital transformation, organisations can gain a competitive edge in today's dynamic and fast-paced business environment.

2.7 Tech Stack

2.7.1 Vensim Software

Vensim software stands as a pivotal tool in our project's tech stack, facilitating the creation of stocks and flows models integral to our logistics interface platform. Vensim, renowned for its robust system dynamics modelling capabilities, provides a user-friendly interface for constructing dynamic simulation models that capture the interplay between various variables within complex systems. Leveraging Vensim's intuitive graphical interface, we developed intricate stock and flow diagrams to depict the accumulation and movement of resources throughout the logistics ecosystem. By incorporating real-time data and system feedback loops, these models enable us to simulate different scenarios and evaluate the potential impact of strategic interventions on system behaviour. Using Vensim, we created a comprehensive stocks and flows model specifically tailored to our logistics application, providing valuable insights into the dynamics of the supply chain and facilitating informed decision-making processes. Vensim's versatility and analytical power have been instrumental in optimising the performance of our logistics interface platform, ultimately enhancing efficiency and driving sustainable growth within the logistics sector.

2.7.2 ReactJS

ReactJS, commonly known as React, is an open-source JavaScript library primarily used for building user interfaces (UIs) for web applications. At its core, React revolves around the concept of components. Components are self-contained, reusable building blocks that encapsulate a piece of UI and its behavior. These components can be composed together to create complex UIs. React's component-based architecture promotes modularity, reusability, and maintainability, making it well-suited for large-scale applications.

Key Features and Advantages

1. *Virtual DOM* - React's use of a virtual DOM enables efficient updates to the actual DOM, minimising unnecessary re-renders and improving performance. Some other frameworks may not employ such a mechanism, leading to less efficient rendering.
2. *Component-Based Architecture* - React's component-based architecture promotes code reusability and modularity, making it easier to manage complex UIs. While many other frameworks also use components, React's approach has been widely praised for its simplicity and flexibility.
3. *JSX* - JSX allows developers to write HTML-like syntax within JavaScript code, making it easier to visualise and write UI components. Other frameworks may use templates or string-based approaches, which can be less intuitive and harder to maintain.

2.7.3 Material UI (MUI)

Material-UI is a popular React UI framework that implements Google's Material Design principles. It provides a set of pre-designed React components, styles, and utilities that developers can use to build attractive and responsive user interfaces quickly and efficiently.

Key Features and Advantages

1. *Material Design* - Material-UI adheres to Google's Material Design guidelines, which offer a comprehensive set of design principles for creating visually appealing and user-friendly interfaces. By using Material-UI, developers can ensure consistency in design across their applications and benefit from a familiar and intuitive user experience.
2. *Component-Based* - Material-UI is built with a component-based architecture, allowing developers to easily compose complex UIs from simple and reusable components. This promotes code reusability, maintainability, and scalability, making it easier to manage large-scale applications
3. *Responsive Design* - Material-UI components are designed to be responsive out of the box, ensuring that applications look and function well across different devices and screen sizes. Developers can leverage built-in responsive design features and media query support to create adaptive layouts with minimal effort.

2.7.4 NodeJS

Node.js is a server-side JavaScript runtime environment that allows developers to run JavaScript code outside the browser, on the server side. It's built on Chrome's V8 JavaScript engine and provides an event-driven, non-blocking I/O model that makes it lightweight and efficient for building scalable and high-performance applications.

Key Features and Advantages

1. *Asynchronous and Non-blocking* - Node.js uses an event-driven, non-blocking I/O model, which allows it to handle multiple connections simultaneously without getting blocked. This makes it highly efficient for handling I/O-heavy operations such as file I/O, network requests, and database operations.

2. *Scalability* - Node.js is well-suited for building highly scalable applications due to its non-blocking architecture. It can handle a large number of concurrent connections with minimal overhead, making it ideal for real-time applications and microservices architectures.
3. *Performance* - Node.js offers high performance and low latency, thanks to its event-driven, non-blocking architecture and the V8 JavaScript engine. It's particularly well-suited for building real-time applications such as chat applications, streaming services, and online gaming platforms.

2.7.5 MongoDB

MongoDB is a popular NoSQL database management system that stores data in flexible, JSON-like documents. It's designed for scalability, high performance, and ease of development and maintenance.

Key Features and Advantages

1. *Schema-less Design* - MongoDB is schema-less, meaning you can store documents without having to pre-define the structure. This flexibility is advantageous for evolving data models and dynamic schemas.
2. *Flexible Query Language* - MongoDB uses a powerful query language that supports complex queries, indexing, and aggregation operations. It also supports ad-hoc queries, making it easy to explore and analyse data.

CHAPTER 3

SYSTEM MODELLING

In the ever-evolving domain of modern logistics, the creation of a unified system demands a thorough comprehension of the intricate interplay between various factors influencing its operation. This chapter delves into our systematic methodology for modelling complex logistics systems, with a specific focus on the utilization of causal loop diagrams as a foundational framework. By exploring the multifaceted interdependencies inherent in logistics operations, we illuminate the path towards the development of a unified logistic system that not only upholds efficiency but also adapts seamlessly to dynamic changes in its operational landscape.

3.1 Utilization of Causal Loop Diagrams:

Causal loop diagrams, within the realm of system modelling, serve as invaluable tools for visualizing the intricate causal relationships between various variables constituting a system. These diagrams enable stakeholders to gain a comprehensive understanding of system dynamics by illustrating how changes in one variable can reverberate throughout the entire system. In the context of our unified logistic system project, we harnessed the power of causal loop diagrams to lay the groundwork for system development and guide our decision-making process.

At the outset of our project, we embarked on constructing a causal loop diagram that served as a blueprint for understanding the complex interdependencies within the logistics system. This diagram provided a visual representation of the causal relationships between key variables, including user satisfaction, system reliability, delivery efficiency, technological integration, and operational costs. By delineating these relationships, the causal loop diagram facilitated a holistic comprehension of the factors influencing the performance of our unified logistic system.

3.2 Variables and Interdependencies:

The variables incorporated into our causal loop diagram were meticulously selected to encompass diverse facets of the unified logistic system, reflecting the multifaceted nature of logistics operations. Each variable played a crucial role in shaping the overall performance and effectiveness of the system. For instance, user satisfaction emerged as a pivotal variable, influenced by factors such as order completion time, response time to queries, and system reliability.

Moreover, our causal loop diagram underscored the intricate interdependencies between these variables, highlighting how changes in one variable could cascade through the system and impact others. For instance, delivery time directly influenced user satisfaction, as prolonged delivery times could lead to decreased satisfaction levels. Simultaneously, delivery time was itself influenced by factors such as route efficiency, driver availability, and order volume.

Similarly, variables like route efficiency and warehouse capacity were revealed to be intricately intertwined, as improvements in route efficiency could enhance delivery times and, consequently, user satisfaction, while optimal warehouse capacity management was essential for ensuring timely order fulfilment and minimizing operational costs.

In essence, our causal loop diagram served as a comprehensive map of the interconnected variables shaping the unified logistic system. By elucidating the relationships and interdependencies between these variables, the diagram provided valuable insights into the systemic dynamics driving the performance of our logistics system, empowering us to make informed decisions and devise strategies for system optimization and enhancement.

3.3 Project Initiation and Progression:

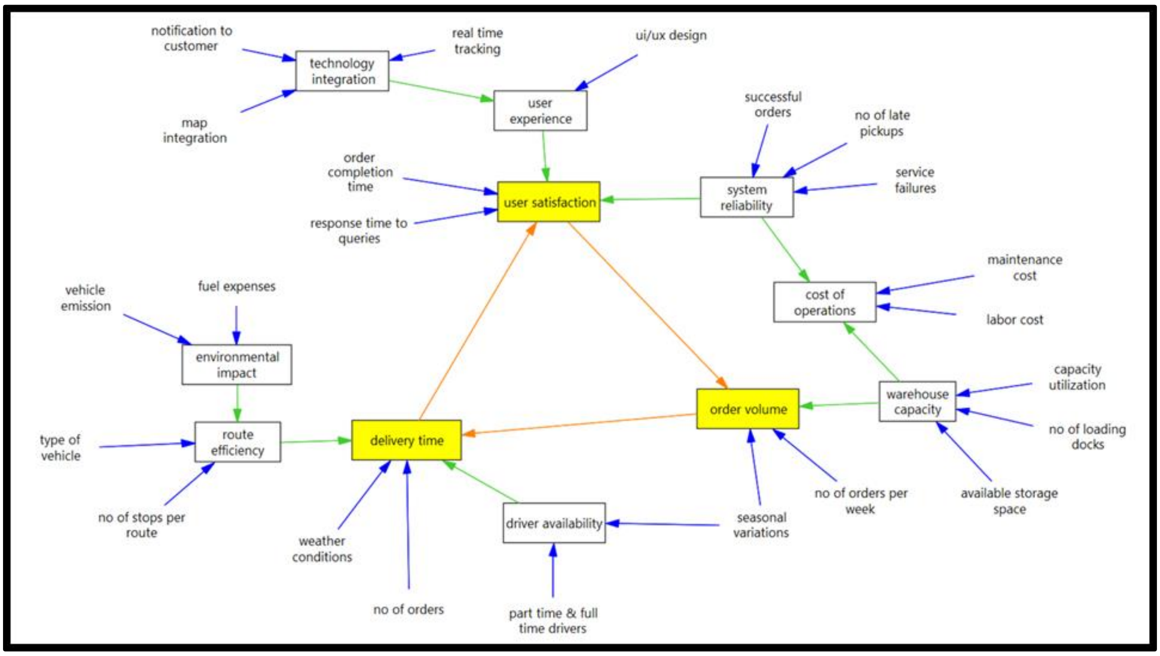

Fig 3.1 Casual loop diagram for unified logistic system

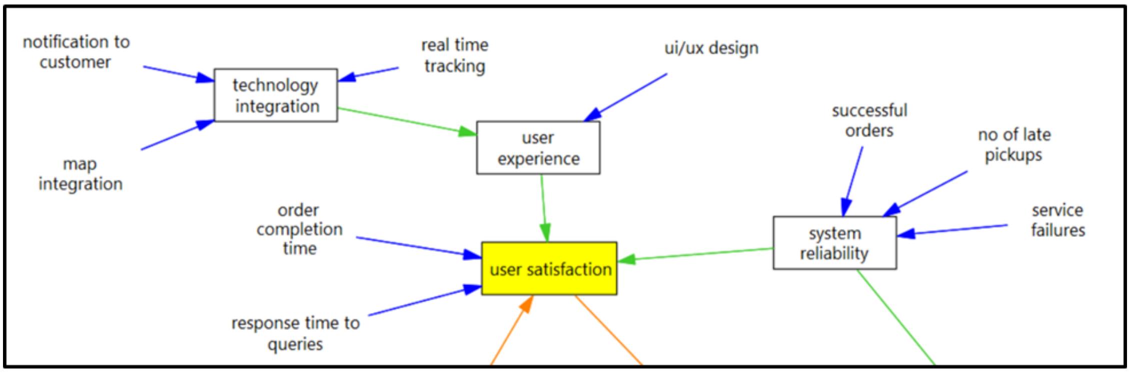

Fig 3.2 Casual loop diagram for user satisfaction

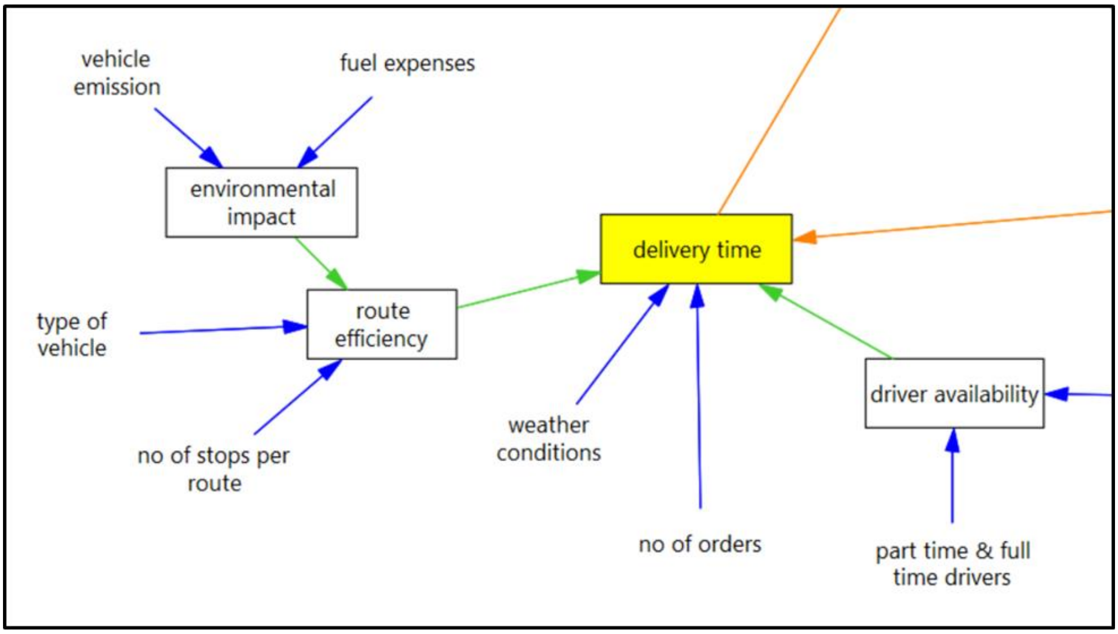

Fig 3.3 Casual loop diagram for delivery time

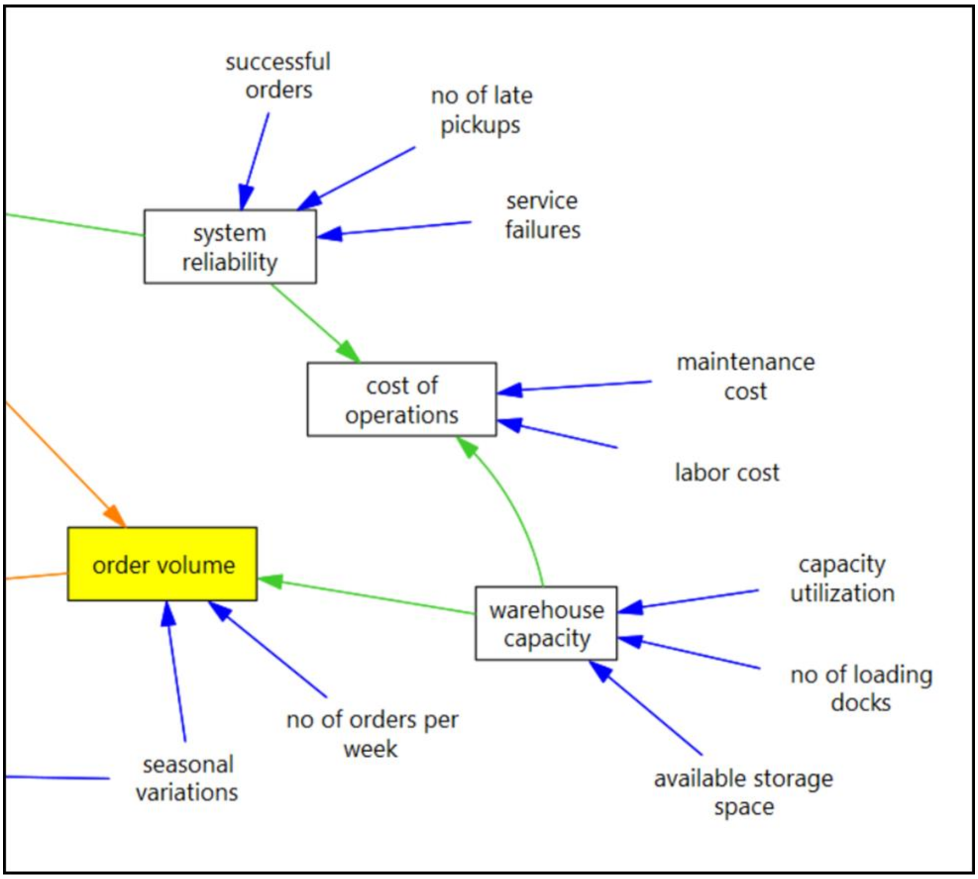

Fig 3.4 Casual loop diagram for order volume

Armed with the causal loop diagram as our guiding blueprint, the initiation of the unified logistic system project marked a strategic endeavour to translate conceptual insights into tangible solutions. Throughout the project lifecycle, from inception to implementation, the system model played an indispensable role in shaping our approach and guiding our decisions. Each stage of development was informed by the systemic understanding afforded by the causal loop diagram, ensuring that our efforts were directed towards addressing the core challenges and maximizing the potential of the logistics system.

For instance, during the design phase, the causal loop diagram served as a compass, guiding the development of the user interface (UI) with a keen focus on enhancing user experience (UX). Insights gleaned from the diagram elucidated the critical factors influencing user satisfaction, such as order completion time, response time to queries, and system reliability. Consequently, UI design considerations were tailored to optimize these factors, thereby fostering a seamless and intuitive user experience.

Furthermore, the system model informed the prioritization of development efforts, enabling us to allocate resources judiciously and tackle the most salient issues first. By identifying leverage points and causal relationships within the logistics system, the causal loop diagram facilitated a strategic approach to system development, ensuring that our endeavours were aligned with the overarching goals of enhancing efficiency, reliability, and user satisfaction.

3.4 Future Scope and Unimplemented Features:

While substantial progress has been made in implementing the core functionalities of the unified logistic system, the causal loop diagram has also served as a catalyst for identifying areas of future exploration and enhancement. Several features and initiatives, outlined within the diagram, present promising avenues for further development and optimization. These include:

1. *Advanced Route Optimization* - Leveraging machine learning algorithms to dynamically optimize delivery routes represents a key area of future development. By harnessing real-time traffic data, weather conditions, and order volume information, we aim to devise algorithms capable of continuously

adapting and refining delivery routes to minimize transit times and enhance overall efficiency.

2. *Environmental Sustainability Measures* -The promotion of environmental sustainability within logistics operations remains a pressing concern. Future iterations of the unified logistic system will prioritize the integration of eco-friendly initiatives, such as the adoption of electric or hybrid vehicles and the optimization of route efficiency to minimize fuel consumption and emissions. These measures align with our commitment to reducing the ecological footprint of logistics operations and fostering a greener, more sustainable future.

3. *Enhanced Driver Management* - Efficient driver management lies at the heart of logistics operations, influencing factors such as delivery times, service reliability, and overall operational efficiency. Future developments in driver management will focus on the implementation of advanced algorithms to optimize driver scheduling and allocation, considering factors such as driver availability, skill sets, and workload distribution. By harnessing data-driven insights, we aim to streamline driver operations and maximize productivity while ensuring a balanced and equitable workload distribution among drivers.

In conclusion, our journey in developing a unified logistic system has been greatly enriched by our systematic approach to system modelling, epitomized by the utilization of causal loop diagrams. Through the meticulous construction and analysis of these diagrams, we have gained invaluable insights into the intricate web of interdependencies shaping logistics operations, paving the way for informed decision-making and strategic development.

The utilization of causal loop diagrams served as a guiding light, illuminating the path forward at each stage of our project. From conceptualization to implementation, these diagrams provided a comprehensive framework for understanding the complex relationships between variables such as user satisfaction, system reliability, delivery efficiency, and operational costs. Armed with this systemic understanding, we were able to prioritize development efforts, allocate resources judiciously, and design solutions that address the core challenges of the logistics landscape.

As we look towards the future, the causal loop diagrams continue to serve as beacons of inspiration, pointing us towards areas of further exploration and enhancement. The identified future scope, encompassing initiatives such as advanced route optimization, environmental sustainability measures, and enhanced driver management, represents promising avenues for innovation and optimization within our unified logistic system.

In essence, our system modelling approach, anchored by causal loop diagrams, has been instrumental in guiding the development of our unified logistic system project. By providing a holistic view of system dynamics and interdependencies, these diagrams have empowered us to make informed decisions, drive meaningful change, and pave the way for a more efficient, resilient, and user-centric logistics ecosystem. As we continue to refine and expand the capabilities of our system, the lessons learned from our system modelling approach will serve as a cornerstone for ensuring its effectiveness and adaptability in meeting the evolving needs of logistics operations in the years to come.

Chapter 4

Methodology

The methodology employed in this project encapsulates a comprehensive approach aimed at developing a user-friendly shipping website that optimises truck selection based on various user inputs while integrating MapMyIndia APIs. This chapter provides a detailed insight into the project setup, frontend and backend development, as well as the seamless integration of MapMyIndia APIs.

4.1 Project Setup:

To kickstart the project, an extensive planning phase was conducted. Key features such as user-friendly interface, real-time truck selection, and seamless MapMyIndia API integration were identified during this phase.

After careful consideration of various options, React.js was chosen for the frontend development due to its component-based architecture, efficient rendering, and strong community support. For the backend, Node.js with Express.js was selected for its non-blocking I/O and scalability. MongoDB, a NoSQL database, was chosen for its flexibility and compatibility with Node.js applications.

The development environment was configured by installing necessary tools such as Node.js, npm (Node Package Manager), and MongoDB. Visual Studio Code was chosen as the primary Integrated Development Environment (IDE) for its robust features and extensions tailored for React.js and Node.js development. Git version control system was utilised for collaborative development, with GitHub serving as the remote repository.

4.2 Frontend Development:

The frontend implementation of the shipping website comprises two distinct dashboards: one for users and another for administrators. The user dashboard is designed to facilitate the input of shipment details, including source and destination locations, product type, and weight, while the admin dashboard serves as a management interface for overseeing logistics operations.

4.2.1 User Dashboard

The user dashboard features a two-step form, implemented using the Form.jsx component. This form allows users to input location details and order specifics seamlessly. The Form.jsx component encompasses several sub-components tailored to different aspects of the user's interaction:

4.2.1.1 Location Details Input

Users can provide their source and destination locations through Form1.jsx. This component offers two options for location input: either via a direct address or through latitude and longitude coordinates. The choice between these options is facilitated by toggling between Form1.jsx and LatLongForm.jsx, enabling users to select the method most convenient for them.

4.2.1.2 Confirmation of Source Location

In order to ensure the accuracy and convenience of inputting source locations, users are provided with visual confirmation options. This verification process is essential for guaranteeing the precision of shipping logistics. The implementation of this feature diverges slightly depending on whether users input their source location directly as an address or via latitude and longitude coordinates.

- **For Form 1.jsx (Address Input)**
  When users opt to provide their source location as a direct address within Form1.jsx, the confirmation process is facilitated by Map1.jsx. This component integrates seamlessly with the form, offering users a visual map interface. Through this interface, users can pinpoint their exact source location by navigating the map and selecting the desired address. Map1.jsx ensures that the address provided by the user is accurately represented, enhancing the overall reliability of the shipping information.
- **For Form 2.jsx (Latitude-Longitude Input)**

Alternatively, when users choose to input their source location using latitude and longitude coordinates within Form2.jsx, the confirmation process is managed by Map2.jsx. This component is specifically designed to accommodate latitude and longitude inputs, providing users with a map interface tailored to this method of location specification. Users can visually confirm their source location by entering the corresponding latitude and longitude coordinates, ensuring alignment between the provided coordinates and the displayed map representation. Map2.jsx enhances the precision of location confirmation, particularly for users who prefer specifying locations using geographic coordinates.

4.2.1.3 Order Details Input

Once location details are confirmed, users proceed to input order specifics through Form2.jsx. This includes selecting the type of product being shipped from a dropdown menu, which dynamically recommends suitable trucks based on the selected item type. Additionally, users can specify the weight of the product being shipped to further refine the selection process.

4.2.1.4 User Dashboard Submission Process

Upon completing the two-step form on the user dashboard, users have the option to submit their shipment details. This action triggers the frontend to process the entered information and send it to the backend server.

- **Backend Processing and UUID Generation**

  Upon receiving the submitted data, the backend server undertakes several tasks, including validating the inputted information and generating a unique identifier (UUID) for the submitted order. This UUID serves as a reference for tracking the shipment throughout its journey.
- **Hub Allocation**

  Simultaneously, the backend system calculates and assigns the nearest hub for handling the shipment based on the provided source and destination locations. This ensures optimal routing and logistical efficiency, minimising transit times and enhancing overall service quality.
- **Redirection to Success Page and Page Content**

Upon successful processing of the shipment request, the user is automatically redirected to a dedicated success page. This page serves as a confirmation of the successful submission of the order and provides users with pertinent details regarding their shipment, including the generated UUID and information about the allocated hub. This information empowers users with visibility into the logistics process and instils confidence in the reliability and transparency of the shipping service.

4.2.2 Admin Dashboard

The admin dashboard provides comprehensive oversight of the shipping service, allowing administrators to manage pickup and drop-off hubs across various states. The dashboard offers functionality to view hubs within specific states, visualise order locations, and optimise truck routes based on selected product types. Leveraging backend algorithms such as the Travelling Salesman Problem (TSP), the admin can efficiently plan truck routes to collect orders from designated hubs.

4.2.2.1 State and Hub Selection

Administrators can navigate through the dashboard to view available states where the shipping service operates. Upon selecting a state, the dashboard displays hubs located within that state, facilitating efficient management of logistics operations. For instance, clicking on Maharashtra reveals hubs in cities like Nagpur, Mumbai, and Aurangabad.

4.2.2.2 Order Visualization and Details

Upon further selection of a hub, a map interface pops up, marking all order locations assigned to that specific hub. This visualisation is implemented in the MarkerMap.jsx component, which displays order markers on the map. Concurrently, a table displays order details, including the originating city/address and the unique identifier (UUID) of the customer who placed the order. This visualisation enables administrators to track orders efficiently and manage logistics effectively.

4.2.2.3 Filtering by ItemType

An additional feature allows administrators to filter orders based on product types. By selecting a specific product type, such as clothing, only orders matching that criteria are displayed on the map and table. This functionality enhances the dashboard's usability by providing targeted information relevant to the administrator's focus.

4.2.2.4 Route Optimization

Using backend algorithms like the Travelling Salesman Problem (TSP), the admin can optimise truck routes for order collection. The MapDirection.jsx component is responsible for displaying the best route for the truck. By considering factors such as distance and order locations, the system generates the most efficient route for truck drivers to follow, minimising travel time and fuel consumption. This optimization enhances operational efficiency and reduces costs associated with order pickups.

4.2.3 Component Structure

The frontend architecture follows a modular approach, organised within the 'components' directory:

- **Form Component:**
  This directory contains Form.jsx, serving as the main component for user interaction. It further encapsulates the 'subForm' directory, housing Form1.jsx, Form2.jsx, and LatLongForm.jsx, each catering to specific aspects of the form input process.
- **Map Component:**
  The 'Map' directory encompasses Map1.jsx and Map2.jsx, responsible for confirming source locations for Form1.jsx and Form2.jsx, respectively. This segregation ensures that the confirmation process aligns with the chosen method of location input, enhancing user experience and accuracy.
  Additionally, the Map component includes MarkerMap.jsx and MapDirection.jsx components, essential for the functionality of the admin dashboard.
  MarkerMap.jsx facilitates the visualisation of order locations for a specific hub, while MapDirection.jsx displays optimised truck routes based on backend

algorithms like the Travelling Salesman Problem (TSP), enhancing logistics management efficiency.

- **Pages Directory:**
  The 'pages' directory contains the Admin.jsx component, serving as the main component for the admin dashboard. This component acts as the entry point for administrative functionalities, providing a cohesive interface for managing hubs, visualising orders, and optimising truck routes.

4.3 Backend Development:

The backend development phase of the project focused on implementing the core functionality of the shipping website, including data management, route optimization, and integration with external APIs. This section provides an overview of the key functions developed to facilitate efficient route selection and logistics management.

4.3.1 Connecting to MongoDB:

The first step in backend development involved establishing a connection to the MongoDB database.

connect (): This function establishes a connection to MongoDB using Mongoose, employing asynchronous handling to prevent event loop blocking. It utilises provided connection URI and options for compatibility and stability. Error handling is integrated to manage potential connection issues, while successful connection is confirmed through console logging.

4.3.2 Backend functions

The backend functions enable seamless logistics management within the shipping website. They encompass features such as nearest hub selection, efficient data storage and retrieval from MongoDB, and route optimization using TSP algorithms. Together, these components streamline order processing, enhance route planning, and optimise truck allocation, contributing to a more efficient and cost-effective shipping experience.

4.3.2.1 findNearestHub (): This function calculates the nearest logistics hub based on a given source location (eLoc_src) and state (state_src). It iterates through hubs within the specified state, retrieving their geographic coordinates from MapMyIndia geocode API.

Utilising MapMyIndia’s distance API, it computes the distance between the source location and each hub, selecting the one with the shortest distance as the nearest hub. The function handles errors gracefully and returns the name of the nearest hub for further processing.

4.3.2.2 assignNearestHub () : This function determines the nearest logistics hub based on either location names or latitude-longitude coordinates provided. Then, it iterates through the provided source locations or coordinates, fetching their respective city names or reverse-geocoding to obtain city names. For each location, it calculates the nearest hub using the **findNearestHub()** function and stores the order details along with the assigned hub in the database. The function returns the assigned hub and a unique user ID for further processing. Error handling is implemented to catch and log any encountered errors.

4.3.2.3 tsp (): This function implements the Travelling Salesman Problem (TSP) algorithm to find the shortest path that visits all cities exactly once and returns to the starting city. It takes a weighted adjacency matrix representing the graph of cities and their distances as input. Using dynamic programming with memoization, it recursively explores all possible paths, efficiently computing the shortest path and its cost.

4.3.2.4 getRoute (): This function asynchronously computes the optimal route between multiple locations using the Travelling Salesman Problem (TSP) algorithm. It initialises a distance matrix (graph) based on the input locations and calculates the distances between them using MapMyIndia’s distance API. Once the distances are obtained, the **tsp ()** function is called to find the minimum cost and path for visiting all locations. Finally, the function returns an object containing the minimum cost (minCost) and the sequence of locations (minPath) representing the optimal route.

4.3.3 Backend API’s

4.3.3.1 /findNearestHub : This route handler processes a POST request, extracting parameters from the request body including source location, latitude-longitude coordinates, flags indicating coordinate usage, truck type, item type, and weight. It then invokes the **assignNearestHub()** function with these parameters to determine the nearest logistics hub based on the provided data. Upon receiving the result, it sends a response containing the assigned hub and a unique user ID.

4.3.3.2 /getOrders : This route handler processes a POST request to retrieve orders based on specific criteria, such as item type or hub location. If the item type is set to 'default', it fetches orders from the database for the specified hub without considering item type constraints. Otherwise, it retrieves orders matching both the hub location and the specified item type. It then constructs a response containing relevant order details, including source location, destination location, and unique order identifier (UUID), before sending it to the client. Error handling is implemented to manage any encountered errors during the database query process.

4.3.3.3 /getOrderRoute : The route handler processes a POST request to compute the optimal route for picking orders based on a given list of pickup locations (eLocs of locations are given as input). It invokes the **getRoute()** function asynchronously to calculate the shortest path. Once the optimal route is determined, the function sends a response containing the result, including the minimum cost and the sequence of locations representing the optimal route, back to the client.

4.3.3.4 /getTruckInfo : This route handler processes a POST request to determine the number of trucks required for picking orders from a specified hub, considering the item type. It retrieves orders matching the provided hub and item type from the database and calculates the total weight of the orders. Based on the weight and the capacity of the truck corresponding to the item type, it computes the number of trucks required to transport the orders. The function then sends the calculated number of trucks as a response to the client.

4.3.4 MapMyIndia API Integration:

4.3.4.1 API Setup:

The integration of MapMyIndia APIs within the project is facilitated through a .env file, which securely stores the necessary credentials required to access the APIs. The .env file contains the client ID and client secret, ensuring the secure authentication and authorization process when making requests to MapMyIndia APIs.

4.3.4.2 APIs Used:

The project leverages several MapMyIndia APIs to enhance its functionality:

Geocoding API: This API is utilised for converting addresses into geographic coordinates (eLocs), enabling accurate location-based services within the shipping website.

Reverse Geocoding API : Used for converting geographic coordinates into human-readable addresses, this API facilitates location identification and address resolution in the application.

Distance Matrix API: By calculating the distances between multiple locations, this API enables route optimization and logistics planning within the shipping website, contributing to efficient truck selection and delivery routing.

4.3.4.3 Authentication and Access Control:

The project ensures secure access to MapMyIndia APIs by utilising the provided client ID and client secret for authentication. Access tokens are generated and managed securely to authenticate API requests, maintaining the confidentiality and integrity of the communication between the application and MapMyIndia services.

4.3.4.4 Endpoint Configuration:

Each API endpoint is configured with the appropriate base URL and parameters required for making requests, such as addresses, geographic coordinates, and transportation modes. These endpoints are seamlessly integrated within the backend functions and route handlers, enabling real-time interaction with MapMyIndia services to provide accurate and reliable location-based functionalities within the shipping website.

CHAPTER 5

IMPLEMENTATION

5.1 Strategic Hub-Centric Logistics Management

Upon booking an order from the frontend, the data swiftly moves to the backend, initiating a crucial process: determining the nearest hub to the order. Our approach revolves around an efficient logistics strategy where each order is initially directed to its nearest hub. Subsequently, from this central hub, the order is dispatched to its ultimate destination. This model optimises delivery routes, enhances efficiency, and ensures timely order fulfilment.

In our system, we've strategically established hubs in four key states, with each state hosting three hubs. These hubs are strategically located within major cities, ensuring comprehensive coverage across the respective states. The selection of hub locations considers factors such as geographical coverage, population density, and accessibility. In our logistics network, we've strategically established hubs in four key states: Maharashtra, Madhya Pradesh, Gujarat, and Rajasthan. Each state hosts three hubs, strategically located within major cities to ensure comprehensive coverage and efficient logistics operations. In Maharashtra, our hubs are situated in Mumbai, Aurangabad, and Nagpur. These cities are pivotal hubs of economic activity, providing strategic access to major transportation routes and serving as key distribution points within the state. Madhya Pradesh boasts hubs in Bhopal, Indore, and Jabalpur. These cities are strategically positioned to serve as central logistics hubs, facilitating efficient transportation and distribution of goods throughout the state. Gujarat, with its dynamic economy, hosts hubs in Ahmedabad, Rajkot, and Surat. These cities are vibrant commercial centres, offering strategic advantages in terms of connectivity, infrastructure, and market access. In Rajasthan, our hubs are located in Jaipur, Udaipur, and Jodhpur. By strategically positioning hubs in these states, we aim to optimise logistics operations, minimise transit times, and ensure timely delivery of goods to customers. Our hub selection criteria consider factors such as geographical coverage, population density, and infrastructure, ensuring efficient logistics management and exceptional service delivery.

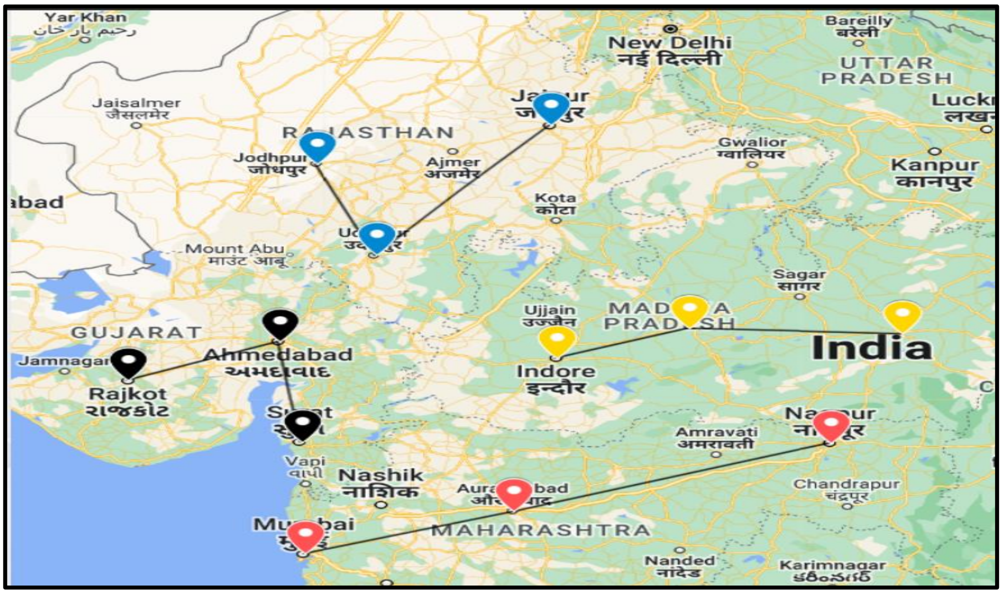

Fig 5.1(a): Hubs Location

To determine the nearest hub for a particular order, we leverage geolocation data and distance calculation algorithms. The `assignNearestHub` function, as implemented in our code, calculates the distance from the order's source location to all hubs within the corresponding state. By iterating through each hub and computing the distance, we identify the hub with the minimum distance from the order's location. This nearest hub is then assigned to the order, ensuring optimal routing and efficient logistics management. This hub-centric approach streamlines logistics operations, minimises transit times, and maximises operational efficiency. Moreover, by leveraging hubs as central points of distribution, we can optimise route planning, reduce transportation costs, and enhance overall supply chain performance.

The implementation of this logistics model is facilitated by advanced technologies and algorithms, such as the one depicted in the provided code snippet. By utilising APIs for geocoding, distance calculation, and hub selection, our system can accurately identify the nearest hub for each order in real-time. This ensures that orders are swiftly routed to their designated hubs, where they are subsequently processed and dispatched for final delivery.

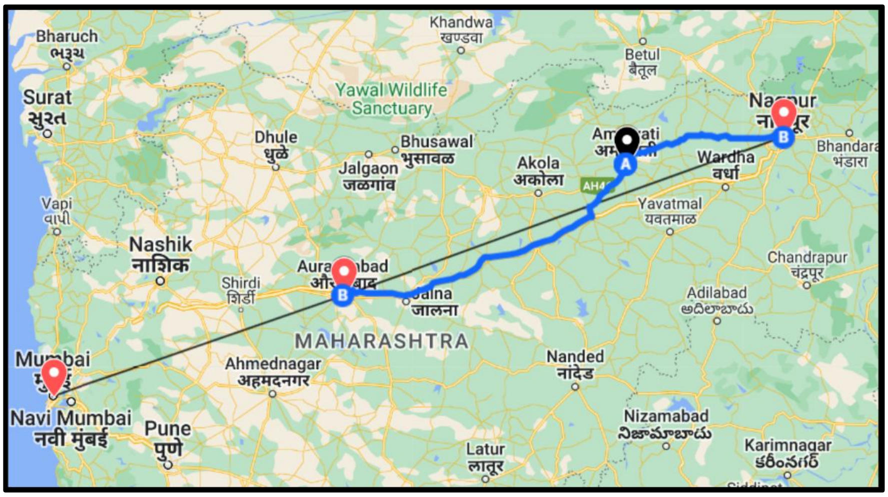

Fig 5.1(b): Distance of a point from two hubs

Furthermore, upon assigning the nearest hub, a unique identifier is generated for the order. This unique identifier serves as a tracking mechanism, enabling real-time monitoring of the order's progress throughout the delivery process. All pertinent information, including the order details, assigned hub, and unique identifier, is meticulously stored in the database, ensuring data integrity and accessibility for future reference.

Overall, our approach to hub selection and order assignment embodies a strategic logistics framework aimed at enhancing operational efficiency, minimising transit times, and delivering exceptional service to our customers. By leveraging technology and data-driven insights, we optimise logistics processes, streamline order management, and uphold our commitment to delivering excellence in logistics services.

5.2 How to book an Order

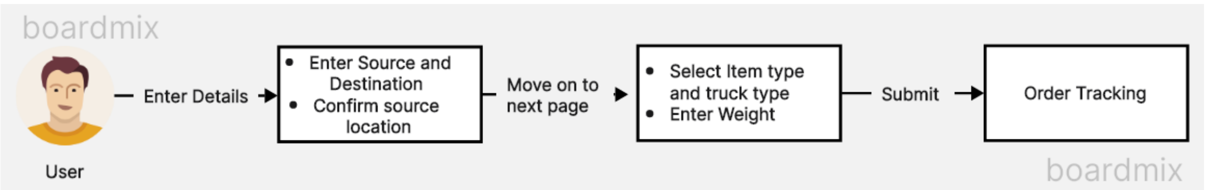

Fig 5.2(a). Steps to book an order

5.2.1 Enter Location Details

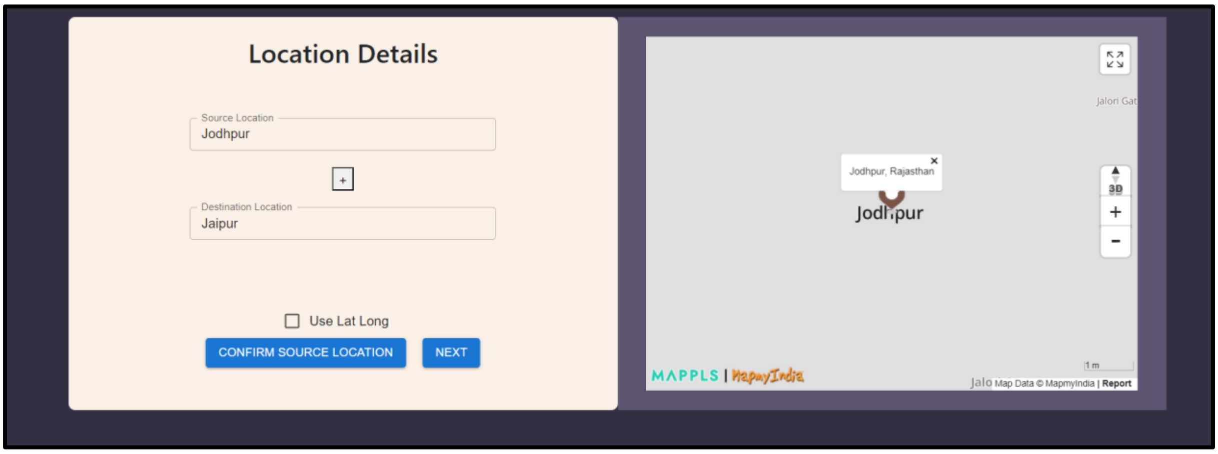

Fig 5.2(b): Location Details as names

To initiate the order booking process, you'll begin by entering location details. This involves specifying both the source and destination locations. Once entered, you can confirm your source location by clicking the "Confirm Source Location" button. A map on the right side will display your precise pickup location, allowing you to verify it accurately.

Fig 5.2(c): Multiple inputs

If you have multiple pickup points for your order, you can enter each of them as separate source locations. All of these pickup points will have the same destination location. This allows to consolidate orders with different pickup locations but the same destination, streamlining the logistics process for efficiency and convenience.

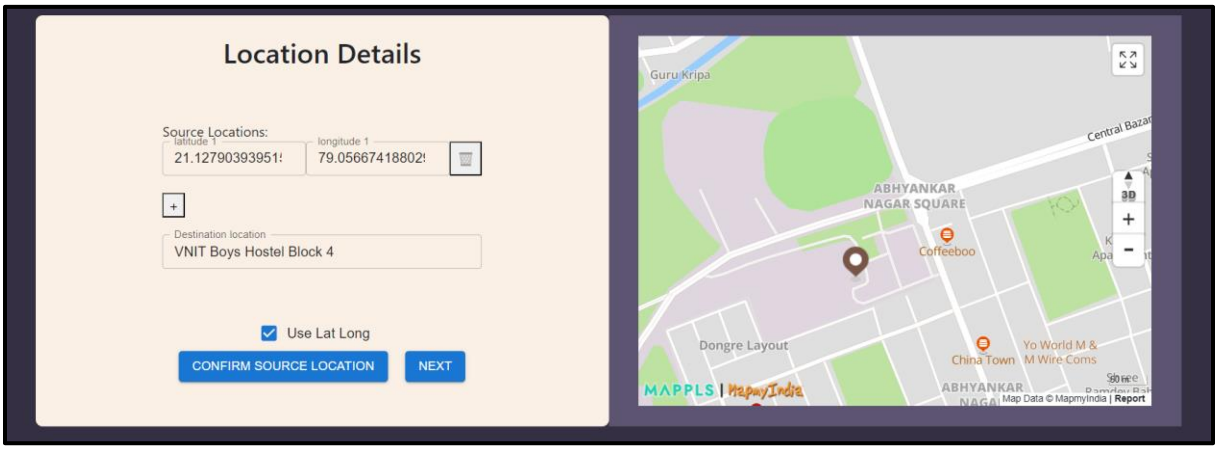

Fig 5.2(d): Location Details as latitude and longitude

In addition to entering location names, we also have the option to input the longitude and latitude coordinates of the product. This allows for more accurate pinpointing of locations, especially in cases where specific addresses might not be available or precise coordinates are preferred.

5.2.2 Enter Order Details

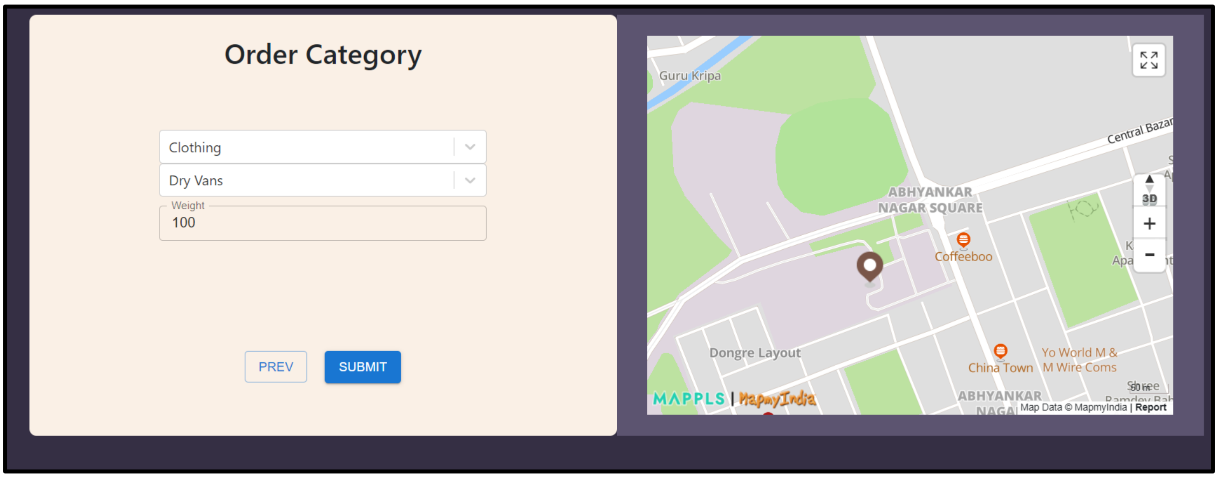

Fig 5.3(a): Order Category

Fig 5.3(b): Product to truck matching

Once location details are confirmed, proceed to the next page to select the type of product. Depending on the selected product, the system will provide you with suitable truck options to accommodate your needs. Following this, input the weight of the product in tons that you wish to transfer, and then submit the data.

5.2.3 Dashboard for Order tracking

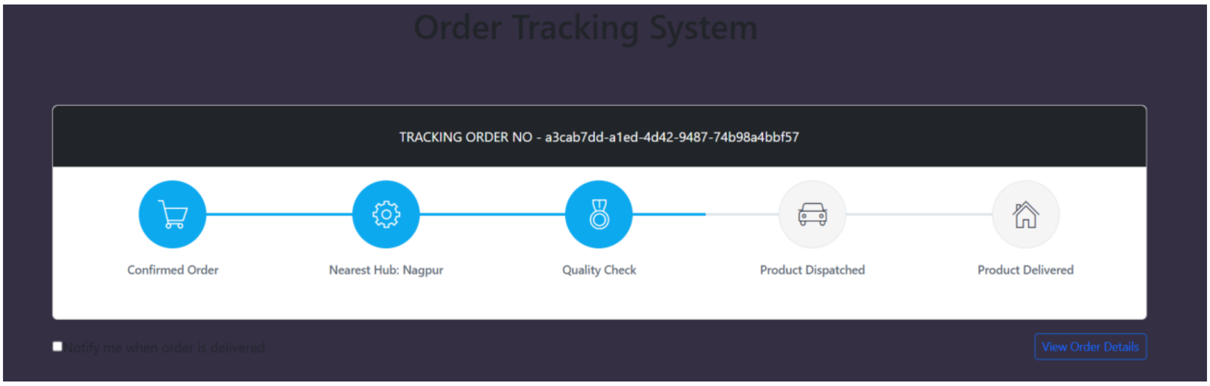

Fig 5.4: Order Tracking Dashboard

Once you've submitted the form, you'll be redirected to a dynamic dashboard within our order tracking system. This dashboard serves as your control centre, offering real-time insights into the status and whereabouts of your order. Moreover, it provides valuable data regarding the nearest hub to your order. It's from this strategic hub that your order begins its journey, branching out to reach its various source locations efficiently.

5.2.4 Processing Order Details

```
{
  srcLoc: [ 'Jodhpur', 'Kota' ],
  srcLatLong: [ { latitude: '21.12790393951505', longitude: '79.05667418802912' } ],
  destLoc: 'VNIT Boys Hostel Block 4',
  itemType: 'Clothing',
  truckType: 'Dry Vans',
  weight: '100',
  latLongUsed: true
}
a3cab7dd-a1ed-4d42-9487-74b98a4bbf57
latLong =  { latitude: '21.12790393951505', longitude: '79.05667418802912' }
Nagpur
eLoc---- 2YZRDY
hub---- Nagpur
```

Fig 5.5: Order Details as JSON

Behind the scenes, our system will utilise the source location to determine the closest hub. This information, along with other order details, will be stored into the MongoDB database. Once the data is securely stored, the backend will generate a unique identification for the order and hub details will be sent back to the frontend so that it can display it.

5.3 Backend

This section will provide an in-depth look at the backend processes involved in managing orders. We'll explore how order data is stored in the database and delve into methods for optimising logistics, including determining the shortest route to efficiently fulfil all orders.

5.3.1 Database Schema

The schema for storing Order and Hub information is outlined as follows:
Order: Each order represents a unique transaction within the logistics system, comprising essential details. With detailed information orders are systematically processed and routed through the logistics network to their respective destinations.

```
const hubSchema = new mongoose.Schema(
  {
    state: String,
    city: String,
  },
  { collection: "Hubs" }
);

const OrderSchema = new mongoose.Schema({
  eloc: String,
  lat: String,
  long: String,
  srcloc: String,
  uuid: String,
  hub: String,
  state: String,
  truckType: String,
  itemType: String,
  weight: String
});
```

Fig 5.6: Database Schema

Eloc: A 6-character digital address provided by MapmyIndia, simplifying complex addresses.

Lat: Latitude of the order location.

Long: Longitude of the order location.

Srcloc: Source location of the order.

Uuid: Unique identifier associated with each product.

State: State in which the order is placed.

TruckType: Type of truck assigned for transportation.

ItemType: Type of item being transported.

Weight: Weight of the product being transported.

Hub: The hub serves as a pivotal point in the logistics network, facilitating the efficient distribution and management of orders. Situated strategically within a specific state, each hub is typically located in a prominent city, ensuring accessibility and connectivity to various source and destination locations. As the central node in the logistics chain, hubs play a crucial role in streamlining operations, consolidating shipments, and optimising transportation routes.

State: Name of the state where the hub is located.

City: City assigned as the hub location.

5.3.2 Managing Order

Fig 5.7: Managing Order Flow

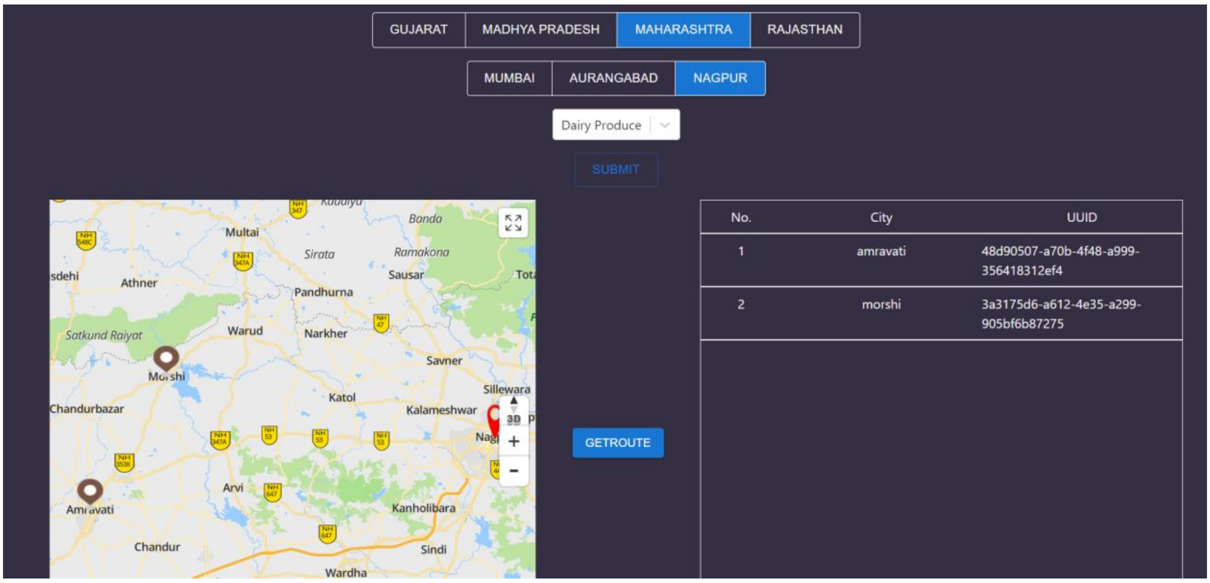

Fig 5.8(a): Admin Page with total orders

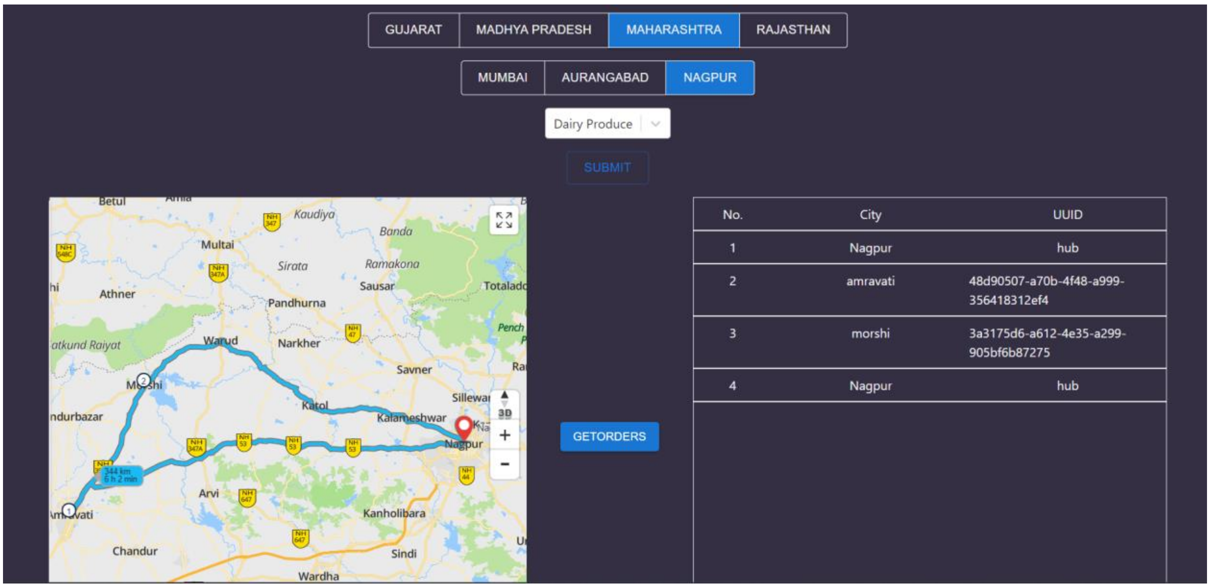

Fig 5.8(b): Admin Page with order route

As an administrator, you have the privilege to access and manage orders destined for specific hubs within our system. Upon logging in, you can choose the desired state and select the corresponding hub to view its associated orders. Additionally, for enhanced clarity, you have the option to filter orders based on product type.Upon submission, the system will present a comprehensive list of orders allocated to the chosen hub. These orders are visually represented on an interactive map, with each source location clearly labelled. Furthermore, the hub itself is prominently marked with a distinctive red label for easy identification. To further optimise logistic planning, the system offers a "Get Route" function. Upon activation, it utilises advanced algorithms to solve the Travelling Salesman Problem, determining the shortest and most efficient route for order delivery. We'll input the locations of all the cities into a Travelling Salesman Problem (TSP) solver. This solver will then crunch the data in the background, determining the most efficient route that visits each city exactly once and returns to the starting point. This process involves finding the minimum path length among all possible permutations of city visits. By utilising TSP algorithms, we're essentially streamlining the task of optimising travel routes, especially in scenarios like logistics, where finding the shortest path among multiple destinations is crucial for minimising time and resources. The resulting optimised route is then visually displayed on the map, providing valuable insights for logistical decision-making.

5.3.3 Travelling Salesman Problem

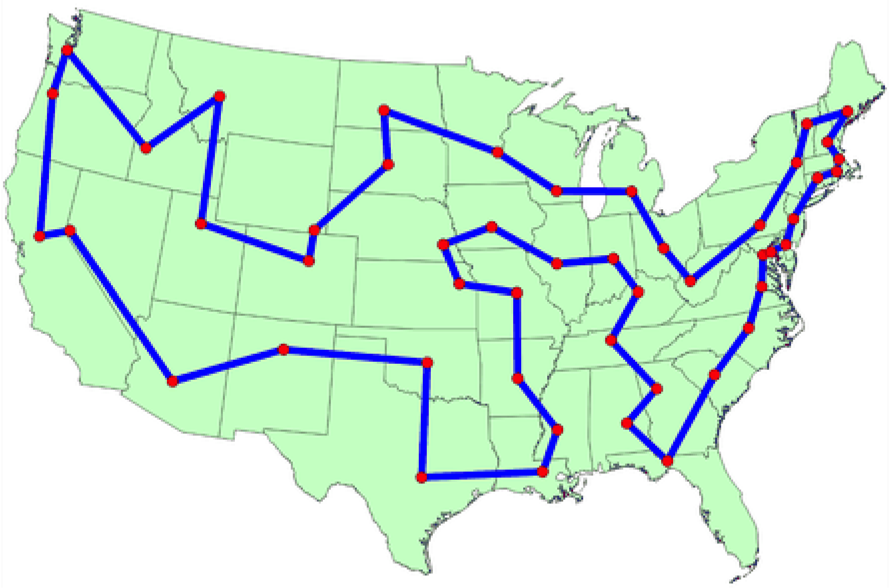

Fig 5.9: Travelling Salesman Problem

The Travelling Salesman Problem (TSP) is a classic problem in computer science and mathematics. It's a combinatorial optimization problem that seeks to find the shortest possible route that visits a set of given cities and returns to the original city. In its simplest form, it can be stated as follows. Given a list of cities and the distances between each pair of cities, the task is to find the shortest possible route that visits each city exactly once and returns to the original city. The problem is known to be NP-hard, meaning that there is no known efficient algorithm that can solve all instances of the problem in polynomial time. However, there are several approximation algorithms and heuristic approaches that can find good solutions for many instances of the problem.

CHAPTER 6

RESULTS AND CONCLUSION

6.1 Results

The frontend development efforts focused on enhancing user experience and data accuracy through the implementation of a user-friendly two-step form for inputting shipment details. This was achieved by integrating React components such as Form.jsx, Form1.jsx, and Form2.jsx, facilitating streamlined form navigation and data input. Interactive map features, powered by Map1.jsx and Map2.jsx, were incorporated to visually confirm source locations, further improving user convenience. Additionally, dynamic truck recommendations based on selected product types were integrated to aid users in making informed decisions and optimizing truck selection, thereby enhancing the overall efficiency of the shipping process.

On the backend, a stable connection to a MongoDB database was established to ensure efficient data storage and retrieval, promoting reliability and scalability. Backend functions such as findNearestHub(), assignNearestHub(), tsp(), and getRoute() were developed to enable seamless logistics management and route optimization. Backend APIs including /findNearestHub, /getOrders, /getOrderRoute, and /getTruckInfo facilitated communication between frontend and backend components, enabling various functionalities. Utilization of route optimization algorithms like the Travelling Salesman Problem (TSP) further enhanced route planning and truck allocation, contributing to improved efficiency and cost-effectiveness in shipping operations.

The integration of MapMyIndia APIs provided significant enhancements to the project's functionality. Through secure setup and storage of necessary credentials in a .env file, the project ensured secure access to MapMyIndia APIs, enhancing overall application security. Leveraging APIs such as Geocoding, Reverse Geocoding, and Distance Matrix, the project enabled precise location-based services, accurate address resolution, and efficient route optimization. By employing client ID, client secret, and access token management, secure authentication and access control were ensured, maintaining confidentiality and integrity in communication between the application and MapMyIndia services. Each API endpoint was configured with appropriate parameters, seamlessly integrated within backend functions and route handlers, enabling real-time interaction with MapMyIndia services to provide accurate and reliable location-based functionalities within the shipping website.

In summary, the integration of frontend enhancements, backend functionalities, and MapMyIndia APIs has significantly improved the user experience, efficiency, and reliability of the shipping website. From streamlined data input and form navigation to seamless logistics management, route optimization, and secure location-based services, the project has successfully delivered a comprehensive solution for enhanced shipping operations.

6.2 Conclusion

The integration of frontend, backend, and MapMyIndia APIs marks a significant achievement in the development of a comprehensive shipping website. This conclusion highlights the accomplishments and significance of the project.

6.2.1 Achievements

1. User-Centric Interface: The frontend implementation prioritised user experience with a streamlined two-step form and interactive map features. This intuitive design enhances usability and ensures accurate data input, improving overall user satisfaction.
2. Efficient Backend Functionality: The backend development phase focused on core functionalities such as data management and route optimization. Stable connections to MongoDB and seamless integration of APIs enable efficient logistics management and route planning, contributing to operational efficiency.
3. Enhanced Functionality with MapMyIndia APIs: Integration of MapMyIndia APIs adds significant value by providing accurate geocoding, reverse geocoding, and distance calculation functionalities. This integration enhances location-based services, optimises routes, and facilitates efficient truck selection and delivery routing.

6.2.2 Significance

1. Efficiency and Cost Reduction: By optimising truck selection and route planning, the platform contributes to efficiency improvements and cost reductions in shipping operations. Users and administrators can make informed decisions, leading to resource optimization and operational efficiency.
2. User Convenience and Experience: The user-centric design of the platform enhances convenience and experience. Features such as dynamic truck recommendations and real-time location-based services improve usability and foster customer satisfaction.
3. Environmental Sustainability: Emphasis on route optimization reduces fuel consumption and emissions, aligning with sustainability goals. By minimising environmental impact, the platform promotes responsible logistics practices.

In conclusion, the successful integration of frontend, backend, and MapMyIndia APIs represents a significant milestone in optimising logistics operations. With a focus on user experience, efficiency, and sustainability, the platform enhances shipping processes and contributes to broader economic and environmental objectives.

CHAPTER 7
FUTURE SCOPE

7.1 Future Work

- Implement sophisticated load optimization algorithms to intelligently and dynamically allocate shipments to trucks based on factors such as weight distribution, volume, and destination.
- Integrate GPS technologies, geofencing, and real-time updates to offer precise location information throughout the entire shipping journey.
- Implement dynamic pricing algorithms that consider real-time factors such as fuel prices, demand-supply dynamics, and market conditions to optimise shipping costs.
- Integrate a load tracking and management system to monitor the status and location of goods in transit. This feature would provide real-time visibility into the movement of cargo, allowing for better coordination and proactive problem-solving in case of delays or discrepancies.
- Create a customer portal where clients can log in to track the status of their shipments, view delivery schedules, and access relevant documents such as invoices and delivery receipts. This portal would enhance transparency and communication between logistics providers and their clients.
- Integrate the logistics platform with inventory management systems to ensure seamless coordination between warehousing and transportation activities. This integration would enable automatic updates to inventory levels, facilitate just-in-time delivery scheduling, and optimise inventory replenishment processes.

REFLECTION ON LEARNING

Throughout the development and learning journey of our project, we've acquired a diverse set of skills, both technical and non-technical. Key among these are:

Soft Skills:

1. *Resilience*: Upholding a determined spirit, we've embraced challenges and persevered in troubleshooting errors.
2. *Problem Solving*: No challenge is insurmountable; we've embraced a solution-oriented approach, finding creative ways to overcome obstacles.
3. *Time Management*: Efficiently managing our time and resources, we've optimised productivity and met project milestones effectively.
4. *Positive Mindset*: Maintaining an optimistic outlook, we've cultivated resilience and motivation, even in the face of adversity.
5. *Curiosity*: With an insatiable appetite for knowledge, we've embraced a culture of continuous learning, constantly exploring and adapting to new concepts and technologies.

Technical Skills:

1. Reactjs
2. Rest APIs
3. Nodejs and Expressjs
4. MongoDB
5. Vensim
6. System Modelling
7. Causal Loop Diagram

REFERENCES

List of references used for this project:

1. "MapPLS Developer Documentation" (https://developer.mappls.com): This resource was instrumental in accessing and comprehending the API functionalities required for integration within the project.

2. The Node.js official documentation (https://nodejs.org/docs/latest/api/) served as a comprehensive resource for acquiring detailed information about the Node.js API and its usage.

3. The MongoDB official documentation (https://www.mongodb.com/docs/) was extensively utilised to access comprehensive information regarding MongoDB's features, functionalities, and usage.

4. The React documentation (https://react.dev/learn) was a primary resource utilised to gain a deep understanding of React.js, its core concepts, and best practices.

5. The Wikipedia page on the Travelling Salesman Problem (TSP) (https://en.wikipedia.org/wiki/Travelling_salesman_problem) provided valuable insights into understanding the TSP and implementing relevant algorithms within the project.

6. The article titled "Logistics Management Systems: How to Implement Them" from Integrio's blog (https://integrio.net/blog/logistics-management-systems-how-to-implement-them) served as a valuable guide for understanding the implementation of logistics management systems, providing insights and strategies applicable to our project.

7. https://vensim.com/documentationindex.html: this documentation provides comprehensive information and resources to help users learn and master the features and functionalities of Vensim.