TRANSPORT TRACKING

A PROJECT REPORT

Submitted by

DENIS REMIJEUS A

PRINCE FRANKILNE A

SUDHARSHAN M

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

K.RAMAKRISHNAN COLLEGE OF ENGINEERING (AUTONOMOUS)

ANNA UNIVERSITY CHENNAI 600 025

DECEMBER 2024

TRANSPORT TRACKING

PROJECT WORK

Submitted by

DENIS REMIJEUS A	**8115U23AM014**
PRINCE FRANKILNE A	**8115U23AM035**
SUDHARSHAN M	**8115U23AM052**

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

Under the guidance of
Mr. M. PONNI VALAVAN
Department of Artificial Intelligence and Machine Learning
K.RAMAKRISHNAN COLLEGE OF ENGINEERING

K. RAMAKRISHNAN COLLEGE OF ENGINEERING
(AUTONOMOUS)
Under
ANNA UNIVERSITY, CHENNAI

K.RAMAKRISHNAN COLLEGE OF ENGINEERING

(AUTONOMOUS)

Under

ANNA UNIVERSITY, CHENNAI

BONAFIDE CERTIFICATE

Certified that this project report ON **“TRANSPORT TRACKING”** is the bonafide work of **DENISREMIJEUS A (8115U23AM014), PRINCE FRANKILNE A (8115U23AM035), SUDHARSHAN M (8115U23AM052)** who carried out the project work under my supervision.

Dr. B. KIRAN BALA M.E,Ph.D,

HEAD OF THE DEPARTMENT

ASSOCIATE PROFESSOR

Department Of Artificial Intelligence

And Machine Learning,

K.Ramakrishnan College Of

Engineering (Autonomous)

Samayapuram,Trichy.

Mr. M. PONNI VALAVAN M.E

SUPERVISOR

ASSISTANT PROFESSOR

Department Of Artificial Intelligence

And Machine Learning,

K.Ramakrishnan College Of

Engineering (Autonomous)

Samayapuram, Trichy.

SIGNATURE OF INTERNAL EXAMINER

NAME :

DATE :

SIGNATURE OF EXTERNAL EXAMINER

NAME :

DATE:

K. RAMAKRISHNAN COLLEGE OF ENGINEERING
(AUTONOMOUS)
Under
ANNA UNIVERSITY, CHENNAI

DECLARATION BY THE CANDIDATES

We declare that to the best of our knowledge the work reported here in has been composed solely by ourselves and that it has not been in whole or in part in any previous application for a degree.

Submitted for the project Viva- Voce held at K. Ramakrishnan College of Engineering on __________

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We thank the almighty GOD, without whom it would not have been possible for us to complete our project.

We wish to address our profound gratitude to **Dr.K.RAMAKRISHNAN**, Chairman, K. Ramakrishnan College of Engineering(Autonomous), who encouraged and gave us all help throughout the course.

We extend our hearty gratitude and thanks to our honorable and grateful Executive Director **Dr.S.KUPPUSAMY, B.Sc., MBA., Ph.D.,** K. Ramakrishnan College of Engineering(Autonomous).

We are glad to thank our Principal **Dr.D.SRINIVASAN, M.E., Ph.D.,FIE., MIIW., MISTE., MISAE., C.Engg,** for giving us permission tocarry out this project.

We wish to convey our sincere thanks to **Dr.B.KIRAN BALA, M.E., M.B.A., Ph.D.,** Head of the Department, Artificial Intelligence and Machine Learning for giving us constant encouragement and advice throughout the course.

We are grateful to **M.PONNI VALAVAN M.E.**, Artificial Intelligenceand Data Science, K. Ramakrishnan College of Engineering (Autonomous), for his guidance and valuable suggestions during the course of study.

Finally, we sincerely acknowledged in no less terms all our staffmembers, our parents and, friends for their co-operation and help at various stages of this project work.

DENIS REMIJEUS A	**8115U23AM014**
PRINCE FRANKILNE A	**8115U23AM035**
SUDHARSHAN M	**8115U23AM052**

INSTITUTE VISION AND MISSION

VISION OF THE INSTITUTE:

To achieve a prominent position among the top technical institutions.

MISSION OF THE INSTIITUTE:

M1: To best owstandard technical education parexcellence through state of the art infrastructure, competent faculty and high ethical standards.

M2: To nurture research and entrepreneurial skills among students in cutting edge technologies.

M3: To provide education for developing high-quality professionals to transform the society.

DEPARTMENT VISION AND MISSION

DEPARTMENT OF CSE(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

Vision of the Department

To become a renowned hub for Artificial Intelligence and Machine Learning Technologies to produce highly talented globally recognizable technocrats to meet Industrial needs and societal expectations.

Mission of the Department

M1: To impart advanced education in Artificial Intelligence and Machine Learning, Built upon a foundation in Computer Science and Engineering.

M2: To foster Experiential learning equips students with engineering skills to Tackle real-world problems.

M3: To promote collaborative innovation in Artificial Intelligence, machine Learning, and related research and development with industries.

M4: To provide an enjoyable environment for pursuing excellence while upholding Strong personal and professional values and ethics.

Programme Educational Objectives (PEOs):

Graduates will be able to:

PEO1: Excel in technical abilities to build intelligent systems in the fields of Artificial Intelligence and Machine Learning in order to find new opportunities.

PEO2: Embrace new technology to solve real-world problems, whether alone or As a team, while prioritizing ethics and societal benefits.

PEO3: Accept lifelong learning to expand future opportunities in research and Product development.

Programme Specific Outcomes (PSOs):

PSO1: Ability to create and use Artificial Intelligence and Machine Learning Algorithms, including supervised and unsupervised learning, reinforcement Learning, and deep learning models.

PSO2: Ability to collect, pre-process, and analyze large datasets, including data Cleaning, feature engineering, and data visualization..

PROGRAM OUTCOMES(POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis:** Identify, formulate, review, research, literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectivelyon complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

This project develops an advanced **Transport Tracking System** integrating GPS tracking, sensor data, and predictive analytics to optimize logistics and supply chain operations. The system provides real-time vehicle and product tracking, while predicting delivery times and monitoring environmental conditions. It incorporates IoT sensors to track vehicle health, product quality, and driver behavior. The system uses predictive analytics to forecast potential delays and improve operational efficiency. Mobile integration enables drivers and customers to interact with real-time updates. The interactive dashboard helps fleet managers monitor and optimize performance. Customer feedback is integrated to enhance service quality. Autonomous vehicle support is also included for future readiness. By combining data collection and analytics, the system improves resource allocation and reduces delivery time. This project aims to streamline transportation processes and improve overall efficiency in logistics.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

GPS	Global Positioning System
IoT	Internet of Things
API	Application Programming Interface
AI	Artificial Intelligence
ML	Machine Learning
ETA	Estimated Time of Arrival
API	Application Programming Interface
KPI	Key Performance Indicator
UX	User Experience
UI	User Interface

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The Transport Tracking Application aims to enhance logistics management by providing real-time visibility, efficient route tracking, and data-driven insights. This project incorporates design thinking to create a user-centered, efficient tracking system that leverages GPS and sensor technologies to optimize the supply chain.

1.2 PURPOSE AND IMPORTANCE

This application addresses the critical need for real-time location tracking and data analysis in logistics. By integrating predictive analytics and interactive dashboards, the project highlights how advanced technology and user-centered design can improve decision-making, reduce costs, and boost customer satisfaction.

Purpose:

To provide real-time tracking of vehicles, products, and drivers, integrate GPS tracking, IoT sensors, and predictive analytics for enhanced logistics, improve delivery accuracy by predicting reach and receiving times, monitor environmental factors (e.g., climate, road conditions) that affect transit.

Importance:

Streamlines transportation processes, improving operational efficiency, Reduces delays and improves on-time deliveries with predictive insights, Ensures product quality during transit with environmental condition monitoring, Improves driver safety and vehicle performance through real-time health tracking.

OBJECTIVES

- **Primary Objectives:** Develop a transport tracking prototype that integrates GPS tracking, sensor data, and data visualization for real-time monitoring and decision-making.
- **Secondary Objectives:** Demonstrate the value of predictive analytics in improving delivery accuracy, operational efficiency, and proactive issue management. Focus on a user-centric design that enhances the experience for drivers, fleet managers, and customers. Provide a framework for future scalability, enabling the integration of new technologies, such as autonomous vehicles, and adapting to evolving industry needs.

1.3 PROJECT SUMMARIZATION

The project integrates design thinking methods with advanced tracking technologies to create an application for real-time tracking, environmental monitoring, and delivery forecasting. The application utilizes GPS tracking and IoT sensors to monitor vehicle and product status in real-time. It incorporates predictive analytics to forecast delivery times and optimize routes. Environmental factors, such as weather and road conditions, are also monitored for improved decision-making. The app provides a user-friendly interface for fleet managers, drivers, and customers. It enhances operational efficiency and ensures timely deliveries. The system is designed to scale with future technologies, such as autonomous vehicles. Ultimately, the application aims to improve logistics and customer satisfaction.

koj

CHAPTER 2

PROJECT METHODOLOGY

2.1 INTRODUCTION TO SYSTEM ARCHITECTURE

The architecture of the transport tracking application supports real-time data processing and analysis. Using a modular approach, the system allows efficient data capture, visualization, and mobile integration.

2.1.1 Efficiency

The system uses GPS and sensors to enable fast, reliable location updates, ensuring up-to-date tracking information for users**.** leverages GPS and sensor technologies to provide fast and accurate location updates, ensuring real-time tracking and efficient delivery management. This enables users to receive timely and reliable tracking information, improving operational efficiency. Utilizes GPS and sensor technology for fast, accurate, and real-time tracking updates. Ensures reliable delivery management, improving operational efficiency with up-to-date information.

2.1.2 Scalability

The architecture is designed to handle increasing data and adapt to various use cases, from simple delivery tracking to complex logistics operations. designed to accommodate growing data volumes and scale across different use cases, from simple delivery tracking to complex logistics operations. It can expand seamlessly to support larger fleets and more sophisticated logistics needs as the business grows.

2.1.3 DETAILED SYSTEM ARCHITECTURE DIAGRAM

The system architecture diagram visually represents the flow of data across key components, including GPS, sensors, and visualization tools. It shows how real-time data is collected from the GPS and sensor devices, processed, and integrated into the system. This data is then displayed through interactive dashboards, providing fleet managers and users with up-to-date information on vehicle location, product status, and environmental conditions. The diagram illustrates the seamless integration of these components to enable efficient tracking and monitoring. component is used to define the system parts GPS, Sensors, Data Processing, and Visualization, User and Fleet Manager interact with the Visualization Dashboard.

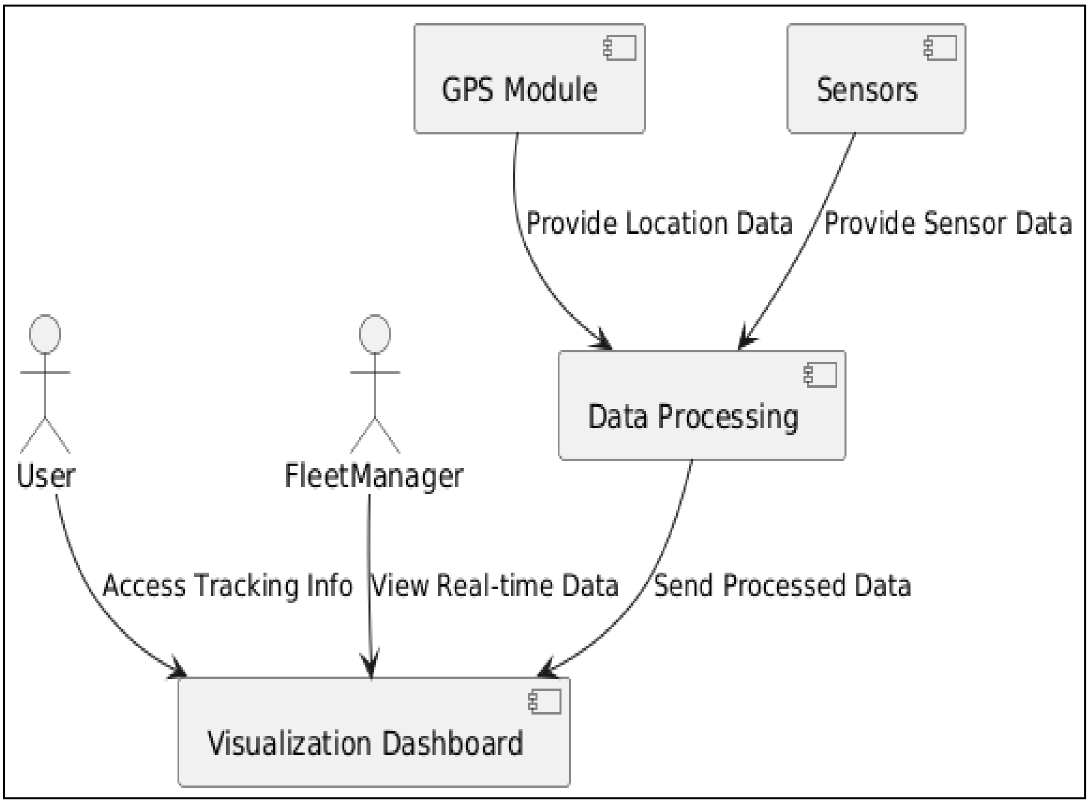


Figure2.1: Architecture Diagram (Sample)

CHAPTER 3

DATA HANDLING AND PREDICTIVE ANALYTICS

3.1 EXPLANATION OF DATA HANDLING TECHNIQUES

The application leverages advanced data handling techniques to efficiently collect and process data from GPS and sensors. By organizing the data into relevant categories, it enables effective tracking and analysis of vehicle movement and environmental conditions. The structured approach ensures that insights are delivered in real-time, supporting optimized decision-making.

- **Data Collection**:
  Real-time data is gathered from GPS devices and IoT sensors monitoring vehicle location, temperature, humidity, and other critical conditions.
- **Data Categorization**:
  Data is organized into categories like location, environmental conditions, and vehicle health for streamlined processing.
- **Efficient Processing**:
  Data is processed and stored in a structured format, enabling quick access and real-time decision-making.
- **Data Integration**:
  All collected data is integrated into a unified platform for continuous monitoring and analysis.

This method ensures efficient tracking and effective use of the collected data for logistics optimization.

3.1.1 COMPARISON WITH OTHER TRACKING TECHNOLOGIES

Compared to traditional tracking technologies, this application offers a more comprehensive solution by integrating sensor data and predictive algorithms. While traditional tracking systems focus mainly on GPS location, this system includes additional data from environmental sensors (e.g., temperature, humidity) for a deeper understanding of conditions affecting transport. Predictive forecasting is another key differentiator, allowing the system to anticipate delays, maintenance needs, and optimal routes. This integration of dynamic data makes the system more adaptive and efficient. Traditional systems lack this level of real-time decision-making capability. Overall, this system offers enhanced tracking, forecasting, and operational insight.

3.2 Advantages and Disadvantages of Predictive Analytics

3.2.1 Advantages

- **Accurate Delivery Forecasts** : Predictive analytics enhances the ability to forecast delivery times, optimizing routes and reducing delays.
- **Improved Operational Efficiency** : Real-time data and predictive insights enable better resource allocation and decision-making, streamlining logistics operations.

3.2.2 Disadvantages

- **Data Processing Demands** : Predictive models require significant computational resources for processing large volumes of data.
- **Dependency on Data Quality** : The accuracy of predictions is highly dependent on the quality and reliability of the data collected from sensors and GPS systems.

CHAPTER 4

DATA COLLECTION AND USER INTERFACE METHODOLOGY

4.1 GPS and Sensor Integration

The system integrates GPS and sensors to capture both real-time **location** and environmental data such as temperature, humidity, and vehicle health. The GPS tracks the exact position of vehicles, while sensors monitor conditions that may affect transportation. This data is processed to provide accurate, up-to-the-minute tracking, allowing for proactive decisions and route adjustments. Key features include real-time location tracking, vehicle condition monitoring, and environmental data collection.

Key features

1. **Real-time Location Tracking** : Provides continuous, precise vehicle location updates via GPS.
2. **Environmental Monitoring** : Tracks temperature, humidity, and other conditions affecting transportation using sensors.
3. **Vehicle Health Monitoring** : Monitors vehicle conditions such as fuel levels and tire pressure in real-time.
4. **Data Synchronization** : Ensures seamless integration and synchronization of GPS and sensor data for accurate tracking.
5. **Proactive Alerts** : Sends notifications based on environmental factors or vehicle health issues to enable timely action.

4.2 Data Visualization Implementation

The application features interactive dashboards that display real-time tracking data, including vehicle locations and environmental conditions. With mobile integration, users can access updates and historical trends anytime, enhancing decision-making for fleet managers, drivers, and customers. This ensures better visibility and improves transportation efficiency by providing actionable insights in real-time.

4.3 Interactive Features and User Testing

The system offers interactive features such as customizable views, real-time alerts, and data filtering for users to engage with tracking data effectively. User testing was conducted to refine the interface, ensuring it's intuitive and meets the needs of users. Feedback from testing helped enhance usability, ensuring a smooth experience for both fleet managers and drivers.

- **Customizable Views**: Users can personalize dashboards to focus on key data, enhancing accessibility and usability.
- **Real-time Alerts**: The system sends notifications for important events, such as delays or environmental changes, to keep users informed.
- **Usability Improvements**: Feedback from user testing led to refinements in the interface, making it more intuitive and improving user satisfaction.
- **Smooth User Experience**: Focused on optimizing the design for both fleet managers and drivers, ensuring ease of use across different user types.
- **Real-time Alerts**: The system sends notifications for important events, such as delays or environmental changes, to keep users informed.
- **Data Filtering**: Users can filter tracking data by specific parameters (e.g., location, time, vehicle status) to gain targeted insights.

CHAPTER-5

MODULES

This section outlines the core modules of the application, each designed to handle specific tasks that collectively enable effective transport tracking and decision-making.

5.1 Initialization:

The **Initialization Module** is responsible for setting up and establishing connections with **GPS devices** and **sensors** at the start of the application. This process ensures the system is ready to capture **real-time data** from the vehicle's location and surrounding environment. It handles the synchronization of all devices and prepares the system to collect and process accurate data as soon as the application is launched, ensuring a smooth start-up for tracking operations.

5.2 Data Collection and Tracking

The **Data Collection and Tracking Module** manages the continuous integration of GPS and sensor data. This module collects real-time location data from GPS and environmental information (e.g., temperature, humidity) from sensors. It processes this information to provide an accurate and up-to-date view of the vehicle's status and external conditions. The system aggregates this data, ensuring that the tracking remains precise and responsive, even in dynamic environments. This module serves as the backbone of the application, ensuring that all collected data is processed and available for visualization.

5.3 Data Visualization and Analytics

The Data Visualization and Analytics Module plays a crucial role in presenting tracking data through interactive dashboards. It allows users to view real-time updates, historical data, and key metrics like vehicle location, environmental conditions, and performance insights. The module includes predictive analytics, which use the collected data to forecast delivery times, route optimization, and potential delays. This feature empowers users to make informed decisions and proactively manage logistics. Additionally, the module is equipped with filtering options to display specific data views, enabling users to monitor particular vehicles or shipments efficiently.

CHAPTER 6

APPLICATION DEVELOPMENT

Creating a basic transport tracking application involves setting up a simple web interface that uses GPS tracking to monitor vehicle locations in real-time. I'll provide a streamlined example of how to create a transport tracking application using HTML , JavaScript , and Node.js to simulate real-time tracking data. The application will feature a backend server to manage simulated data and a frontend to visualize the transport locations.

6.1 Overview

This application will have:

1. Frontend for visualization.
2. Backend to simulate GPS location data.

6.2 Project Structure

Here's a basic structure for the project:

```php
transport-tracking/
├── server.js            # Backend server
├── public/
│   ├── index.html       # Frontend webpage
│   └── script.js        # Frontend JavaScript for map and tracking
└── package.json         # Node.js dependencies
```

Step 1: Set Up the Backend (Node.js)

1. **Initialize Node Project**

Open a terminal in the project folder and run:

```bash
npm init -y
```

2. **Install Dependencies**

We'll use **Express** for the server.

```bash
npm install express
```

3. **server.js** : Simulating GPS Data

In `**server.js**`, set up an endpoint that provides location data for a vehicle.

```javascript
const express = require('express');
const app = express();
const PORT = 3000;

// Serve static files from 'public' directory
app.use(express.static('public'));

// Simulated GPS coordinates (latitude, longitude)
let latitude = 37.7749;   // Starting latitude
let longitude = -122.4194; // Starting longitude

// Endpoint to get current location data
app.get('/location', (req, res) => {
    // Simulate GPS coordinate changes
    latitude += (Math.random() - 0.5) * 0.001;
    longitude += (Math.random() - 0.5) * 0.001;
    res.json({ latitude, longitude });
```

```
});

app.listen(PORT, () => {
    console.log(`Server running on http://localhost:${PORT}`);
});
```

This code will simulate GPS coordinates changing over time, providing the frontend with new coordinates every time it fetches the `**/location**` endpoint.

Step 2: Frontend (HTML and JavaScript for Map and Tracking)

1. Frontend HTML (index.html) : Create a simple interface with a map.

```
Copy code
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Transport Tracking</title>
    <!-- Link to Mapbox for mapping (you can use Leaflet as well if preferred) -->
    <link href='https://api.mapbox.com/mapbox-gl-js/v2.9.1/mapbox-gl.css' rel='stylesh
    <style>
        body { margin: 0; padding: 0; }
        #map { width: 100vw; height: 100vh; }
    </style>
</head>
<body>
    <div id="map"></div>
    <script src="script.js"></script>
    <script src='https://api.mapbox.com/mapbox-gl-js/v2.9.1/mapbox-gl.js'></script>
</body>
</html>
```

Replace `'YOUR_MAPBOX_ACCESS_TOKEN'` in the script.js with your Map box token.

2. Frontend JavaScript (script.js) : Fetch location data and update the map in real time.

javascript Copy code

```javascript
// Initialize the map with Mapbox
mapboxgl.accessToken = 'YOUR_MAPBOX_ACCESS_TOKEN';
const map = new mapboxgl.Map({
    container: 'map',
    style: 'mapbox://styles/mapbox/streets-v11',
    center: [-122.4194, 37.7749], // Initial position [longitude, latitude]
    zoom: 12
});

let marker = new mapboxgl.Marker().setLngLat([-122.4194, 37.7749]).addTo(map);

// Function to fetch updated location and move the marker
async function updateLocation() {
    try {
        const response = await fetch('/location');
        const data = await response.json();
        const { latitude, longitude } = data;
```

```javascript
        // Update marker location
        marker.setLngLat([longitude, latitude]);
        map.setCenter([longitude, latitude]);
    } catch (error) {
        console.error('Error fetching location:', error);
    }
}

// Update location every 2 seconds
setInterval(updateLocation, 2000);
```

Step 3: Run the Application

1. Start the server:

```bash
node server.js
```

2. Open your browser and go to `**http://localhost:3000**`. You should see a map with a marker representing the vehicle's current location, updating every 2 seconds.

This setup provides a foundation for a basic transport tracking application that simulates real-time location tracking. You could further expand this project by:

- Integrating real GPS data.
- Adding user authentication.
- Including route optimization features.

CHAPTER 7
CONCLUSION & FUTURE SCOPE

7.1 CONCLUSION

The **Transport Tracking project** effectively demonstrates the integration of **design thinking** into logistics solutions, focusing on user-centered development. By combining **GPS**, **sensors**, and **predictive analytics**, the application offers a comprehensive tracking system that significantly improves operational efficiency. Real-time data collection and visualization provide valuable insights, allowing fleet managers to make informed decisions. The system's ability to forecast delivery times and monitor vehicle health enhances overall logistics performance. The integration of mobile accessibility ensures users can stay informed at all times. Ultimately, the project delivers a streamlined solution for optimizing transport operations. With future improvements, the system has the potential to evolve into a more advanced tool with autonomous vehicle capabilities and AI-powered optimizations.

7.2 FUTURE SCOPE

- **Advanced Features**: Integrating **autonomous vehicles** for more complex and automated tracking systems.
- **Sustainability**: Developing **eco-friendly logistics routes** to reduce environmental impact.
- **AI and Machine Learning**: Leveraging **AI** for better **predictive accuracy** and **route optimization** to further enhance operational efficiency.

APPENDICES

APPENDIXA-SOURCECODE

```
import { useState, useEffect } from 'react'
import { MapContainer, TileLayer, Marker, Popup } from 'react-leaflet'
import 'leaflet/dist/leaflet.css'
import L from 'leaflet'

// Fix for default marker icon in Leaflet with Next.js
delete L.Icon.Default.prototype._getIconUrl
L.Icon.Default.mergeOptions({
  iconRetinaUrl: 'https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.3.1/images/marker-icon-2x.png',
  iconUrl: 'https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.3.1/images/marker-icon.png',
  shadowUrl: 'https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.3.1/images/marker-shadow.png',
})

type Vehicle = {
  id: string
  name: string
  position: [number, number]
  temperature: number
  climate: string
  productState: string
  lastReceived: string
  lastSent: string
}

const initialVehicles: Vehicle[] = [
  {
    id: '1',
```

```
    name: 'Truck 001',
    position: [51.505, -0.09],
    temperature: 22,
    climate: 'Normal',
    productState: 'In Transit',
    lastReceived: '2023-05-10T10:30:00Z',
    lastSent: '2023-05-10T10:35:00Z',
  },
  {
    id: '2',
    name: 'Van 002',
    position: [51.51, -0.1],
    temperature: 18,
    climate: 'Cool',
    productState: 'Delivered',
    lastReceived: '2023-05-10T11:00:00Z',
    lastSent: '2023-05-10T11:05:00Z',
  },
]

export default function TransportTrackingSystem() {
  const [vehicles, setVehicles] = useState<Vehicle[]>(initialVehicles)
  const [selectedVehicle, setSelectedVehicle] = useState<Vehicle | null>(null)

  useEffect(() => {
    // Simulating real-time updates
    const interval = setInterval(() => {
    setVehicles(prevVehicles =>
        prevVehicles.map(vehicle => ({
          ...vehicle,
          position: [
            vehicle.position[0] + (Math.random() - 0.5) * 0.001,
            vehicle.position[1] + (Math.random() - 0.5) * 0.001,
          ],
```

```
        temperature: Math.round((vehicle.temperature + (Math.random() - 0.5)) * 10) / 10,
      }))
    )
  }, 5000)

  return () => clearInterval(interval)
}, [])

return (
  <div className="flex h-screen">
    <div className="w-1/3 p-4 overflow-y-auto">
      <h1 className="text-2xl font-bold mb-4">Transport Tracking System</h1>
      <div className="space-y-4">
        {vehicles.map(vehicle => (
          <div
            key={vehicle.id}
            className="p-4 border rounded cursor-pointer hover:bg-gray-100"
            onClick={() => setSelectedVehicle(vehicle)}
          >
            <h2 className="text-lg font-semibold">{vehicle.name}</h2>
            <p>Temperature: {vehicle.temperature}°C</p>
            <p>Climate: {vehicle.climate}</p>
            <p>Product State: {vehicle.productState}</p>
            <p>Last Received: {new Date(vehicle.lastReceived).toLocaleString()}</p>
            <p>Last Sent: {new Date(vehicle.lastSent).toLocaleString()}</p>
          </div>
        ))}
      </div>
    </div>
    <div className="w-2/3">
      <MapContainer center={[51.505, -0.09]} zoom={13} style={{ height: '100%', width:
'100%' }}>
        <TileLayer url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png" />
        {vehicles.map(vehicle => (
```

```
          <Marker key={vehicle.id} position={vehicle.position}>
            <Popup>
              <h2 className="text-lg font-semibold">{vehicle.name}</h2>
              <p>Temperature: {vehicle.temperature}°C</p>
              <p>Climate: {vehicle.climate}</p>
              <p>Product State: {vehicle.productState}</p>
            </Popup>
          </Marker>
        ))}
      </MapContainer>
    </div>
    {selectedVehicle && (
      <div className="absolute top-0 right-0 m-4 p-4 bg-white border rounded shadow-lg">
        <h2 className="text-xl font-bold mb-2">{selectedVehicle.name}</h2>
        <p>Temperature: {selectedVehicle.temperature}°C</p>
        <p>Climate: {selectedVehicle.climate}</p>
        <p>Product State: {selectedVehicle.productState}</p>
        <p>Last Received: {new Date(selectedVehicle.lastReceived).toLocaleString()}</p>
        <p>Last Sent: {new Date(selectedVehicle.lastSent).toLocaleString()}</p>
        <button
          className="mt-2 px-4 py-2 bg-blue-500 text-white rounded hover:bg-blue-600"
          onClick={() => setSelectedVehicle(null)}
        >
          Close
        </button>
      </div>
    )}
  </div>
 )
}

// Simulated data update interval (in milliseconds)
const UPDATE_INTERVAL = 5000
```

```
export default function TransportTracker() {
  const [vehicleData, setVehicleData] = useState({
  gps: { latitude: 0, longitude: 0 },
    temperature: 0,
    productState: "Good",
    climate: "Normal",
    lastReceived: new Date(),
    lastSent: new Date(),
    vehicleInfo: {
      id: "TRK-001",
      model: "FreightLiner Cascadia",
      year: 2023,
      driver: "John Doe"
    }
  })

  useEffect(() => {
    const updateData = () => {
      setVehicleData(prevData => ({
        ...prevData,
        gps: {
          latitude: prevData.gps.latitude + (Math.random() - 0.5) * 0.01,
          longitude: prevData.gps.longitude + (Math.random() - 0.5) * 0.01
        },
        temperature: Math.round((prevData.temperature + (Math.random() - 0.5) * 2) * 10) / 10,
        productState: Math.random() > 0.9 ? "Check Required" : "Good",
        climate: ["Normal", "Humid", "Dry"][Math.floor(Math.random() * 3)],
        lastReceived: new Date(),
        lastSent: new Date()
      }))
    }

    updateData() // Initial update
    const interval = setInterval(updateData, UPDATE_INTERVAL)
```

```
    return () => clearInterval(interval)
  }, [])

// Mock data for demonstration
const mockVehicles = [
  { id: 1, name: 'Truck 001', lat: 51.505, lng: -0.09, temp: 2, fuel: 75, alert: false },
  { id: 2, name: 'Truck 002', lat: 51.51, lng: -0.1, temp: -1, fuel: 45, alert: true },
  { id: 3, name: 'Truck 003', lat: 51.515, lng: -0.09, temp: 4, fuel: 90, alert: false },
]

export default function VehicleTrackingDashboard() {
  const [vehicles, setVehicles] = useState(mockVehicles)
  const [selectedVehicle, setSelectedVehicle] = useState(null)

  useEffect(() => {
    // In a real application, you would fetch real-time data here
    const interval = setInterval(() => {
      // Simulate vehicle movement
      setVehicles(prevVehicles =>
      prevVehicles.map(vehicle => ({
          ...vehicle,
          lat: vehicle.lat + (Math.random() - 0.5) * 0.001,
          lng: vehicle.lng + (Math.random() - 0.5) * 0.001,
        }))
      )
    }, 5000)

    return ()

  return (
    <div className="container mx-auto p-4">
      <h1 className="text-3xl font-bold mb-6">Transport Tracking System</h1>

      <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
```

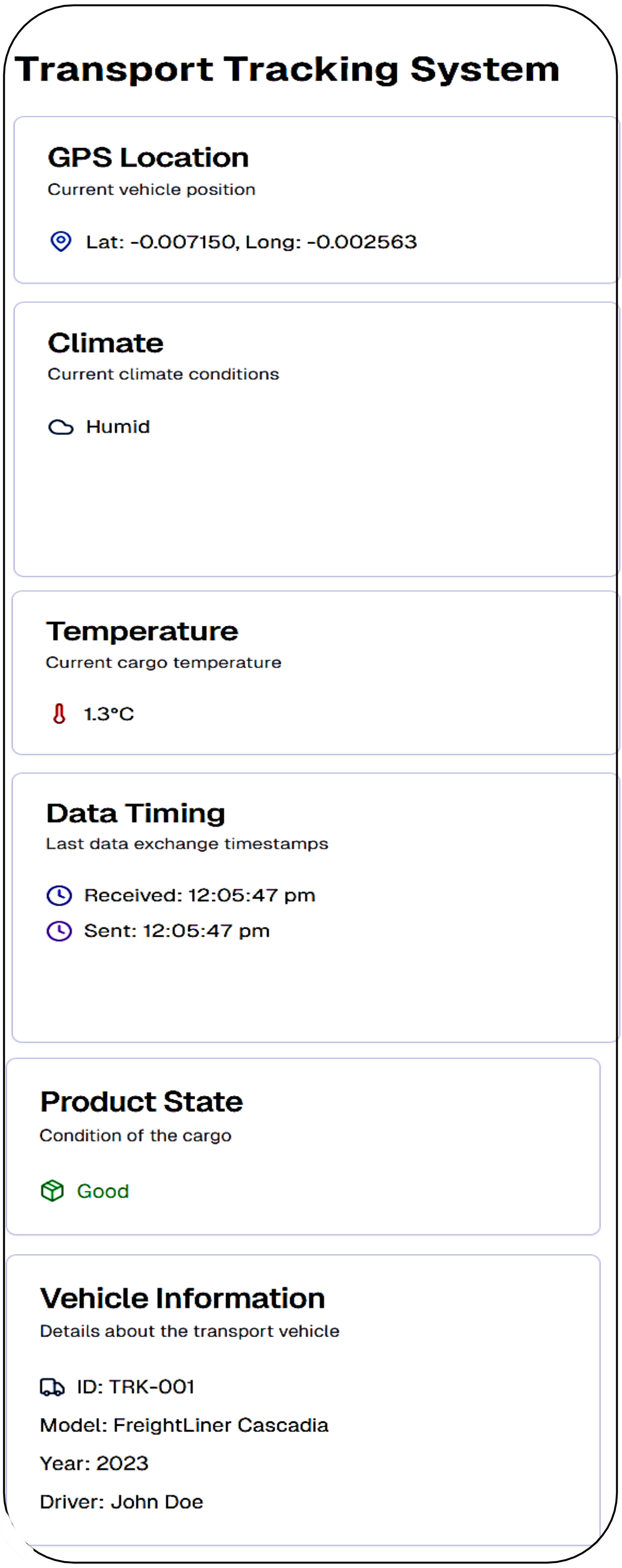
```
<Card>
  <CardHeader>
    <CardTitle>GPS Location</CardTitle>
    <CardDescription>Current vehicle position</CardDescription>
  </CardHeader>
  <CardContent>
    <div className="flex items-center space-x-2">
      <MapPin className="h-5 w-5 text-blue-500" />
      <span>
        Lat: {vehicleData.gps.latitude.toFixed(6)},
        Long: {vehicleData.gps.longitude.toFixed(6)}
      </span>
    </div>
  </CardContent>
</Card>

<Card>
  <CardHeader>
    <CardTitle>Temperature</CardTitle>
    <CardDescription>Current cargo temperature</CardDescription>
  </CardHeader>
  <CardContent>
    <div className="flex items-center space-x-2">
      <Thermometer className="h-5 w-5 text-red-500" />
      <span>{vehicleData.temperature}°C</span>
    </div>
  </CardContent>
</Card>
<Card>
  <CardHeader>
    <CardTitle>Product State</CardTitle>
    <CardDescription>Condition of the cargo</CardDescription>
  </CardHeader>
  <CardContent>
```

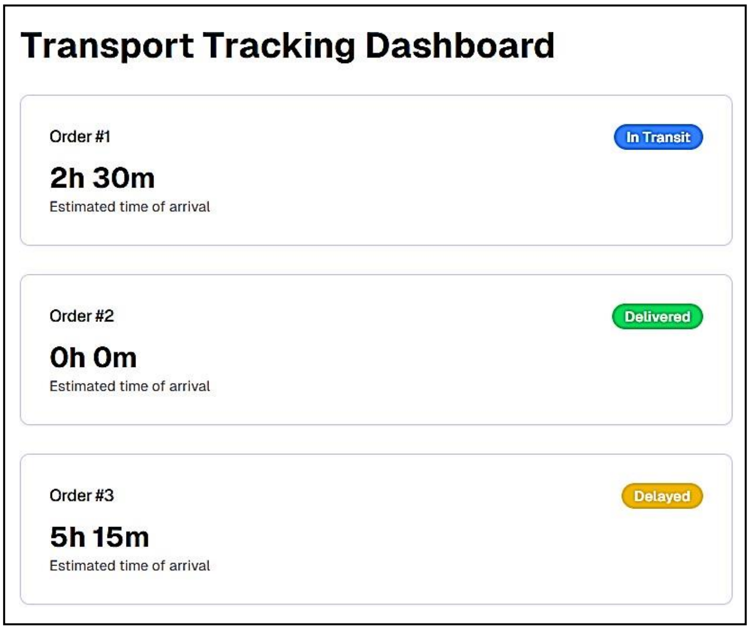
```
          <div className="flex items-center space-x-2">
            <Package className="h-5 w-5 text-green-500" />
            <span className={vehicleData.productState === "Good" ? "text-green-500" : "text-
yellow-500"}>
              {vehicleData.productState}
            </span>
          </div>
        </CardContent>
      </Card>
      <Card>
        <CardHeader>
          <CardTitle>Climate</CardTitle>
          <CardDescription>Current climate conditions</CardDescription>
        </CardHeader>
        <CardContent>
          <div className="flex items-center space-x-2">
            <Cloud className="h-5 w-5 text-gray-500" />
            <span>{vehicleData.climate}</span>
          </div>
        </CardContent>
      </Card>

      <Card>
        <CardHeader>
          <CardTitle>Data Timing</CardTitle>
          <CardDescription>Last data exchange timestamps</CardDescription>
        </CardHeader>
        <CardContent>
          <div className="space-y-2">
            <div className="flex items-center space-x-2">
              <Clock className="h-5 w-5 text-indigo-500" />
              <span>Received: {vehicleData.lastReceived.toLocaleTimeString()}</span>
            </div>
            <div className="flex items-center space-x-2">
```

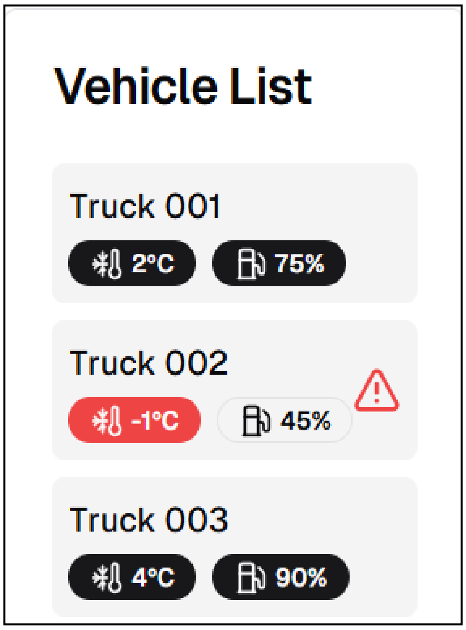
```
            <Clock className="h-5 w-5 text-purple-500" />
            <span>Sent: {vehicleData.lastSent.toLocaleTimeString()}</span>
          </div>
        </div>
      </CardContent>
    </Card>
    <Card>
      <CardHeader>
        <CardTitle>Vehicle Information</CardTitle>
        <CardDescription>Details about the transport vehicle</CardDescription>
      </CardHeader>
      <CardContent>
        <div className="space-y-2">
          <div className="flex items-center space-x-2">
            <Truck className="h-5 w-5 text-gray-500" />
            <span>ID: {vehicleData.vehicleInfo.id}</span>
          </div>
          <div>Model: {vehicleData.vehicleInfo.model}</div>
          <div>Year: {vehicleData.vehicleInfo.year}</div>
          <div>Driver: {vehicleData.vehicleInfo.driver}</div>
        </div>
      </CardContent>
    </Card>
  </div>
  <div className="mt-6 flex justify-center">
    <Button className="flex items-center space-x-2">
      <RotateCw className="h-5 w-5" />
      <span>Refresh Data</span>
    </Button>
  </div>
 </div>
)
}
```

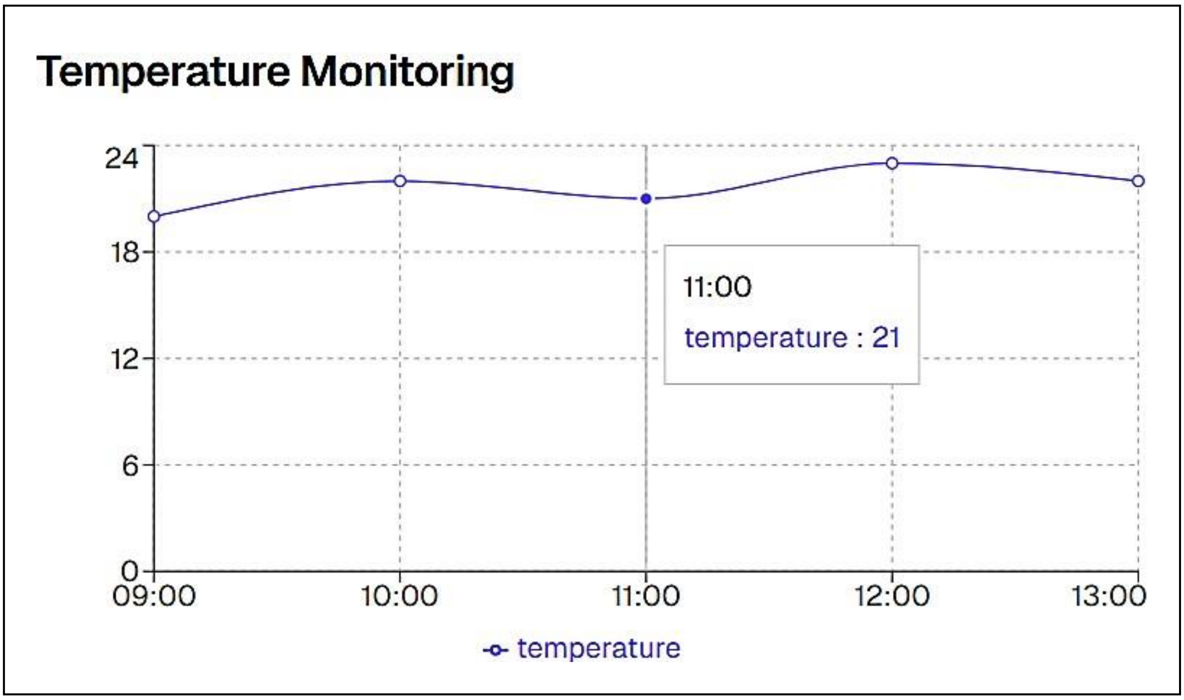
APPENDIX B

SCREENSHOTS RESULT AND DISCUSSION




Transport Tracking Dashboard
Order #1
In Transit
2h 30m
Estimated time of arrival
Order #2
Delivered
0h 0m
Estimated time of arrival
Order #3
Delayed
5h 15m
Estimated time of arrival


Vehicle List
Truck 001
2°C
75%
Truck 002
-1°C
45%
Truck 003
4°C
90%


Temperature Monitoring
24
18
12
6
0
09:00
10:00
11:00
12:00
13:00
11:00
temperature : 21
temperature

Truck 001

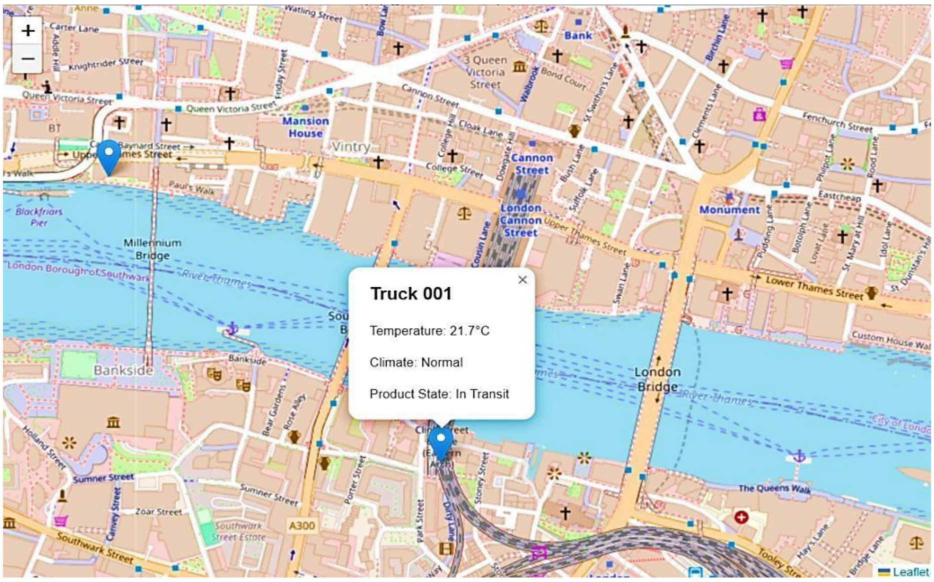
Temperature: 21.7°C

Climate: Normal

Product State: In Transit

Last Received: 10/5/2023, 4:00:00 pm

Last Sent: 10/5/2023, 4:05:00 pm



Van 002

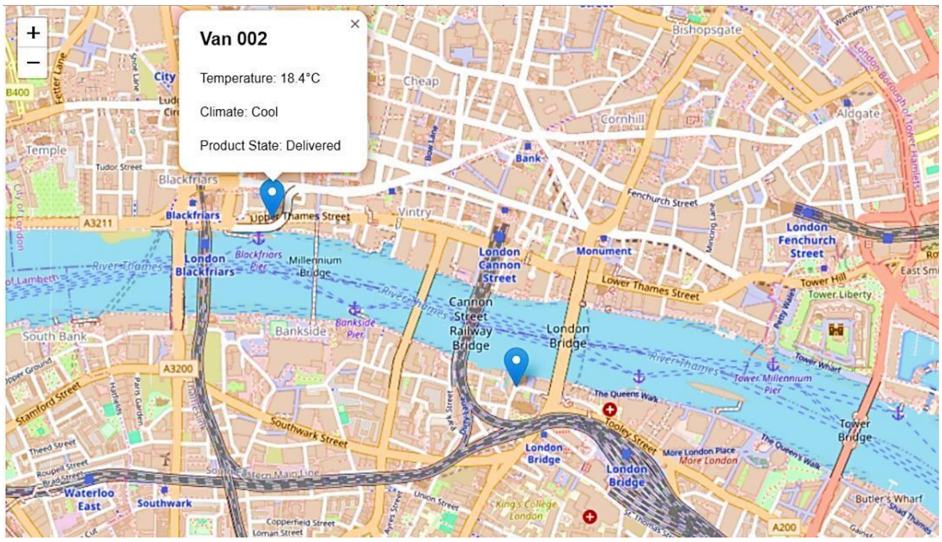
Temperature: 18.4°C

Climate: Cool

Product State: Delivered

Last Received: 10/5/2023, 4:30:00 pm

Last Sent: 10/5/2023, 4:35:00 pm



REFERENCES

1. **Aitken, R. (2018). Digital Supply Chains**: A Revolution in the Supply Chain. Palgrave Macmillan. (This book provides insights into the digital transformation of supply chains, including the role of tracking technologies in enhancing logistics efficiency)

2. **Baldacci, R., & Maniezzo, V. (2019).** "Vehicle Routing Problems in Logistics: Algorithms and Applications." Transportation Research Part B: Methodological, 125, 34-56. (This research explores vehicle routing and tracking problems, offering methods for optimizing transport logistics)

3. **Deng, Q., & Zheng, Q. (2020).** Real-Time Big Data Analytics for Predictive Transportation Tracking. (IEEE Xplore Discusses how big data and predictive analytics can improve real-time tracking in transportation systems)

4. **Goodchild, A., & Toy, J. (2021).** "GPS and Sensor Integration for Real-Time Transport Tracking." International Journal of Logistics Research and Applications, 24(3), 219-232. (Examines how GPS and sensors are integrated into tracking systems to enhance real-time data collection and location accuracy)

5. **Kandilli , C., & Bayraktar, E. (2018).** Implementing IoT and Sensor Networks in Logistics Springer. (This book explains IoT technologies, including GPS and sensors, in logistics and their impact on supply chain visibility)

6. **Lee, H. L., & Lee, D. (2020).** "The Future of Logistics: AI and Predictive Analytics in Transport Tracking." Journal of Transportation and Supply Chain Innovation, 14(1), 87-101.
7. **Meena, P. L., & Barve, A. (2019).** "The Role of GPS Tracking in Last-Mile

Delivery Optimization." Journal of Supply Chain Management, 45(2), 108-119. (Discusses how GPS tracking optimizes last-mile delivery, focusing on improving customer satisfaction and reducing delivery times)

8. **Mohapatra, P., & Murthy, S. (2017).** GPS and Location-Based Services in Transportation. Morgan Kaufmann. (A comprehensive guide on GPS applications in transportation, covering the technical aspects and benefits of location-based tracking systems)

9. **Singh, R., & Sharma, M. (2021).** "Evaluating Sensor Integration for Environmental and Location Monitoring in Transport Tracking." Sensors, 21(4), 1147.(Examines how sensors monitor environmental conditions and location data in transport tracking, improving data quality and reliability)

10. **Van der Meer, J., & Van Wee, B. (2020).** "Challenges and Future Trends in Transport Tracking and Logistics Optimization." Journal of Advanced Transportation, 53(6), 1221-1234.(Covers the current challenges in transport tracking and highlights emerging trends, such as autonomous vehicles and sustainability in logistics)