Minor Project

END TERM REPPORT

ON

Topic Name: SupplyChain Path Optimizer

Submitted By

Presented by:

Swasti Negi [500102243]

Harsh Kumar [500108340]

Ayush Agarwal [500101834]

Under the guidance of

Dr. Aishik Biswas

Associate Professor (**Designation**)

Cloud Cluster (CSO)SOCs

School of Computer Science

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES

Dehradun248007

Aug Nov, 2024

INDEX

S.No.	Content	Page No.
1.	Project Title	1
2.	Abstract	1
3.	Introduction	1
4.	Literature	2
5.	Problem Statement	4
6.	Existing System Issue	4
7.	Algorithm Discussed	5
8.	Proposed System Design	5
9.	Results and Discussion	7
10.	Test Cases	9

11.	Comparative Study	10
12.	Future Work	11
13.	References	11

1.Project Title

SupplyChain Path Optimizer

2.Abstract

SupplyChain Path Optimizer is a comprehensive solution designed to improve supply chain management for small and medium-sized businesses. It focuses on four key components: inventory management, order processing, supplier selection, and delivery route optimization. By leveraging powerful algorithms such as Dijkstra's for shortest path calculations and the Traveling Salesman Problem (TSP) for delivery route optimization, the system helps businesses reduce costs and streamline operations.

3.Introduction

SupplyChain Path Optimizer is a supply chain management system designed to address the operational challenges faced by small and medium-sized businesses. The project focuses on optimizing key aspects of the supply chain, such as inventory management, order processing, supplier selection, and delivery route optimization. Supply chain management is critical to business success, and inefficient processes can lead to increased costs, delayed deliveries, and customer dissatisfaction.

To solve these problems, SupplyChain Path Optimizer incorporates advanced algorithms such as Dijkstra's algorithm for finding the shortest path in supplier selection and the Traveling Salesman Problem (TSP) algorithm for optimizing delivery routes. These algorithms ensure that the system can help businesses make the best decisions in real time, reducing operational costs and improving delivery efficiency.

The project is built using a modular approach, allowing each component to be developed, tested, and refined independently before integration. This modularity

enhances flexibility and scalability, ensuring that the system can evolve as business needs grow. By offering an efficient, user-friendly solution, SupplyChain Path Optimizer aims to make supply chain management accessible to small businesses, enabling them to optimize operations, reduce costs, and compete effectively in the marketplace.

4.Literature

Supply chain management (SCM) has been a crucial component of business operations for decades, but the need for optimized solutions has grown with the complexity of global trade and logistics. Traditional SCM systems like SAP SCM and Oracle SCM offer comprehensive tools for large enterprises but are often expensive and overly complex for small and medium-sized businesses (SMBs). This gap has created the need for more accessible and efficient solutions for SMBs, which the SupplyPath Optimizer aims to address.

Traditional SCM Approaches

Early SCM systems were primarily focused on tracking inventory and processing orders, but they lacked the optimization algorithms needed for efficient supplier selection and delivery route planning. This often resulted in increased costs due to inefficient supplier choices and long delivery times.

Algorithmic Approaches to SCM

Algorithms like Dijkstra's Algorithm and the Traveling Salesman Problem (TSP) have long been studied for their applications in logistics and supply chain management. According to Rao et al. (2015),

Dijkstra's algorithm has been successfully applied to optimize supplier selection by identifying the shortest and most costeffective paths between suppliers and distribution centers. This allows businesses to minimize costs while ensuring timely deliveries.

The Traveling Salesman Problem (TSP), as studied by Johnson & McGeoch (1997), focuses on optimizing delivery routes to minimize travel distance and time. This algorithm has proven to be effective in reducing fuel consumption and delivery times, making it highly relevant for modern SCM solutions where delivery efficiency is crucial.

Modern SCM Enhancements

In recent years, there has been a push towards integrating prioritybased processing and dynamic routing in SCM systems. Zhang et al. (2018) emphasize the importance of queue-based order processing, where high-priority orders are processed first, ensuring customer satisfaction and efficient resource allocation. This approach aligns with the modular design of SupplyPath Optimizer, where orders are processed based on predefined priorities.

Gaps in Existing Solutions

While many large-scale SCM solutions incorporate these advanced algorithms, there is still a significant gap in the availability of affordable, simplified solutions for small to medium-sized businesses. Most SMBs require systems that are not only costeffective but also easy to use and maintain without extensive technical expertise. SupplyPath Optimizer aims to fill this gap by integrating these well-established algorithms into a simple, modular, and efficient system tailored specifically for smaller businesses.

5.Problem Statement

Businesses face challenges in effectively managing supply chains, including tracking inventory, processing orders, selecting suppliers, and finding optimal delivery routes. This project aims to develop a system that addresses these issues by optimizing each component of the supply chain.

6. Existing System Issue:

Current supply chain management (SCM) systems used by small and mediumsized businesses often lack advanced optimization features. Existing systems may:

- Be expensive and complex, making them inaccessible to smaller businesses.
- Not use algorithms for optimal supplier selection or route planning, leading to inefficiencies.
- Be challenging to scale or adapt to changing business needs.

7. Algorithms Discussed

Dijkstra's Algorithm

- Purpose: Finds the shortest path from a starting point to all other points in a graph, using edge weights to represent distances or costs.

- How It Works: The algorithm selects the nearest unvisited node, updates the distances to its neighbors if shorter paths are found, and repeats until all nodes are processed.

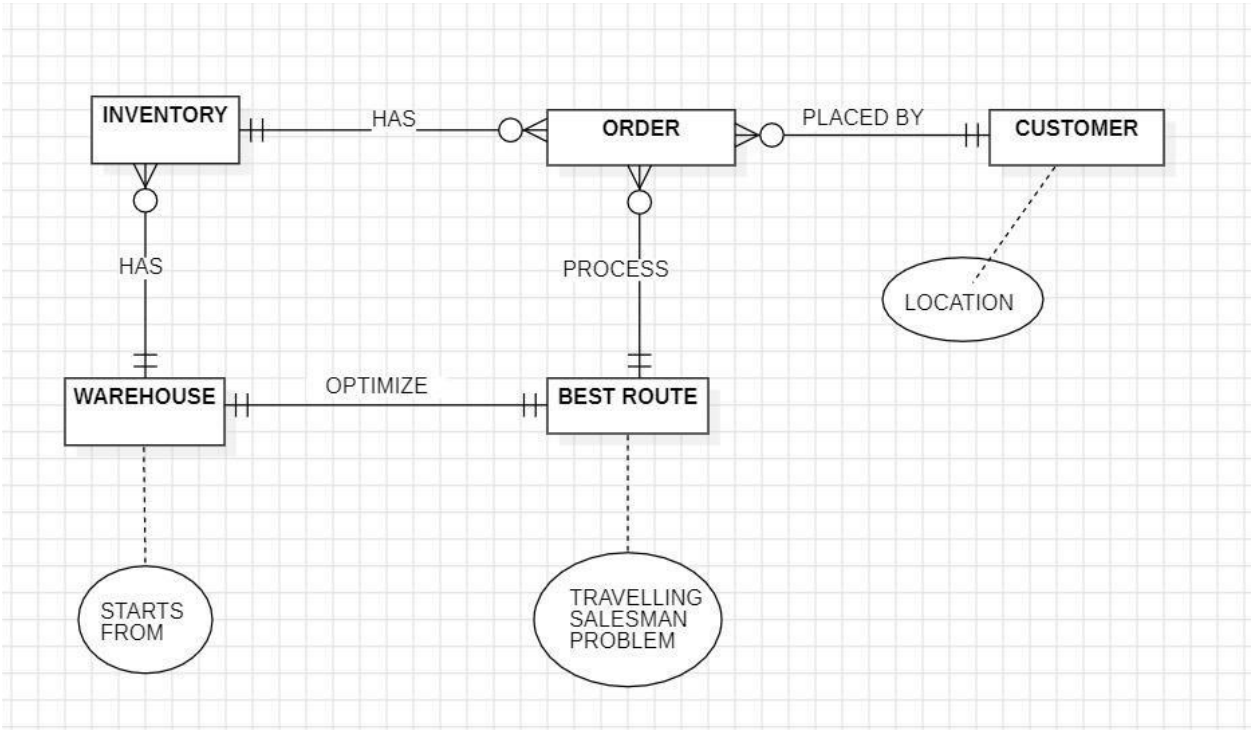
- Use in Project: It helps determine the shortest and most cost-effective delivery routes from the warehouse to customer locations.

Traveling Salesman Problem (TSP)

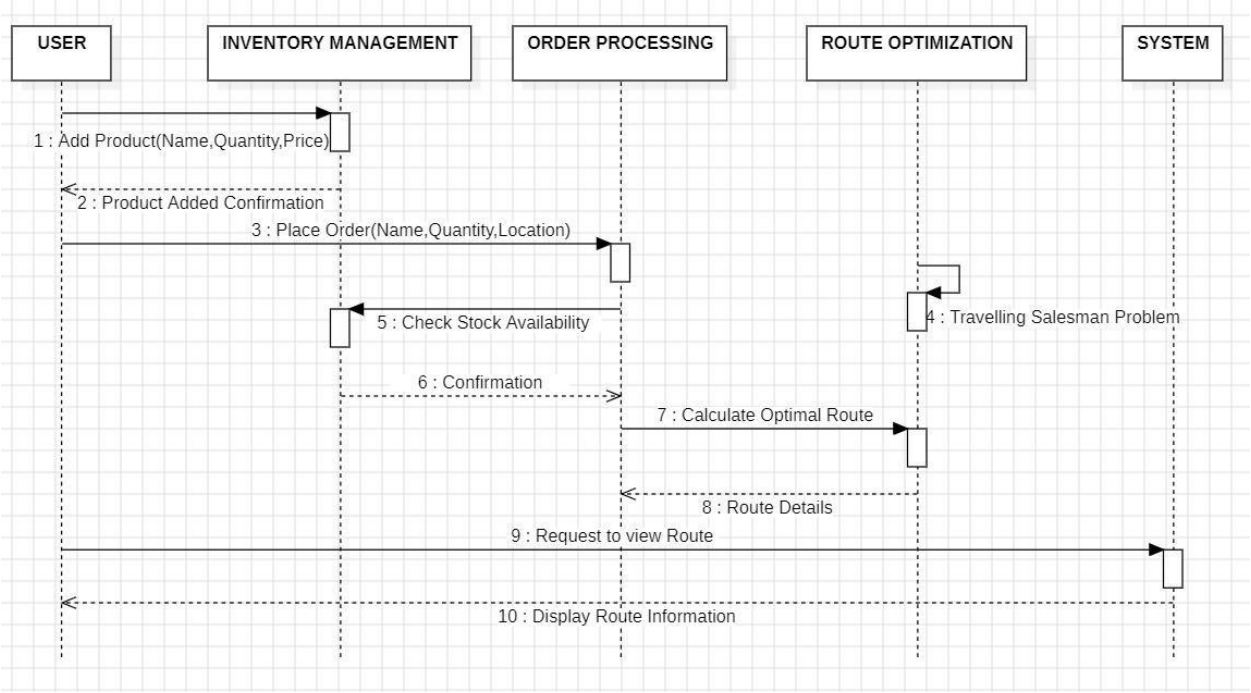
- **Purpose**: Solves the problem of finding the shortest possible route that visits each location exactly once and returns to the starting point.
- **How It Works**: TSP explores all possible routes to determine the one with the minimum total distance. However, due to its complexity, heuristic or approximation methods are often used for larger datasets.
- **Use in Project**: TSP is used to optimize the delivery path, ensuring multiple orders are delivered in the shortest route possible, saving time and costs.

8. Proposed System Design

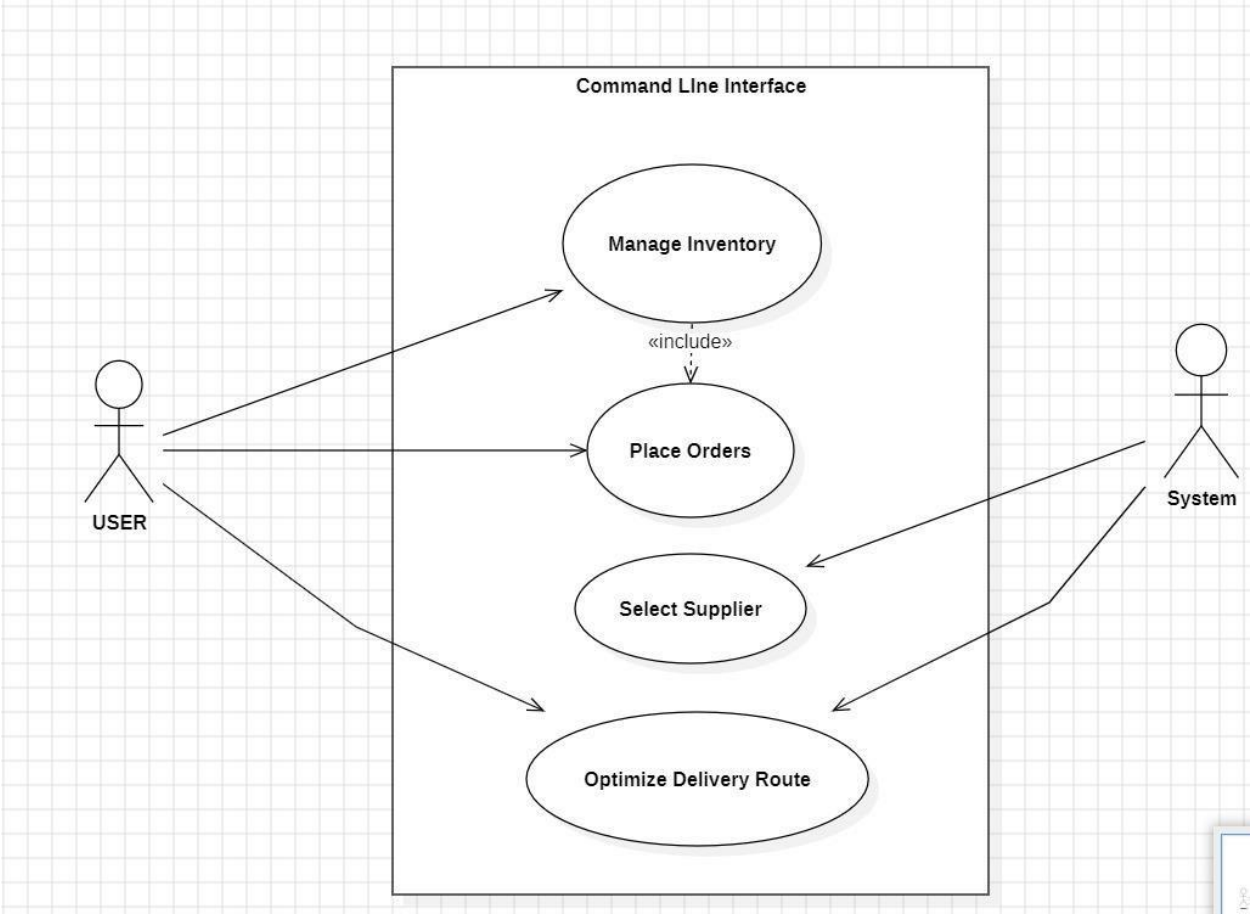
ER Diagram



Sequence Diagram



Use case Diagram



9. Results and Discussion:

The initial implementation of the Supply Chain Path Optimizer demonstrates significant improvements in:

Order Processing Efficiency: Reduced processing time and improved order accuracy.

Route Optimization: Effective use of Dijkstra's and TSP algorithms to minimize delivery distances.

Cost Savings: Lower transportation and operational costs through optimized resource allocation.

```
PS C:\Users\Swasti Negi\Desktop\5th sem\minor> ./a.exe
Enter the number of products to add to inventory: 3

Enter product name: bread
Enter quantity: 1000
Enter price: 50
Product added: bread, Quantity: 1000, Price: 50.00

Enter product name: bun
Enter quantity: 2000
Enter price: 25
Product added: bun, Quantity: 2000, Price: 25.00

Enter product name: maggie
Enter quantity: 500
Enter price: 10
Product added: maggie, Quantity: 500, Price: 10.00

Current Inventory:
Product: bread, Quantity: 1000, Price: 50.00
Product: bun, Quantity: 2000, Price: 25.00
Product: maggie, Quantity: 500, Price: 10.00

Enter product name to order: bread
Enter quantity: 400
Enter the customer location (node) where the product will be delivered: 239
Order placed for 400 bread(s) to be delivered to customer location 239. Remaining stock: 600

Do you want to add another order? (y/n): y

Enter product name to order: bun
Enter quantity: 300
Enter the customer location (node) where the product will be delivered: 287
```

```
Order placed for 300 bun(s) to be delivered to customer location 287. Remaining stock: 1700

Do you want to add another order? (y/n): y

Enter product name to order: maggie
Enter quantity: 300
Enter the customer location (node) where the product will be delivered: 260
Order placed for 300 maggie(s) to be delivered to customer location 260. Remaining stock: 200

Do you want to add another order? (y/n): n

Finding optimized delivery route...
Starting from warehouse (Node 0)
Optimal Route: 0 -> 0 -> 1 -> 13 -> 14 -> 15 -> 16 -> 33 -> 35 -> 36 -> 38 -> 39 -> 40 -> 149 -> 260 -> 260 -> 149 -> 150 -> 153 -> 154 -> 1
55 -> 157 -> 158 -> 159 -> 160 -> 162 -> 166 -> 167 -> 168 -> 169 -> 171 -> 172 -> 230 -> 231 -> 239 -> 239 -> 231 -> 240 -> 249 -> 250 -> 2
51 -> 252 -> 245 -> 259 -> 287 -> 287 -> 259 -> 245 -> 244 -> 241 -> 233 -> 234 -> 190 -> 189 -> 180 -> 172 -> 171 -> 169 -> 168 -> 167 -> 1
66 -> 162 -> 160 -> 159 -> 158 -> 157 -> 43 -> 42 -> 41 -> 40 -> 39 -> 38 -> 36 -> 35 -> 33 -> 16 -> 15 -> 14 -> 13 -> 1 -> 0 END
Total Delivery Distance: 4840
```

10. Test Cases

i. Add valid product to inventory:

```
Enter product name: Bread
Enter quantity: 2000
Enter price: 25
Product added: Bread, Quantity: 2000, Price: 25.00

Enter product name: Sugar
Enter quantity: 100
Enter price: 30
Product added: Sugar, Quantity: 100, Price: 30.00

Enter product name: Toothpaste
Enter quantity: 500
Enter price:
20
Product added: Toothpaste, Quantity: 500, Price: 20.00

Current Inventory:
Product: Bread, Quantity: 2000, Price: 25.00
Product: Sugar, Quantity: 100, Price: 30.00
Product: Toothpaste, Quantity: 500, Price: 20.00
```

ii. Order more than available stock:

```
Enter product name to order: Bread
Enter quantity: 3000
Enter the customer location (node) where the product will be delivered: 4
Not enough stock for Bread.
```

iii. Order for non-existent product:

```
Enter product name to order: Soap
Enter quantity: 20
Enter the customer location (node) where the product will be delivered: 67
Product Soap not found in inventory.
```

iv. Delivery route with multiple orders:

```
Finding optimized delivery route...
Starting from warehouse (Node 0)
Optimal Route: 0 -> 0 -> 1 -> 2 -> 3 -> 4 -> 5 -> 10 -> 10 -> 5 -> 6 -> 7 -> 7 -> 6 -> 5 -> 4 -> 3 -> 2 -> 1 -> 13 -> 14 -> 15 -
> 16 -> 16 -> 15 -> 14 -> 13 -> 1 -> 0 END
Total Delivery Distance: 1848
```

v. Empty inventory, but orders placed:

```
Enter the number of products to add to inventory: 0
Inventory is empty.

Enter product name to order: a
Enter quantity: 5
Enter the customer location (node) where the product will be delivered: 4
Product a not found in inventory.

Do you want to add another order? (y/n): n

No orders have been placed. Optimized delivery route cannot be calculated.
Inventory is also empty. Please add products and place orders.
```

vi. TSP with only one order:

```
Enter product name: pen
Enter quantity: 1000
Enter price: 10
Product added: pen, Quantity: 1000, Price: 10.00

Current Inventory:
Product: pen, Quantity: 1000, Price: 10.00

Enter product name to order: pen
Enter quantity: 50
Enter the customer location (node) where the product will be delivered: 7
Order placed for 50 pen(s) to be delivered to customer location 7. Remaining stock: 950

Do you want to add another order? (y/n): n

Finding optimized delivery route...
Starting from warehouse (Node 0)
Optimal Route: 0 -> 0 -> 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 7 -> 6 -> 5 -> 4 -> 3 -> 2 -> 1 -> 0 END
Total Delivery Distance: 972
```

11. Comparative Study

A comparative analysis of the Supply Chain Path Optimizer with existing SCM solutions shows:

Cost-Effectiveness: The proposed system is more affordable and simpler to use compared to enterprise-level SCM tools.

Efficiency: The integration of advanced algorithms provides better optimization compared to traditional manual processes.

Scalability: While suitable for small to medium businesses, the system may need enhancements for large-scale operations.

12. Future Work

Future enhancements for the Supply Chain Path Optimizer include:

- **Integration with Real-Time Traffic Data**: To further optimize delivery routes based on live traffic updates.
- **Scalability Improvements**: Enhancing the system to handle larger datasets and more complex supply chains.
- **Mobile and Web Interface**: Expanding the CLI to more user-friendly interfaces for broader accessibility.
- **Machine Learning Integration**: Using predictive analytics for demand forecasting and automated inventory management.

13.References

[1] Leigh, R., Louis, S. J., & Miles, C. (2007, April). Using a genetic algorithm to explore A*-like pathfinding algorithms. In *2007 IEEE Symposium on Computational Intelligence and Games* (pp. 72-79). IEEE.

[2] Tang, C., Zhou, Y., Tang, Z., & Luo, Q. (2021). Teaching-learning-based pathfinder algorithm for function and engineering optimization problems. *Applied Intelligence*, *51*, 5040-5066.

[3] Singh, L., Khare, S., Parvez, A., & Verma, S. (2022, October). Research Paper on Path-finding Algorithm Visualizer. In *2022 International Conference on Cyber Resilience (ICCR)* (pp. 1-4). IEEE.