A.S.T.R.O.
(Advanced Supply and Trade Resource Optimisation)

Project Group 03

Amel Chandra	(U2109009)
Bharath S	(U2109018)
Joepaul Vilsan	(U2109033)
Shane George Salphie	(U2109063)

Under the Guidance of

Dr. Nikhila T. Bhuvan,
Assosciate Professor,
Department of CSBS

RSET
RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

Dept. of Computer Science and Business Systems,
RSET

INTRODUCTION

- **Introduction**
  - ASTRO helps small vendors with purchasing and delivery challenges.
  - Vendors struggle with high costs, poor supply chains, and low bargaining power.
  - Key features:
    - Order aggregation for bulk discounts.
    - Demand forecasting for better planning.
    - Optimized supplier selection for cost savings.
  - Uses Genetic Algorithm for efficient deliveries.
  - Compares supplier selection methods for best deals.
  - Provides a scalable, cost-effective supply chain solution.
  - Boosts competitiveness and sustainability for vendors.

PROBLEM STATEMENT

- Design a collaborative platform which
  - Enables multiple vendors to aggregate orders, facilitating bulk purchasing and cost savings.
  - Demand forecasting to predict product needs accurately
  - Implement route optimization techniques to establish an efficient delivery network.
- This approach aims to enhance supply chain efficiency, reduce operational costs, and strengthen the market position of small-scale vendors.

OBJECTIVES

- **Objectives of the Project**
  - Order Aggregation : Allow vendors to combine purchases for bulk discounts.
  - Demand Forecasting : Predict demand to optimize purchasing.
  - Logistics Optimization : Reduce delivery costs using Green Routing Optimization.
  - Supplier Selection : Identify cost-effective and reliable suppliers.
  - Vendor Competitiveness : Help small vendors compete with large retail chains.

- **Importance and relevance of the project**

ASTRO helps small-scale vendors reduce costs, improve supply chain efficiency, and compete with large retail chains. By enabling order aggregation, demand forecasting, and optimized logistics, the platform ensures better pricing, faster deliveries, and smarter purchasing decisions. It promotes cost-effectiveness, scalability, and sustainability, making it highly relevant in today's competitive market.

PROJECT SCOPE

- **Project Scope: Boundaries and Focus**
  - ASTRO is designed to enhance procurement and logistics for small-scale vendors by leveraging order aggregation, demand forecasting, and supplier selection optimization. The platform aims to reduce procurement costs, streamline deliveries, and improve vendor-supplier collaboration, making supply chain management more efficient, data-driven, and cost-effective.

- **What is Covered**
  - **Order Aggregation** : Enables vendors to combine orders for bulk purchase discounts.
  - **Demand Forecasting** : Utilizes data-driven analytics to predict demand and optimize procurement.
  - **Logistics Optimization** : Implements Green Routing Optimization for cost-efficient and timely deliveries.
  - **Supplier Coordination** : Streamlines vendor-supplier interactions for efficient order processing

- **What is Excluded**
  - **Inventory Management** : Does not handle warehouse stock tracking or inventory control.
  - **Retail Consumer Sales** : Exclusively focuses on vendor-supplier transactions, not direct customer sales.
  - **Financial Transactions** : Does not process payments, financing, or lending services.

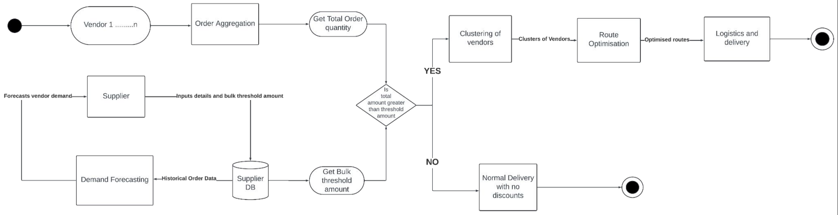
METHODOLOGY

- **Tools & Technologies**
  - **Frontend :** Flutter
  - **Backend :** FastAPI Python
  - **Programming Languages :** Python and Dart
  - **Cloud :** Firebase
  - **APIs & Integrations :** Google Maps API

- **Dataset**
  - Store Item Demand Forecasting Challenge - Kaggle Dataset (Kaggle Link)
  - Random Google Maps Business Locations - South India

- **Algorithms Used**
  - **Route Optimization :** Genetic Algorithm
  - **Order Aggregation :** DBSCAN alongwith Genetic Algorithm
  - **Demand Forecasting :** NeuralProphet (Link)

SYSTEM DESIGN



IMPLEMENTATION

- **Data Collection & Storage**
  - Orders are stored in Google Firestore with details like supplier ID, vendor location, order quantity, and status.
  - Suppliers set a Minimum Order Quantity (MOQ) for aggregation.

- **Order Aggregation using DBSCAN alongwith GENETIC ALGORITHM**
  - Location-Based Clustering:Orders from vendors are clustered using DBSCAN based on real-world (Haversine) distances.
  - Dynamic Parameter Tuning:A Genetic Algorithm optimizes DBSCAN's parameters (eps, min_samples) for better clustering accuracy..
  - MOQ Validation & Fallback:Clusters are accepted only if they meet Minimum Order Quantity (MOQ); otherwise, orders remain pending. Default DBSCAN is used if order volume is low.

IMPLEMENTATION

- **Order Aggregation using DBSCAN alongwith GENETIC ALGORITHM**
  - **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**
    - Unsupervised clustering algorithm.
    - Groups dense points; marks low-density points as outliers.
    - No need to predefine the number of clusters (unlike k-means).
  - **Why Use DBSCAN Instead of Other Algorithms?**
    - Better for Geospatial Clustering: Excels at location-based clustering.
    - Automatic Cluster Discovery: No need to predefine the number of clusters.
    - Handles Noise & Outliers: Avoids forcing outliers into clusters.
    - Flexible Shapes: Detects irregular cluster shapes (unlike k-means).
  - Why Use GENETIC ALGORITHM
    - Traditional DBSCAN requires fixed eps and min_samples, which don't adapt to changing vendor order patterns.
    - To overcome this, we combine DBSCAN with a **Genetic Algorithm (GA)** that **automatically optimizes** these parameters for each dataset.

IMPLEMENTATION

- **Order Aggregation using DBSCAN Clustering**
  - **How DBSCAN alongwith GENETIC ALGORITHM Works?**
    - **Input**: Order coordinates (latitude, longitude), and quantities.
    - **Distance Calculation**: Compute **Haversine distance matrix** between all vendor locations.
    - **Optimization**: Run Genetic Algorithm to find optimal eps and min_samples for DBSCAN.
    - **DBSCAN Clustering**:
      - Run DBSCAN using the GA-tuned parameters.
      - Identify clusters of geographically close orders.
    - **MOQ Validation**:
      - Each cluster is accepted **only if it meets MOQ (plus buffer)**.
      - Outliers or low-quantity clusters are kept **pending**.
  - **Advantages Over Traditional DBSCAN**
    - **Dynamic Parameter Tuning**: Automatically finds the best eps and min_samples using Genetic Algorithm.
    - **MOQ-Aware Clustering**: Ensures clusters are only accepted if they meet Minimum Order Quantity.
    - **Adaptive to Varying Data Volumes**: Switches between optimized and default DBSCAN based on order count.
    - **Real-World Distance Support**: Uses Haversine distance for accurate location-based clustering.
    - **Robust Outlier Handling**: Effectively filters noise and avoids misgrouping of outliers.

IMPLEMENTATION

- **Route Optimization using Ant Colony Optimization (ACO) :**
  - Aggregated orders require efficient delivery routes.
  - ACO algorithm is used to optimize the shortest and most cost-effective route from supplier to vendors.
  - This minimizes fuel costs and delivery time.

- **Demand Forecasting using NeuralProphet :**
  - Historical order data is analyzed using NeuralProphet (an advanced time-series forecasting model).
  - Forecasts future demand patterns, helping suppliers plan inventory and pricing.

IMPLEMENTATION

- **Route Optimization using Ant Colony Optimization (ACO) :**
  - **Ant Colony Optimization (ACO) for Route Optimization**
    - A metaheuristic optimization algorithm inspired by the foraging behavior of ants.
    - It is used to find the optimal route for logistics and transportation problems.
  - **Why Use ACO?**
    - Efficient Routing: Finds cost-effective and time-efficient paths.
    - Dynamic Adaptation: Adjusts to real-time changes in routes.
    - Scalable: Works with large datasets of locations.
    - Energy-Efficient: Minimizes fuel costs and distance traveled.
  - **How ACO Works?**
    - Define Parameters:
      - Pheromone Influence (): Importance of past solutions.
      - Heuristic Influence (): Importance of distance.
      - Evaporation Rate: Prevents unlimited pheromone accumulation.

IMPLEMENTATION

- **Route Optimization using Ant Colony Optimization (ACO) :**
  - **How ACO Works?**
    - Route Construction:
      - Ants are placed at random locations.
      - Each ant builds a route probabilistically based on:
      - Pheromone trails (past successful paths).
      - Heuristic value (shorter distances, lower fuel cost).
  - **ACO in this Project**
    - Extracts Order Locations (latitude & longitude).
    - Constructs Distance Matrix using Google Maps API.
    - Optimizes Fuel Cost by considering cost per liter.
    - Builds Routes using probability-based selection.
    - Applies 2-Opt Local Search for further route optimization.
    - Ensures Eco-Friendly Routing by minimizing unnecessary travel.
    - Stops Early if no improvement is detected to save computation time.

IMPLEMENTATION

- **Route Optimization using Genetic Algorithm (GA):**
  - **Genetic Algorithm (GA) for Route Optimization**
    - Inspired by natural selection, GA evolves multiple solutions to find the best delivery routes.
    - More efficient than traditional routing by exploring multiple possibilities in parallel.
  - **Why Use Genetic Algorithms?**
    - Efficient Optimization: Faster than brute-force search.
    - Handles Constraints: Adapts to time windows & traffic.
    - Scalable: Works with large datasets.
    - Avoids Local Minima: Prevents getting stuck in suboptimal solutions.
  - **How It Works:**
    - Key Components:
      - Population: Set of possible routes.
      - Fitness Function: Evaluates route efficiency.
      - Selection: Picks best routes for reproduction.
      - Crossover: Combines parent routes.
      - Mutation: Introduces random changes for diversity.

IMPLEMENTATION

- **Route Optimization using Genetic Algorithm (GA):**
  - **How It Works:**
    - Initialize: Generate routes (Greedy Nearest Neighbor, 2-Opt).
    - Evaluate: Compute distance, favor shorter routes.
    - Select: Pick best routes (Tournament Selection).
    - Crossover & Mutation: Mix routes (OX, ERX) + random changes.
    - Refine: Apply 2-Opt for optimization.
    - Converge: Stop when no further improvement.
  - **GA in this Project**
    - Extracts order locations (latitude & longitude).
    - Computes Distance Matrix using Google Maps API & Haversine Formula.
    - Initializes diverse route solutions for efficiency.
    - Runs GA with adaptive mutation rates for optimization.
    - Applies 2-Opt Local Search for final refinement.
    - Selects the best route with minimal delivery time & cost.

CHALLENGES

- **Order Aggregation Challenges**
  - **Issue:**
    - Clustering vendor orders using DBSCAN required careful tuning of parameters like eps (neighborhood radius) and min_samples.
    - Additionally, clustering had to satisfy **Minimum Order Quantity (MOQ)** constraints while reducing outliers.
  - **Solution:**
    - Replaced manual tuning with **Genetic Algorithm (GA)** to dynamically optimize eps and min_samples for DBSCAN.
    - GA uses a fitness function that maximizes valid clusters while minimizing noise.
    - Integrated MOQ checks post-clustering to ensure only qualifying clusters are confirmed for aggregation.
    - Enhanced outlier handling by isolating orders that don’t meet MOQ, marking them as pending for future rounds.
- **Route Optimization Challenges**
  - **Issue:**
    - Genetic Algorithm required careful tuning of parameters such as population size, mutation rate, and crossover strategy to effectively handle real-world constraints like traffic patterns, delivery time windows, and vendor/supplier availability.

.

CHALLENGES

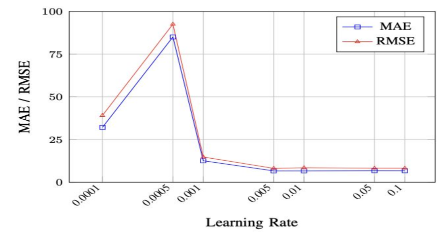
- **Solution:**
  - **Optimized GA parameters to balance exploration and exploitation in the route search process.**
  - **Optimized GA parameters to balance exploration and exploitation in the route search process.**
  - **Integrated domain-specific heuristics into the fitness function to account for real-time delivery constraints such as time windows, capacity limits, and location clustering.**
  - **Ensured diversity in the population using mutation and elitism to avoid premature convergence.**

Demand Forecasting Issue

- **Issue:**
  - Forecasting demand using NeuralProphet required high-quality historical data.
- **Solution:**
  - Applied data cleaning techniques (removing inconsistencies, handling missing values).
  - Used hyperparameter tuning to improve forecast accuracy.

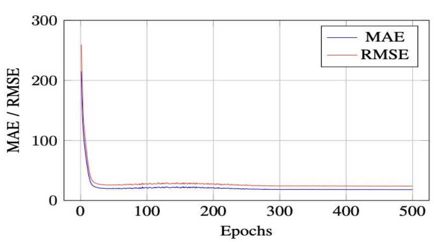
RESULTS AND EVALUATION

- We used Prophet and NeuralProphet for comparison, but we chose NeuralProphet for our work due to its active development and performance benefits.
- Prophet is discontinued, while NeuralProphet is actively maintained and improved.Bug fixes and feature enhancements are still happening in NeuralProphet.
- NeuralProphet is built on PyTorch, it can leverage GPUs, making it faster for large datasets compared to Prophet as it uses Stan which does not support GPU computation.



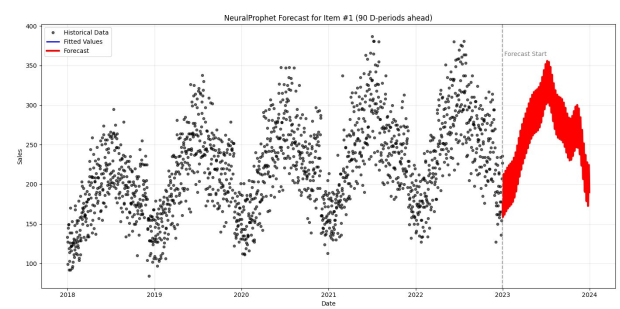
RESULTS AND EVALUATION

Effect of Epochs on MAE and RMSE



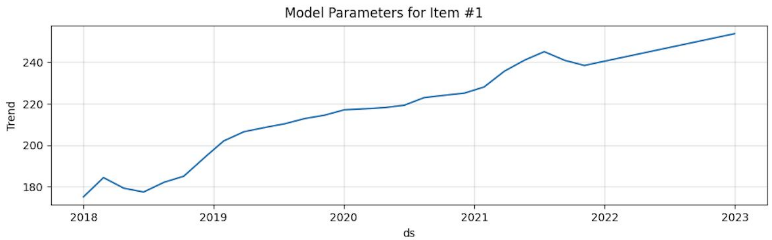
RESULTS AND EVALUATION

NeuralProphet Demand Forecasting



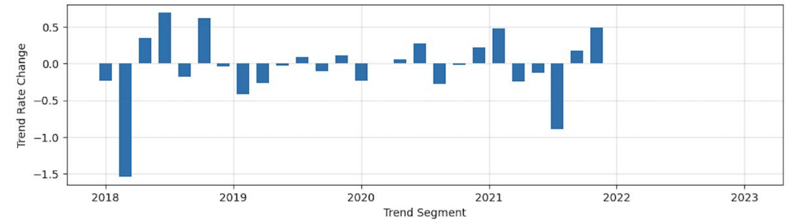
RESULTS AND EVALUATION

Yearly Seasonality



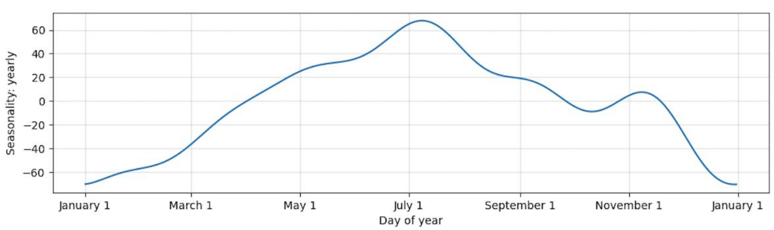
RESULTS AND EVALUATION

Long-Term Trend Analysis



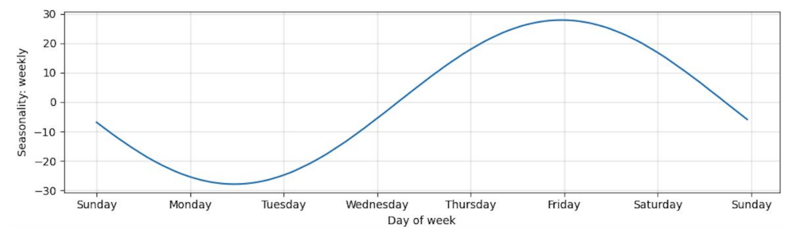
RESULTS AND EVALUATION

Yearly Seasonality



RESULTS AND EVALUATION

Weekly Seasonality



RESULTS AND EVALUATION

We compared Genetic Algorithm (GA), Ant Colony Optimization (ACO), and the Christofides Algorithm for solving the route optimization problem in our system.

Genetic Algorithm (GA) uses evolutionary techniques such as crossover, mutation, and selection to iteratively evolve better routes. It is highly adaptable, capable of handling complex constraints like delivery windows, capacity limits, and varying cluster sizes.

ACO is designed specifically for routing problems and uses pheromone trails to reinforce promising paths. While it can dynamically adapt to optimal routes, we found it prone to premature convergence, especially in scenarios with fewer iterations or complex constraints.

In our evaluation, GA consistently outperformed ACO, finding shorter and more feasible delivery routes. This may be due to GA's ability to explore the solution space more effectively, avoiding local optima through genetic variation.

The Christofides Algorithm, while useful for theoretical approximation of the Travelling Salesman Problem (TSP), lacked flexibility. Its static nature made it less suitable for real-world delivery constraints where adaptability is crucial.

Based on these comparisons, Genetic Algorithm was chosen for route optimization due to its superior adaptability, scalability, and ability to deliver high-quality solutions under real-world conditions.

RESULTS AND EVALUATION

Algorithm	Vendors	Dist. (km)	Cost (Rs.)	Time (s)
ACO	20	5197.54	29811.51	23.26
	50	8333.63	47618.80	50.43
	100	9648.29	54444.64	148.69
GA	20	4703.90	22320.78	20.53
	50	6993.30	33770.55	60.75
	100	8581.06	40956.56	142.21

RESULTS AND EVALUATION

We evaluated Agglomerative Clustering, K-means, DBSCAN, and a Genetic Algorithm-enhanced DBSCAN (GA-DBSCAN) to group orders geographically. GA-DBSCAN outperformed all others in silhouette scores and noise reduction for datasets of 100 orders and above.

• **DBSCAN** works well for smaller datasets due to low overhead.

• **GA-DBSCAN** adaptively selects parameters (ε and MinPts), ensuring optimal clustering even under dynamic order distributions.

• **K-means** and **Agglomerative** produced higher cluster counts but lacked flexibility in handling noise.

RESULTS AND EVALUATION

Algorithm	Orders	Time (s)	Clusters	Outliers	Silh. Score
DBSCAN	25	0.002	7	16	0.350
	50	0.002	13	32	0.330
	100	0.003	28	68	0.392
	150	0.004	39	99	0.380
	200	0.005	63	118	0.375
	300	0.009	85	187	0.361
	350	0.010	98	219	0.355
GA-DBSCAN	25	0.020	0	25	N/A
	50	0.025	0	50	N/A
	100	0.025	11	89	0.735
	150	0.030	12	138	0.780
	200	0.036	39	156	0.726
	300	0.048	35	262	0.715
	350	0.055	45	301	0.629
K-means	25	0.030	6	18	0.358
	50	0.004	12	34	0.324
	100	0.004	26	72	0.392
	150	0.005	37	103	0.390
	200	0.032	61	122	0.385
	300	0.007	84	189	0.363
	350	0.008	94	227	0.350
Agglomerative	25	0.002	6	18	0.341
	50	0.002	12	34	0.337
	100	0.002	27	70	0.393
	150	0.004	38	101	0.358
	200	0.005	61	122	0.359
	300	0.007	84	189	0.359
	350	0.009	95	225	0.349

RESULTS AND EVALUATION

Cost-Benefit Analysis

In a 50-vendor scenario, our collaborative route model showed:

- **Distance savings**: 80.4%
- **Route cost savings**: 80.4%
- **Fuel cost savings**: 80.3%

Metric	Trad.	Collab.	Savings
Distance (km)	35,606	6,993	28,613 (80.4%)
Route Cost (Rs.)	36,929	7,255	29,675 (80.4%)
Fuel (L)	1,811	357	1,454 (80.3%)
Fuel Cost (Rs.)	171,135	33,770	137,366 (80.3%)

CONCLUSION

- **Summary of the Work Done**
  - ASTRO was developed to enhance procurement and logistics for small-scale vendors.
  - Implemented DBSCAN with GENETIC algorithm for order aggregation, Genetic Algorithm for route planning, and NeuralProphet for demand forecasting.
- **Limitations of the Project**
  - Accuracy of demand forecasting depends on data availability.
  - Real-world implementation may require additional infrastructure and funding.
  - Inventory management is not covered in the current scope.
- **Suggestions for Future Work**
  - Improve supplier selection with advanced ML models.
  - Integrate real-time tracking for better delivery monitoring.
  - Expand dataset coverage to improve forecasting accuracy.

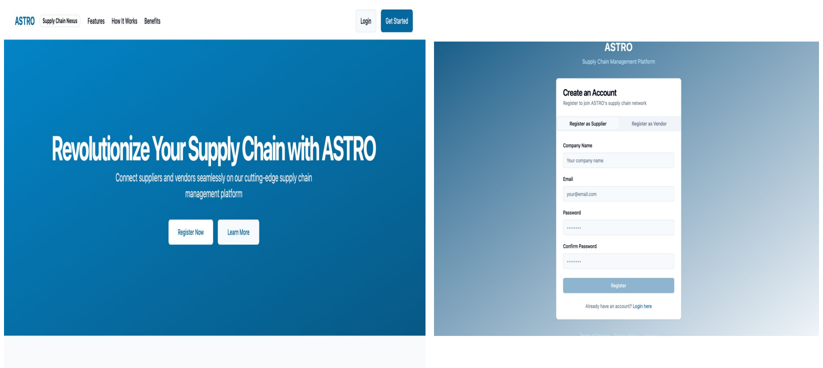
REFERENCES

- X. Wang and L. Wang, **"Green routing optimization for logistics distribution with path flexibility and service time window,"** in 15th International Conference on Service Systems and Service Management (ICSSSM), 2018, pp. 1–5.
- E. M. U. S. B. Ekanayake, S. P. C. Perera, W. B. Daundasekara, and Z. A. M. S. Juman, **"A modified ant colony optimization algorithm for solving a transportation problem,"** J. Adv. Math. Comput. Sci., vol. 35, no. 5, pp. 83–101, Aug. 2020.
- J. Zhou, **"Research on multi-objective green vehicle path optimization based on whale algorithm,"** in International Conference on Networking, Informatics and Computing (ICNETIC), 2023, pp. 385–389.
- K. A. Mamoun, L. Hammadi, A. E. Ballouti, A. G. Novaes, and E. S. D. Cursi, **"Vehicle routing optimization algorithms for pharmaceutical supply chain: A systematic comparison,"** Transp. Telecommun. J., vol. 25, no. 2, pp. 161–173, 2024.
- A. Maroof, B. Ayvaz, and K. Naeem, **"Logistics optimization using hybrid genetic algorithm (HGA): A solution to the vehicle routing problem with time windows (VRPTW),"** IEEE Access, vol. 12, pp. 36974–36989, 2024.

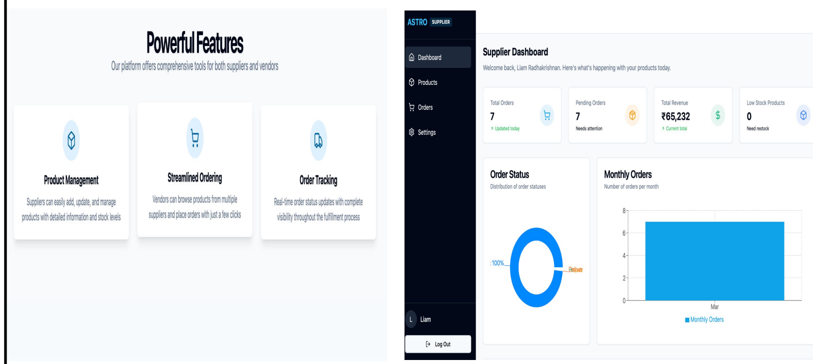
REFERENCES

- S. J. Taylor and B. Letham, **"Forecasting at scale,"** PeerJ Preprints, vol. 5, p. e3190v2, Sep. 2017.
- O. Triebe, H. Hewamalage, P. Pilyugina, N. Laptev, C. Bergmeir, and R. Rajagopal, "NeuralProphet: Explainable forecasting at scale," 2021.
- X. Ai, Y. Yue, H. Xu, and X. Deng, **"Optimizing multi-supplier multi-item joint replenishment problem for non-instantaneous deteriorating items with quantity discounts,"** PLOS ONE, vol. 16, no. 2, pp. 1–22, Feb. 2021.
- S. B. Jouida and S. Krichen, **"A genetic algorithm for supplier selection problem under collaboration opportunities,"** J. Exp. Theor. Artif. Intell., vol. 34, no. 1, pp. 53–79, 2022.
- M. Seifbarghy, M. Shoeib, and D. Pishva, **"Coordination of a single-supplier multi-retailer supply chain via joint ordering policy considering incentives for retailers and utilizing economies of scale,"** Scientia Iranica, 2022.

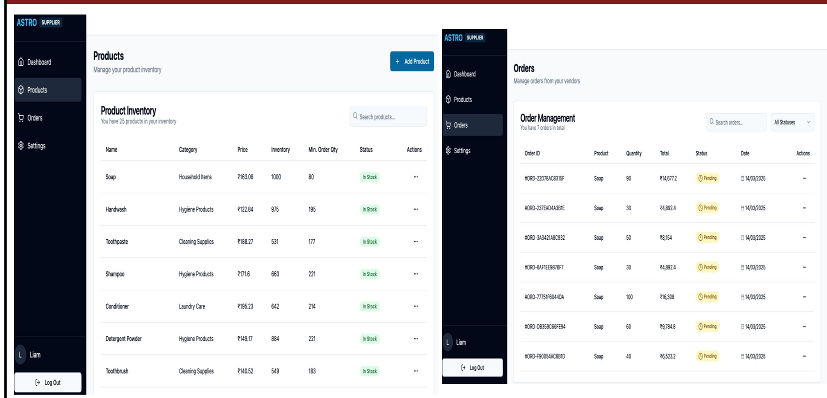
SCREENSHOTS



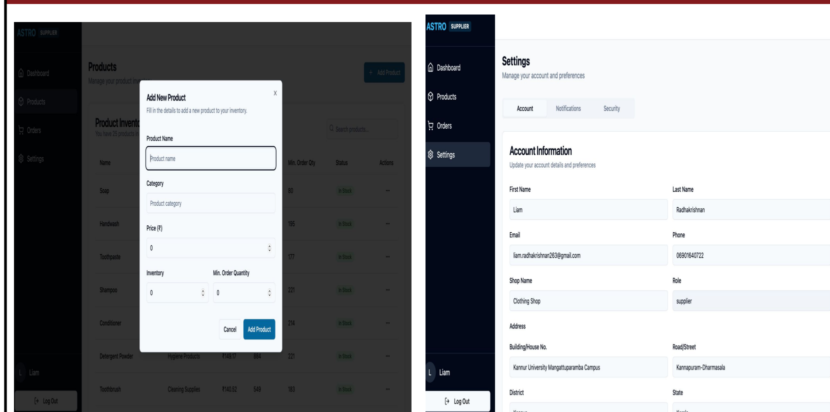
SCREENSHOTS



SCREENSHOTS



SCREENSHOTS



THANK YOU