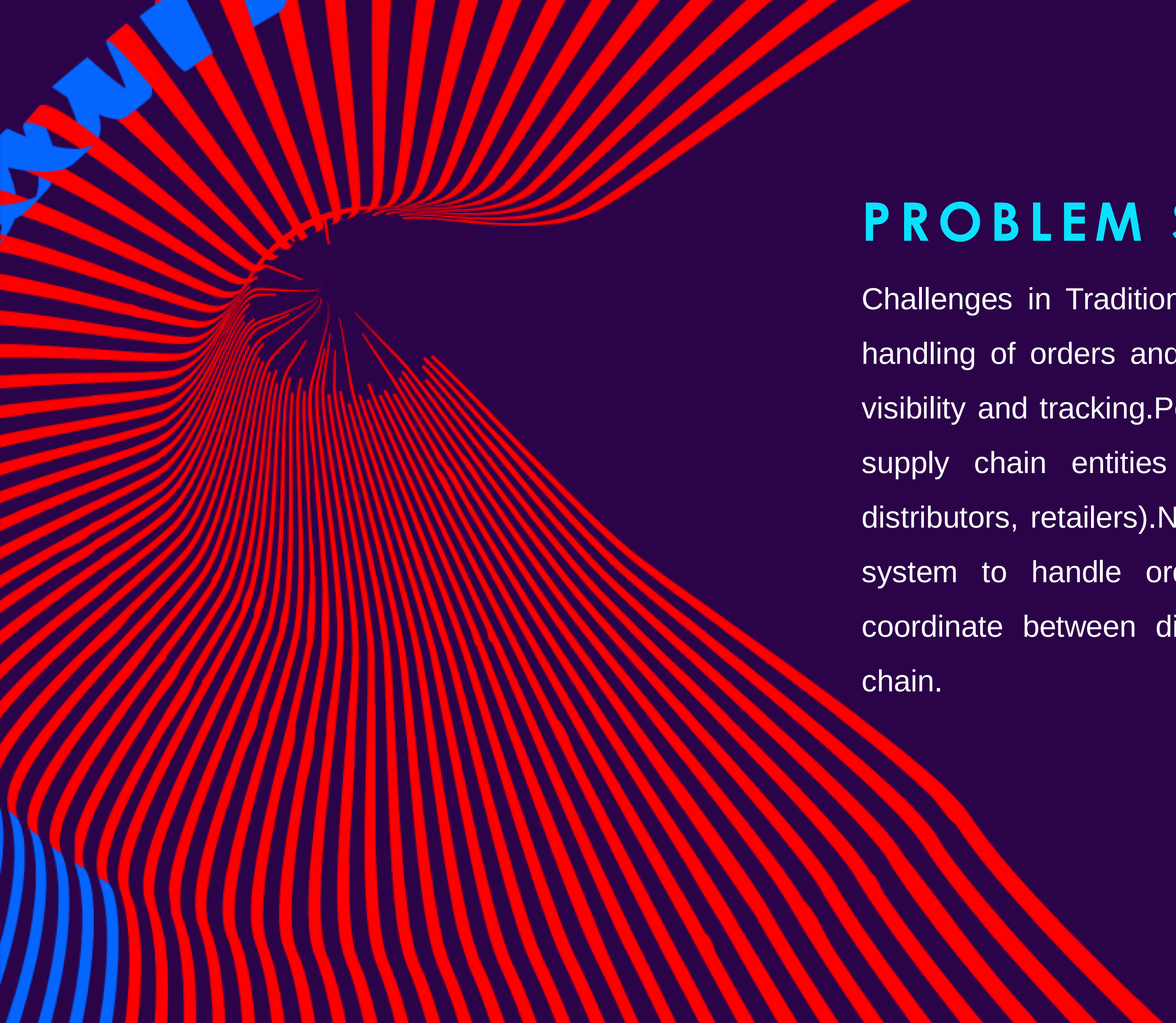
SUPPLY CHAIN MANAGEMENT SYSTEM

Fetch.ai

DESCRIPTION

Challenges in Traditional Supply Chains:Inefficient handling of orders and inventory.Lack of real-time visibility and tracking.Poor communication between supply chain entities (suppliers, manufacturers, distributors, retailers).Need: A streamlined, efficient system to handle orders, track materials, and coordinate between different roles in the supply chain.



PROBLEM STATEMENT

Challenges in Traditional Supply Chains:Inefficient handling of orders and inventory.Lack of real-time visibility and tracking.Poor communication between supply chain entities (suppliers, manufacturers, distributors, retailers).Need: A streamlined, efficient system to handle orders, track materials, and coordinate between different roles in the supply chain.

SOLUTION OVERVIEW

A comprehensive Supply Chain Management System integrating advanced technologies to improve operational efficiency.Key Features:Automated Order Placement and TrackingReal-time Communication Between Supply Chain RolesData-Driven Decision Making with Interactive Visualizations

TECH STACK: FRONT-END

REACT.JS	Framework for building dynamic user interfaces.Three.js & React Three
MATERIAL UI	A React component library that provides pre-designed UI components, ensuring a consistent and professional design.
FIBER	For creating interactive 3D visualizations.
GLASSMORPHISM	A design trend involving frosted glass effects for a modern, sleek look, implemented through CSS for a visually appealing UI.

TECH STACK: BACK-END

FLASK

A lightweight Python framework for building web applications and APIs. It handles routing, data processing, and integrates with UAgents.

UAgents

A framework for building multi-agent systems. It manages the different roles (Supplier, Manufacturer, Distributor, Retailer) and their interactions through HTTP endpoints.

Other tools

For integration and features related to decentralized systems

FRONTEND IMPLEMENTATION

USER INTERFACE:

Glassmorphism Design:

Utilized glassmorphism for a modern and appealing UI, providing a frosted glass effect with transparency.

Interactive Charts:

Integrated Chart.js for dynamic and interactive visualizations, enabling users to track order status and performance metrics in real-time.

Components:

- **Order Placement:** Input fields and buttons for placing orders, sending materials, and checking status.
- **Order History**: List displaying historical order data with statuses and details.Status
- **Updates**: Real-time status updates and feedback.

UI/UX

User Interface

REACT.JS COMPONENT

Modular components for different functionalities (order placement, status checking).

MATERIAL UI INTEGRATIO N

Consistent styling with pre-designed components for buttons, input fields, and containers.

GLASS-MORPHISM

Achieved using CSS, creating a frosted glass effect for a modern and visually appealing look.

UI/UX

User Experience

RESPONSIVE DESIGN

Ensures compatibility with different devices and screen sizes.

INTERACTIVE ELEMENTS

Includes buttons, input fields, and dynamic charts for enhanced user interaction.

CHARTS AND GRAPHS

Interactive charts for visualizing data (e.g., order history, material status).

BACKEND IMPLEMENTATION

Flask Framework:

Routing: Defines endpoints for order placement, material handling, and status checking.

Error Handling: Robust error handling mechanisms to ensure smooth operations and informative responses.

UAgents:

Supplier Agent: Manages supplier interactions and processes incoming orders.

Manufacturer Agent: Handles manufacturing tasks and material processing.

Distributor Agent: Coordinates distribution of goods.

Retailer Agent: Manages retail operations and customer interactions.

Contract Management: State machine contracts for managing transitions between different supp

chain states.

APPROACH AND METHODOLOGY

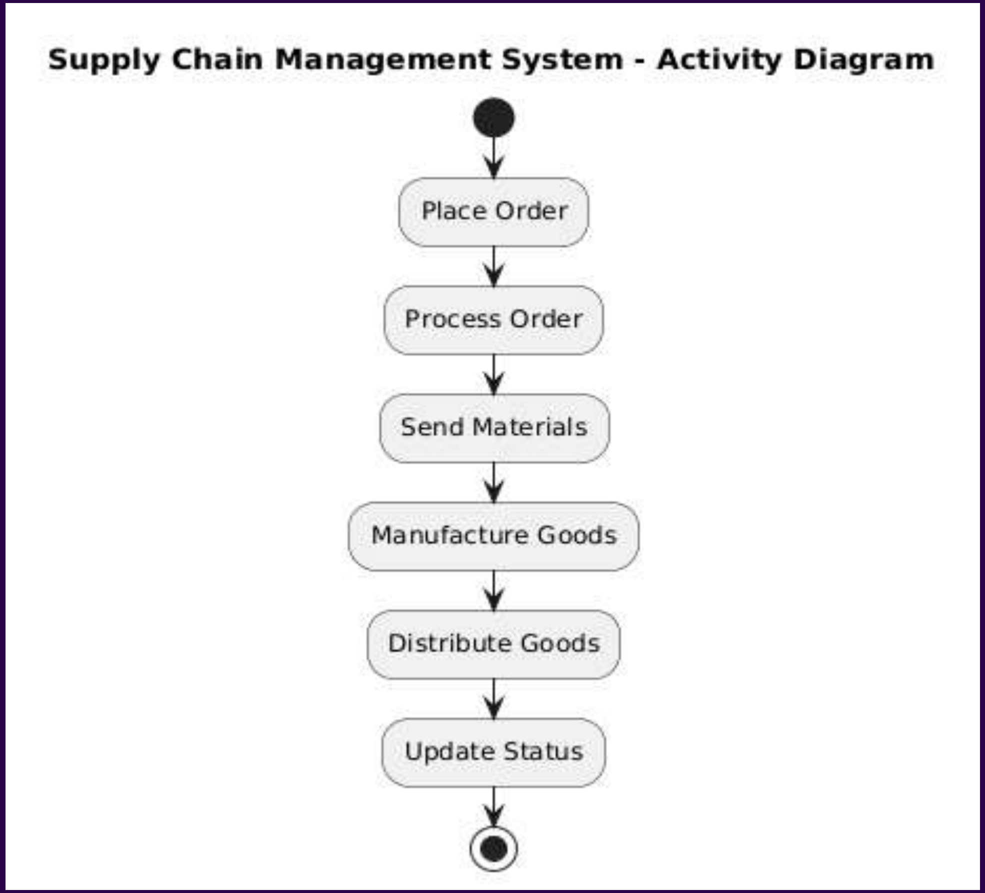
<u>**Development Phases**</u>

- Phase 1: Requirement Analysis and Design
- Phase 2: Backend Development with Uagents
- Phase 3: Frontend Development with React and Interactive Elements
- Phase 4: Integration and Testing

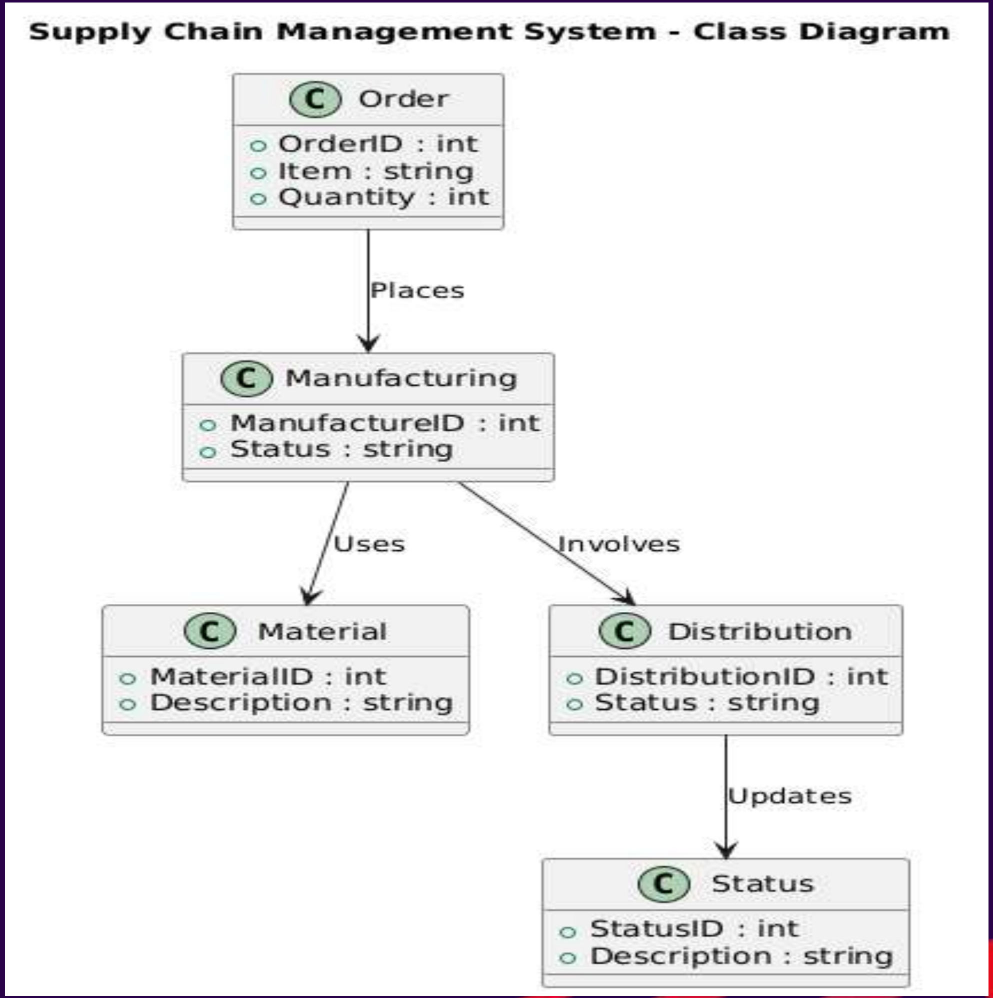
Integration: Ensured seamless communication between frontend and backend through well-defined API endpoints.

Testing: Comprehensive testing of all functionalities to ensure reliability and performance.

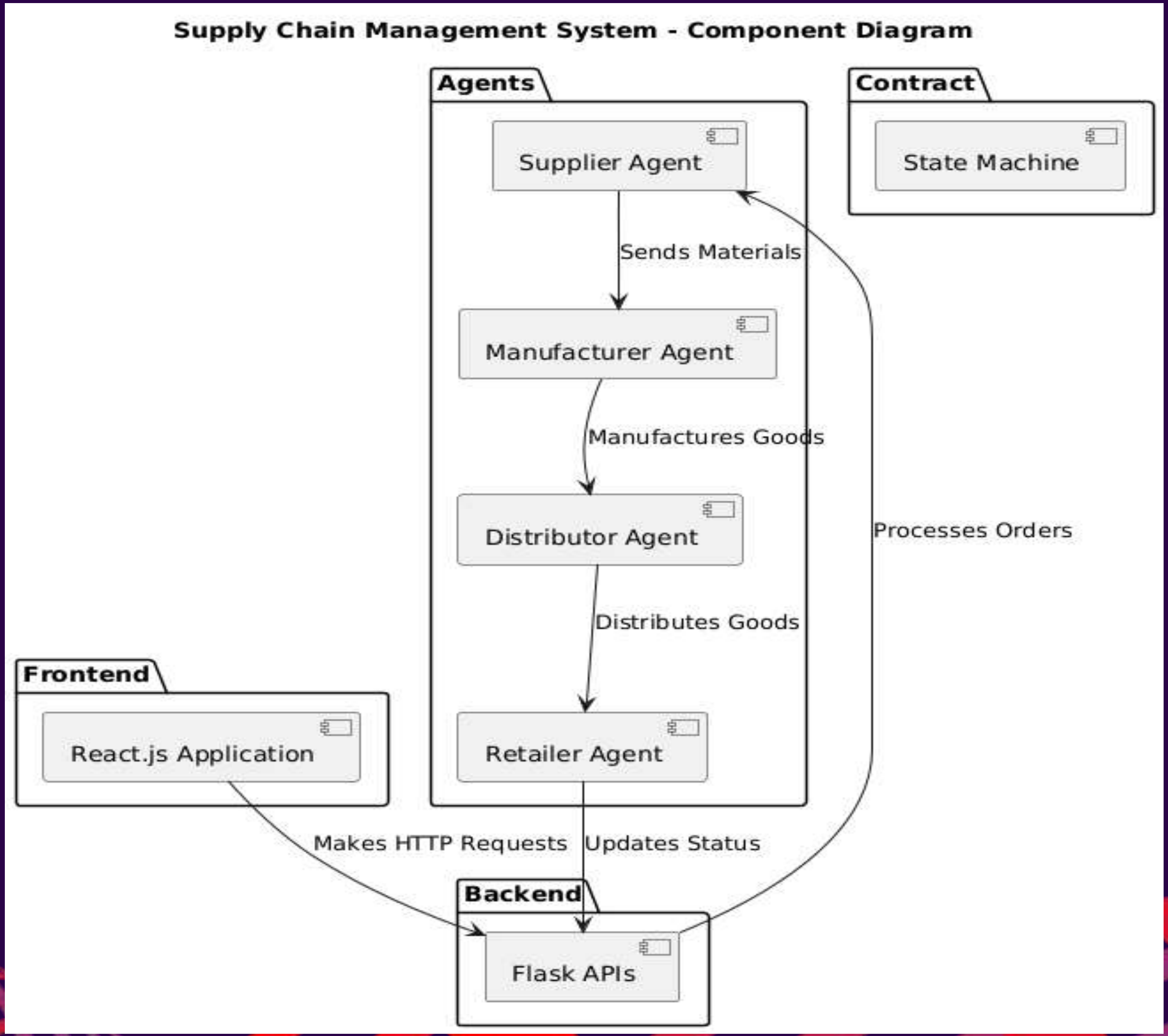
ACTIVITY DIAGRAM



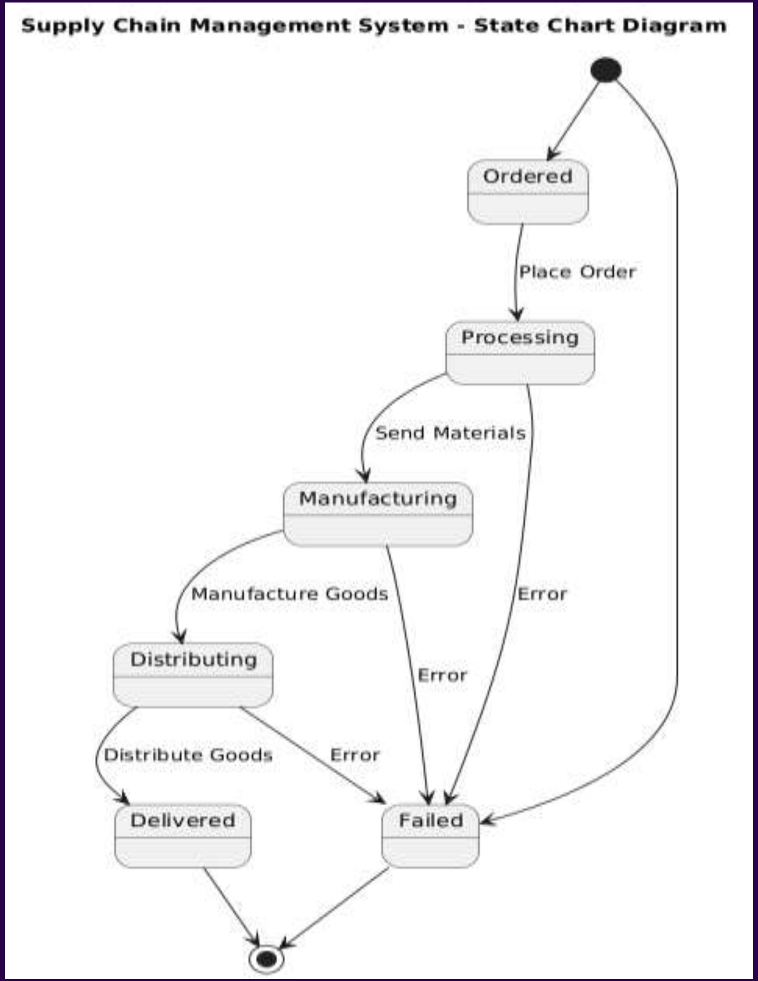
CLASS DIAGRAM

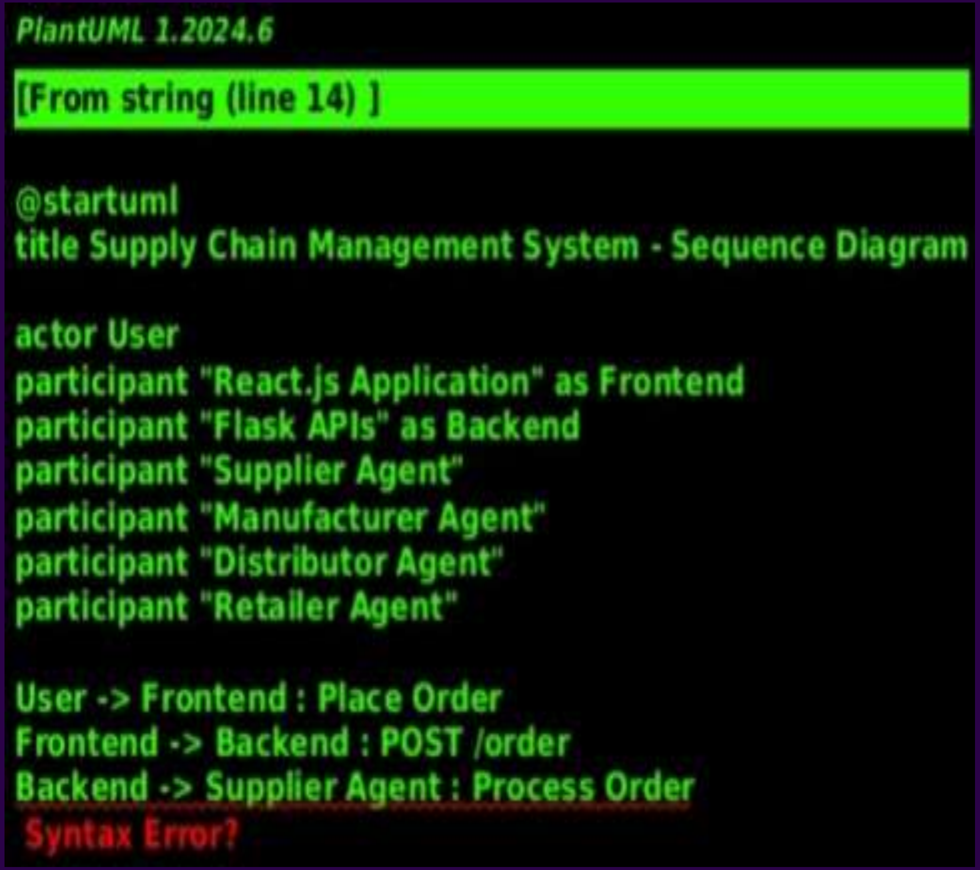


COMPONENT DIAGRAM



STATE TRANSITION DIAGRAM





```
PlantUML 1.2024.6
[From string (line 14) ]

@startuml
title Supply Chain Management System - Sequence Diagram

actor User
participant "React.js Application" as Frontend
participant "Flask APIs" as Backend
participant "Supplier Agent"
participant "Manufacturer Agent"
participant "Distributor Agent"
participant "Retailer Agent"

User -> Frontend : Place Order
Frontend -> Backend : POST /order
Backend -> Supplier Agent : Process Order
 Syntax Error?
```

FEATURES

ORDER MANAGEMENT

Users can place orders through the frontend interface. The order details are sent to the backend, processed, and stored. - Order history is displayed, showing item names, quantities, and statuses.

MATERIAL HANDLING

Automated processes for sending and receiving materials between agents, enhancing efficiency and reducing manual effort.

FEATURES

MANUFACTURING & DISTRIBUTION

Agents manage the production and distribution of goods, ensuring timely fulfillment and tracking.

REAL-TIME STATUS

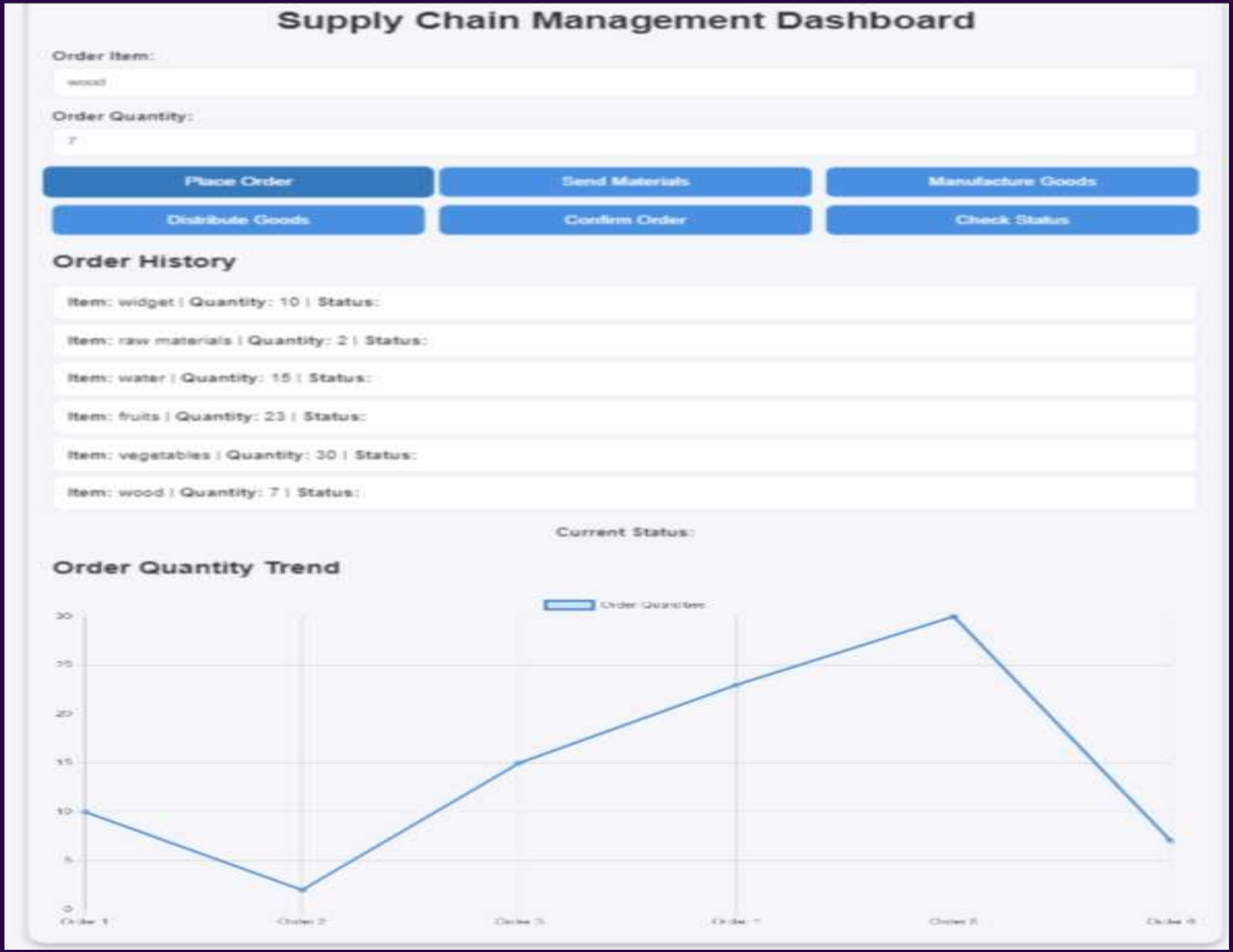
Provides real-time updates on order status, material handling, and other supply chain processes.

OUR COMPETITION GRAPHIC

CHALLENGES FACED	SOLUTIONS IMPLEMENTED
Difficulties in synchronizing frontend and backend components.	Resolved integration issues by refining API endpoints and frontend logic.
Ensuring robust error handling and logging for smooth operation.	Implemented comprehensive error handling and logging mechanisms to track and address issues effectively.

IMPLEMENTATION

https://fetchai-dapp.netlify.app/



Supply Chain Management Dashboard

Order Item:

wood

Order Quantity:

7

Order History

Item: widget | Quantity: 10 | Status:

Item: raw materials | Quantity: 2 | Status:

Item: water | Quantity: 15 | Status:

Item: fruits | Quantity: 23 | Status:

Item: vegetables | Quantity: 30 | Status:

Item: wood | Quantity: 7 | Status:

Current Status:

Order Quantity Trend

RESULTS & IMPACT

Key metrics & Impacts

Key metrics

Efficiency Improvement:

Reduced order processing time by [specific percentage or time], streamlining operations.

User Engagement:

Enhanced user interaction through dynamic and responsive UI elements.

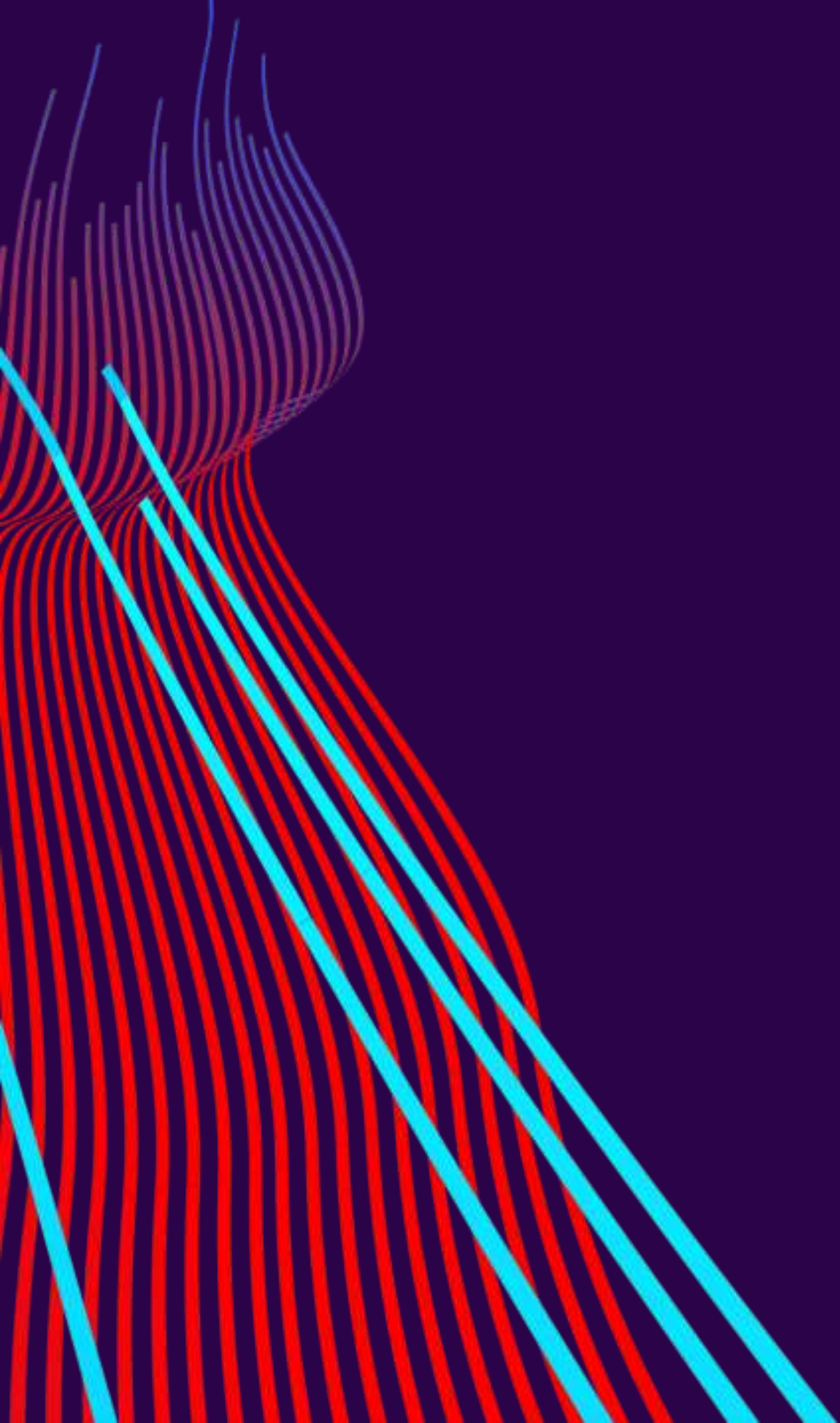
Impact

Operational Efficiency:

Streamlined supply chain processes with improved transparency and reduced delays.

User Feedback:

Positive responses from users and stakeholders regarding the system's functionality and design.



FUTURE WORK

ADVANCED ANALYTICS

Develop analytics and reporting features for deeper insights into supply chain performance.

Extended Integration

Integrate with additional supply chain partners and platforms for broader applicability.

System Scalability

Scale the system for larger and more complex supply chains.

Blockchain Integration

Explore blockchain technology for added security and decentralization.

CONCLUSION

Summary:

Recap of the project’s objectives, approach, and outcomes.
Emphasize key achievements and improvements.

Acknowledgments:

Thank collaborators and Fetch.ai for their support and the opportunity to present this project.

MARKET OPPORTUNITY OVERVIEW

VVS LAXMAN

GitHub:

https://github.com/Vvslaxman

M POOJITH CHOWDARY

GitHub:

https://github.com/Poojith-Chowdary

M BUDDU UDAY

GitHub:

https://github.com/Morampudi-Buddu-Uday

N MAHIDEEP

GitHub:

https://github.com/matrix-09

THANK YOU