Industry 4.0 Warehouse Management System

R25-062

Meet our team

Mayukha Palihena

IT21079672

Adilu Abeydeera

IT21822780

Sidath Tharana

IT21822094

Visura Suranjith

IT21318184

Dr. Samantha Rajapaksha

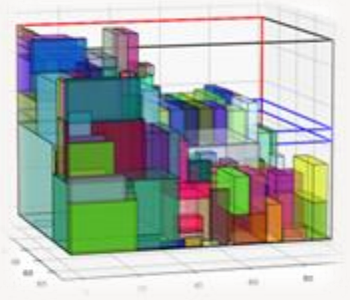
Supervisor

Dr. Dinuka Wijendra

Co-Supervisor

Introduction

- **Overview:**
  - Warehouses are vital components of modern supply chains. Optimizing warehouse operations leads to significant cost savings, improved efficiency, and better service delivery.
  - This research focuses on applying Industry 4.0 technologies to improve warehouse management, ensuring the efficient use of space, optimized order picking, better safety measures, and real-time monitoring of stock movements.
- **Challenges Addressed:**
  - Space utilization inefficiencies
  - Time-consuming order picking processes
  - Insufficient fire detection systems
  - Lack of real-time stock movement monitoring



Research Questions

- **Limited Fire Detection:** How can we provide real-time fire alerts near critical areas like racks, hindering rapid response ?

- **Inefficient Space Utilization:** Lack of intelligent systems for dynamically optimizing warehouse space leads to inefficiencies. How can we address that ?

- **Route Inefficiency:** How can we save time and energy by minimizing warehouse route problems ?

- **Stock Instabilities:** How to predict incidents which address sudden movements in stock, leading to undetected inventory issues ?

Main Objective

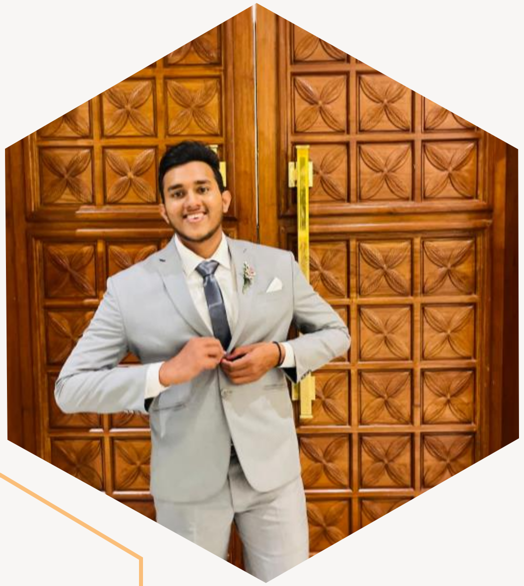
To develop an optimized warehouse management system that enhances space utilization, improves inventory turnover adaptability, streamlines year-end evaluations and mitigates risks associated with sensitive goods

Sub Objectives

- Maximize warehouse space utilization by leveraging algorithms to dynamically allocate inventory.
- Develop an optimized order picking and forklift path system that reduces time, enhances safety, and minimizes congestion.
- Implementing systems to detect fire hazards based on proximity to critical zones and trigger real-time alerts.
- Analyse and predict for abnormal stock movements, improving safety and operational efficiency.

Aspect 01

Optimizing the arrangement of products within the best location of a warehouse, to maximize space utilization.



Introduction

Why is product arrangement optimization important?

- Efficient use of warehouse space ensures cost savings.
- Proper arrangement reduces retrieval time and operational inefficiencies.
- Aligns with the overall objective of maximizing warehouse CBM utilization.

Problems

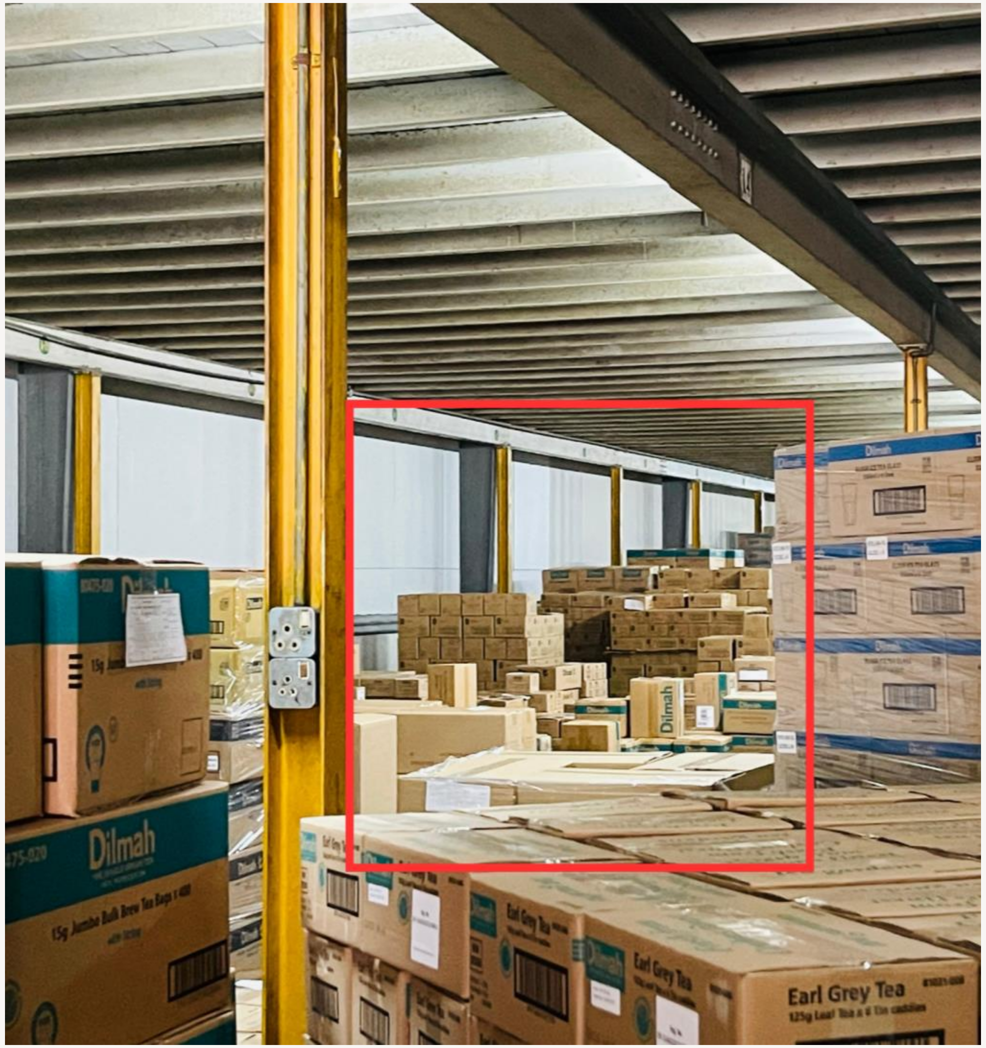
Disorganized layout, leaving gaps and unused areas leads to space wastage

Need for a systematic method to arrange products within predefined locations.

Problems

Difficulty in accommodating products while maintaining accessibility

Designed for horizontal storage, leaving significant vertical space underutilized.



Research Gap

	Research A [1]	Research B [2]	Research C [3]	Research D [4]	Proposed Solution
Product Arrangement in Location to Maximize CBM Utilization	✗	✗	✗	✗	✓
Real-Time Visualization of Product Arrangement (2D/3D)	✗	✗	✗	✓	✓
Algorithms for Product Arrangement	✓	✓	✗	✗	✓
Usability and Accessibility in Storing Products	✓	✗	✓	✗	✓

Novelty

Optimizing product arrangements within the best warehouse location, this research leverages CBM-specific strategies, real-time 2D visualization.

Container space optimization: - Leveraging maximum space available within a rack without wasting any space. Drawing the optimized space with allocated objects.

Objectives

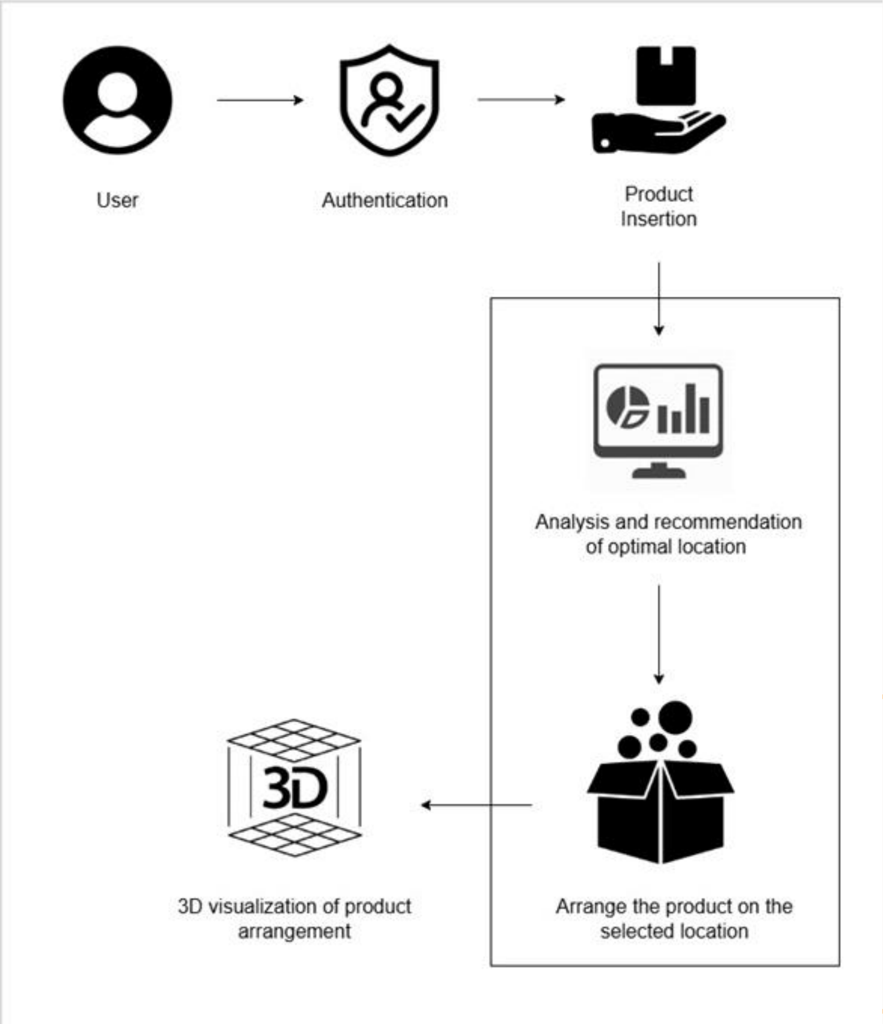
Main Objective

Optimizing the arrangement of products within the best location of a warehouse, to maximize space utilization.

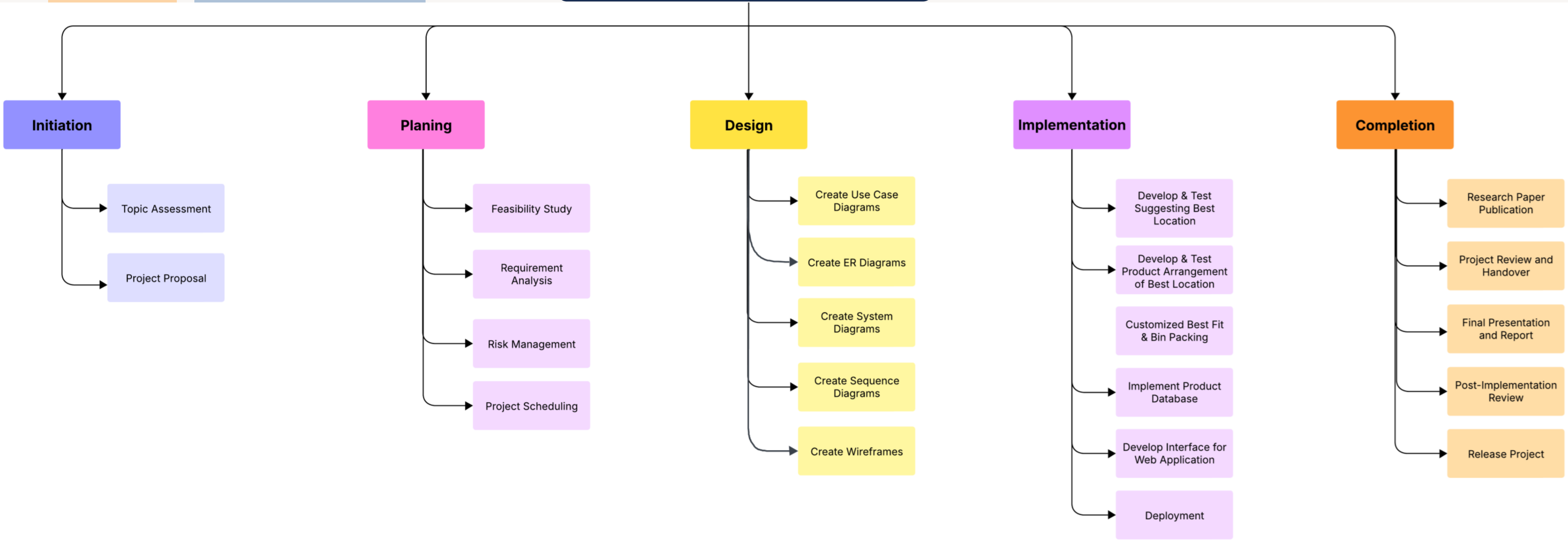
Sub Objectives

- Create predictive models based on CBM-specific data to optimize product storage arrangements dynamically
- Provide real-time 2D visual representations of optimized product arrangements
- Enhance Usability and Accessibility

Methodology

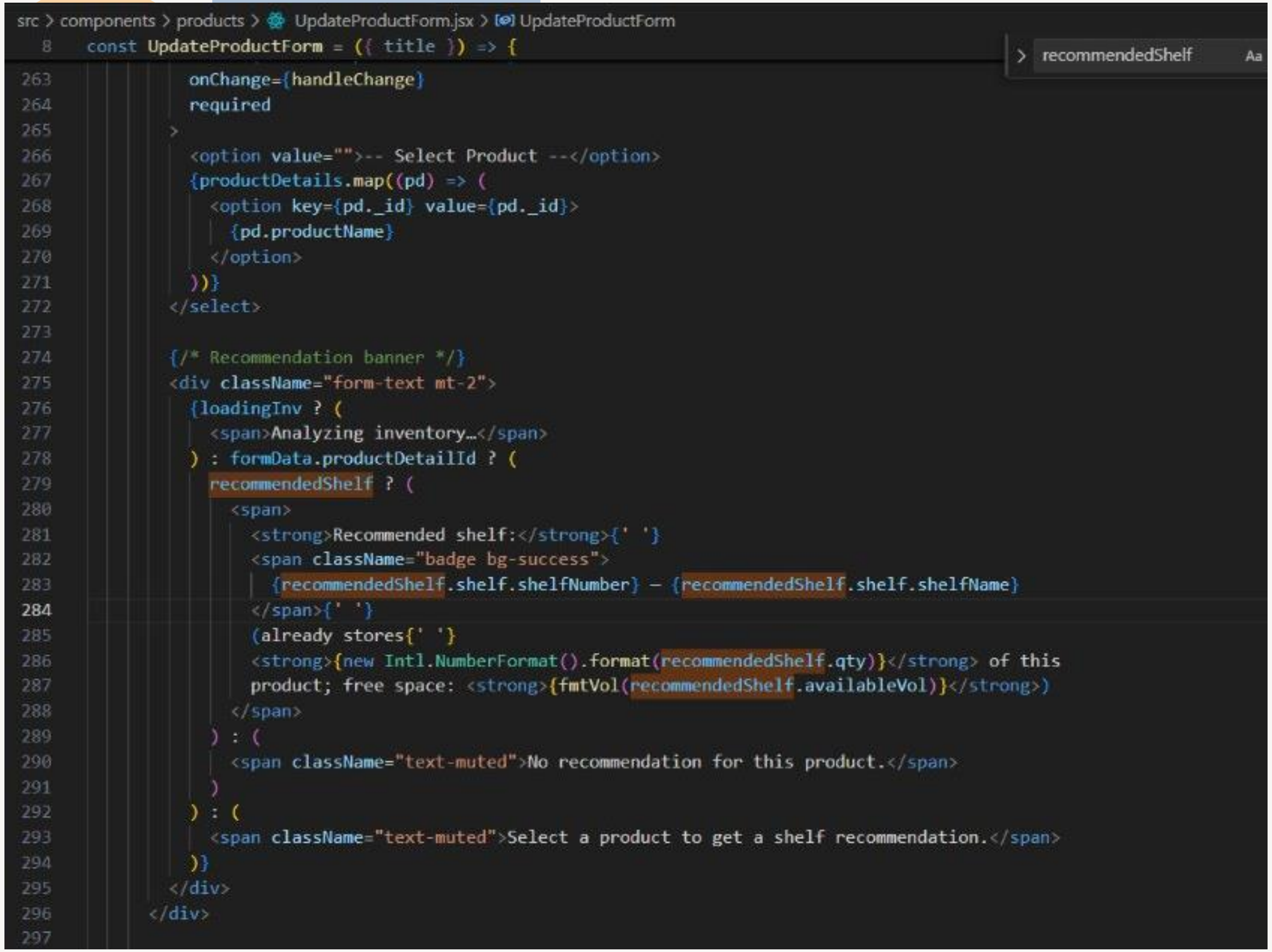

User
Authentication
Product Insertion
Analysis and recommendation of optimal location
Arrange the product on the selected location
3D visualization of product arrangement
3D

Work Breakdown Structure



- **Initiation**
  - Topic Assessment
  - Project Proposal
- **Planing**
  - Feasibility Study
  - Requirement Analysis
  - Risk Management
  - Project Scheduling
- **Design**
  - Create Use Case Diagrams
  - Create ER Diagrams
  - Create System Diagrams
  - Create Sequence Diagrams
  - Create Wireframes
- **Implementation**
  - Develop & Test Suggesting Best Location
  - Develop & Test Product Arrangement of Best Location
  - Customized Best Fit & Bin Packing
  - Implement Product Database
  - Develop Interface for Web Application
  - Deployment
- **Completion**
  - Research Paper Publication
  - Project Review and Handover
  - Final Presentation and Report
  - Post-Implementation Review
  - Release Project

Current Progress



src > components > products > UpdateProductForm.jsx > UpdateProductForm

```jsx
const UpdateProductForm = ({ title }) => {
        onChange={handleChange}
        required
      >
        <option value="">-- Select Product --</option>
        {productDetails.map((pd) => (
          <option key={pd._id} value={pd._id}>
            {pd.productName}
          </option>
        ))}
      </select>

      {/* Recommendation banner */}
      <div className="form-text mt-2">
        {loadingInv ? (
          <span>Analyzing inventory…</span>
        ) : formData.productDetailId ? (
          recommendedShelf ? (
            <span>
              <strong>Recommended shelf:</strong>{' '}
              <span className="badge bg-success">
                {recommendedShelf.shelf.shelfNumber} – {recommendedShelf.shelf.shelfName}
              </span>{' '}
              (already stores{' '}
              <strong>{new Intl.NumberFormat().format(recommendedShelf.qty)}</strong> of this
              product; free space: <strong>{fmtVol(recommendedShelf.availableVol)}</strong>)
            </span>
          ) : (
            <span className="text-muted">No recommendation for this product.</span>
          )
        ) : (
          <span className="text-muted">Select a product to get a shelf recommendation.</span>
        )}
      </div>
    </div>
```

- Shelf Best Suggestion

Recommends the shelf where the selected product already has the highest quantity (co-location).

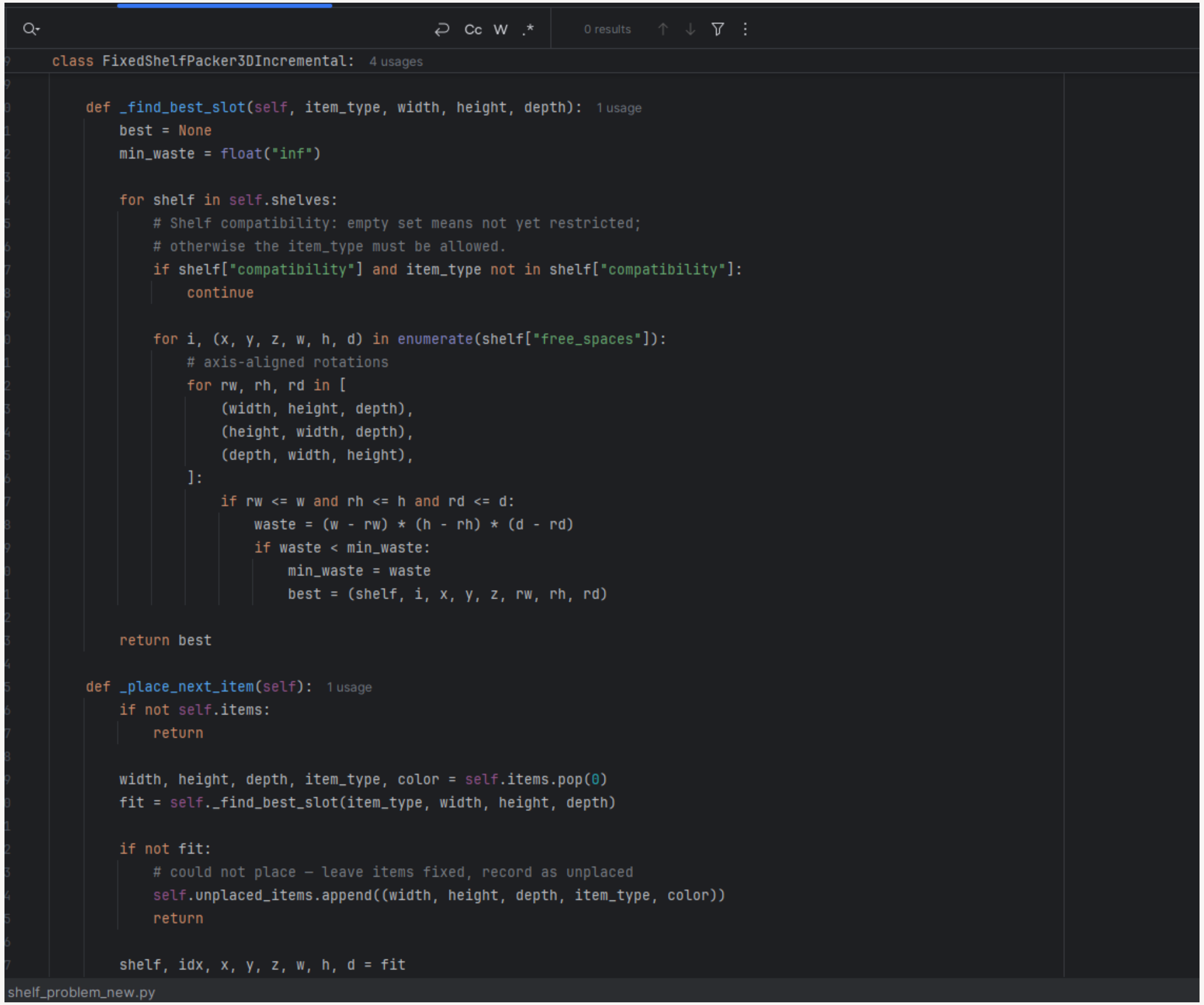
Shows available volume for the chosen shelf.

- Full Shelf Optimization

Packs all products for a shelf set using the incremental 3D packer (compatibility respected).

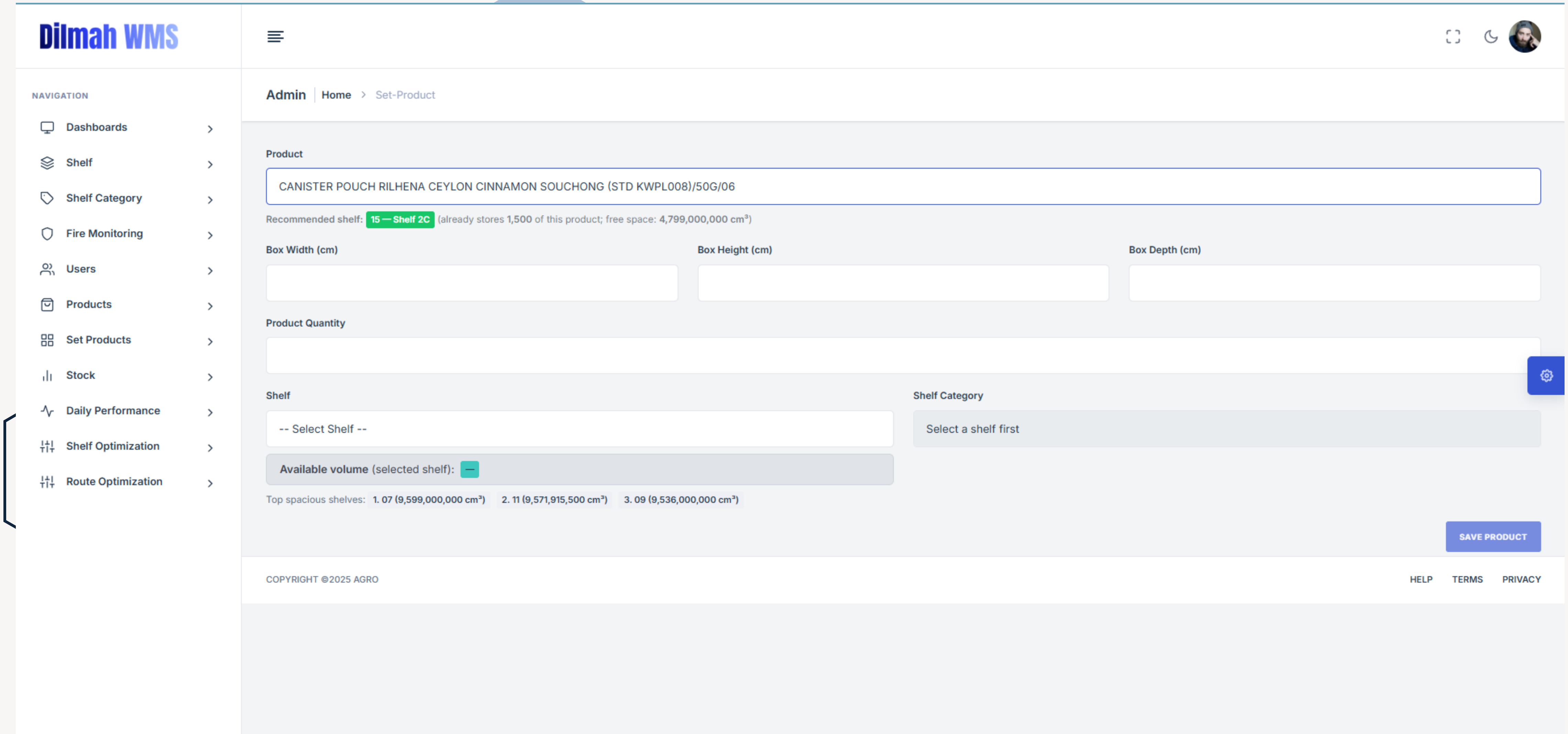
Returns placed_items + free_spaces + unplaced_items JSON.

Optional one-frame GIF/PNG visualization for reports.



```python
class FixedShelfPacker3DIncremental:

    def _find_best_slot(self, item_type, width, height, depth):
        best = None
        min_waste = float("inf")

        for shelf in self.shelves:
            # Shelf compatibility: empty set means not yet restricted;
            # otherwise the item_type must be allowed.
            if shelf["compatibility"] and item_type not in shelf["compatibility"]:
                continue

            for i, (x, y, z, w, h, d) in enumerate(shelf["free_spaces"]):
                # axis-aligned rotations
                for rw, rh, rd in [
                    (width, height, depth),
                    (height, width, depth),
                    (depth, width, height),
                ]:
                    if rw <= w and rh <= h and rd <= d:
                        waste = (w - rw) * (h - rh) * (d - rd)
                        if waste < min_waste:
                            min_waste = waste
                            best = (shelf, i, x, y, z, rw, rh, rd)

        return best

    def _place_next_item(self):
        if not self.items:
            return

        width, height, depth, item_type, color = self.items.pop(0)
        fit = self._find_best_slot(item_type, width, height, depth)

        if not fit:
            # could not place — leave items fixed, record as unplaced
            self.unplaced_items.append((width, height, depth, item_type, color))
            return

        shelf, idx, x, y, z, w, h, d = fit
```

Frontend Devlopment



Admin | Home > Set-Product

Product

CANISTER POUCH RILHENA CEYLON CINNAMON SOUCHONG (STD KWPL008)/50G/06

Recommended shelf: 15 — Shelf 2C (already stores 1,500 of this product; free space: 4,799,000,000 cm^3)

Box Width (cm)

Box Height (cm)

Box Depth (cm)

Product Quantity

Shelf

-- Select Shelf --

Shelf Category

Select a shelf first

Available volume (selected shelf): —

Top spacious shelves: 1. 07 (9,599,000,000 cm^3) 2. 11 (9,571,915,500 cm^3) 3. 09 (9,536,000,000 cm^3)

COPYRIGHT ©2025 AGRO

Future Improvements

- **Incremental Free-Space Allocation (finish)**

  Place only new items into **persisted `free_spaces`** without moving existing placements.

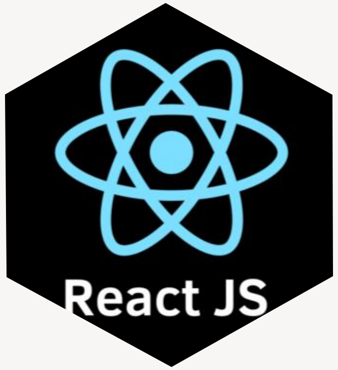
- **Full vs Incremental: comparison view**

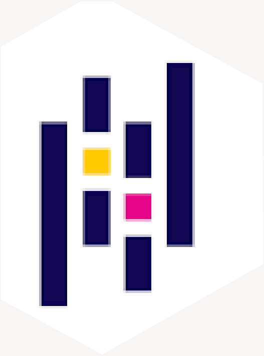
  Report **Utilization%**, **moved items/volume, unplaced**, and **runtime.**

- **Algorithm enhancements.**

- Plan to build a **worker-friendly interface** that directly maps package IDs or barcodes to their exact shelf location.

Technologies





References

[1] Bartholdi, J.J.; Hackman, S.T. "Warehouse and Distribution Science". 2019. Available online: https://www.warehouse-science.com/book/index.html (accessed on 30 May 2020)

[2] Ignacio Angulo, J.D; Jenny Fajardo, H.R "Optimization of Warehouse Layout for the Minimization of Operation Times". September 2021, Hybrid Artificial Intelligent Systems (pp.649-658)

[3] Bartholdi JJ, Hackman S (2008) "Allocating space in a forward pick area of a distribution center for small parts". IIE Trans 40(11):1046–1053. https://doi.org/10.1080/07408170802167662

[4] J. Gu, M. Goetschalckx, and L. F. McGinnis, "Research on warehouse operation: A comprehensive review," *European Journal of Operational Research*, vol. 203, no. 3, pp. 539–549, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S037722170900641X

Aspect 02

Order picking route optimization

Introduction

- When a Warehouse user has multiple items to get when fulfilling an order, it can be a time-consuming process, with efficient routes, we can reduce time and energy wasted by the worker.
- This main goal is to implement a pathfinding algorithm used to find the most efficient route for picking orders in a warehouse, considering time & effectiveness.

Research Problem

- How can we optimize worker productivity and efficiency by mapping the path the user needs to take on the Warehouse

Objectives

Main Objective

- Develop an efficient route optimization system for warehouse order pickers and forklifts, using a custom algorithm solution, to reduce time and improve operational efficiency.

Sub Objectives

- Implement the Algorithm which will compute the shortest most efficient path for the user.
- Adjusting the path according the congestions within the warehouse
- Map the routes of the warehouse, to the warehouse layout.

Research Gap

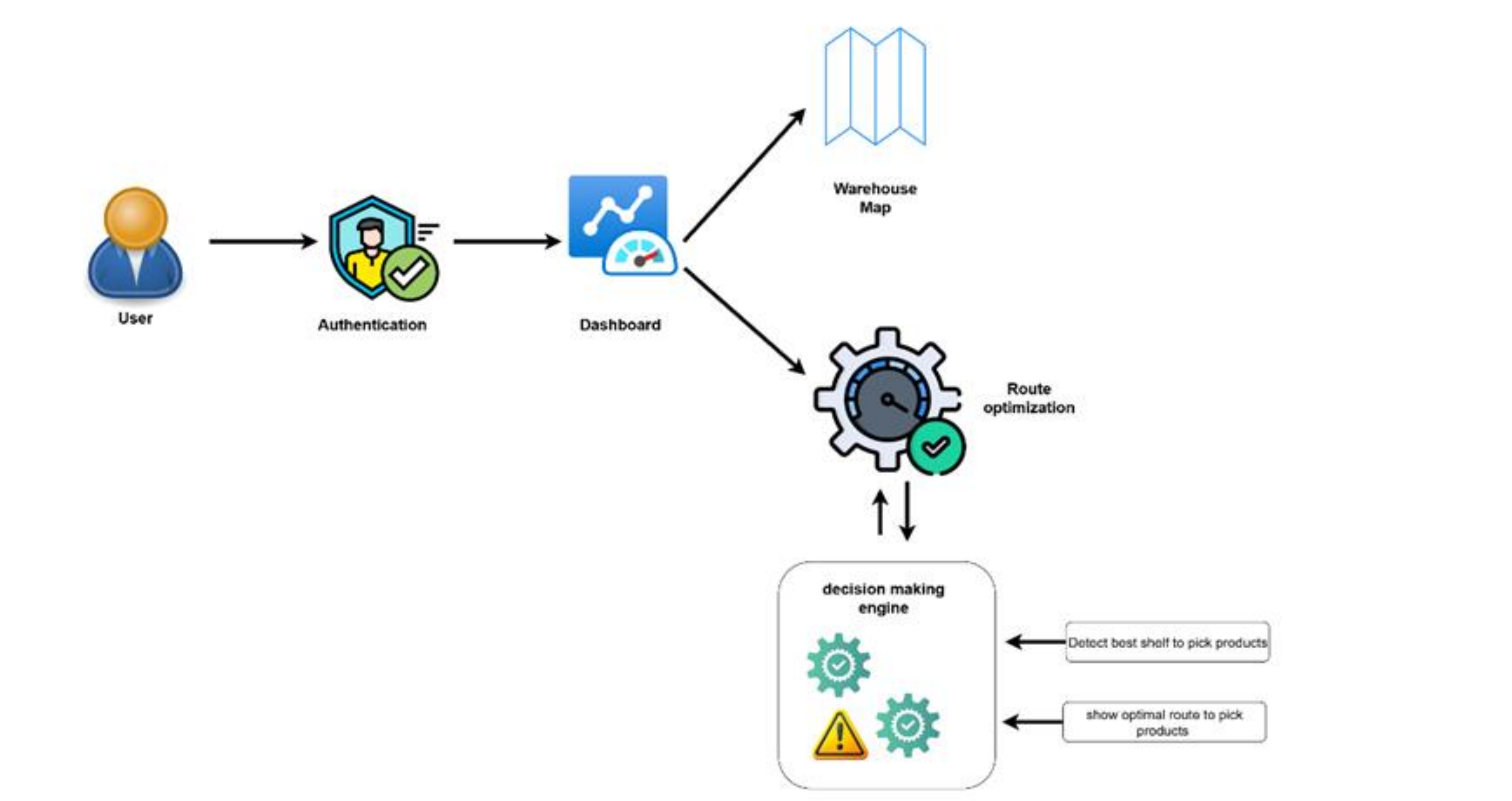
- **Path Mapping**
  - Mapping the path to fulfill the order, is a function that is not available in the existing system
- **Dynamic Path Adjustment:**
  - Current systems do not dynamically adjust to changes in the warehouse environment, such as congestion or maintenance zones.

Novelty

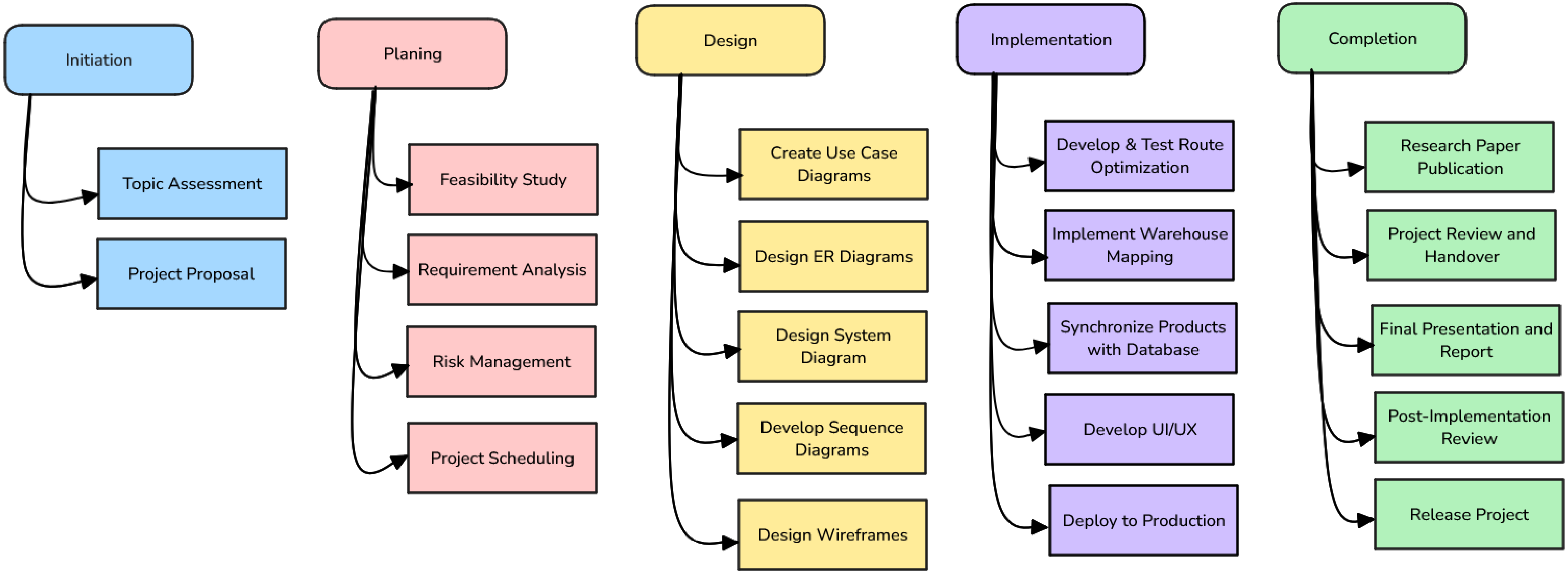
Novelty of this system lies in its dynamic optimization of warehouse navigation. Key novel aspects include:

- Custom Path Finding Algorithm
- Collision Avoidance
- Maintenance Zone Integration

Methodology



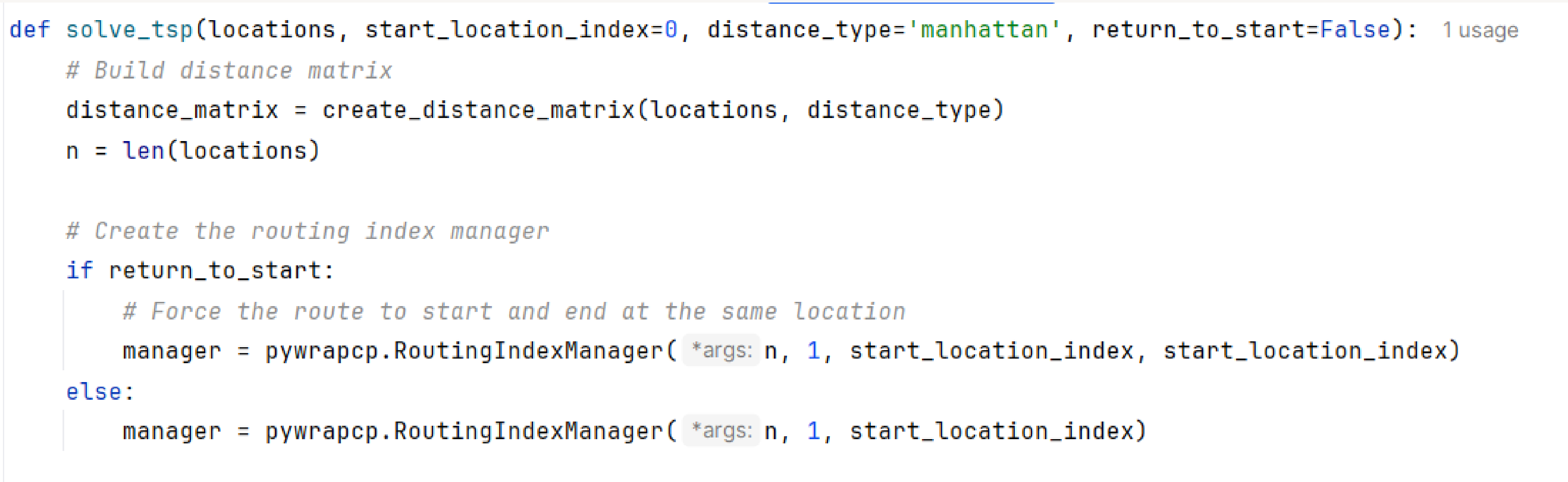
Work Breakdown Structure



Current Progress

- **Generate Optimal Order Picking Route**
- **Avoid Congestions/Stationary Workers within Route**

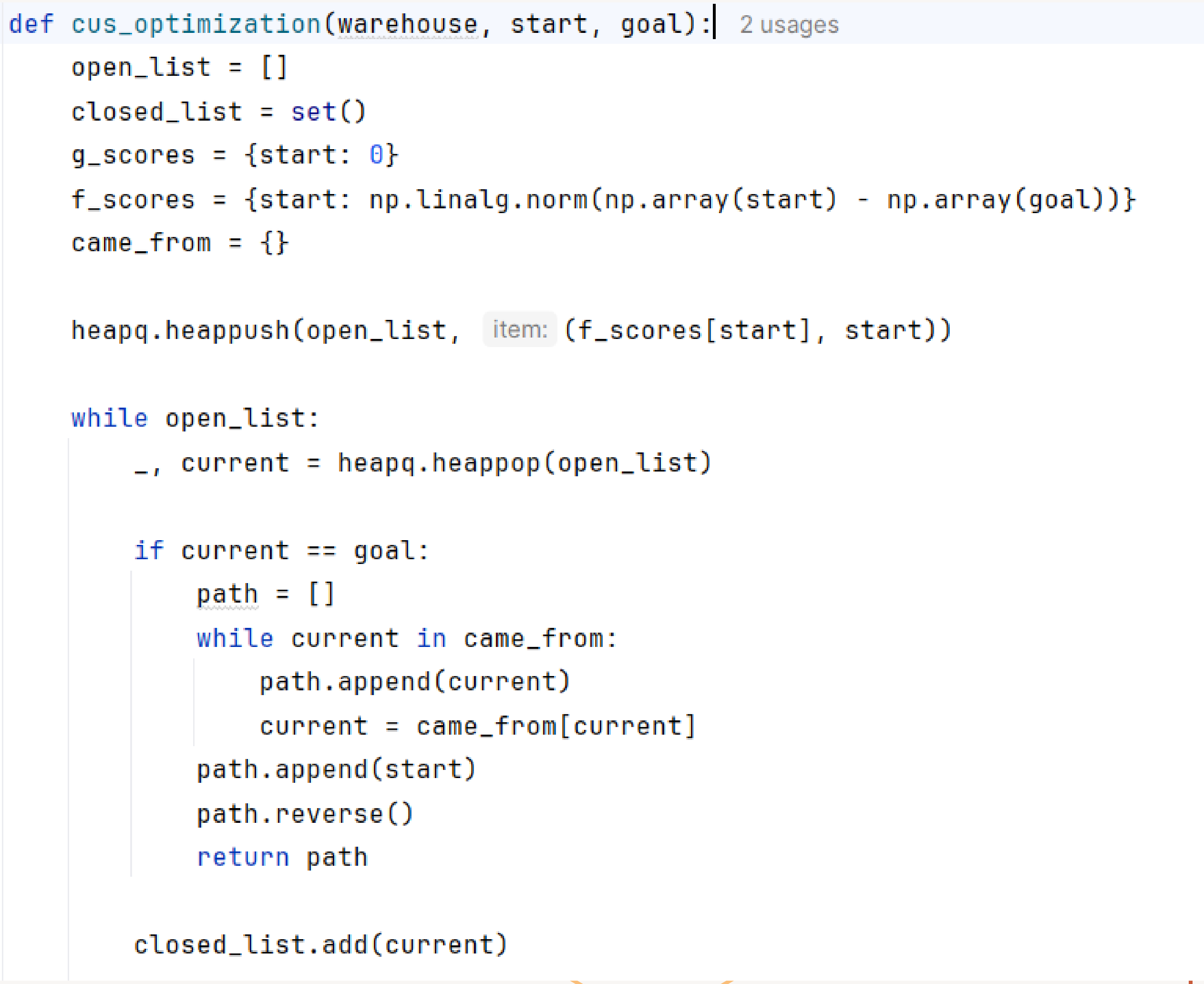
Current Progress



```
def solve_tsp(locations, start_location_index=0, distance_type='manhattan', return_to_start=False):  1 usage
    # Build distance matrix
    distance_matrix = create_distance_matrix(locations, distance_type)
    n = len(locations)

    # Create the routing index manager
    if return_to_start:
        # Force the route to start and end at the same location
        manager = pywrapcp.RoutingIndexManager( *args: n, 1, start_location_index, start_location_index)
    else:
        manager = pywrapcp.RoutingIndexManager( *args: n, 1, start_location_index)
```

- TSP Route Sequencing

Current Progress

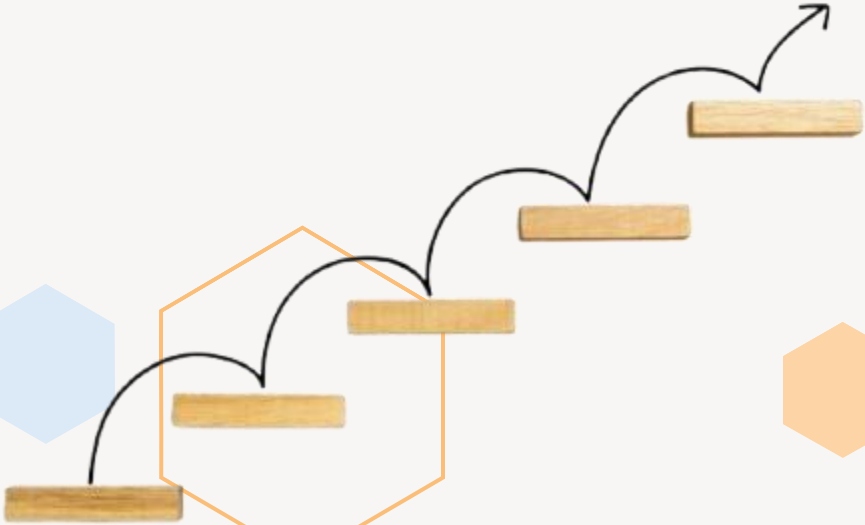


```
def cus_optimization(warehouse, start, goal):  2 usages
    open_list = []
    closed_list = set()
    g_scores = {start: 0}
    f_scores = {start: np.linalg.norm(np.array(start) - np.array(goal))}
    came_from = {}

    heapq.heappush(open_list,  item: (f_scores[start], start))

    while open_list:
        _, current = heapq.heappop(open_list)

        if current == goal:
            path = []
            while current in came_from:
                path.append(current)
                current = came_from[current]
            path.append(start)
            path.reverse()
            return path

        closed_list.add(current)
```

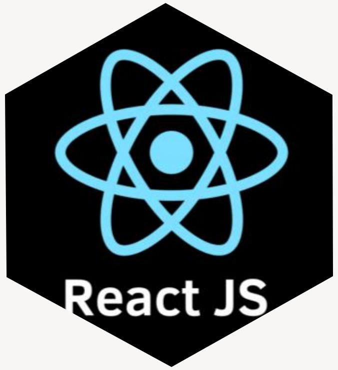
- Grid-Based Pathfinding

Future Improvements

- Improved UI/UX for the mapped route.
- Route Mapping Glitch fixes.



Technologies



React JS

AWS Lambda

Express JS

Google OR

Python

MongoDB

References

- Li, X., et al. (2023). "Optimized Pathfinding in Smart Warehouses." *International Journal of Logistics Management*.
- Y. Zhang, L. Zhang, and W. Li, "Real-time Dynamic Path Planning in Automated Warehouses," *Journal of Warehouse Technology*, 2021.
- "GPS Tracking in Warehouse Management: Real-time Positioning and Path Optimization," *Journal of Supply Chain Management*, 2022.

Aspect 03

Vision-Based Fire Detection and Prevention System for Warehouse Safety Using Real-Time Camera Surveillance

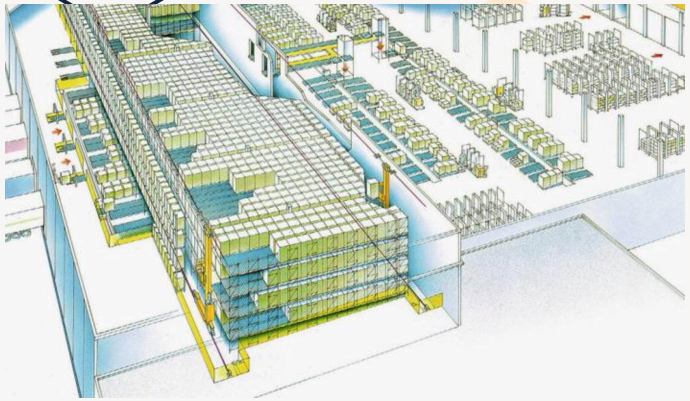
Tharana P A S - IT21822094

Introduction

- Warehouses face high fire risks that traditional systems often detect too late. This project introduces a real-time, camera-only solution using AI and CCTV to detect fires, assess shelf proximity, classify fire size, and predict spread. Integrated IoT components can trigger fire alarms or suppression systems in the predicted spreading direction, making the approach smart, scalable, and cost-effective for warehouse safety.

Research Problem

- Most fire detection systems are slow and reactive—they only respond after the fire has started growing.
- They don’t show where the fire is or how close it is to important assets like shelves.
- They don’t estimate fire size, which is critical for response planning.
- They cannot predict fire spread, leaving staff unaware of how fast or where it may move.
- No current system provides a complete, camera-only solution for fire detection, risk analysis, and prediction in warehouse environments.
- **When fire is spreading, connected fire alarms must be triggered automatically to ensure rapid staff evacuation and activation of suppression protocols.**



Objectives

Main Objective

- To develop a smart, real-time fire monitoring system using AI and existing warehouse CCTV cameras, providing early alerts and fire risk analysis.

Sub Objectives

•Detect fire (flames or smoke) from video feeds

•Identify shelves or objects near the fire

•Measure the size of the fire (small, medium, large)

•Predict the direction the fire might spread

•Display all information clearly on a user dashboard

•Trigger fire alarms automatically when a fire risk is detected to ensure timely alerts and evacuation

Research Gap

Gap	Description
No full camera-based system	Cameras alone don’t detect fire, size, spread, and nearby risks.
Ignores fire surroundings	Fire environment (shelves, items) is not analyzed.
No fire spread prediction	Fire movement and paths are not predicted.
High hardware cost	Extra sensors/IoT devices increase setup and maintenance.
Fire size not linked to alerts	Fire size rarely affects emergency actions or warnings.
Not indoor-ready	Spread models suit outdoors, not complex indoor layouts.
No all-in-one system	Detection, monitoring, and alerts are separate tools.
No alarm integration by fire direction	Fire spread predictions don’t trigger directional alarms.

Novelty

The proposed system uses **existing surveillance cameras** with **no extra hardware needed**. It can **detect fire, measure its size, and identify nearby shelf risks**. By **predicting fire spread**, the system **supports safer evacuation and faster response**. It is **connected to fire alarms**, allowing alerts to trigger based on the **direction of fire spread**. The solution is **cost-effective, easy to install, and scalable** for different warehouse layouts. This makes it particularly **effective in complex indoor environments** and **important for Sri Lanka**, where such technology is currently not in use.

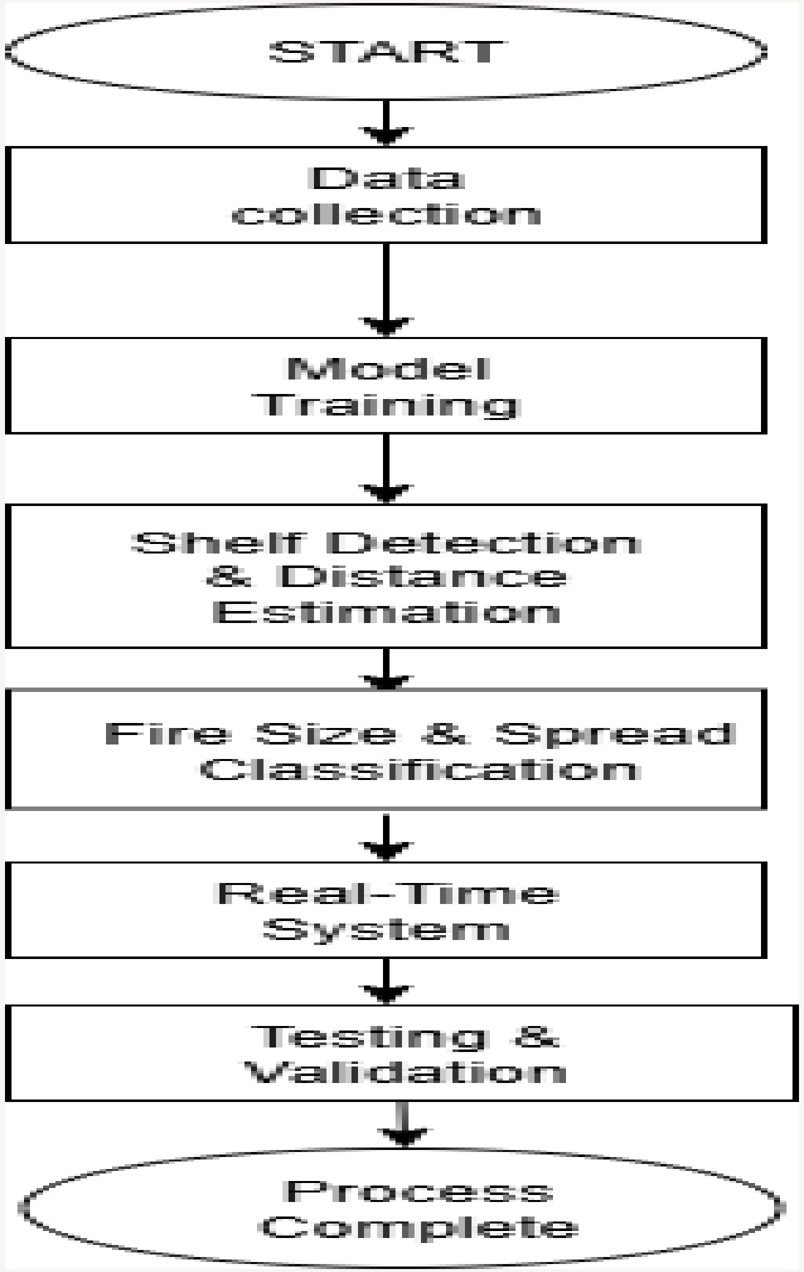
Methodology

Use AI models like YOLO and CNN to detect fire and shelves

Estimate fire size with image processing

Predict spread using video sequences

Combine all features into one real-time dashboard for monitoring



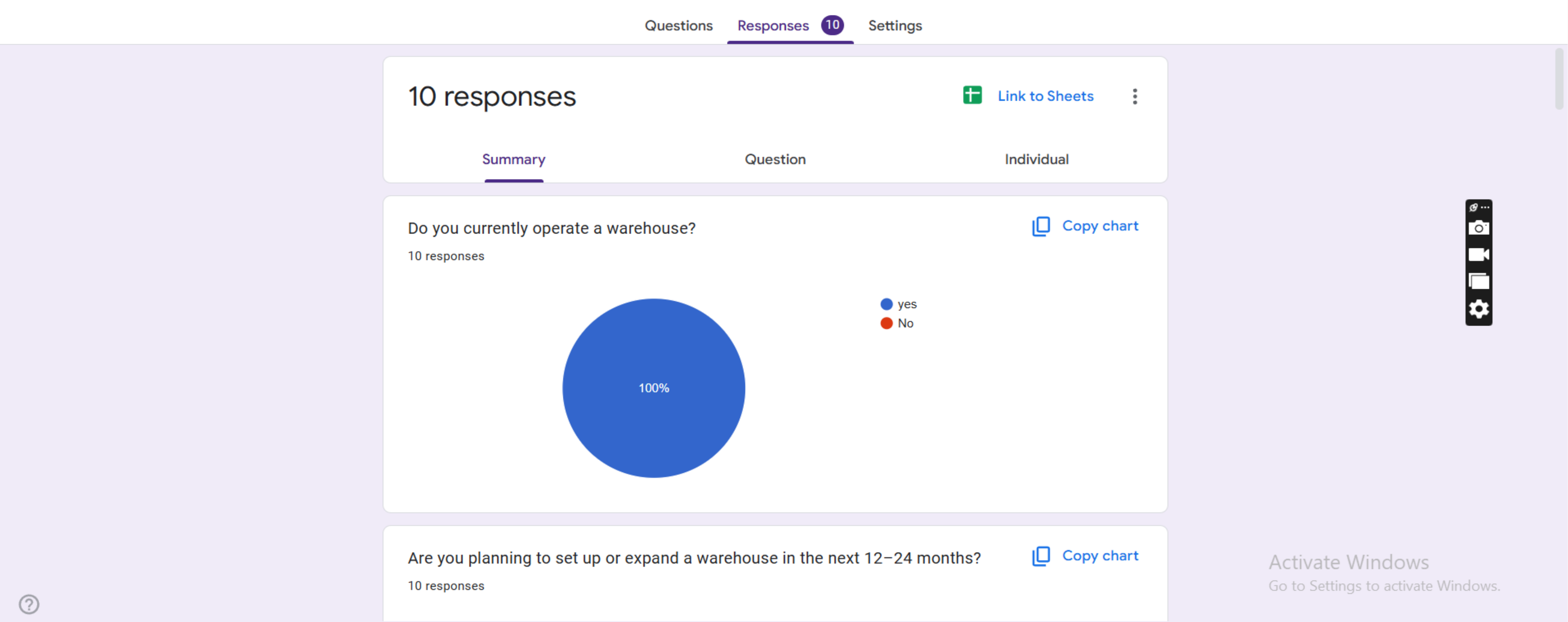
Current Progress

most existing systems are expensive and don't give enough information.

From a survey of 10 warehouse owners, we learned they need an affordable, smart solution.

That's why we developed a CCTV-based fire monitoring system that detects fire, estimates shelf distance, classifies fire size, and predicts fire spread in real-time using deep learning."

Google Form Results



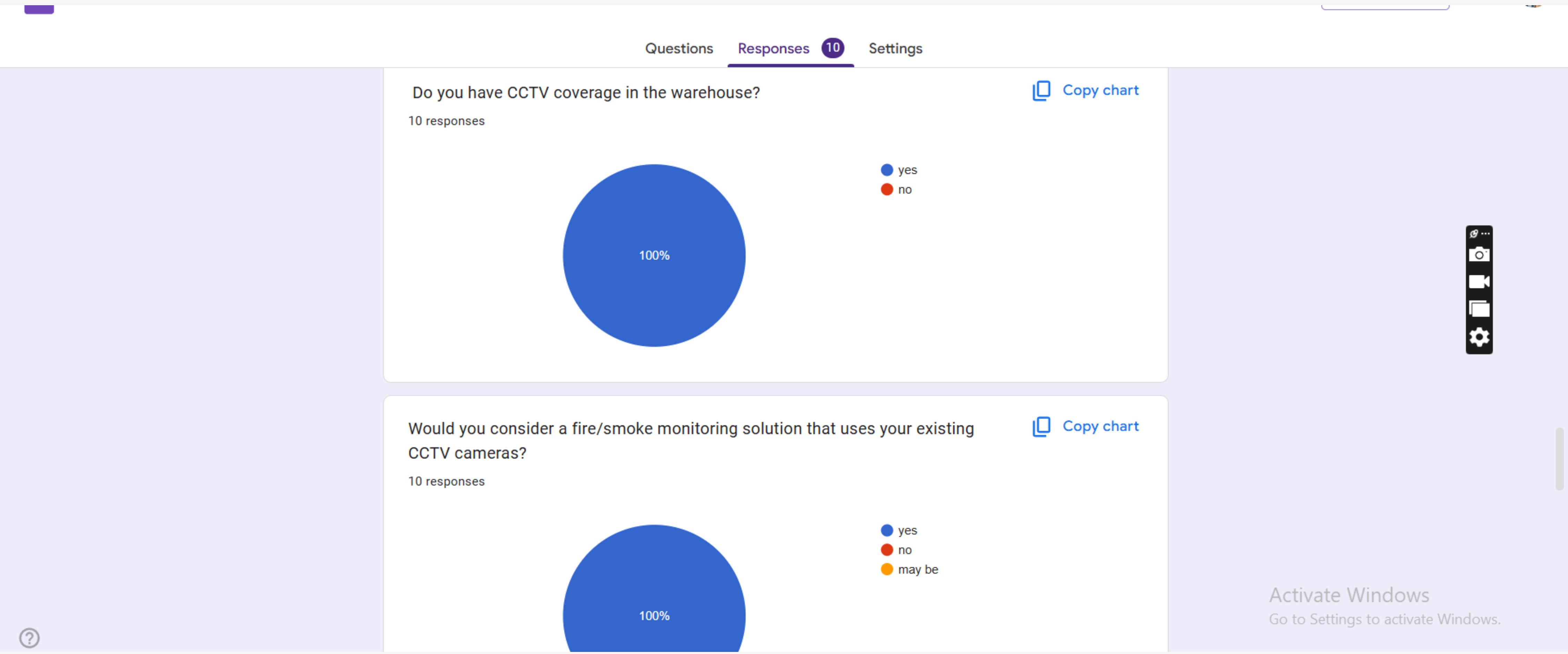
10 responses

Do you currently operate a warehouse?

10 responses

Are you planning to set up or expand a warehouse in the next 12–24 months?

10 responses

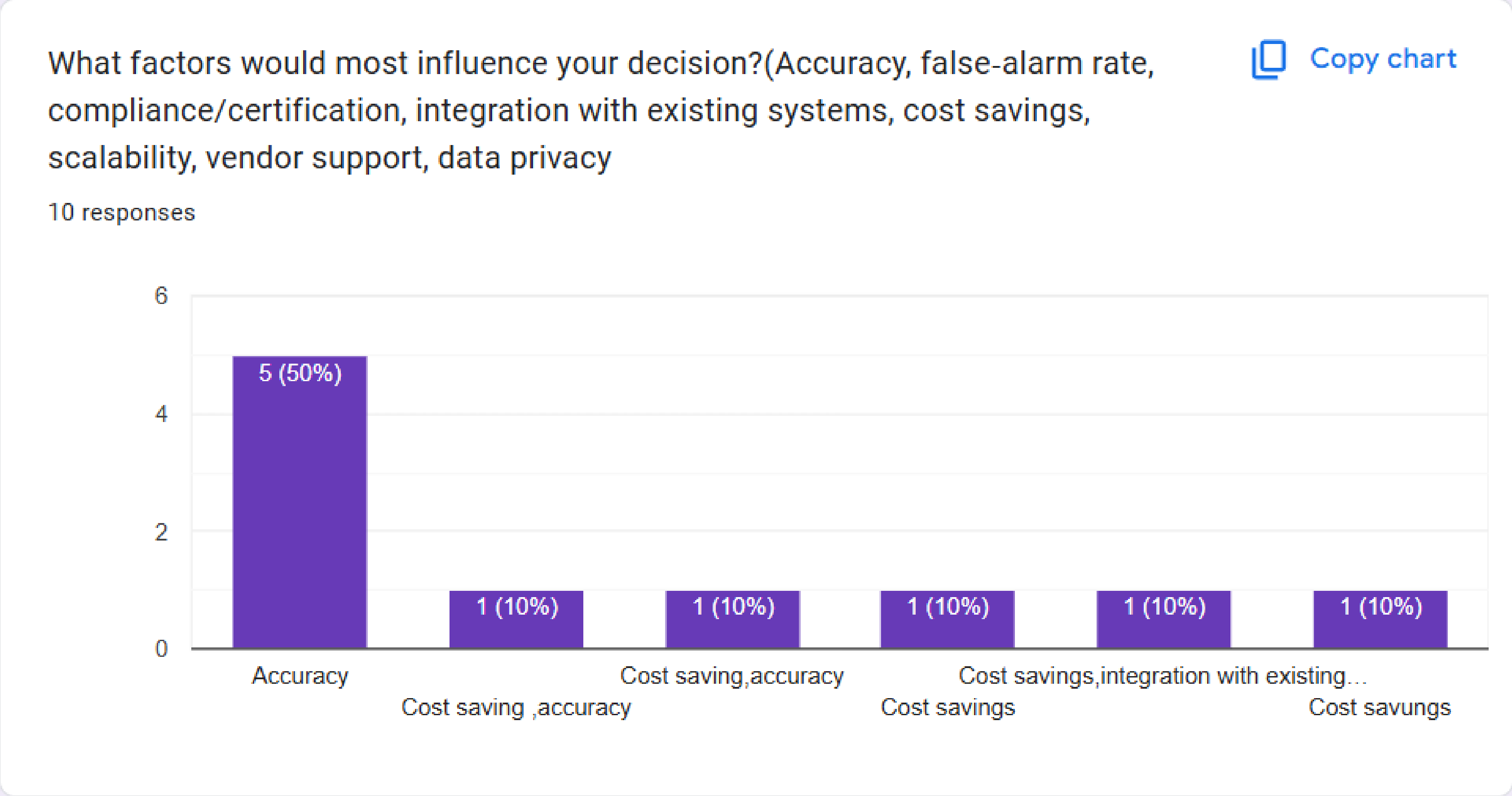


Do you have CCTV coverage in the warehouse?

10 responses

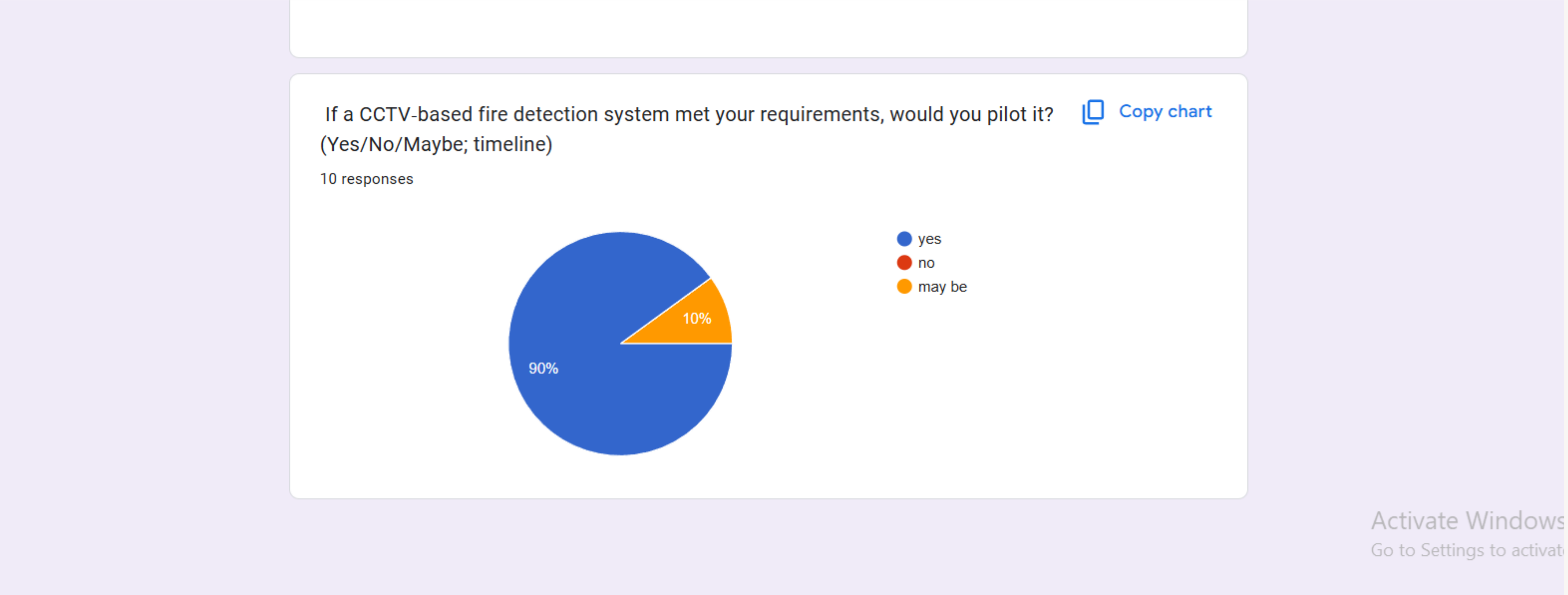
Would you consider a fire/smoke monitoring solution that uses your existing CCTV cameras?

10 responses



What factors would most influence your decision?(Accuracy, false-alarm rate, compliance/certification, integration with existing systems, cost savings, scalability, vendor support, data privacy

10 responses



If a CCTV-based fire detection system met your requirements, would you pilot it? (Yes/No/Maybe; timeline)

10 responses

Now you can see they are fond to system like that ,with existing CCTV and accuracy ,cost saving method

Relevant Laws / Regulations in Sri Lanka

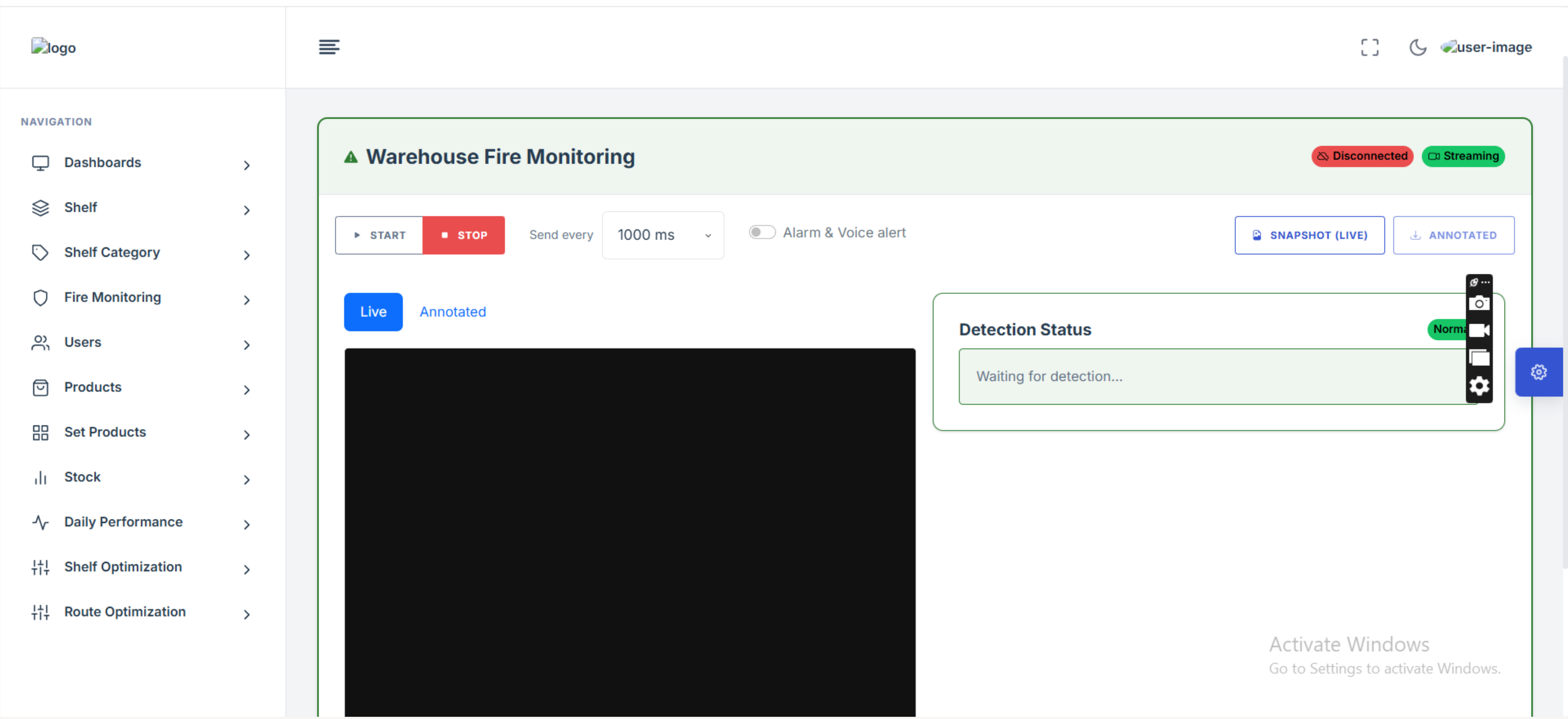
Factories Ordinance Section 39-42A: For “factories” (which may include large industrial warehouses), the Ordinance requires “effective provision for giving warning in case of fire, which shall be clearly audible throughout the building.

LawNet +1 Also requires fire-fighting appliances, and ability for escape,

LawNet +1 Local Authority Building / Fire Safety Certificate Requirements Municipal / city councils include fire exit signs,

manual fire alarm or automatic alarm required to get a license to approval. from Municipal Council

Before input



Warehouse Fire Monitoring

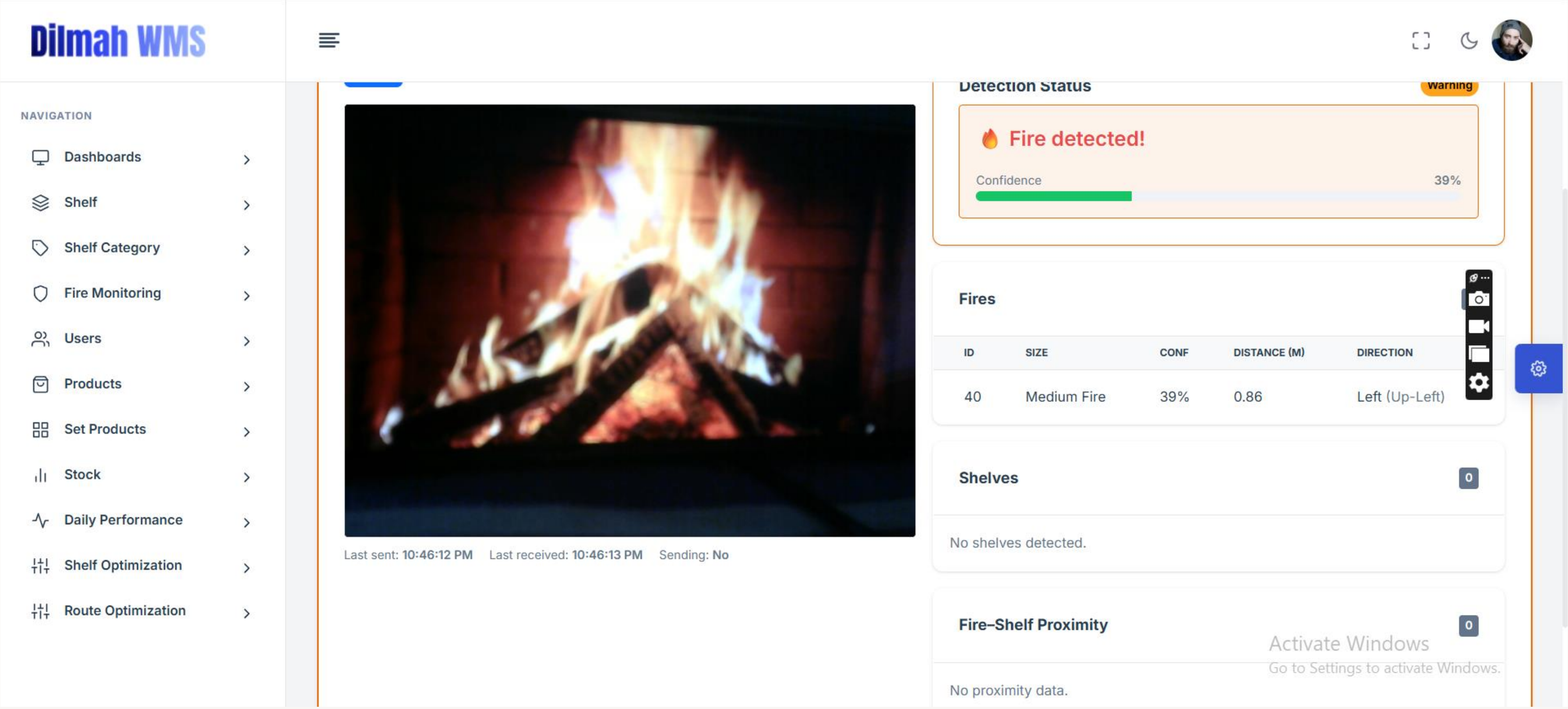
Send every 1000 ms

○ Alarm & Voice alert

Detection Status

Waiting for detection...

After input



Last sent: 10:46:12 PM Last received: 10:46:13 PM Sending: No

Detection Status Warning

🔥 Fire detected!

Confidence 39%

Fires

ID	SIZE	CONF	DISTANCE (M)	DIRECTION
40	Medium Fire	39%	0.86	Left (Up-Left)

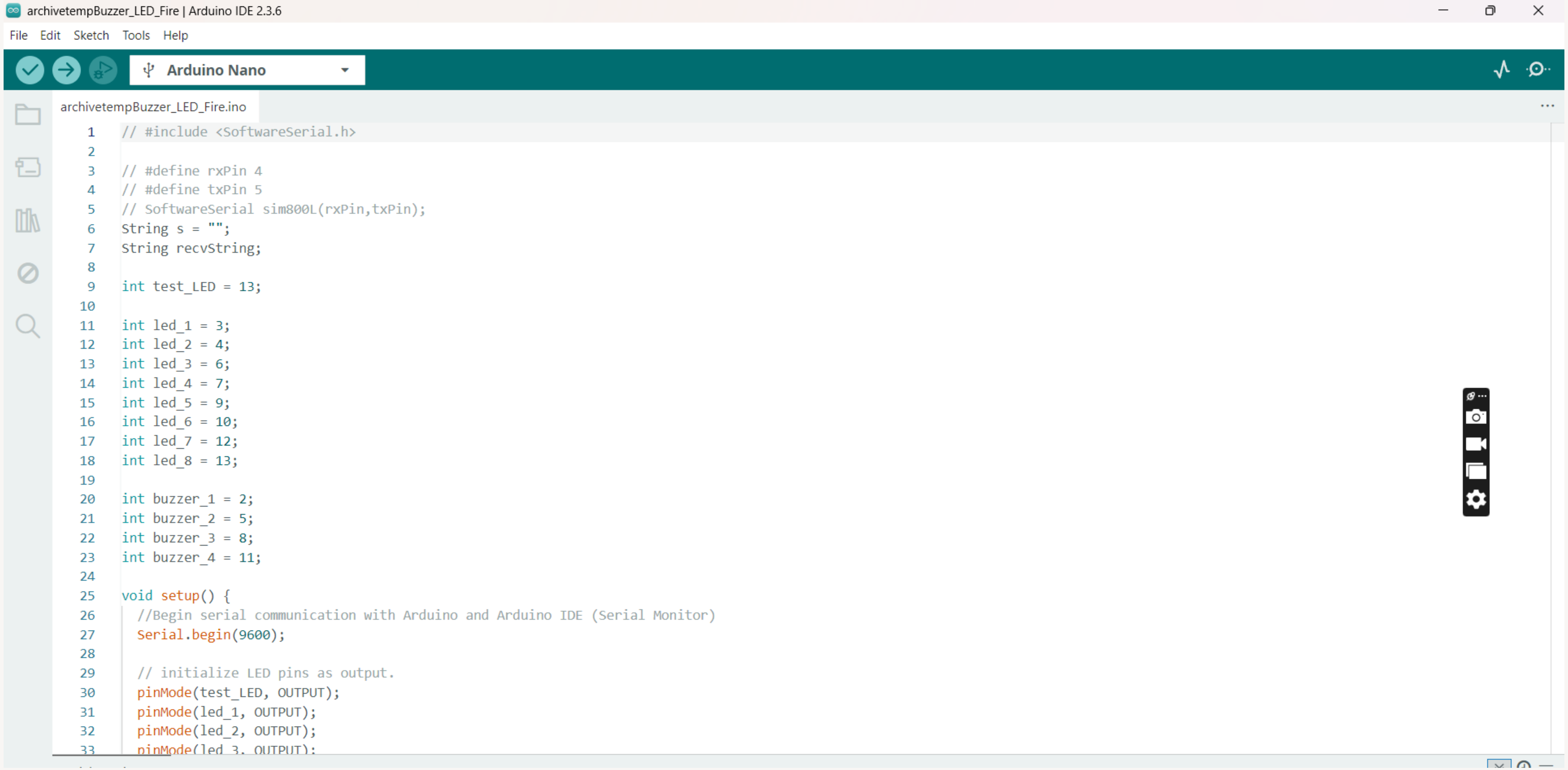
Shelves 0

No shelves detected.

Fire–Shelf Proximity 0

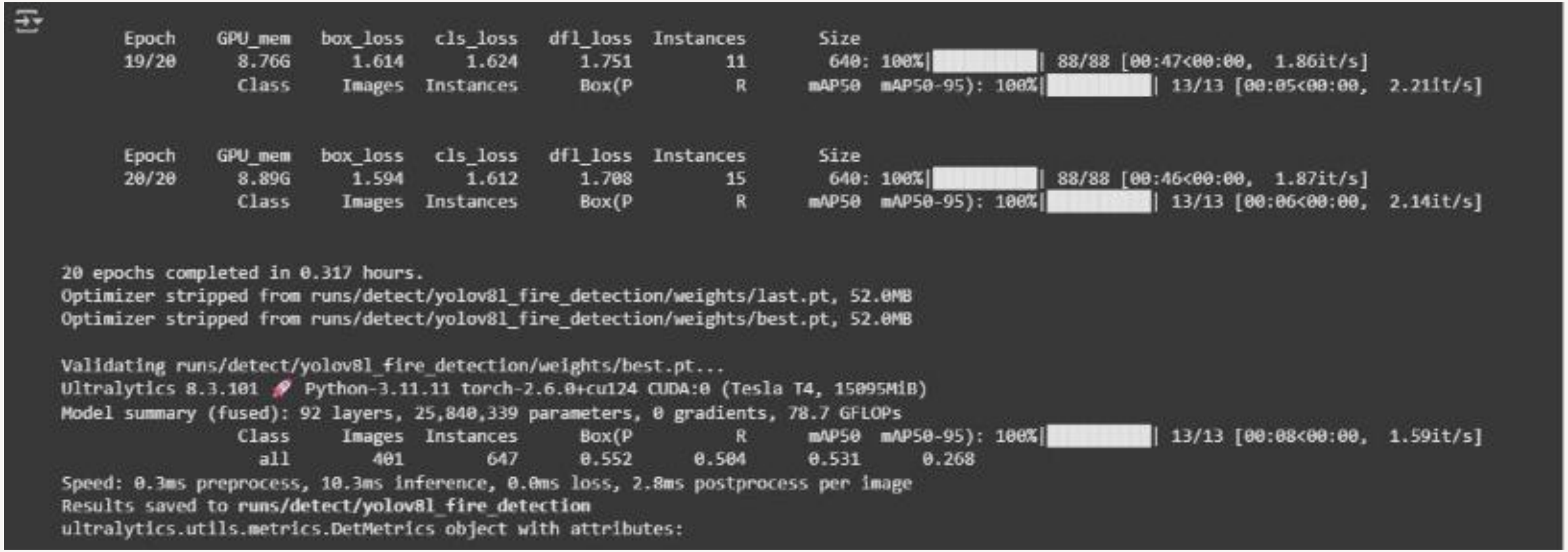
No proximity data.

Arduino code


archivetempBuzzer_LED_Fire | Arduino IDE 2.3.6
File
Edit
Sketch
Tools
Help
Arduino Nano
archivetempBuzzer_LED_Fire.ino

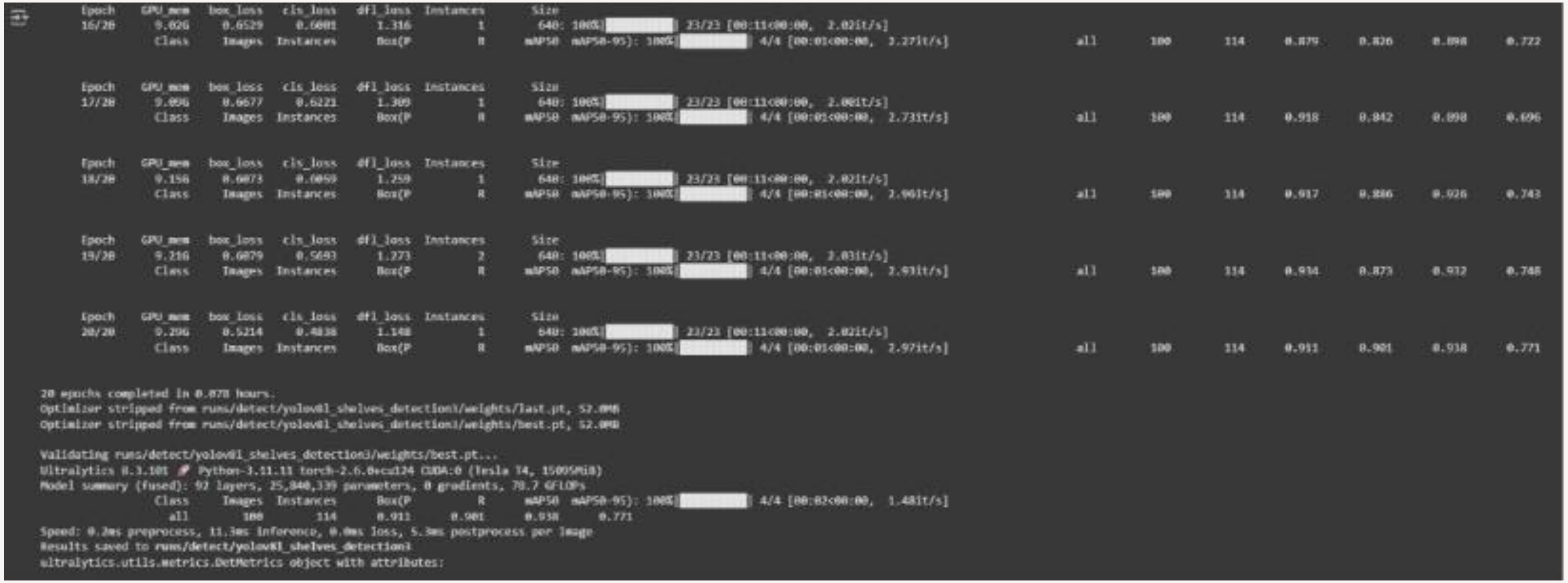
```
// #include <SoftwareSerial.h>

// #define rxPin 4
// #define txPin 5
// SoftwareSerial sim800L(rxPin,txPin);
String s = "";
String recvString;

int test_LED = 13;

int led_1 = 3;
int led_2 = 4;
int led_3 = 6;
int led_4 = 7;
int led_5 = 9;
int led_6 = 10;
int led_7 = 12;
int led_8 = 13;

int buzzer_1 = 2;
int buzzer_2 = 5;
int buzzer_3 = 8;
int buzzer_4 = 11;

void setup() {
  //Begin serial communication with Arduino and Arduino IDE (Serial Monitor)
  Serial.begin(9600);

  // initialize LED pins as output.
  pinMode(test_LED, OUTPUT);
  pinMode(led_1, OUTPUT);
  pinMode(led_2, OUTPUT);
  pinMode(led_3, OUTPUT);
```

Fire detection training accuracy



```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      19/20      8.76G      1.614      1.624      1.751         11        640: 100%|██████████| 88/88 [00:47<00:00,  1.86it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 13/13 [00:05<00:00,  2.21it/s]


      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      20/20      8.89G      1.594      1.612      1.708         15        640: 100%|██████████| 88/88 [00:46<00:00,  1.87it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 13/13 [00:06<00:00,  2.14it/s]


20 epochs completed in 0.317 hours.
Optimizer stripped from runs/detect/yolov8l_fire_detection/weights/last.pt, 52.0MB
Optimizer stripped from runs/detect/yolov8l_fire_detection/weights/best.pt, 52.0MB

Validating runs/detect/yolov8l_fire_detection/weights/best.pt...
Ultralytics 8.3.101 🚀 Python-3.11.11 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
Model summary (fused): 92 layers, 25,840,339 parameters, 0 gradients, 78.7 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 13/13 [00:08<00:00,  1.59it/s]
                   all        401        647      0.552      0.504      0.531      0.268
Speed: 0.3ms preprocess, 10.3ms inference, 0.0ms loss, 2.8ms postprocess per image
Results saved to runs/detect/yolov8l_fire_detection
ultralytics.utils.metrics.DetMetrics object with attributes:
```

Shelf training loose and accuracy



```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      16/20      9.02G     0.6529     0.6001      1.316          1        640: 100%|██████████| 23/23 [00:11<00:00,  2.02it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 4/4 [00:01<00:00,  2.27it/s]                   all        100        114      0.879      0.826      0.898      0.722

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      17/20      9.09G     0.6677     0.6221      1.309          1        640: 100%|██████████| 23/23 [00:11<00:00,  2.00it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 4/4 [00:01<00:00,  2.73it/s]                   all        100        114      0.918      0.842      0.898      0.696

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      18/20      9.15G     0.6073     0.6059      1.259          1        640: 100%|██████████| 23/23 [00:11<00:00,  2.02it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 4/4 [00:01<00:00,  2.96it/s]                   all        100        114      0.917      0.886      0.926      0.743

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      19/20      9.21G     0.6079     0.5693      1.273          2        640: 100%|██████████| 23/23 [00:11<00:00,  2.03it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 4/4 [00:01<00:00,  2.93it/s]                   all        100        114      0.934      0.873      0.932      0.748

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      20/20      9.29G     0.5214     0.4838      1.148          1        640: 100%|██████████| 23/23 [00:11<00:00,  2.02it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 4/4 [00:01<00:00,  2.97it/s]                   all        100        114      0.911      0.901      0.938      0.771

20 epochs completed in 0.078 hours.
Optimizer stripped from runs/detect/yolov8l_shelves_detection3/weights/last.pt, 52.0MB
Optimizer stripped from runs/detect/yolov8l_shelves_detection3/weights/best.pt, 52.0MB

Validating runs/detect/yolov8l_shelves_detection3/weights/best.pt...
Ultralytics 8.3.101 🚀 Python-3.11.11 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
Model summary (fused): 92 layers, 25,840,339 parameters, 0 gradients, 78.7 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 4/4 [00:02<00:00,  1.48it/s]
                   all        100        114      0.911      0.901      0.938      0.771
Speed: 0.2ms preprocess, 11.3ms inference, 0.0ms loss, 5.3ms postprocess per image
Results saved to runs/detect/yolov8l_shelves_detection3
ultralytics.utils.metrics.DetMetrics object with attributes:
```

Future improvements

- IOT integration with system

Functional and Non-Functional Requirements

Functional Requirements:

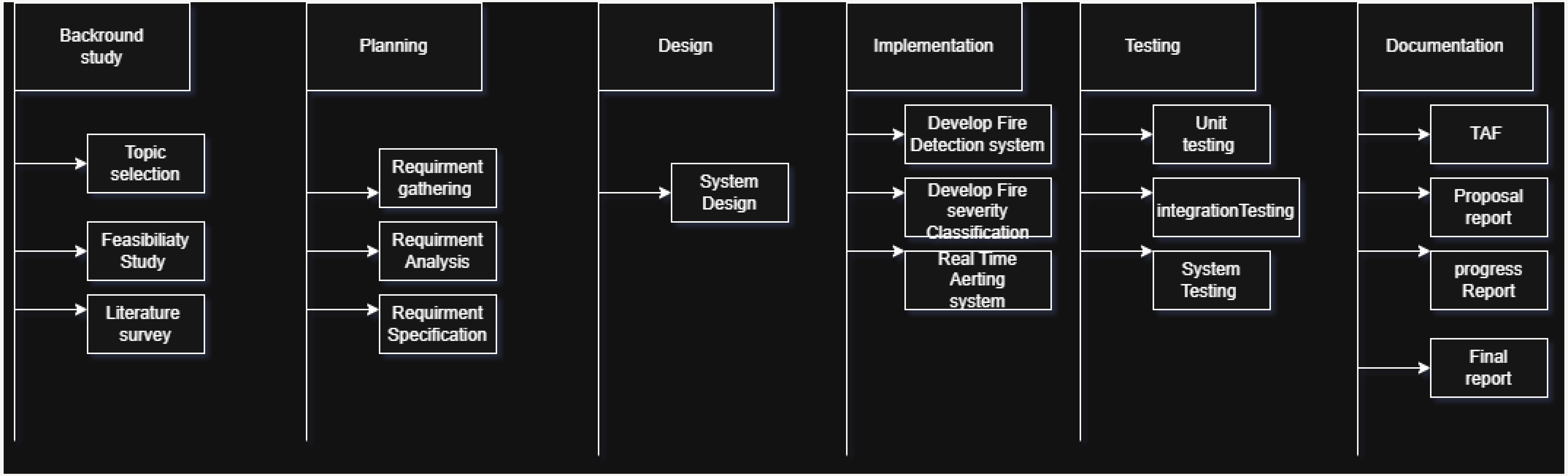
- Real-time fire and shelf detection
- Fire size classification
- Spread prediction
- Alerts and notifications
- Live dashboard with visual updates

- Non-Functional Requirements:

- Fast and accurate performance
- Easy to use and understand
- Safe and secure data handling
- Works with existing warehouse setups
- Reliable and scalable system

Tools and technology

Work Break-Down Structure



References

Fernando, M. (2020). *Fire Safety Systems in Sri Lankan Industries: Challenges and Opportunities*. Journal of Industrial Safety, 12(3), 45-58.

- Wickramasinghe, S., & Gunawardena, T. (2021). *Computer Vision for Industrial Safety: Applications in Sri Lanka*. Proceedings of the International Conference on AI and Automation.
- Kumar, R., & Patel, A. (2019). *Fire Detection and Surveillance Systems: A Review*. Journal of Artificial Intelligence, 31(7), 1021-1037.
- Zeng, H., & Wang, D. (2018). *Real-Time Fire Detection Using Deep Learning*. International Journal of Computer Vision and Pattern Recognition, 20(5), 101-114.
- Mohan, S., & Das, P. (2022). *The Role of AI in Enhancing Fire Safety: A Review*. International Journal of AI Applications, 16(2), 65-82.

Aspect 04

STOCK MOVEMENT FORECASTING & ANOMALY DETECTION AND WORKER PERFORMANCE MONITORING

Silva V S - IT21318184

Introduction

- This project proposes a data driven warehouse management system that provides two most essential features:

  (a) stock movement anomaly detection through time-series forecasting.

  (b) worker performance monitoring through activity logs and productivity analysis.

By integrating these key features into a single dashboard, this project provides warehouse managers a decision-support system that can both proactively recognize risks in stock flows and enhance people performance.

Research Problems

1. **Integration with Historical Data:**

2.Current inventory systems often focus on current data, leaving a gap in predictive analysis using past data to drive proactive decision-making.

2. Forecasting vs. Anomaly Detection
Most existing warehouse research emphasizes demand forecasting rather than detecting short-term stock anomalies such as theft, misplacement, or irregular flows.

3. Workforce–Inventory Integration
Worker performance monitoring is usually studied separately from stock management. There is no established framework linking worker productivity data with inventory movement anomalies. How can these two dimensions be combined into a single decision-support system?

4. Predominantly Reactive Systems
Commercial WMS solutions are largely reactive, flagging errors only after thresholds are breached. How can predictive analytics be used to transition from reactive alerts to proactive anomaly prevention?

5. Unified Dashboard Gap

Warehouse managers currently rely on fragmented systems for inventory and workforce data. How can a single unified dashboard be developed to integrate anomaly detection and worker performance monitoring for faster, data-driven decisions?

6. Limited Application of Industry 4.0 Principles

While Industry 4.0 research emphasizes AI, predictive analytics, and cyber-physical systems, there is a lack of practical implementation in warehouses. How can machine learning models be effectively operationalized within low-cost, scalable systems?

Main Objective

- This research aims to design and develop an improved warehouse management system that uses prediction and evidence to instruct both inventory movement anomaly detection, and worker performance, under one framework, within a data-driven system for a more effective warehouse operation with an aim of working towards industry 4.0 in line with operational efficiency.

Sub Objectives

Data Aggregation and Preprocessing:

- Collect and integrate historical inventory movement records and employee activity logs from existing WMS systems.
- Develop data cleaning and normalization processes to ensure high-quality inputs for analysis.

Baseline Modeling and Anomaly Detection:

- Establish dynamic statistical baselines (e.g., using time-series forecasting) for normal stock movements.
- Implement and optimize anomaly detection algorithms to flag movements that exceed the baseline thresholds.

Creating a worker performance monitoring tool:

- Fairly assess productivity compared to employee's inactivity.
- Provide managers useful information to better manage their workforce and training.

Visualization and Alerting Mechanism:

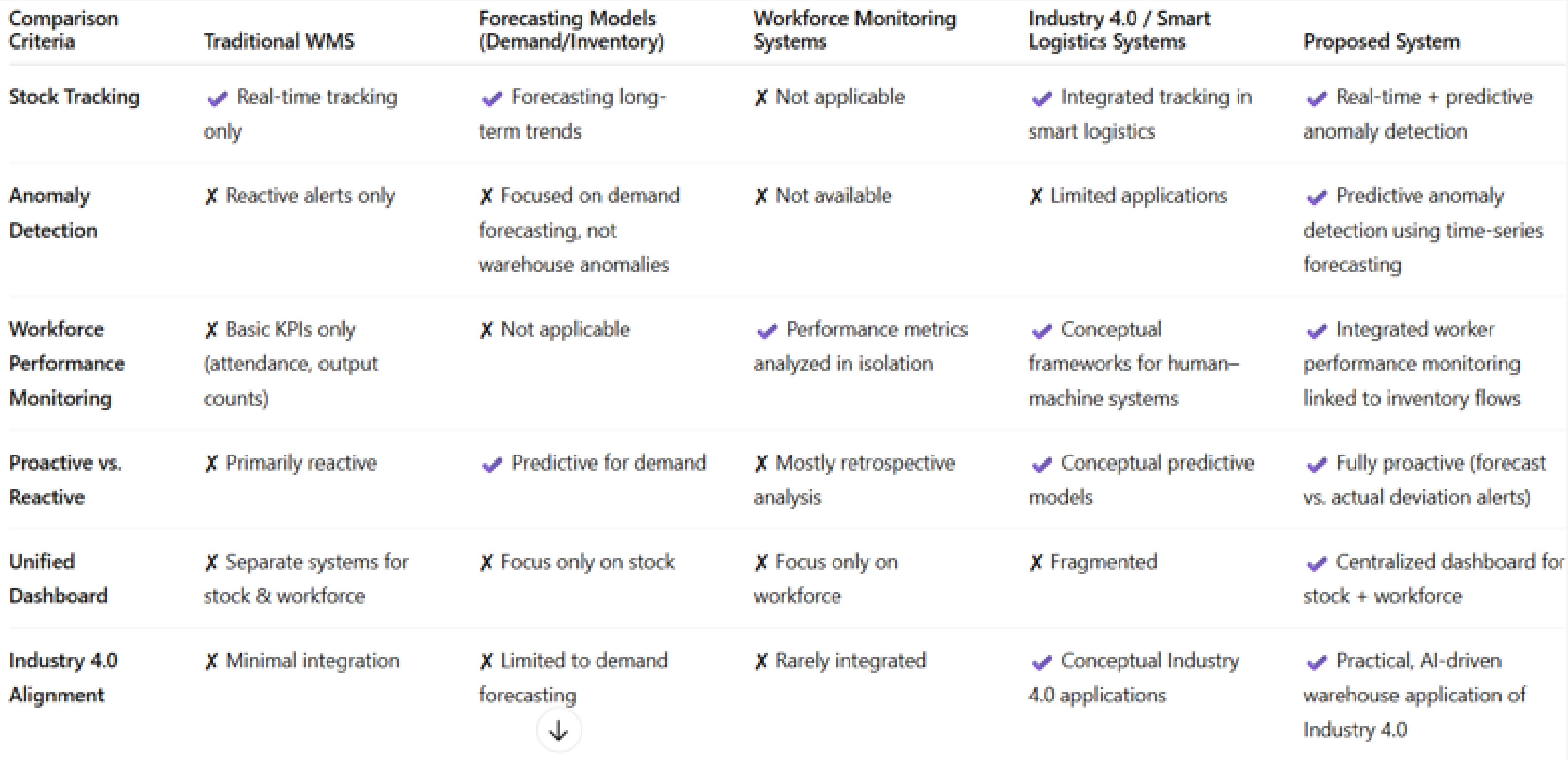
- Design a user-friendly dashboard that visually maps anomalies onto warehouse rack layouts.
- Integrate automated notification systems (e.g., email, SMS) to alert managers when anomalies are detected.

System Evaluation and Performance Optimization:

- Evaluate the system's prediction accuracy and responsiveness using defined metrics (e.g., false positive/negative rates, cost savings, and response times).

Research Gaps

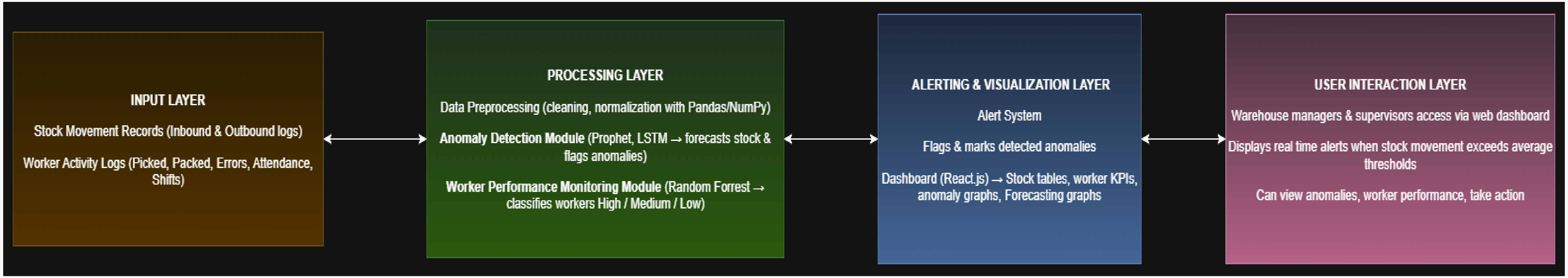
- **Leverage Historical Data:** Few studies have investigated the use of historical inventory movement data for predictive anomaly detection.
- **Threshold Determination:** Limited research exists on establishing statistically sound thresholds to trigger alerts based on past data.
- **Cost-Effective Solutions:** There is an opportunity to develop systems that provide actionable insights without the cost and complexity of installing and maintaining real-time IoT sensor networks.
- **Predominantly Reactive Warehouse Systems:** Most commercial WMS are reactive, alerting only after errors occur, rather than providing predictive insights. This post-incident approach increases operational risk and costs.
- **Fragmentation Between Workforce Monitoring and Warehouse Operations:** Workforce performance is rarely linked to inventory movement, preventing managers from seeing its impact on operational anomalies.
- **Lack of Unified Dashboards for Decision-Making:** Managers lack a single dashboard combining inventory anomalies and employee performance, slowing decision-making.
- **Limited Application of Industry 4.0 Principles in Warehouses:** While Industry 4.0 research highlights cyber-physical systems, predictive analytics, and big data in logistics and manufacturing, most studies remain theoretical with little direct application to warehouse operations.



Comparison Criteria	Traditional WMS	Forecasting Models (Demand/Inventory)	Workforce Monitoring Systems	Industry 4.0 / Smart Logistics Systems	Proposed System
Stock Tracking	✓ Real-time tracking only	✓ Forecasting long-term trends	✗ Not applicable	✓ Integrated tracking in smart logistics	✓ Real-time + predictive anomaly detection
Anomaly Detection	✗ Reactive alerts only	✗ Focused on demand forecasting, not warehouse anomalies	✗ Not available	✗ Limited applications	✓ Predictive anomaly detection using time-series forecasting
Workforce Performance Monitoring	✗ Basic KPIs only (attendance, output counts)	✗ Not applicable	✓ Performance metrics analyzed in isolation	✓ Conceptual frameworks for human–machine systems	✓ Integrated worker performance monitoring linked to inventory flows
Proactive vs. Reactive	✗ Primarily reactive	✓ Predictive for demand	✗ Mostly retrospective analysis	✓ Conceptual predictive models	✓ Fully proactive (forecast vs. actual deviation alerts)
Unified Dashboard	✗ Separate systems for stock & workforce	✗ Focus only on stock	✗ Focus only on workforce	✗ Fragmented	✓ Centralized dashboard for stock + workforce
Industry 4.0 Alignment	✗ Minimal integration	✗ Limited to demand forecasting	✗ Rarely integrated	✓ Conceptual Industry 4.0 applications	✓ Practical, AI-driven warehouse application of Industry 4.0

2 Research Gap summary

Methodology



INPUT LAYER

Stock Movement Records (Inbound & Outbound logs)

Worker Activity Logs (Picked, Packed, Errors, Attendance, Shifts)

PROCESSING LAYER

Data Preprocessing (cleaning, normalization with Pandas/NumPy)

Anomaly Detection Module (Prophet, LSTM → forecasts stock & flags anomalies)

Worker Performance Monitoring Module (Random Forrest → classifies workers High / Medium / Low)

ALERTING & VISUALIZATION LAYER

Alert System

Flags & marks detected anomalies

Dashboard (React.js) → Stock tables, worker KPIs, anomaly graphs, Forecasting graphs

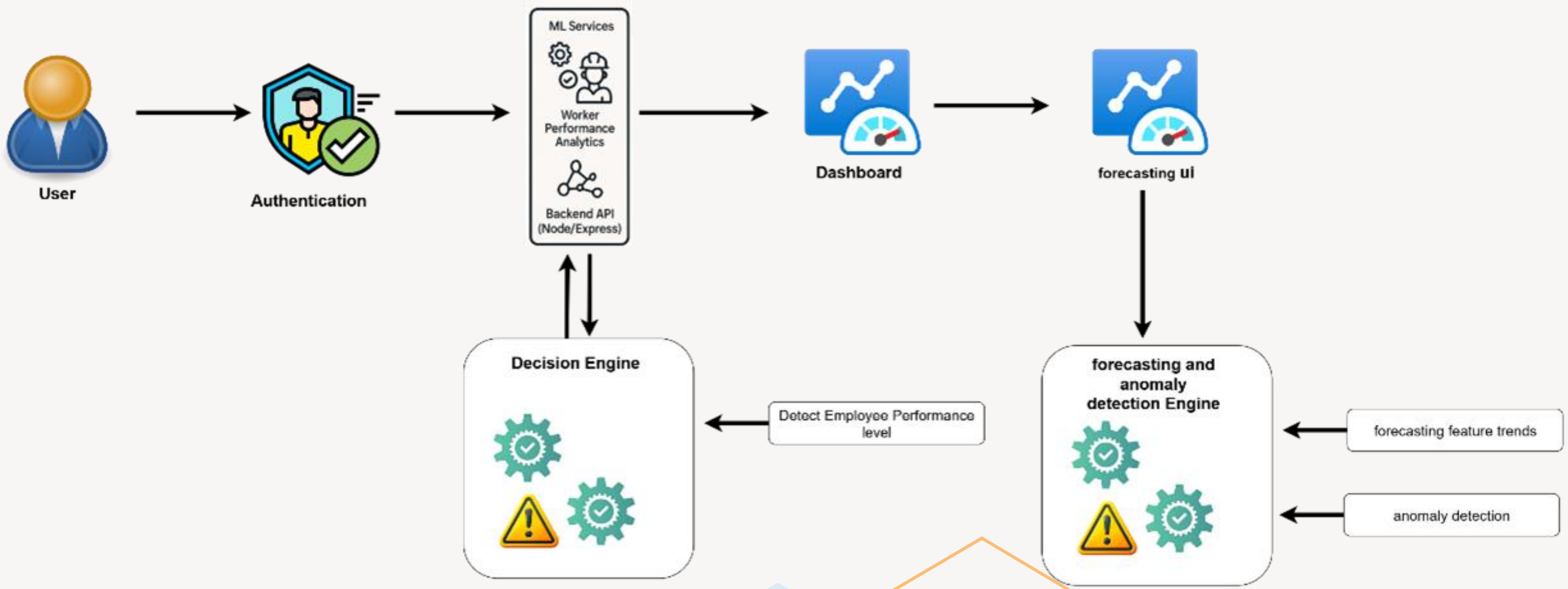
USER INTERACTION LAYER

Warehouse managers & supervisors access via web dashboard

Displays real time alerts when stock movement exceeds average thresholds

Can view anomalies, worker performance, take action

System Diagram



User

Authentication

ML Services

Worker Performance Analytics

Backend API (Node/Express)

Dashboard

forecasting ui

Decision Engine

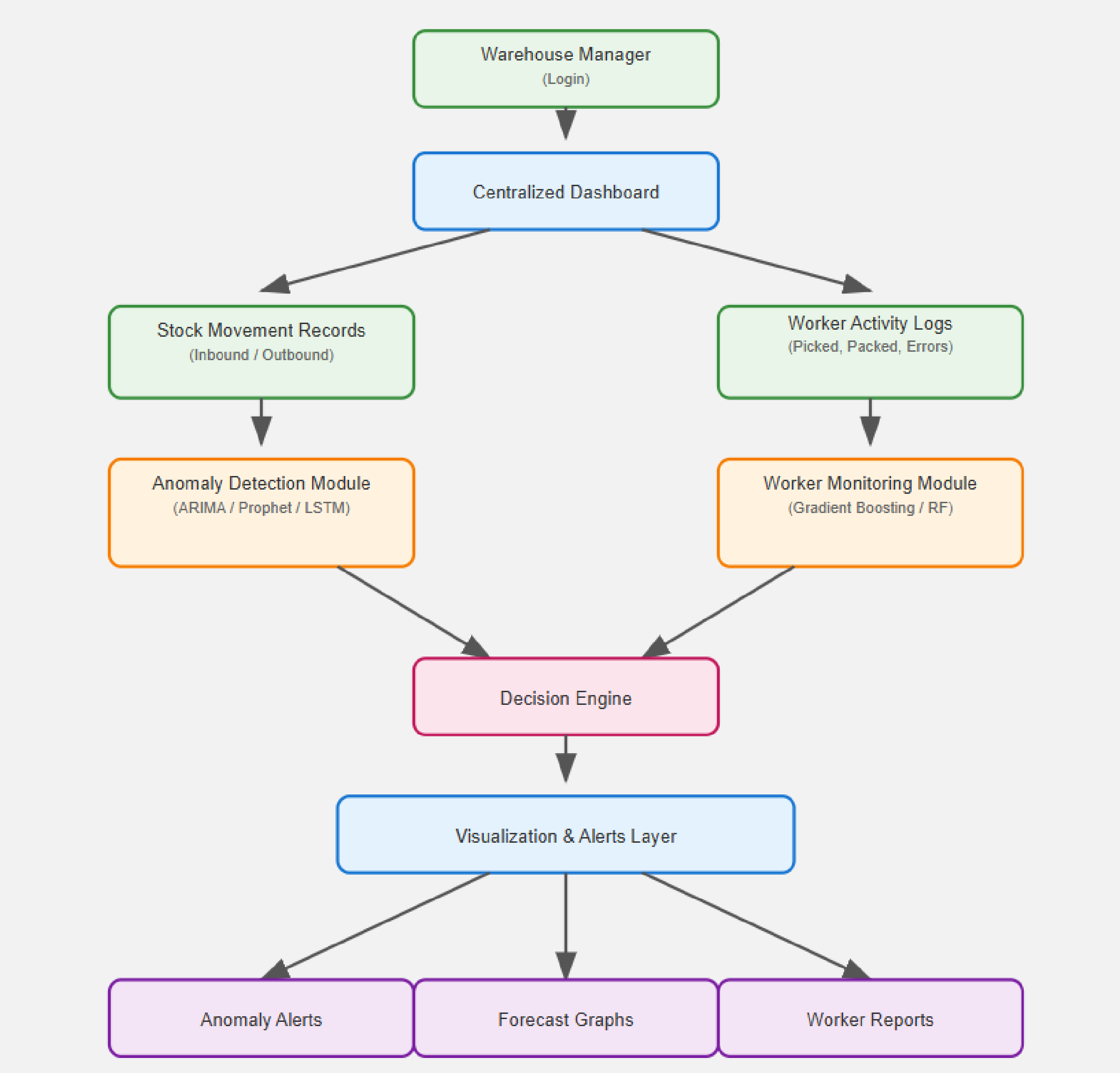
Detect Employee Performance level

forecasting and anomaly detection Engine

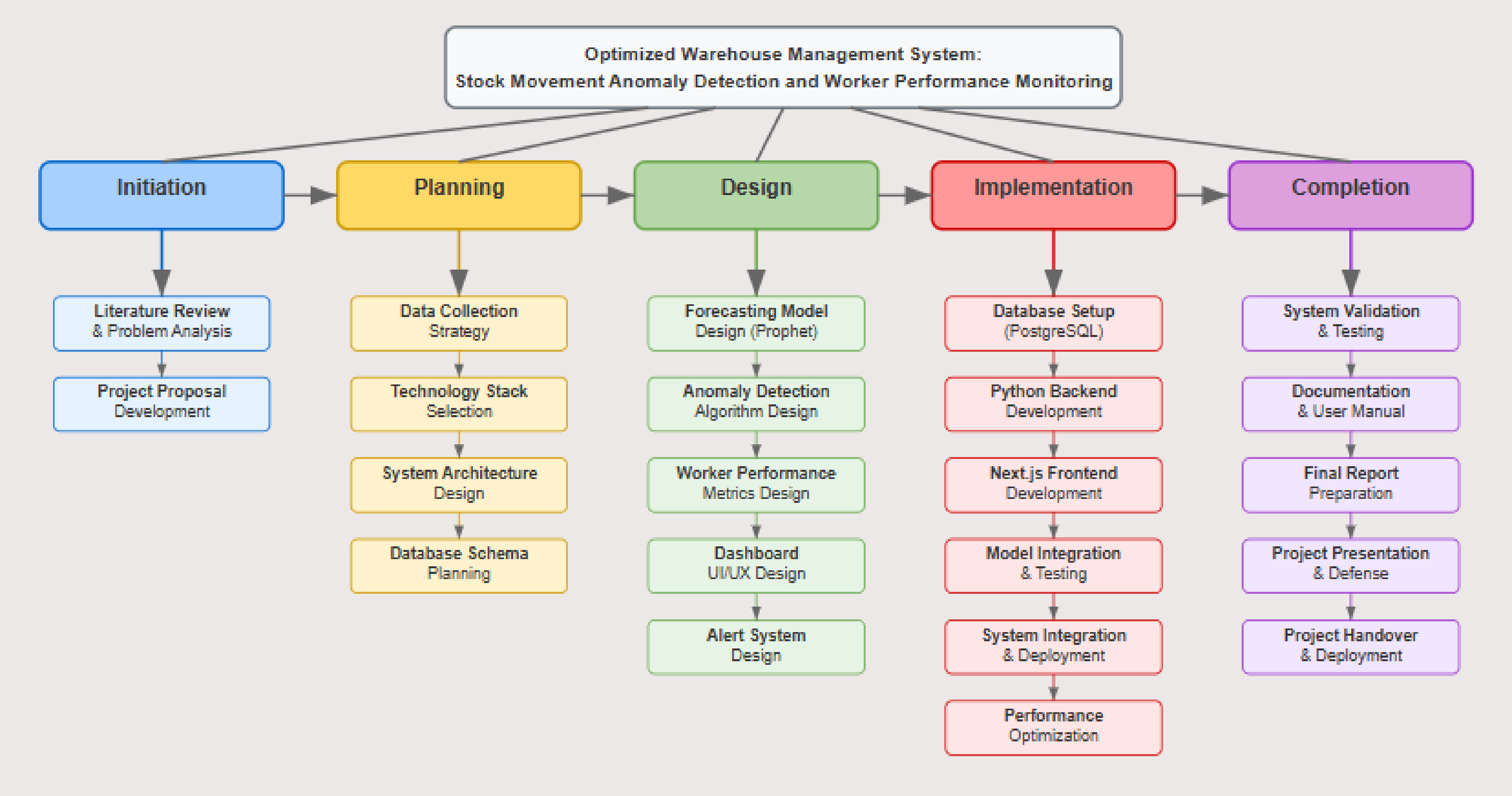
forecasting feature trends

anomaly detection

Flow Chart


Warehouse Manager
(Login)
Centralized Dashboard
Stock Movement Records
(Inbound / Outbound)
Worker Activity Logs
(Picked, Packed, Errors)
Anomaly Detection Module
(ARIMA / Prophet / LSTM)
Worker Monitoring Module
(Gradient Boosting / RF)
Decision Engine
Visualization & Alerts Layer
Anomaly Alerts
Forecast Graphs
Worker Reports

Work Break-Down Structure



Optimized Warehouse Management System:
Stock Movement Anomaly Detection and Worker Performance Monitoring

Initiation
- Literature Review & Problem Analysis
- Project Proposal Development

Planning
- Data Collection Strategy
- Technology Stack Selection
- System Architecture Design
- Database Schema Planning

Design
- Forecasting Model Design (Prophet)
- Anomaly Detection Algorithm Design
- Worker Performance Metrics Design
- Dashboard UI/UX Design
- Alert System Design

Implementation
- Database Setup (PostgreSQL)
- Python Backend Development
- Next.js Frontend Development
- Model Integration & Testing
- System Integration & Deployment
- Performance Optimization

Completion
- System Validation & Testing
- Documentation & User Manual
- Final Report Preparation
- Project Presentation & Defense
- Project Handover & Deployment

Current Progress

- Data Collection & Preprocessing completed
- Forecasting & Anomaly Detection Module Implemented
- Worker Performance Monitoring Module Implemented for classification

Future Improvements

- Add Alerting system
- Add Employee performance graphs
- UI Improvements

Functional Requirements

Historical Data Collection:

- Ability to gather historical inventory movement data and daily worker activities from ERP/WMS systems.

Data Preprocessing:

- Data cleaning, normalization, and integration processes.

Forecasting and Baseline Modeling:

- Load pre-trained forecasting models for both inbound and outbound quantities and generate future forecasts based on historical trends and apply rolling averages to smooth predictions.

Deviation/Anomaly Detection:

- Compare real observed data against forecasted values and flag deviations where actual dispatch or inbound quantities differ from forecasted values by more than a configurable percentage threshold.

Worker Performance Monitoring

- Apply Random Forrest classification model to categorize workers (High, Medium, Low performance).
- Generate worker performance scores and trends over time and allow supervisor validation by manual rating overrides.

Alerting and Visualization:

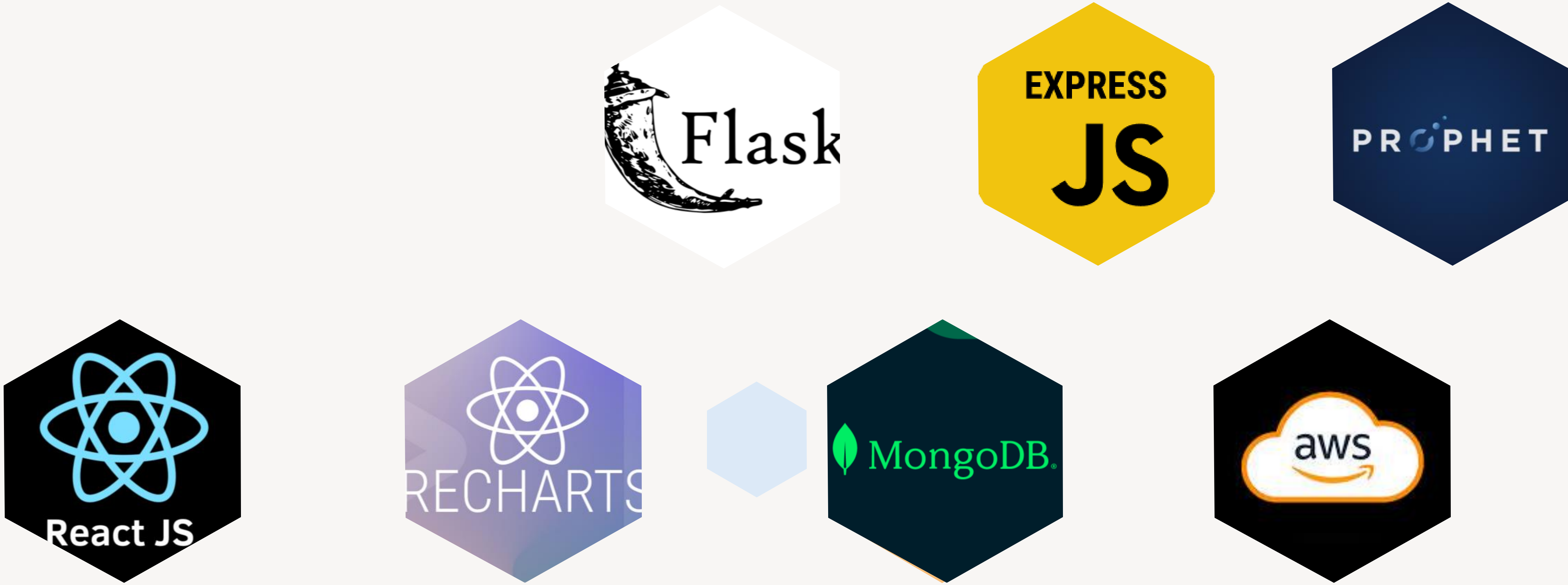
- Automated alert system that triggers notifications (email, SMS, in-app) when anomalies are detected.
- Generates aggregated summaries and visualizations for administrators and options for users to customize alert thresholds and frequency.

Non-Functional Requirements

- **Performance:** The system should process data and generate alerts within a predefined response time (e.g., under 5 seconds for each anomaly detection cycle).
- **Scalability:** Must handle large volumes of historical data and scale with growing data size and number of inventory nodes.
- **Reliability and Availability:** Ensure continuous system operation with high uptime (e.g., 99.9% availability).
- **Security:** Ensure data confidentiality and integrity with encryption and access controls.
- **Usability:** User-friendly dashboard with intuitive navigation and clear visualizations. And minimal training required for warehouse staff to effectively use the system.

Tools, Technologies & algorithms

- Data Processing Libraries (Pandas, NumPy)
- Time-series forecasting(Prophet)
- Random Forest



References

[1] A. Gunasekaran, N. Subramanian, and E. W. T. Ngai, "Managing relationships in supply chains: Emerging issues and challenges," International Journal of Production Economics, vol. 167, pp. 312–325, 2015.

[2] G. Marchet, M. Melacini, S. Perotti, and E. Tappia, "Development of a framework for managing warehouse operations," International Journal of Production Economics, vol. 204, pp. 164–178, 2018.

[3] M. A. Waller and S. E. Fawcett, "Data science, predictive analytics, and big data: a revolution that will transform supply chain design and management," Journal of Business Logistics, vol. 34, no. 2, pp. 77–84, 2013.

[4] R. de Koster, T. Le-Duc, and K. J. Roodbergen, "Design and control of warehouse order picking: A literature review," European Journal of Operational Research, vol. 182, no. 2, pp. 481–501, 2007.

[5] H. Qin and D. A. Nembhard, “Workforce flexibility in production systems: literature review andfuture directions,” International Journal of Production Research, vol. 53, no. 21, pp. 6360–6386,2015.

[6] Y. Li, L. Wang, and H. K. Chan, “Big data analytics in logistics and supply chainmanagement: A review of the literature and applications,” International Journal of ProductionResearch, vol. 57, no. 15–16, pp. 4854–4876, 2019.

[7] E. W. T. Ngai, et al., “RFID research: Anacademic literature review (1995–2005) and future research directions,” International Journal ofProduction Economics, vol. 112, no. 2, pp. 510–520, 2009.

[8] K. L. Choy, et al., “A knowledge-based logistics operations planning system for mitigating risk in warehouse order fulfillment,”Decision Support Systems, vol. 59, pp. 219–230, 2014.

[9] S. Nahavandi, “Industry 4.0 and therole of artificial intelligence in smart supply chain,” IEEE Engineering Management Review, vol.47, no. 1, pp. 5–9, 2019. 58 | Page

[10] E. Hofmann and M. Rüsch, “Industry 4.0 and thecurrent status as well as future prospects on logistics,” Computers in Industry, vol. 89, pp. 23–34, 2017.

[11] R. F. Babiceanu and R. Seker, “Big Data and virtualization for manufacturingcyber-physical systems: A survey of the current status and future outlook,” Computers inIndustry, vol. 81, pp. 128–137, 2016.

How to Promote

- Social Media Marketing
- E-mail Marketing
- Event Sponsorships
- Web-Ads
- YouTube Advertising
- Leaflets

Commercialization

Advantages:

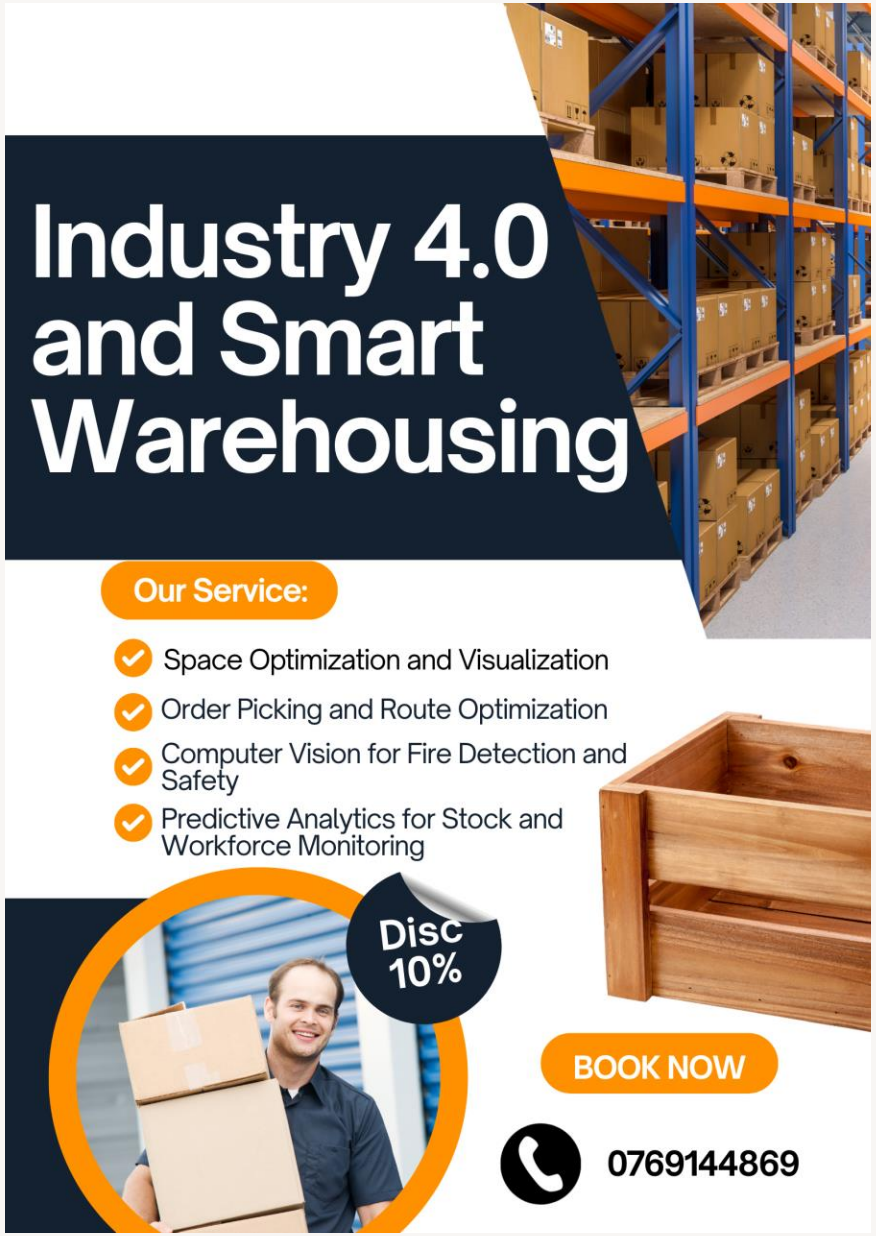
- User Friendly Interfaces
- Remote Location Logins

Target Audience:

- Warehouse Owners
- Warehouse Service Businesses

Marketplace:

- Using Social Media Platforms
- Conduct Awareness Programs to Warehouse Stakeholders



Industry 4.0 and Smart Warehousing

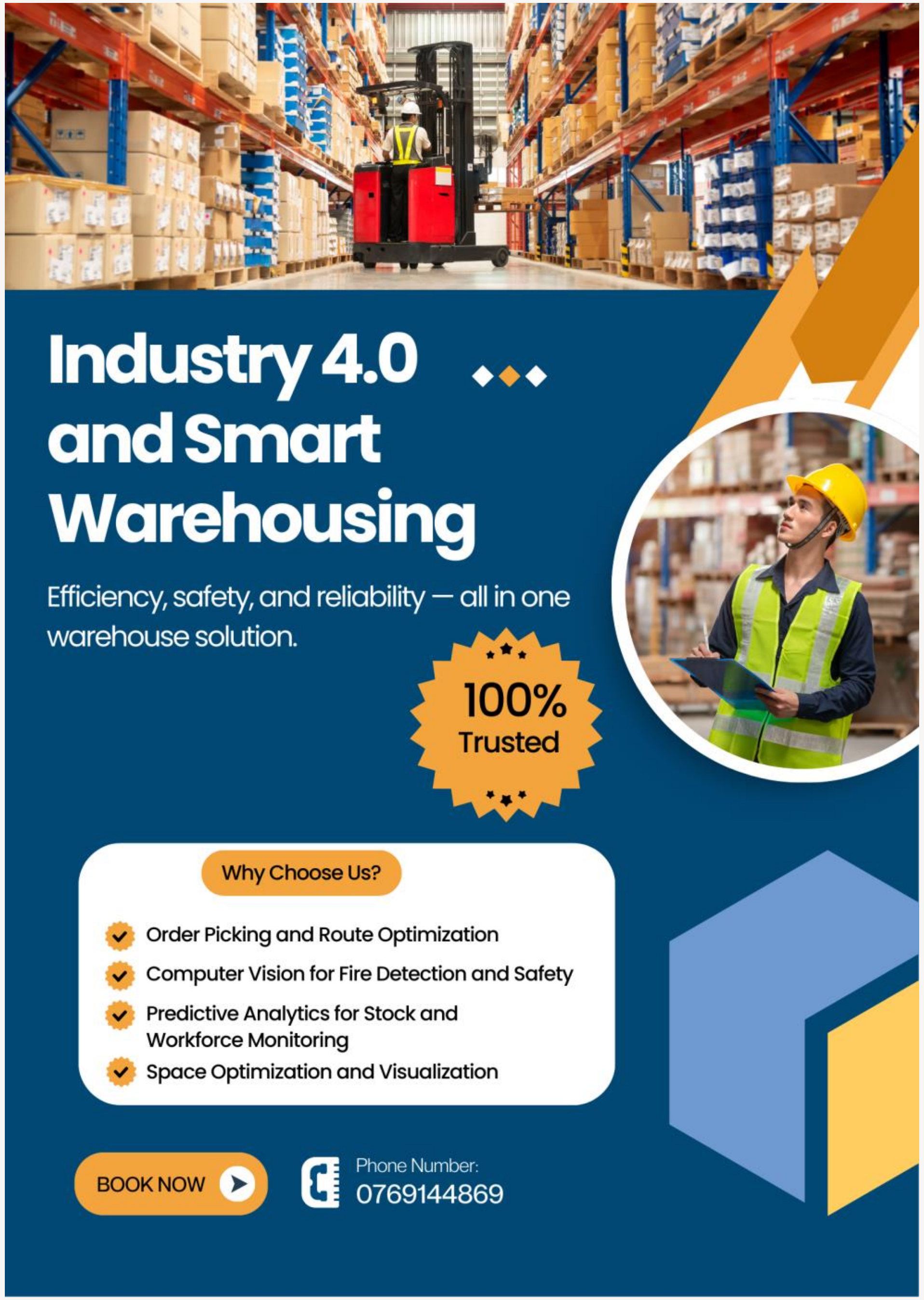
Our Service:

- ✓ Space Optimization and Visualization
- ✓ Order Picking and Route Optimization
- ✓ Computer Vision for Fire Detection and Safety
- ✓ Predictive Analytics for Stock and Workforce Monitoring

Disc 10%

BOOK NOW

☎ 0769144869



Industry 4.0 and Smart Warehousing

Efficiency, safety, and reliability — all in one warehouse solution.

100% Trusted

Why Choose Us?

- ✓ Order Picking and Route Optimization
- ✓ Computer Vision for Fire Detection and Safety
- ✓ Predictive Analytics for Stock and Workforce Monitoring
- ✓ Space Optimization and Visualization

BOOK NOW ▶

Phone Number:
0769144869

Budget

Hardware Costs:

- Cameras – LKR 20,000
- IOT Device – LKR 10,000

Hosting Costs:

- AWS s3 instance – LKR 30,000 * 12
- AWS Lambda instances – LKR 20,000 * 12

Other Expenses

- Misc – LKR 10,000

Total Estimated Budget CapEx: LKR 40,000/=
Total Estimated Budget OpEx Yearly: LKR 600,000/=

Project Requirements

Functional Requirements

1. Real-Time Monitoring: The system must enable users to monitor warehouse operations, including product arrangements and space utilization, in real time.
2. Dynamic Product Storage Optimization: Use Machine Learning models to dynamically suggest optimal product arrangements based on CBM, product turnover rates, and constraints.
3. Interactive User Interface: Develop an intuitive web-based application using React and Python Django (or Node.js with Express.js) for seamless user interaction.
4. Inventory Security Features: Provide safeguards for sensitive and high-value goods by implementing risk management measures such as access controls and alert systems.
5. Customizable Storage Strategies: Allow users to input constraints (e.g., product fragility, weight, or stacking rules) to tailor the storage optimization process.

Project Requirements

Non-Functional Requirements

1. Performance: The system should provide real-time responses for 3D visualization and storage optimization, with minimal latency.
2. Usability: The user interface must be intuitive, enabling users with varying levels of technical expertise to interact with the system effectively.
3. Scalability: The system should support increasing warehouse sizes, product volumes, and additional features without compromising performance.
4. Compatibility: Ensure compatibility across major web browsers and devices, including desktops, laptops, and tablets.
5. Reliability: The system must operate reliably under various conditions, ensuring data accuracy and uninterrupted service.

References

[1] T. D. Rupasinghe and S. Dissanayake (2018) An integrated warehouse design and optimization modelling approach to enhance supply chain performance.

[2] **Chung, S. H., & Lee, H. (2017).** "Optimization of warehouse space utilization for container storage." *Journal of Manufacturing Science and Engineering, 139*(2), 021004.

[3] Li, X., et al. (2023). "Optimized Pathfinding in Smart Warehouses." International Journal of Logistics Management.

[4] Y. Zhang, L. Zhang, and W. Li, "Real-time Dynamic Path Planning in Automated Warehouses," Journal of Warehouse Technology, 2021.

[5] **Zhao, X., & Lee, D. (2016).** "Predictive analytics for monitoring stock movements in warehouses." *International Journal of Production Research, 54*(10), 3023-3035.

[6] **Yang, Y., & Zhang, X. (2020).** "AI-based fire detection systems in warehouse management: A review." *Computers, Materials & Continua, 64*(2), 1071-1089

Thank you...