SmartShop AI

Prepared By

Meshaal Al-Zahrani

Abdulmajid Al-Nasser

Badr Al-Joudi

Fahad Al-Qahtani

Turki Al-Omran

Group (3)

CSC 443

Table of contents

Introduction

Managing inventory efficiently is one of the biggest challenges faced by retail stores today. Many shop owners still rely on manual estimates or basic software to decide what to order and when. This often leads to two major problems: running out of best-selling items—which means lost sales and unhappy customers—or overstocking products that don’t sell quickly—which ties up money and increases storage costs.

To tackle these issues, we introduce SmartShop, an intelligent inventory management system designed to take the guesswork out of stock control. Using machine learning, the system analyzes historical sales data, seasonal trends, and supplier delivery times to accurately predict future product demand. It then automatically generates smart reorder alerts, telling store managers exactly what to buy and when, all through an easy-to-use dashboard.

By adopting SmartShop, stores can significantly reduce instances of being understocked or overstocked. This leads to better use of capital, lower operating costs, and improved profit margins. Most importantly, it ensures that customers find what they need, when they need it—strengthening loyalty and driving repeat business.

This report provides a full overview of the SmartShop project, including its objectives, system design, technical approach, implementation plan, and expected benefits. Our goal is to demonstrate how modern AI technology can make inventory management simpler, smarter, and more sustainable for retailers of all sizes.

1.1 System Description

SmartShop is an intelligent inventory management system designed to help retail stores automate and optimize their stock ordering process. The system uses artificial intelligence to analyze past sales data and accurately predict future product demand. Based on these predictions, it automatically generates smart restocking alerts, telling store managers exactly what to order and when. This eliminates manual guesswork, prevents both stock-outs and overstocking, and ultimately saves the business time and money while improving customer satisfaction. The solution will provide these insights through an easy-to-use web dashboard, making advanced inventory management accessible for any shop owner.

1.2 Problem Statement

Many stores have a hard time managing their inventory. They often run out of best-selling items, causing them to lose sales and disappoint customers. At the same time, they sometimes buy too much of products that don’t sell quickly. This leaves money stuck in storage instead of being used for the business. These problems happen because store owners often have to guess how much stock to order, instead of using smart data-based decisions.

1.3 Project Objectives

The main goal of this project is to build a system that helps stores manage their inventory smarter. Here’s what we aim to achieve:

- **Predict Demand Accurately:** Use past sales data to forecast how much of each product will be sold in the future.
- **Automate Reordering:** Tell store owners exactly when and how much to order, so they never run out or overstock.
- **Provide a Simple Dashboard:** Create an easy-to-use screen where store owners can see stock levels, predictions, and alerts.
- **Save Time and Money:** Help stores reduce wasted inventory, avoid lost sales, and use staff time more efficiently.

Requirement Analysis

The Requirement Analysis phase is a critical step in the SmartShop development process. It involves a detailed examination of all gathered requirements to ensure they are clear, consistent, complete, and technically feasible before moving to the design stage. We have analyzed both functional requirements (what the system should do) and non-functional requirements (how the system should perform). This analysis ensures that the system will effectively address inventory management challenges while remaining practical to implement and use. The following sections break down how requirements have been categorized, prioritized, and validated to guide the development of a solution that meets business objectives while remaining within technical constraints.

2.1 User Requirements

1. Shop Owner / Manager

- Needs a simple dashboard that shows most important inventory information.
- Needs to receive clear alerts when it's time to reorder specific products.
- Needs to see predictions about what will sell well in the coming weeks.
- Needs to understand how much money savings by avoiding overstocking and stockouts.
- Needs the system to be secure and protect business data.
- Needs to be able to access the system from computer or phone.

2. Store Staff / Employees

- Need to quickly check current stock levels for products.
- Need to see which items need to be restocked soon.
- Need to easily record new inventory when it arrives.
- Need a simple way to update product information when needed.
- Need to be able to use the system with minimal training.

3. All Users

- Need the system to be easy to understand and use.
- Need it to work quickly without delays.
- Need reliable information that can be trusted.
- Need the system to be available when needed.
- Need it to help save time on inventory management tasks.

4. Technical Requirements for Users

- Need the system to work on an existing computer or mobile device.
- Need it to work with current internet browser.

- Need to be able to upload sales data easily.
- Need the system to connect with an existing inventory system if possible.
- Need to be able to print or export reports when necessary.

2.2 Data Requirements

- **Input Data Requirements**

The raw information the system needs to collect from the shop

Data Type	**Description**	**Source**	**Format & Example**
Historical Sales Data	The most critical data. A record of what products were sold and when.	Point-of-Sale (POS) System	**CSV/Excel File** with columns: Date, Product ID, Product Name, Quantity Sold, Unit Price
Product Master List	Basic information about every product the shop sells.	Inventory Management System or Manual List	**CSV/Excel File** with columns: Product ID, Product Name, Category, Supplier ID, Cost Price, Selling Price
Supplier Information	Details about each supplier and their delivery performance.	Purchasing Records	**CSV/Excel File** with columns: Supplier ID, Supplier Name, Contact Info, Average Lead Time (days)
Current Inventory Levels	A snapshot of how much stock is currently available for each product.	Manual Count or Stock Management System	**CSV/Excel File** with columns: Product ID, Current Stock

- **Output Data Requirements**

The valuable information the system generates for the user

Data Type	**Description**	**Format & Presentation**
Demand Forecasts	Predictions of how much of each product will be sold in the future (e.g., next 4 weeks).	**Visual Chart** on the dashboard (line graph). **Downloadable Report** (PDF/CSV).
Reorder Alerts	Actionable notifications listing which products to order and in what quantity.	**Dashboard List** with "Order Now" buttons. **Email/SMS Notifications** for critical alerts.
Inventory Reports	Summaries of inventory health, including value, turnover rates, and performance over time.	**Dashboard Charts** (e.g., bar charts, pie charts). **Scheduled PDF Reports** (e.g., weekly, monthly).

Supplier Performance Reports	Insights into which suppliers are reliable and which are often delayed.	**Dashboard Table** ranking suppliers by lead time and reliability.

2.3 Functional Requirements:

1. **Data Input and Management**
   - The system must allow users to upload or connect their historical sales data.
   - It should store and manage product details, supplier information, and inventory levels.
2. **Demand Prediction**
   - The system must use AI to analyze past sales and predict future demand for each product.
   - Predictions should account for seasonal trends, holidays, and promotions.
3. **Automated Reordering Alerts**
   - The system must automatically calculate reorder points and quantities.
   - It should send alerts to users when it’s time to restock specific items.
4. **Dashboard and Reporting**
   - Users must be able to view inventory levels, predictions, and alerts on a simple dashboard.
   - The system should generate basic reports on sales trends and inventory performance.
5. **User Management**
   - The system must support secure login for store owners and staff with appropriate access levels.

2.4 Non-Functional Requirements:

1. **Usability**
   - The interface must be simple, clear, and easy to use, even for non-technical users.
   - Dashboard navigation should be intuitive, with no more than three clicks to access key features.
   - The system should provide tooltips or guided tutorials for first-time users.

2. **Reliability**

   - The system must be available and work correctly with minimal downtime.
   - Target uptime should be ≥ 99.5% annually.
   - The system should gracefully handle errors and provide helpful messages to users.

3. **Security**

   - User and business data must be protected and stored securely.
   - All data transmissions should use encryption (e.g., HTTPS).
   - User passwords must be hashed and never stored in plain text.
   - Role-based access control should ensure users only see data relevant to their role.

4. **Performance**

   - The system should process data and generate predictions quickly, even with large datasets.
   - Dashboard pages should load in under 3 seconds.
   - AI model predictions should complete within 1–5 minutes for standard datasets.
   - The system should support at least 50 concurrent users without slowdowns.

5. **Scalability**

   - The system should be able to handle growth, such as more users, products, or higher sales volumes.
   - It should support scaling up to 10,000+ products and 100+ users.
   - Database and server architecture must allow easy expansion as needed.

6. **Compatibility**

   - The system should work on common web browsers (Chrome, Firefox, Safari, Edge).
   - The dashboard should be accessible on both desktop and mobile devices.

7. **Maintainability**

   - The code should be well-documented and modular for easy updates.
   - The system should allow easy updates to AI models or business rules without disrupting service.

8. **Data Accuracy**

   - Demand forecasts should have a target accuracy of ≥ 85% compared to actual sales.
   - Inventory data must reflect real-time or near-real-time stock levels.

2.5 Methodology

The development of the system will follow a structured methodology that combines the iterative, flexible approach of **Agile** development with the disciplined, phase-driven structure of the **Waterfall** model. This hybrid approach is chosen to ensure the project is well-planned and documented while remaining adaptable to changes and new insights, particularly during the AI model development phase. The methodology is broken down into five distinct phases

Phase	**Key Activities**	**Primary Outcome**
1. Planning	Requirements gathering, feasibility study, project planning	Project Charter & Requirement Specs
2. Design	Architectural design, UI/UX mockups, algorithm selection	Technical Design Document
3. Development	Agile sprints, backend/frontend development, AI model training	Functional Application
4. Deployment	System testing, user training, cloud deployment	Live, Operational System
5. Maintenance	Monitoring, support, model retraining, updates	Continuous System Improvement

2.6 AI Algorithms & Technology

SmartShop uses artificial intelligence (AI) to act as a smart assistant for the inventory. It works by learning from the store's past sales data.

– **It Predicts the Future:** The AI analyzes what was sold in the past to forecast what likely to sell in the future. It automatically factors in things like seasonal trends (e.g., selling more sunscreen in summer) and upcoming holidays.

– **It Makes Smart Decisions:** Instead of guessing when to reorder products, the AI calculates the perfect time and the ideal quantity to order. It does this by combining its sales predictions with knowledge of how long suppliers take to deliver.

– **The Result:** automatic alerts telling exactly what to restock and when. This prevents running out of popular items (avoiding lost sales) and prevents from over-ordering slow-moving products (freeing up cash).

The core of SmartShop is powered by **Machine Learning (ML)**, a type of Artificial Intelligence that allows the system to learn from data and make predictions. The project applies specific ML techniques to solve the precise problems of inventory management. Here are the AI technologies that will be used:

1. Time Series Forecasting for Demand Prediction

This is the main AI function.

- **What it is:** A technique specifically designed to analyze data points collected over time (like daily sales) to predict future values.
- **How it works:** The system will analyze the historical sales data to identify patterns, such as:
  - **Trends:** Is the product selling more or less over time?
  - **Seasonality:** Are there weekly or yearly cycles? (e.g., higher sales on weekends, or for specific seasons).
  - **Special Events:** How do holidays or promotions affect sales?
- **Specific Models & Libraries:**
  - **Prophet:** A library created by Facebook. It is very user-friendly and particularly good at handling the strong seasonal patterns common in retail sales. This will be the primary tool.
  - **SARIMA (Seasonal AutoRegressive Integrated Moving Average):** A classic statistical model for time series forecasting. It will be used to validate predictions or for products with less complex patterns.
  - **Scikit-learn:** A popular Python library that provides tools for simpler linear regression models that can also be useful.

2. Supervised Machine Learning

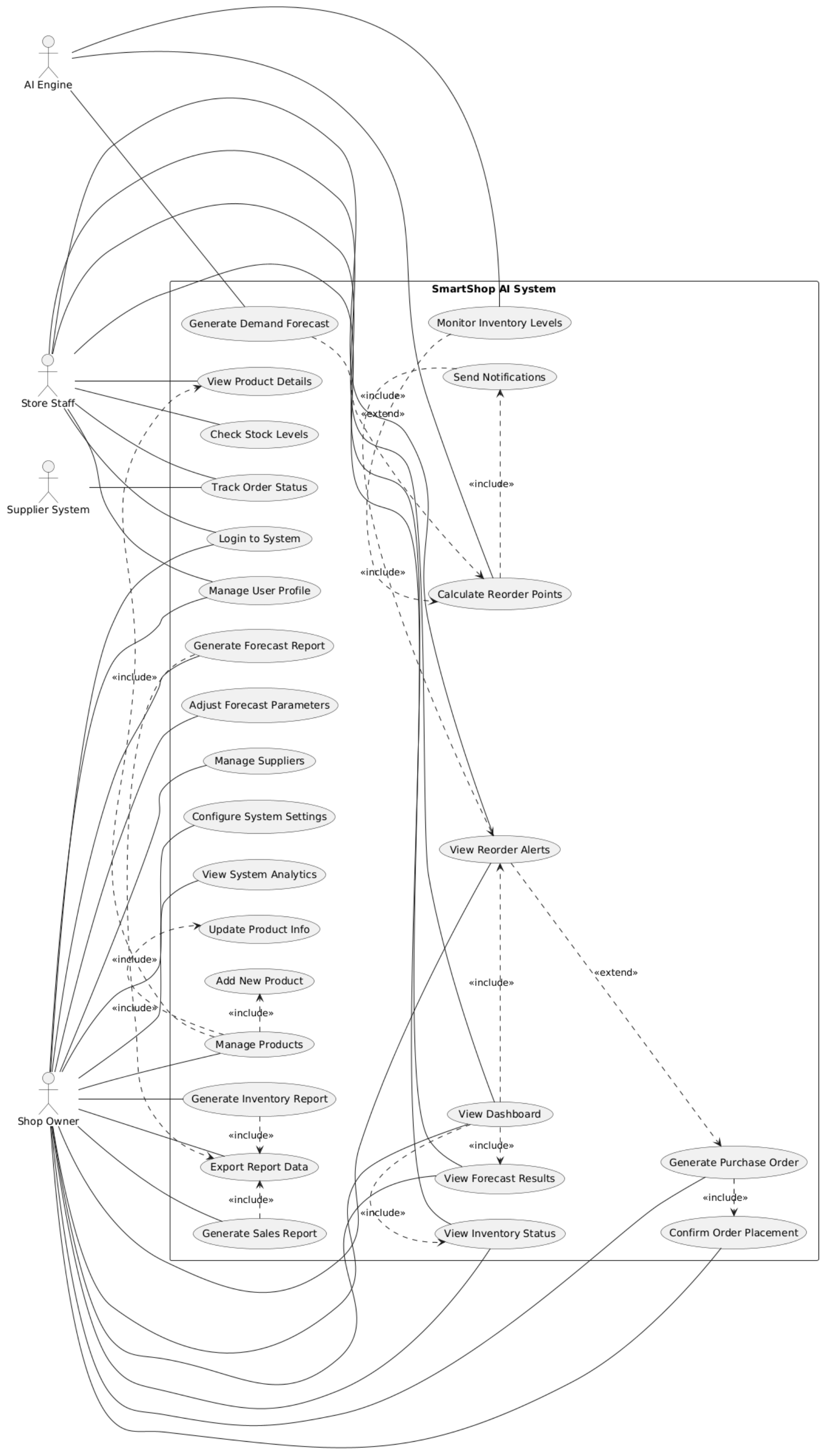
This is the category of ML that both Prophet and SARIMA fall under.

- **What it is:** The model is "trained" on historical data where we already know the outcome. For example, we show the model sales data from last year and it learns the patterns that led to sales figures this January. Then, we ask it to predict sales for next January.
- **Simple Analogy:** It's like teaching a student with past exam papers and then giving them a new test.

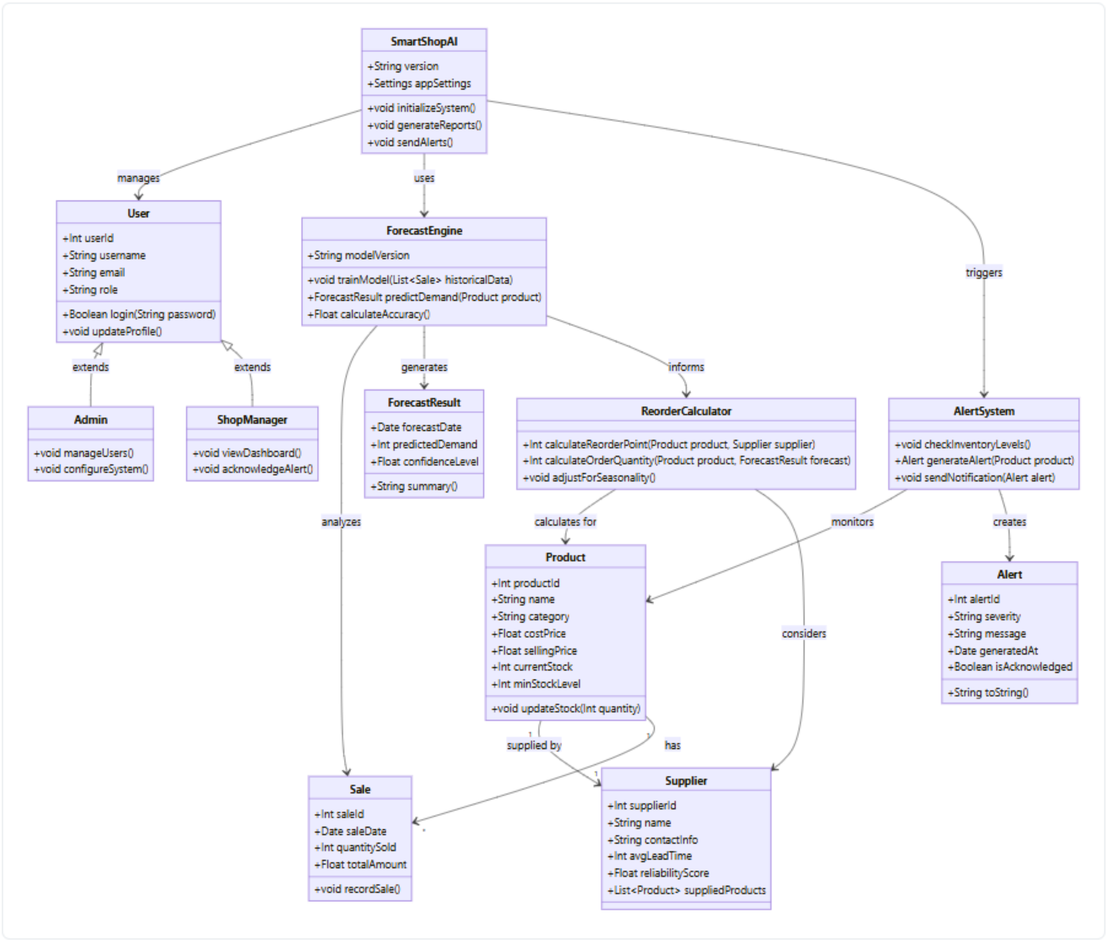
2.6.1 AI Workflow:

1. **Input:** The system imports shop's historical sales data.
2. **Model Training:** The AI (using Prophet primarily) analyzes this data to learn the unique sales patterns for each product.
3. **Prediction:** The trained model generates a forecast of future demand for the coming weeks.
4. **Action:** This forecast is automatically used to calculate smart reorder points and quantities.

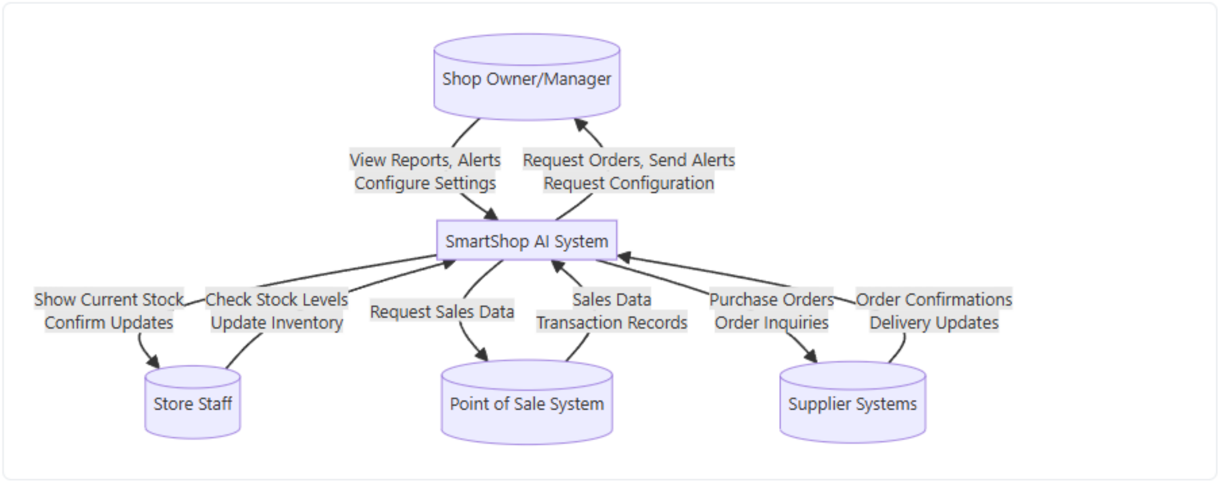
2.7 Use Case Diagram



2.8 System Class Diagram



2.9 Data Flow Diagram



System Design

The system design for SmartShop outlines how the different components of the system will work together to meet the project's goals. This includes the overall architecture, how data will flow through the system, and how users will interact with it. The design focuses on simplicity, scalability, and reliability to ensure the system is both powerful and easy to use. Below, we break down the key parts of the system and how they connect to help stores manage inventory smarter.

3.1 System Architecture

The system is built with four main parts that work together. Here's a simple breakdown:

1. **User Interface (Dashboard)**
   - This is the screen users see and interact with.
   - Built using **React.js**, a modern tool for creating clean and fast web interfaces.
   - Shows inventory levels, predicts future sales, and sends restock alerts.

2. **Backend**
   - The part that does the thinking behind the scenes.
   - Built with **Python** because it's great for AI and data tasks.
   - Uses machine learning models like **Facebook Prophet** to predict sales.
   - Handles user requests and sends information to the frontend.

3. **Database**
   - Stores all important information like sales history, product details, and supplier info.
   - Uses **PostgreSQL**, a reliable and easy-to-use database system.

4. **Cloud Hosting**
   - The system runs on the cloud (like **Google Cloud**), so it's secure, accessible from anywhere, and can grow as the business grows.

Here's how these parts work together:

- The **dashboard** lets the user see what's happening.
- The **backend** uses AI to analyze past sales and predict what to order.
- The **database** safely stores all the data.
- Everything lives on the **cloud**, so it's always available.

3.2 Tools & Technology

SmartShop will be built using a modern, reliable, and scalable set of technologies. Here's a simple breakdown of what we will use and why:

1. Frontend

This is the visual part of the system that the shop owner interacts with in their web browser.

- **React.js:** A popular JavaScript library for building fast and interactive user interfaces. It allows us to create a dynamic dashboard that updates in real-time without refreshing the page.
- **Chart.js:** A simple yet flexible JavaScript charting library to create clear visual graphs of sales trends, inventory levels, and forecasted demand.

2. Backend

This is the server-side logic that processes data, runs the AI models, and communicates with the database and frontend.

- **Python:** The programming language of choice because it has the best and most extensive libraries for AI and data analysis.
- **Flask/Django:** These are Python-based web frameworks. They will handle user login, website traffic, and requests from the frontend. (Flask is lighter; Django has more built-in features).

3. Database

This is where all the data is stored securely.

- **PostgreSQL:** A robust, open-source relational database. It's perfect for storing structured data like product lists, sales transactions, user details, and supplier information. It's reliable and handles large amounts of data well.

4. Deployment

To make the system accessible from anywhere, it will be hosted on a cloud platform.

- **Cloud Platform (Google Cloud):** We will use a cloud service like Google Cloud Platform (GCP). This provides:
  - **Virtual Server:** A computer in the cloud to run backend and database.
  - **Storage:** Secure space to store all data backups.

3.3 System Specifications & Cost Estimation

- **Hardware Requirements (Cloud-Based)**

Since SmartShop is a web application, it runs on internet servers (the cloud). The shop does not need to buy any physical hardware. The cloud provider supplies all the computing power needed.

Component	Description	Purpose	Monthly Cost
General Purpose Server	Good processing power & memory	Runs the main application and website	$80 - $150
Database Server	Optimized for fast data storage and retrieval	Stores all product, sales, and user data securely	$90 - $180
File Storage	Basic storage space for backups and reports	Keeps safe copies of important data	$5 - $15
Total Estimated Monthly Cost			**~$175 - $345**

- **Software Requirements**

The system is built using reliable and often free software tools to keep costs low.

Component	Technology / Tool	Cost
Website Frontend	React.js	Free
Server & AI Backend	Python, Django	Free
Machine Learning	Facebook Prophet	Free
Database	PostgreSQL	Free
Development Tools	VS Code	Free
Other Services	Domain name	$15 - $30
Total Estimated Monthly Cost		**$15 - $30**

Summary of Total Estimated Costs

Cost Type	Description	Monthly Estimate
Cloud Hosting (Hardware)	Renting server space and power	$175 - $345
Software & Services	Domain, security, and monitoring	$15 - $30
Total Operational Cost		**~$190 - $375**

3.4 Feasibility Study

1. Technical Feasibility

– **Assessment:** The proposed technology stack (React, Python, PostgreSQL, cloud hosting) is mature, widely adopted, and well-supported. The machine learning models required (e.g., Prophet) are proven for demand forecasting tasks and are available as open-source libraries. There are no significant technical barriers or unknown risks to building this system.

– **Verdict: Highly Feasible.**

2. Economic Feasibility

– **Assessment:** The system is designed to be cost-effective. Using open-source software eliminates licensing fees. The primary cost is cloud hosting, which operates on a pay-as-you-go model, avoiding large upfront capital expenditure. The financial benefits—reduced stockouts, less wasted capital tied up in excess inventory, and saved staff time—are expected to outweigh the operational costs significantly for any medium-to-large retail business.

– **Verdict: Highly Feasible.** The Return on Investment (ROI) is strong.

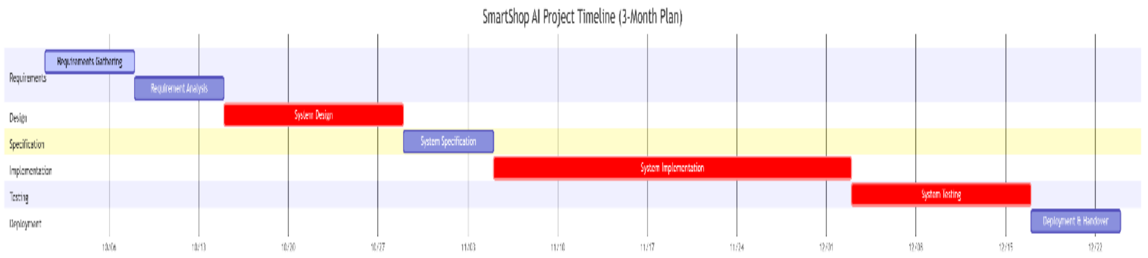
3. Operational Feasibility

– **Assessment:** The system is designed for ease of use. The intuitive dashboard requires minimal training, ensuring smooth adoption by store owners and staff. It automates tedious manual tasks (like calculating reorder points), freeing up employees for more valuable work. It solves a clear and painful operational problem for the target users.

– **Verdict: Highly Feasible.**

4. Scheduling Feasibility

- **Assessment:** The project is well-scoped and can be broken down into clear phases (e.g., design, backend development, frontend development, AI integration, testing). A team of 2-3 developers could build a Minimum Viable Product (MVP) within 3**-4 months**.
- **Verdict: Feasible.** The timeline is realistic.

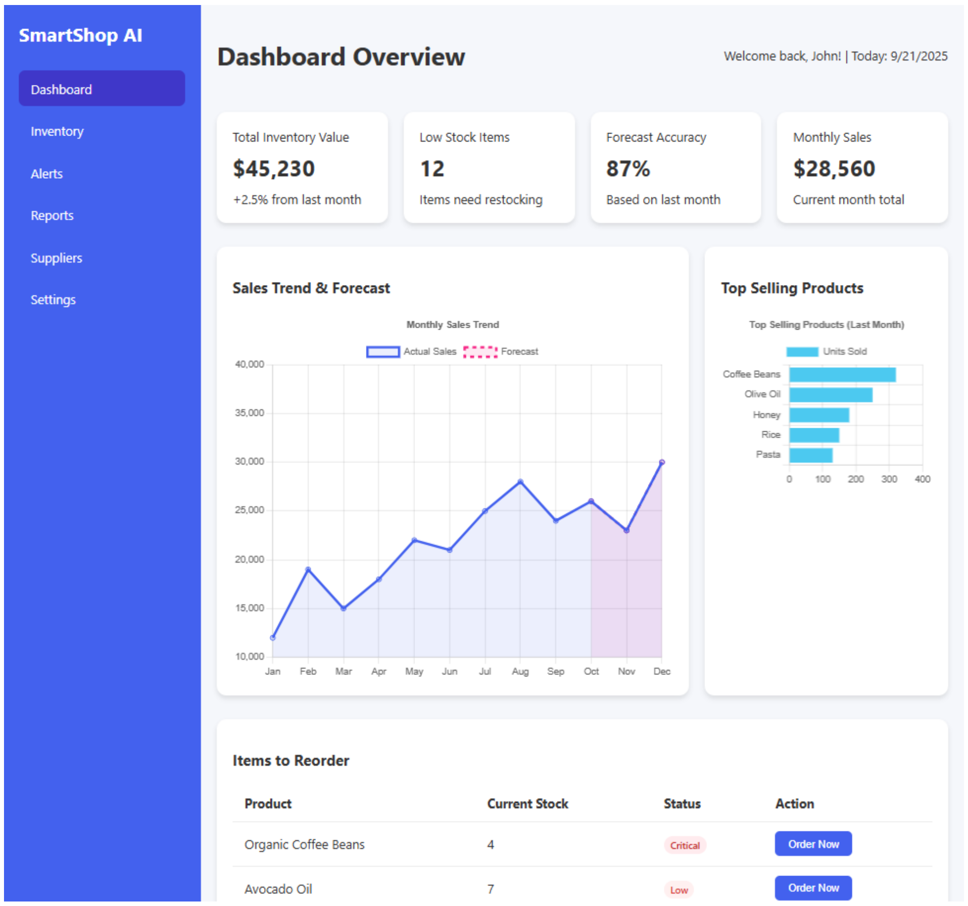


The system is **feasible** to develop and deploy. It is technically sound, economically justified, will be readily adopted by its users, and can be completed in a reasonable timeframe. The project is recommended to proceed to the development phase.

System Implementation

This section outlines the plan for building and deploying the SmartShop system. It describes the main steps we will take to turn the design into a working product, from setting up the initial development environment to launching the final application for store owners to use. The focus is on a clear, organized approach to ensure the system is reliable, meets all requirements, and is delivered on time. The implementation will be divided into key phases to manage progress effectively.

- **Dashboard Overview**



Dashboard Overview

Welcome back, John! | Today: 9/21/2025

Total Inventory Value
$45,230
+2.5% from last month

Low Stock Items
12
Items need restocking

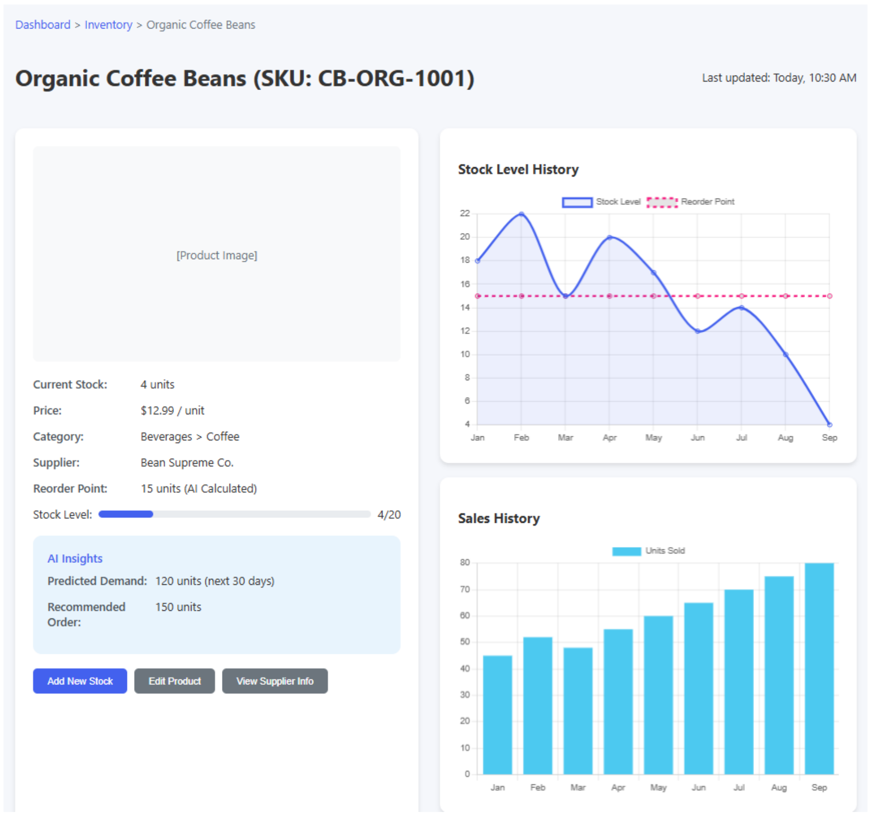
Forecast Accuracy
87%
Based on last month

Monthly Sales
$28,560
Current month total

Items to Reorder

Product	Current Stock	Status	Action
Organic Coffee Beans	4	Critical	Order Now
Avocado Oil	7	Low	Order Now

- **Product Detail Page**



Dashboard > Inventory > Organic Coffee Beans

Organic Coffee Beans (SKU: CB-ORG-1001)

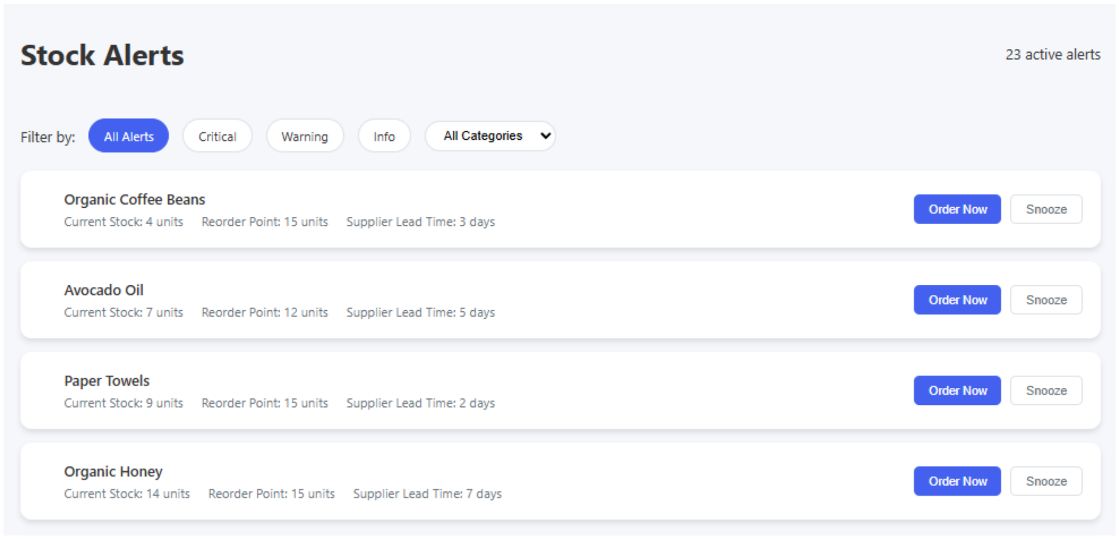
Last updated: Today, 10:30 AM

Current Stock:	4 units
Price:	$12.99 / unit
Category:	Beverages > Coffee
Supplier:	Bean Supreme Co.
Reorder Point:	15 units (AI Calculated)
Stock Level:	4/20

AI Insights

Predicted Demand:	120 units (next 30 days)
Recommended Order:	150 units

- **Alerts & Notifications Center**



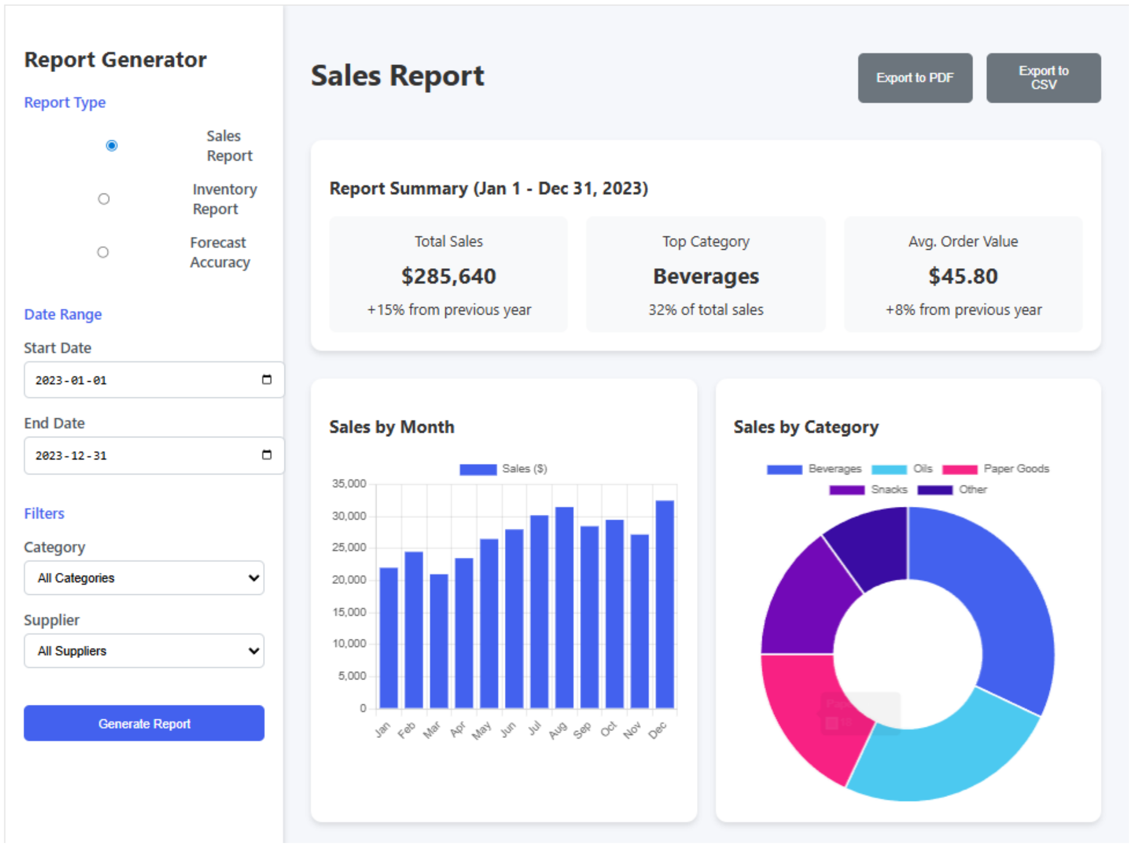
Stock Alerts

23 active alerts

Filter by: All Alerts | Critical | Warning | Info | All Categories

Organic Coffee Beans	Current Stock: 4 units	Reorder Point: 15 units	Supplier Lead Time: 3 days
Avocado Oil	Current Stock: 7 units	Reorder Point: 12 units	Supplier Lead Time: 5 days
Paper Towels	Current Stock: 9 units	Reorder Point: 15 units	Supplier Lead Time: 2 days
Organic Honey	Current Stock: 14 units	Reorder Point: 15 units	Supplier Lead Time: 7 days

- **Reporting & Analytics Hub**



Report Generator

Report Type

[✓ Sales Report]
Inventory Report
Forecast Accuracy

Date Range

Start Date: 2023-01-01

End Date: 2023-12-31

Filters

Category: All Categories

Supplier: All Suppliers

Sales Report

Report Summary (Jan 1 - Dec 31, 2023)

Total Sales	Top Category	Avg. Order Value
$285,640	Beverages	$45.80
+15% from previous year	32% of total sales	+8% from previous year

Sales by Month

Sales by Category

System Testing

System Testing is a critical phase in the SmartShop development lifecycle where the complete, integrated system is evaluated against specified requirements. This phase ensures that all components work together as intended and that the system meets both functional and non-functional requirements before deployment.

4.1 Objectives of System Testing

- Verify that the entire system functions according to design specifications
- Ensure all integrated components work together seamlessly
- Validate system performance under various conditions
- Confirm data integrity and security measures
- Verify usability and user experience meet expectations

4.2 Types of Testing

1. Functional Testing

- Inventory Management Testing: Verify stock level updates, reorder calculations, and alert generation
- Forecast Accuracy Testing: Validate AI prediction accuracy against historical data
- User Role Testing: Ensure proper access controls for different user types (owner, staff)
- Report Generation Testing: Confirm all reports generate correctly with accurate data

2. Non-Functional Testing

- Performance Testing: Test system response under heavy load and data volume
- Security Testing: Validate data encryption, authentication, and authorization mechanisms
- Usability Testing: Ensure the interface is intuitive for non-technical users
- Compatibility Testing: Verify functionality across different browsers and devices

3. AI-Specific Testing

- Model Validation Testing: Verify forecasting algorithm accuracy with various datasets
- Data Pipeline Testing: Ensure clean data flow from input to prediction output
- Edge Case Testing**:** Test system behavior with unusual data patterns or missing information

4.3 Testing Process:

- Test Planning: Develop detailed test cases covering all requirements
- Test Environment Setup: Configure testing environment mirroring production
- Test Execution: Run comprehensive test suites and document results
- Defect Reporting: Log and prioritize identified issues for resolution
- Regression Testing: Verify fixes don't introduce new problems
- Performance Benchmarking: Establish baseline performance metrics
- User Acceptance Testing: Involve actual users to validate system meets business needs

4.4 Testing Deliverables

- Test plan and test cases documentation
- Defect reports and resolution status
- Performance benchmark reports
- Security assessment report
- User acceptance testing sign-off
- Final test summary report

System Testing ensures that the system will perform reliably in real-world conditions, providing accurate forecasts and reliable inventory management capabilities that meet user expectations and business requirements.

Conclusion

The SmartShop project presents a comprehensive and innovative solution to the critical inventory management challenges faced by modern retail businesses. By leveraging advanced machine learning algorithms, the system transforms traditional, reactive inventory management into a proactive, data-driven process that significantly reduces both stock-outs and overstock situations.

Throughout this report, we have detailed how SmartShop addresses core business needs through its intelligent forecasting capabilities, user-friendly dashboard interface, and automated alert system. The system's ability to analyze historical sales data, predict future demand patterns, and calculate optimal reorder points represents a significant advancement over manual inventory management methods.

The implementation of SmartShop will deliver substantial benefits to retail operations, including improved inventory turnover, reduced carrying costs, enhanced cash flow management, and increased customer satisfaction through better product availability. The system's design prioritizes usability, ensuring that both shop owners and staff can effectively utilize its capabilities with minimal training.

The technical architecture combines robust backend processing with an intuitive frontend interface, creating a scalable solution that can grow with business needs. The selection of appropriate AI methodologies, including Facebook Prophet for demand forecasting and dynamic reorder point calculation, ensures accurate and actionable insights.

As we move forward with development and implementation, SmartShop is positioned to become an indispensable tool for retail businesses seeking to optimize their inventory management processes. The system not only addresses current operational challenges but also provides a foundation for future enhancements, such as integration with additional data sources, advanced predictive analytics, and expanded reporting capabilities.

In conclusion, SmartShop represents a significant step forward in inventory management technology, offering retailers a powerful tool to increase efficiency, reduce costs, and improve overall business performance through the practical application of artificial intelligence and data analytics.