AI Inventory Demand Forecaster – Product Requirements Document (PRD)

1. Product Overview

The AI Inventory Demand Forecaster is a predictive system that helps businesses accurately forecast product demand using AI/ML models.
It prevents stockouts, reduces overstock, and improves supply chain efficiency.

This tool will generate:

Demand predictions (daily/weekly/monthly)

Inventory restock alerts

Visual charts for trends

Downloadable reports

Insights on fast-moving and slow-moving items

This will be built in a 7-day sprint as part of the Afritech 30-Day AI Challenge.

2. Problem Statement

Small and mid-sized businesses struggle with:

Overstock (wasted money and storage cost)

Stockouts (lost sales and angry customers)

No proper data to know what sells and when

Manual inventory tracking

No forecast to guide restock planning

These inefficiencies reduce revenue and weaken decision-making.

3. Goal of the Product

To help businesses make data-driven decisions by forecasting product demand using AI, giving them clarity on what to stock, when to stock it, and how much to stock automatically.

4. Target Users

Primary Users:

Retail store owners

E-commerce sellers

Supermarkets

Pharmacy owners

Small business operators

Inventory managers

Secondary Users:
Supply chain teams

Data analysts

Product managers

5. Key Features (MVP for Week 1)
1. Data Upload Module

Users can upload CSV/Excel files containing past sales data.

2. Data Validation
System checks if data has:

Date

Product name

Quantity sold

Price (optional)

3. AI Demand Forecasting Engine
Uses Time-Series models (Prophet, ARIMA, LSTM or AutoML)

Predicts sales for next 7–30 days

Identifies demand patterns and seasonality

4. Analytics Dashboard
Displays:

Forecast chart (line graph)

Product demand ranking

Restock recommendations

Trend analysis

5. Downloadable Reports
Export predictions to PDF or CSV.

6. Basic UI
Simple clean interface for Week 1 prototype.

6. Unique Value Proposition
This forecaster is:

Simple enough for non-technical business owners

Powered by AI for accuracy

Fast–predicts within seconds

Practical– insights are actionable

Lightweight– can be used on any device

Tailored for African SMBs (unique advantage)

7. Technical Requirements

Frontend (Prototype Week 1)

Streamlit or Gradio

Python

Basic HTML/CSS (optional)

Backend / AI Engine

Python

Pandas

NumPy

Scikit-learn

Facebook Prophet or ARIMA or AutoML (e.g., skforecast)

Matplotlib/Plotly for charts

Data Storage (Prototype)

No database required for MVP.
Use temporary uploaded files only.

Integration (optional future)

Google Sheets API

POS systems

Cloud databases

8. Success Metrics

MVP (Week 1):

Working forecast model

At least 1 prediction chart displayed

CSV upload that runs smoothly

Clear product insights generated

Overall:
90% correct forecast trends

Reduce stockouts by at least 25%

Reduce overstock by 20%

Weekly active users increase

9. Assumptions

Users already have historical sales data

Data is clean or can be cleaned automatically

Users can upload CSV files

MVP will be web-based, not mobile

10. Open Questions

Should the forecast be daily, weekly, or monthly?

Should we add notifications (email/SMS)?

Do we need a POS integration?

Should the system support multiple store locations?

11. 7-Day Build Plan

Day 1: Requirements + Brainstorming

Define scope, assign tasks, research models, design simple UI.

Day 2: Data + Research

Collect sample datasets, study similar tools, prepare preprocessing.

Day 3: Model Planning

Choose forecasting model, write pseudo-code, draft UI.

Day 4: Build Prototype

Implement model + upload system + basic UI.

Day 5: Testing

Try different datasets, fix bugs, refine charts.

Day 6: Polish Demo

Add better UI, prepare recording, improve visuals.

Day 7: Final Demo

Share prototype + PRD + GitHub link.

12. Sample Data Structure

Date	name	quantity	sold	price
2023-01-05	Rice	5kg	12	5000
2023-01-06	Rice	5kg	9	5000

13. References

Walmart Kaggle Sales Dataset

Time Series Forecasting Playbook

Facebook Prophet Documentation

14. Final Deliverables

By the end of Week 1,we would have:

A working AI Demand Forecaster prototype

A clean PRD.md

A GitHub repo containing code and documentation

A demo video for the community

A LinkedIn update tracking progress daily