YIELD SYNC

MOBILE APPLICATION FOR SMARTER FARMING

Project Id – 25-26J-169

Project Proposal Report

ANANDI L.B.I

IT22296764

BSc. (Hons) Degree in Information Technology Specialized in

Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology Sri

Lanka

August 2025

PRICE PREDICTION AND DEMAND FORECASTING

MOBILE APPLICATION FOR SMARTER FARMING

Project Proposal Report

Price Prediction and Demand Forecasting: Mobile Application for Smarter Farming

ANANDI L.B.I

IT22296764

BSc. (Hons) Degree in Information Technology Specialized in

Information Technology

Department of Information Technology

Sri Lanka Institute of Information

Technology Sri Lanka

August 2025

Declaration of the Candidate & Supervisor

To my knowledge, the proposal I herewith submit is to the best of my capability and has not been presented for any other degree or professional program. All work used herein is my own; where other people's work has been quoted, it has been duly acknowledged and referenced

Group Member Name	Student ID	Signature
Anandi L B I	IT22296764	

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

Name of supervisor: Mrs. Sanvitha Kasthuriarachchi

Name of co-supervisor: Mrs. Sasini Hathurusinghe

	Name	Signature
Supervisor	Mrs. Sanvitha Kasthuriarachchi	

Abstract

Sri Lankan crop markets are shaped by Maha/Yala seasonality, festival-driven demand (Sinhala & Tamil New Year, Vesak), and frequent weather shocks. Smallholder farmers often sell without timely market signals, leading to mistimed sales and sub-optimal prices. This study proposes Yeild Sync, a Sri Lanka–tailored crop price and demand forecasting system that converts fragmented market and context data into short-term, actionable insights for when and where to sell.

The methodology integrates multi-year market prices and sales volumes by crop–market with district production, seasonal calendars, festival/holiday dates, and weather variables. After data cleaning and feature engineering (lags, rolling trends/volatility, season/harvest and festival flags, weather anomalies), we benchmark classical, machine-learning, and deep sequence models (e.g., SARIMAX/Prophet, gradient-boosted trees, LSTM/TFT). The system produces multi-horizon forecasts at 1, 2, 4, 8, and 12 weeks with calibrated prediction intervals and an optional SELL/HOLD signal derived from transparent, season-aware rules. Outputs are delivered via a mobile-first interface (Sinhala/Tamil/English) with SMS/USSD for low-connectivity settings.

Expected outcomes include improved selling timing and realized prices, clearer visibility of near-term demand trends (rising/steady/falling), and reduced wastage for perishable crops (e.g. beet root, radish). The study also targets reproducible pipelines and honest uncertainty, enabling advisors and co-ops to trust and operationalize the forecasts. We conclude that localized, explainable price and demand forecasting can strengthen market decisions and income stability for Sri Lankan smallholders.

Key words: Crop price forecasting, demand forecasting, time-series, seasonality, Sri Lanka, festivals, agricultural markets, decision support, multi-horizon prediction, uncertainty intervals.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

01.INTRODUCTION

Sri Lanka's crop markets are shaped by pronounced seasonality (the Maha and Yala seasons), festival-driven demand surges (notably Sinhala & Tamil New Year and Vesak), and frequent weather shocks. Smallholder farmers often lack timely, actionable market signals, which leads to mistimed sales and sub-optimal prices

Against this backdrop, the project develops a Crop Price & Demand Forecasting and Decision-Alert System. It integrates multi-year historical prices, market demand/sales volumes, district-level production, seasonal planting/harvesting calendars, weather variables (rainfall, temperature, humidity, extreme events), and major holidays/festivals. For each crop market pair, the system generates rolling forecasts 1, 2, 4, 8, and 12 weeks ahead, complete with prediction intervals. A decision layer then translates these forecasts into clear sell/hold alerts using threshold rules and a simple expected-profit model that accounts for storage, spoilage, and transport costs

Outputs are delivered through a lightweight web/mobile dashboard (Sinhala, Tamil, and English) with SMS/USSD fallback for low-connectivity areas. The overarching aim is to help smallholders time their sales better, especially around harvest peaks and festival-driven demand so they can reduce avoidable losses and improve income.

1.1 Background & Literature Survey

Forecasting Approaches

Agricultural markets are characterized by the number of ups and downs, hinging on many things ranging from seasonal changes, weather, and variations in the world economy. This thus means that good models for predicting prices are what can make farmers able to make smart choices. The Auto Regressive Integrated Moving Average (ARIMA) model is one of the most common models used in this area. In agricultural price forecasting, classic time-series models Naïve and Seasonal-Naïve, ARIMA/SARIMA, ETS/TBATS, and Prophet remain the go-to starting point. They capture trends and seasonality with clear, transparent components and set a reliable baseline. When relationships become more complex, machine-learning regressors and SVR learn nonlinear links between past prices and external drivers (e.g., rainfall, holidays, harvest timing). For multi-horizon predictions and richer covariates, deep learning methods LSTM/GRU, Temporal Convolutional Networks (TCN), N-BEATS, Temporal Fusion Transformer, and DeepAR model long-range patterns and can produce probabilistic forecasts (e.g., prediction intervals), which are crucial for risk-aware decisions.

In the predictive analytics stream, Long Short-Term Memory (LSTM) networks enhance ARIMA. An LSTM is an RNN type suitable for sequences of data in cases when there are long-term dependencies. This is very important to agriculture because patterns exhibited in price levels are most often seasonally oriented. Other studies have recommended that LSTM networks outperform classical models like ARIMA when the time series has complex and long-range patterns.[4]

Demand Modeling

Demand (sales volume) behaves differently from price and often shows bursts around festivals or during harvest peaks. Count models such as Poisson and Negative Binomial including zero-inflated variants for many zeros offer a principled way to model such data. Where richer features are available, gradient-boosting methods and sequence models handle nonlinearities and time dynamics effectively. Simple price–demand elasticity formulations add interpretability: they show how sensitive buyers are to price changes and help translate forecasts into practical guidance.

External Signals & Features

Prices and demand in Sri Lanka are strongly shaped by maha/yala seasons, harvest windows, weather conditions and anomalies (and their lags), district-level production, and event calendars (public holidays and festivals). Encoding these as features season flags, holiday windows, lagged rainfall/temperature, production indicators typically improves forecast accuracy. These signals help the model distinguish, for example, a temporary weather shock from a predictable seasonal dip, and to anticipate demand spikes around Sinhala & Tamil New Year or Vesak.

Evaluation and Actionability

Sound evaluation uses leakage-free rolling or expanding-origin cross-validation that respects time order, along with multi-horizon metrics such as MAE, RMSE, MAPE/sMAPE, and WAPE. For probabilistic models, interval coverage and sharpness indicate how trustworthy the uncertainty estimates are. A notable gap in prior studies is the final mile: turning forecasts into clear decision rules tied to profit , and low-bandwidth channels like SMS. This project explicitly addresses that gap, pairing forecasts with transparent rules and farmer-friendly delivery.

1.2 Research Gap

Most tools farmers see today look at one thing at a time usually past prices and ignore other signals like weather, local production, the Maha/Yala cycle, and festival effects. They also predict price but rarely predict demand alongside it, and they stop at accuracy scores instead of giving clear, profit-aware SELL/HOLD advice. Uncertainty is often vague: a single "90% band" for everyone, not a calibrated band for each market and each forecast horizon that farmers can trust. Testing is sometimes flawed (future data leaking into training), and delivery assumes good internet and English, leaving out users who need SMS/USSD. Finally, the data plumbing is weak missing values, outliers, unit mismatches, and regime shifts aren't handled consistently. Our project tackles these gaps head-on with a unified, multi-source pipeline; joint price-and-demand forecasting; per-market, per-horizon calibrated intervals; profit-focused, hysteresis-based decisions; leakage-safe evaluation; and a tri-lingual interface that works even on basic phones.

Table 1. 1 Research Gap Table

Aspect	Common practice	Identified Gap	Proposed Solution
Data Integration	Use a single stream (usually past prices)	Ignore weather, production, seasonality, and festivals	Unified multi-source pipeline: prices, demand, district production, weather, Maha/Yala calendar, festivals
Calendar & Festivals	Binary holiday dummies; coarse "season" flags	No lead-up/lag effects; weak season phase info	Numeric days-to-festival feature + Maha/Yala phase
Targets (price vs demand)	Model price only	Misses volume spikes and supply effects	Joint forecasting of price and demand (arrivals/sales)
Evaluation	Random splits, leakage-prone CV	Overstated accuracy; not horizon-aware	Rolling/expanding-origin backtests; horizon-wise metrics (MAE/RMSE/WAPE), pinball/CRPS, coverage & width

1.3 Research Problem

Smallholder farmers in Sri Lanka struggle to time sales for rice, beetroot, radish, red onion. Prices for these crops swing with the Maha/Yala seasons, festival demand (Sinhala & Tamil New Year, Vesak), and weather shocks. Vegetables like red onion, radish, and beetroot are highly perishable, while rice store better, but even they face sharp, district-level price moves. The signals farmers need are scattered across bulletins and websites and rarely translated into simple, trustworthy guidance.

This project frames the research problem as a decision problem, not just a forecasting one. For each crop market pair, it aims to roll price and demand forecasts for 1, 2, 4, 8, and 12 weeks ahead with prediction intervals, using multi-source data: historical market prices, sales volumes, district production, Maha/Yala calendars, weather (rainfall, temperature, humidity, extremes), and holiday/festival effects.

A decision layer then converts forecasts into clear sell/hold recommendations. Rules are risk-aware and crop-specific combining expected future price, interval uncertainty, storage and transport costs, and expected spoilage. Recommendations are delivered via a lightweight web/mobile dashboard in Sinhala, with SMS for low-connectivity users, and include short "why this alert" explanations.

Scope is limited to forecasting and alerting for the seven listed crops across major wholesale/retail markets. Agronomic advice (e.g., fertilizer plans) and on-farm logistics are out of scope. Success will be judged by forecast accuracy (interval coverage), the quality of sell/hold decisions (precision and expected income uplift versus simple baselines), and operational reliability (low latency, high uptime, timely SMS delivery). By unifying fragmented data, accounting for crop-specific perishability, and bridging predictions to actionable decisions, the study targets the core pain point: helping Sri Lankan farmers choose when to sell rice, beetroot, radish and red onion to reduce losses and improve income.

02.OBJECTIVES

2.1 Main Objective

The main function is to turn scattered market signals into clear, timely selling guidance for farmers. It continuously gathers multi-year prices, recent sales volumes, the Maha/Yala calendar, weather, and festival days and, for each crop-market pair (rice, beetroot, radish, red onion), predicts prices and demand for the next 1, 2, 4, 8, and 12 weeks with quantified uncertainty. A decision layer then converts those forecasts into a simple, crop-aware sell/hold recommendation by weighing expected future price against real costs and risks (storage, transport, spoilage) and the farmer's minimum acceptable price. Each alert includes a short "why this alert" explanation (e.g., harvest peak, holiday demand, rainfall pattern) and is delivered in Sinhala via a lightweight web/mobile interface with SMS fallback for low-connectivity areas.

2.2 Specific Objectives

1) **Development of Integrated Forecasting Models:** Design and benchmark multi-horizon models (e.g., Seasonal Naïve, SARIMAX/Prophet, LSTM) to predict prices and demand 1, 2, 4, 8, and 12 weeks ahead for each crop–market pair.

2) **Data Collection and Integration:** Gather and harmonize multi-year market prices, sales volumes, district production, Maha/Yala calendars, weather (rainfall /temperature /humidity /extremes), and holiday/festival calendars; build a repeatable ETL and feature store.

3) **Data Quality & Exploratory Analysis:** Detect and treat missing values/outliers, standardize units and frequencies, and quantify seasonality, holiday effects, weather sensitivity, and cross-market linkages for the seven crops.

4) **Feature Engineering:** Create informative predictors lags, rolling statistics, momentum, weather lags, Maha/Yala and festival windows, cross-market spreads and define perishability profiles per crop.

5) **Decision & Alert Engine Implementation:** Implement risk-aware rules with hysteresis and user-set minimum prices; generate concise "why this alert" explanations referencing key drivers (seasonality, holidays, weather).

6) **System Integration and User Interface Development:** Develop a lightweight, (Sinhala) web dashboard with clear charts and summaries, plus SMS fallback for low-connectivity users.

03.METHODOLOGY

Yeild Sync fuses multi-year daily/weekly market prices and sales volumes for each target crop–market pair (rice, beetroot, radish, red onion), district production totals, the Maha/Yala planting–harvest calendar, festival/holiday dates (e.g., Sinhala & Tamil New Year, Vesak), and weather observations (rainfall, temperature, humidity, extreme events) aligned to producing districts.

3.1 Development of Integrated Forecasting Models

- **Data Sources:** Gather multi-year daily/weekly prices and sales volumes for each crop–market, plus district production, Maha/Yala calendars, festival/holiday dates (Sinhala & Tamil New Year, Vesak), and weather (rainfall, temperature, humidity, extremes). These are the core drivers of Sri Lankan market movement.

   Variables: Capture crop/variety, market, district, date, price type (wholesale/retail), volume, production, season/harvest flags, festival indicators, weather station/levels, and data provenance.

- **Data Acquisition:** Use scheduled APIs or scrapers for price/demand bulletins weather APIs for daily observations; static calendars for seasons/festivals. Allow CSV uploads via a template auto geo-tag markets/district.

3.2 Preprocessing

- **Data Cleaning:** Remove duplicates and malformed rows; reconcile crop/market codes; validate ranges (e.g., no negative prices).

- **Normalization & alignment:** Standardize to LKR/kg and MT; align mixed daily/weekly series without leaking future data; optionally deflate prices for long spans.

3.3 Feature Extraction

We convert raw histories into compact signals the models can learn from. This includes lags of price and demand (yesterday/last week/last month), rolling averages and volatility over 7/14/28-day windows, and calendar flags for Maha/Yala stages and harvest windows. To reflect short, sharp demand changes, we add festival lead/lag windows (e.g., $-21/-14/-7/0$ days). Weather matters too, so we include recent levels and anomalies (and their lags). Finally, market/district identifiers (or simple embeddings) capture local effects, and a light perishability marker note crops that tend to lose value quickly.

3.4 Diagram Model Development

We treat forecasting as a multi-horizon task, producing price and demand predictions for 1, 2, 4, 8, and 12 weeks. To stay grounded and fair, we benchmark three families.

- **Price Prediction Model:**
  Develop and train models such as ARIMA/SARIMAX and LSTM/GRU (optionally Prophet/TFT) to forecast market prices for 1, 2, 4, 8, and 12 weeks. Inputs include historical prices, Maha/Yala season flags, festival effects, weather, production, and cross-market spreads.

- **Demand Forecasting Model:**
  Develop and train models such as ARIMA/SARIMAX and LSTM/GRU (optionally Prophet/TFT) to forecast market prices for 1, 2, 4, 8, and 12 weeks. Inputs include historical prices, Maha/Yala season flags, festival effects, weather, production, and cross-market spreads.

- **Integration**

  Yeild Sync unifies price and demand models behind a single inference path. For each crop–market, it pulls the latest features, routes them to the current champion model, and calibrates prediction intervals so stated confidence matches reality. It returns consistent package point forecasts, 10/50/90 bands, and demand trend with model version and timestamp. An optional step converts these outputs into a transparent SELL/HOLD recommendation; responses are cached for speed and fall back to a seasonal baseline if a feed degrades.

3.5 System Implementation

- **System Design**

  A web-first pipeline ingests prices, sales, production, Maha/Yala calendars, weather, and festivals into a feature store, then serves 1/2/4/8/12-week price & demand forecasts with calibrated intervals via a low-latency API. An optional decision engine applies a seasonal rule plus a simple storage/spoilage check to produce SELL/HOLD; the stack uses champion–challenger models, scheduled retraining, caching, RBAC/encryption, and Sinhala/Tamil/English support.

- **User Interface**

  Farmers enter crop/market, harvest window, quantity, and a minimum price; the UI shows forecast charts with 10/50/90 bands and a compact demand trend. When enabled, it adds a SELL/HOLD card with a one-line reason and suggested sell window, plus seasonal/festival context mobile-first, multilingual, and able to show the last forecast offline.

3.6 Evaluation and Validation

- **Model Evaluation**

  We will use rolling-origin cross-validation per crop–market at 1/2/4/8/12-week horizons, benchmarked against a seasonal-naïve baseline. Reported metrics include MAE, RMSE, sMAPE/WAPE (and Poisson/NegBin deviance for counts) plus interval

  quality via PICP and sharpness.Stress tests cover festival weeks, harvest peaks, weather extremes, and missing days; a champion–challenger process and drift monitors control promotion and retraining.

- **User Testing**

  Short sessions with farmers (Sinhala/Tamil) will assess whether they can read the forecast bands, identify the sell window, and understand a sample SMS/USSD alert. We will track task success, time-to-answer, and clarity of the one-line "why," then refine wording and visuals based on feedback.

- **Impact Analysis**

  Using a pre–post or difference-in-differences design, we will measure realized price uplift, income uplift per kg/batch (accounting for storage/spoilage), timing adherence to recommended windows, and alert precision/adoption with special focus around New Year and Vesak. Results will include confidence intervals and sensitivity analyses across crops and markets.

Backend

1. **Server Infrastructure**

- Web Server: Nginx, Apache
- Application Server: Python (Flask) ,Node.js, Django
- Database Server: PostgreSQL, MongoDB

2. Real-Time Data Processing

- Streaming Platform: Apache Kafka, RabbitMQ
- Processing Engine: Apache Flink, Spark Streaming

3. APIs

- REST API: Forecast retrieval, decision/alert generation, model/feature metadata.
- WebSocket API: Live dashboard updates and admin monitors
- SMS/USSD Gateway: Outbound notifications and simple menu flows

Algorithm

4. Price Prediction

- Feature Extraction: Lagged prices (t−1/−7/−14/−28), rolling mean/volatility (7/14/28), Maha/Yala & harvest flags, festival lead windows (−21/−14/−7/0), weather levels/anomalies + short lags, cross-market price spreads, markets.
- Modeling: Train SARIMAX/Prophet, LightGBM/XGBoost/CatBoost, and LSTM/GRU (use TFT/N-BEATS where history is deep). Use direct (per-horizon) and/or multi-output setups; select per crop–market via rolling-origin CV.

5. Demand Forecasting

- Feature Engineering: Lagged demand and rolling sums (7/14/28) price level/momentum, season/harvest flags, festival lead/lag, weather levels/anomalies & lags, district production index, market/district IDs, zero/closure indicators.
- Modeling: Poisson/Negative-Binomial (zero-inflated if needed),LightGBM/XGBoost (count/quantile loss), and LSTM/TFT for multi-horizon demand. Train with rolling-origin CV; track MAE/WAPE and count deviance.

Database

- **Primary Databases**: MongoDB PostgreSQL/TimescaleDB: Canonical time-series for prices, sales/demand, weather, plus reference tables (crops, markets, districts, Maha/Yala calendars, festivals). Partition by date/market, enforce units (LKR/kg) and keys, and index on (crop, market, date) for fast reads.

- **Feature Store** : Materialized tables/views with lags, rolling stats, season/harvest & festival flags, weather anomalies. Snapshot features at each training cut-off so training and inference use identical definitions.

- **Forecast Store**: Per crop–market–horizon records with point forecasts + q10/q50/q90, model version, timestamp, and quality flags (coverage, data freshness). Optimized for quick API retrieval.

This methodology delivers a Sri Lanka–specific price and demand forecasting system that fuses multi-year prices/sales with Maha/Yala, festival, and weather signals to produce 1/2/4/8/12-week forecasts with calibrated intervals and an optional SELL/HOLD alert. Success hinges on clean, well-tracked data; rolling time-series cross-validation with champion challenger model management; reliable uncertainty calibration and fallbacks to seasonal baselines; and a mobile-first, Sinhala/Tamil/English UI that explains why each recommendation is made.

The Processing & Recommendation Engine gets this and makes recommendations with the help of machine learning models, crop nutrient demands and kept on record agricultural guidelines. Data exchange between the application and the database (MongoDB), that hosts historical data, fertilizer recommendations and soil records is managed by the backend server.

3.7 System Overview Diagram

Yeild Sync transforms fragmented market, seasonal, and weather signals into multi-horizon price and demand forecasts (1/2/4/8/12 weeks) and a simple SELL/HOLD recommendation with a clear reason and confidence.

It ingests external data (historical prices and sales, district production, seasonal and festival calendars, weather) plus farmer context (crops/markets, location and harvest window, quantity, optional minimum price and storage/spoilage). An ingestion layer cleans and normalizes data (LKR/kg) and a feature store builds lags, rolling stats, season/harvest and festival flags, weather levels/lags, and market/district effects. Models are trained with rolling-origin CV registered and served via a low-latency service that also returns prediction intervals. An optional decision layer converts forecasts into SELL/HOLD using a seasonal rule and a simple profit-aware check with a risk buffer and hysteresis
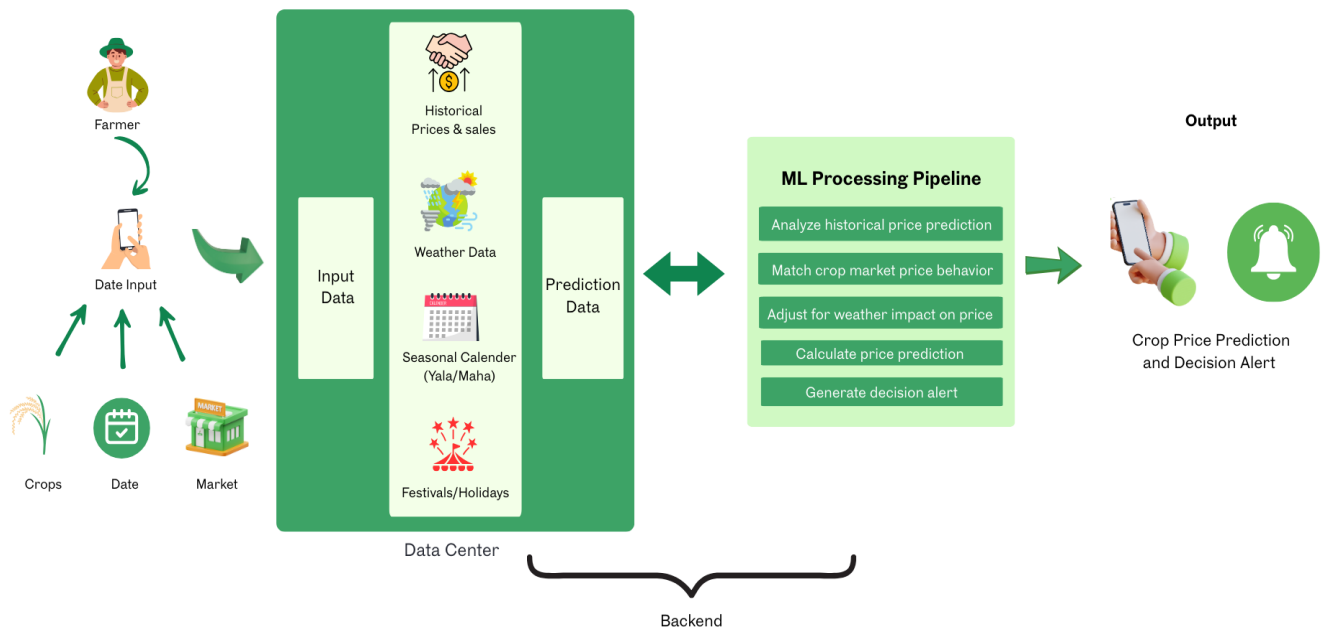

Figure 3. 1 System Overview Diagram of the Price Prediction and Demand Forecasting Component.

04.Project Requirements

For a software solution focused on crop price & demand forecasting with an optional sell/hold alert for Sri Lankan markets, the requirements are:

4.1. Functional Requirements

1. Price Prediction:

- system must generate multi-horizon price forecasts for each crop–market (1, 2, 4, 8, 12 weeks) using models such as ARIMA/SARIMAX/Prophet, LightGBM/XGBoost, and LSTM/GRU.
- Provide farmers with real-time price forecasts to aid in decision-making.

2. Demand Forecasting

- Predict sales/demand over the same horizons with count-aware models (e.g., Poisson/NegBin, GBMs, LSTM/TFT).
- Surface a simple trend signal (rising / steady / falling) and intervals

3. Decision-Alert Engine

- Convert forecasts into a SELL/HOLD recommendation using a seasonal k·σ threshold and risk buffer (hysteresis).
- Show a one line rationale (e.g.seasonal mean; Vesak in 7 days") and allow user settings: minimum price, risk level, quiet hours.

4.2. Non-Functional Requirements

1. Performance

- Ensure the system processes large datasets and provides feedback with minimal latency, offering real-time predictions and recommendations.
- Optimize the system to handle multiple simultaneous users efficiently.

2. Scalability

- Design the system to support scalability, allowing it to handle increased data volumes and a growing number of users, particularly during peak agricultural seasons.

3. Security

- Implement robust data encryption and access controls to protect user data, ensuring compliance with data protection regulations.
- Regularly update security protocols to guard against emerging threats.

4.3. User Requirements

1. Farmers

- Mobile first (Sinhala) UI to enter crops, markets, harvest window; save preferences and opt in to SMS alerts.
- See 1/2/4/8/12-week price forecasts with bands, a demand trend, and an optional SELL/HOLD with a one-line reason; view/compare selected markets and cache the last forecast offline.

2. Agricultural Advisors: -

- Advanced dashboard with market comparisons, seasonal baseline vs. forecast, confidence metrics (coverage/interval width), and alert history for supported farmers.

4.4. System Requirements

1. **Hardware**

   - **Servers**: High-performance servers capable of handling data processing, model training, and real-time analytics.
   - **User Devices**: Access through various devices including smartphones, tablets, and desktop computers.

2. **Software**

   - **Operating System:** Linux or Windows Server for backend; cross- platform support for client devices.
   - **Database Systems**: PostgreSQL or MySQL for relational data management; MongoDB for handling unstructured data and real- time analytics.
   - **Programming Languages**: Python, Java, or Node.js for backend development; JavaScript and React for frontend development.
   - **Machine Learning Frameworks**: TensorFlow, PyTorch, or Keras for developing and deploying predictive models.
   - **Streaming and Processing Tools**: Apache Kafka or RabbitMQ for real-time data processing; Apache Flink or Spark Streaming for analyzing incoming data.
   - **Security Tools:** Implement TLS/SSL for data encryption and OAuth for secure authentication.

4.5 Use Cases

Use cases describe the interactions between users for various tasks:

Use Case 1 – Crop & Market Context

- **Actors:** Farmer
- **Description:** The farmer selects crop and market, sets the location/harvest window and quantity, and (optionally) a minimum acceptable price; chooses language (Sinhala/Tamil/English) and alert preferences.
- **Outcome:** The request is saved; features are refreshed for the chosen crop market pairs.

Use Case 2 – View Price & Demand Forecasts

- **Actors:** Farmer
- **Description:** The system returns 1/2/4/8/12-week price forecasts with prediction bands (e.g., 10/50/90) and a demand trend indicator (rising/steady/falling), plus seasonal/festival context.
- **Outcome:** The farmer sees clear charts and a timestamped summary for planning.

Use Case 3 –Receive SELL/HOLD Recommendation

- **Actors:** Farmer
- **Description:** Based on forecasts and uncertainty, the decision engine applies a seasonal rule (and, if provided, simple storage/spoilage inputs) to generate SELL/HOLD with a one-line reason. Alerts are delivered in-app and via SMS respecting quiet hours.
- **Outcome:** The farmer gets a timely, actionable recommendation and suggested sell window.

Use Case 4 – Compare Markets & Pick a Sell Window

- **Actors:** Farmer
- **Description:** The farmer compares selected markets by expected price (with intervals) and inspects the calendar to choose a favorable week within the forecast horizon.
- **Outcome:** A chosen market and tentative sell date are saved for reminders and follow-up.

4.6 Test Cases

Test Case 1 – Data Input Validation

- **Objective:** Ensure the input form validates crop, market, harvest window/date, quantity, and (optional) minimum price and alert preferences; prevent future leakage dates.

- **Expected Result:** The system blocks submission with clear messages for invalid fields (e.g., unknown market, negative quantity, reversed dates), highlights errors inline, and saves only valid requests.

Test Case 2 – Forecast & Interval Accuracy

- **Objective:** Verify price and demand forecasts on a holdout period using rolling origin backtests across 1/2/4/8/12 week horizons.

- **Expected Result:** Accuracy meets targets (e.g., MAE/RMSE/WAPE thresholds by crop market), and prediction intervals achieve ~90–95% PICP with reasonable sharpness.

Test Case 3 – Decision Alert & Notification Delivery

- **Objective:** Ensure the SELL/HOLD rule triggers correctly and alerts are delivered on schedule, respecting quiet hours.

- **Expected Result:** The farmer receives a single, deduplicated alert (SMS) containing crop, market, horizon, timestamp, and link to the forecast.

05.WORK BREAKDOWN CHART

1) Requirements Analysis

We will speak with farmers and market officers to understand when and where they sell, then assemble Sri Lanka–specific data (prices/sales, Maha Yala calendars, festivals, weather, district production). We'll set clear success targets forecast error, interval coverage, alert precision, and income uplift and lock them into the project charter.

2) System Design

we will design simple but reliable architecture: raw data, cleaned tables, feature store, forecast store, backed by a model registry and low-latency Forecast/Decision APIs. Wireframes for a mobile first interface will prioritize clarity for quick, on-farm use, while security, monitoring, caching, and fallback behavior are specified up front.

3) Development

We will build ETL pipelines with data-quality checks, engineer features that reflect local dynamics (lags, rolling trends, Maha/Yala and festival flags, weather anomalies), and train price and demand models with rolling backtests. Prediction intervals will be calibrated, and an optional SELL/HOLD rule will be implemented with clear rationale text.

4) Integration & Refinement

Data and models will be stitched into one serving path so forecasts return in $\leq$ **2 s**. We'll tune thresholds/hysteresis to avoid flip-flops, add market comparisons and seasonal context, and polish the reason strings.

5) Testing

Unit/integration tests will cover ETL, features, APIs, and permissions model tests will verify accuracy and interval coverage across 1/2/4/8/12-week horizons, including stress tests for harvest peaks, festival weeks, and missing feeds. Small farmer pilots will confirm the charts, bands, and SMS messages are easy to use.

6) **Deployment & Maintenance**

We'll deploy the cloud stack with live monitoring (latency, uptime, drift, coverage), schedule retraining, and set up data-freshness alerts. Farmers and advisors will be onboarded, feedback will drive updates, and safe rollback plus a seasonal-naïve fallback will ensure continuity.
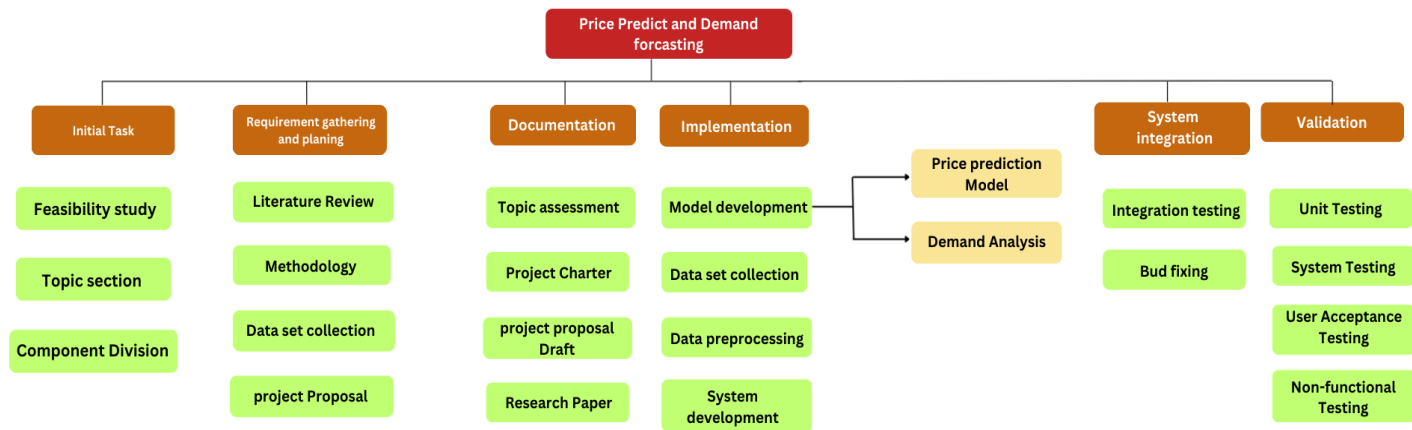

Figure 6. 1 Work Breakdown Chart.

06.GANTT CHART

No	Assessment (Milestones)	2025-2026															
		3 (25)	4 (25)	5 (25)	6 (25)	7 (25)	8 (25)	9 (25)	10 (25)	11 (25)	12 (25)	1 (26)	2 (26)	3 (26)	4 (26)	5 (26)	6 (26)
1	**Project Discussion Workshop**	█															
2	**Topic Evaluation**		█	█	█	█											
2.1	Select a Topic			█													
2.2	Select a supervisor		█	█													
2.3	Topic Evaluation Form Submission				█	█											
3	**Project Proposal Report**																
3.1	Project Proposal Presentation							█									
3.2	Create Project Proposal (Individual)						█	█									
3.3	Create Project Proposal (Group)							█									
4	**Develop the System**																
4.1	Identifying Functions						█	█									
4.2	Database Designing								█								
4.3	Implementation								█	█	█						
4.4	Unit testing																
4.5	Integration Testing																
5	**Progress Presentation - I**																
5.1	Project Status document										█						
5.2	Create Presentation Document										█						
5.3	Progress Presentation – I (50%)										█						
6	**Research Paper**																
6.1	Create the Research Paper							█	█	█	█	█	█	█			
7	**Progress Presentation - II**												█	█			
7.1	Create Presentation Document												█	█			
7.2	Progress presentation – II (90%)													█			
8	**Final Report Submission**																
8.1	Final Report Submission													█	█	█	█
8.2	Application Assessment																
8.3	Project Status Document																
8.4	Student Logbook						█	█	█	█	█	█	█	█	█	█	█
9	**Final Presentation & Viva**																
9.1	Create Final Presentation													█	█	█	
9.2	Final Report Submission														█	█	

07. DESCRIPTION OF PERSONNEL AND FACILITIES

7.1 Facilitators

- **Mrs. Sanvitha Kasthuriarachchi**– Sri Lanka Institute of Information Technology (SLIIT)
- **Mrs. Sasini Hathurusinghe** - Sri Lanka Institute of Information Technology (SLIIT)

08. DESCRIPTION OF BUDGET

Table 8. 1 Budget

Component	Cost
Travelling Cost	Rs. 3,000
Internet Cost	Rs. 2,500

09. COMMERCIALIZATION

The Price & Demand Forecasting & Decision-Alert function in the Yield Sync system will be commercialized by identifying priority user segments and validating the market opportunity. This function is valuable across the agri value chain because it delivers timely, localized 1/2/4/8/12-week forecasts with uncertainty and simple SELL/HOLD cues, helping users time sales, cut waste, and plan logistics. Primary players include farmers seeking better prices and income stability, agricultural advisors/extension officers who guide selling decisions, and organizations improving market access and efficiency such as co-ops/FPOs, collectors/wholesalers, agritech platforms, and relevant public/NGO programs.

9.1 Target Market

Target Audience
- Small and commercial Farmers
- Agricultural Consultants
- Agricultural Technology Companies
- Agricultural Research Institutions

10. REFERENCE

[1] G. A. R. J. Hyndman, Forecasting: Principles and Practice (3rd ed.), Melbourne, Australia: OTexts, 2021.

[2] S. Ö. A. N. L. T. P. B. Lim, "Temporal Fusion Transformers for Interpretable Multi-horizon Time-Series Forecasting," in *Temporal Fusion Transformers for Interpretable Multi-horizon Time-Series Forecasting*, Springer Nature, 2021, pp. 1054-1066.

[3] D. C. N. C. Y. B. B. N. Oreshkin, "N-BEATS: Neural Basis Expansion Analysis for Interpretable Time-Series Forecasting," *Nature Machine Intelligence,* vol. 3, 2020.

[4] D. C. N. C. Y. B. B. N. Oreshkin, "N-BEATS: Neural Basis Expansion Analysis for Interpretable Time-Series Forecasting," *Proc. ICLR,* 2020.

[5] C. G. T. Chen, "XGBoost: A Scalable Tree Boosting System," in *Proc. 22nd ACM SIGKDD*, New York, USA, 2016.

[6] H. (. K. A. R. a. T. Institute), "Weekly Food Commodities Bulletin," HARTI, Colombo, Sri Lanka, 2015-2025.

[7] S. L. Department of Census and Statistics (DCS), "Paddy Statistics: Maha and Yala Seasons," DCS Sri Lanka, Colombo, Sri Lanka, 2024/2025.

[8] S. L. Department of Meteorology, "Online Weather Data and Forecasts," Department of Meteorology, Colombo, Sri Lanka, 2025.

[9] H. S. J. Z. F. Sun, "Agricultural Product Price Forecasting Methods: A Review," *Agriculture (MDPI),* vol. 13, no. 9, p. Article 1671, 2023.

[10] S. B. A. N. Angelopoulos, "Conformal Prediction: A Gentle Introduction," *Foundations and Trends in Machine Learning,* vol. 16, no. 4-5, pp. 494-591, 2023.

[11] B. L. S. J. Taylor, "Forecasting at Scale," *The American Statistician,* vol. 72, no. 1, pp. 37-45, 2018.

[12] Q. M. T. F. e. a. G. Ke, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Red Hook, NY, USA*, Red Hook, NY, USA, 2017.

[13] G. B. R. Koenker, "Regression Quantiles," *Econometrica,* vol. 46, no. 1, pp. 33-50, 1978.

11. APENDIX

5%   Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

🔴 **17** Not Cited or Quoted 4%
Matches with neither in-text citation nor quotation marks

🟠 **0** Missing Quotations 0%
Matches that are still very similar to source material

🟡 **4** Missing Citation 1%
Matches that have quotation marks, but no in-text citation

🟢 **0** Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

4% 🌐 Internet sources

2% 📖 Publications

3% 👤 Submitted works (Student Papers)

Match Groups

🔴 **17** Not Cited or Quoted   4%
Matches with neither in-text citation nor quotation marks

💬 **0**   Missing Quotations   0%
Matches that are still very similar to source material

📄 **4**   Missing Citation   1%
Matches that have quotation marks, but no in-text citation

🔖 **0**   Cited and Quoted   0%
Matches with in-text citation present, but no quotation marks

Top Sources

4%   🌐 Internet sources
2%   📖 Publications
3%   👤 Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	
export.arxiv.org		<1%

2	Internet	
ouci.dntb.gov.ua		<1%

3	Student papers	
Sri Lanka Institute of Information Technology		<1%

4	Internet	
publikationen.bibliothek.kit.edu		<1%

5	Internet	
www.utwente.nl		<1%

6	Internet	
arxiv.org		<1%

7	Student papers	
islingtoncollege		<1%

8	Internet	
vfast.org		<1%

9	Internet	
www.dovepress.com		<1%

10	Publication	
Uvin Eksith Senadheera, Dikkumburage Jasintha Jayasanka, Choolaka Hewawasa...		<1%

11	Student papers	
Florida Gulf Coast University		<1%

12	Student papers	
Munster Technological University (MTU)		<1%

13	Internet	
www.frontiersin.org		<1%

14	Internet	
www.fujipress.jp		<1%

15	Internet	
www.fynd.academy		<1%

16	Internet	
www.ocera.org		<1%