

Proposed System

Weekly Date

9/9/24 to 21/9/24

Objective:

To design a smart, ML-based system that recommends the best-suited fertilizer to the farmer based on soil condition, weather, and crop type. The system ensures sustainable fertilizer use to boost yield and protect the environment.

Working of the System:

1. User Registration & Login

- Secure login with Firebase/Auth0
- Optional multi-factor authentication
- JWT-based session management

2. Farmer Profile Setup

- Name, contact, GPS farm location
- Crop types (e.g., wheat, rice)
- Irrigation system info

3. Weather Integration

- Real-time weather via OpenWeatherMap API
- Data: temperature, rainfall, humidity, 7-day forecast
- Location check with geofencing

4. Soil Data Input

- Manual: Enter NPK, pH, moisture
- IoT: Bluetooth soil sensors (optional)
- Fetch past data from SQLite

5. Farm Details

- Area in acres/hectares
- Crop growth stage (e.g., vegetative)
- Previous yield info

6. AI Fertilizer Prediction

- Input: [N, P, K, pH, moisture, crop, rainfall, area]
- XGBoost model predicts:
 - Fertilizer Type (e.g., 20-10-10)
 - Quantity (per acre/hectare)
 - Application schedule

7. Smart Report Generation

- PDF report with:
 - Soil Health
 - Fertilizer Calculator
 - Weather Impact
 - Sustainability Score
- Share via WhatsApp/Email

8. Result & Feedback

- Dashboard with:
 - Nutrient gap analysis
 - Fertilizer cost vs yield gain
- Thumbs up/down feedback to improve model

Algorithm Used: XGBoost (Extreme Gradient Boosting)

XGBoost

- Excellent for tabular agricultural data
- Fast and optimized for performance
- High accuracy with less computation
- Handles missing values and outliers
- Prevents overfitting with regularization

Steps:

1. Collect historical crop, soil, and fertilizer data
2. Train XGBoost model using features like NPK values, pH, temperature, humidity, and crop type
3. Accept user input (soil & environmental conditions)
4. Predict suitable fertilizer using the model
5. Display result to user

Technologies Used:

- **Frontend:** React or Flutter (for app)
- **Backend:** Python (Flask/Django)
- **ML:** XGBoost
- **Database:** Firebase or SQLite
- **Weather Data:** OpenWeatherMap API
- **Authentication:** Firebase/Auth0
- **Report Generator:** Python (ReportLab / PDFkit)

Advantages:

- High accuracy
- Fast and efficient
- Handles missing data
- Prevents overfitting (via regularization)

Architecture / Flow:

Sustainable Fertilizer Usage Optimizer Flowchart

