AgroSmart – Precision Farming Dashboard

Project Objective

To empower farmers with real-time data, actionable insights, and intelligent recommendations by providing a comprehensive dashboard for monitoring soil conditions, weather patterns, crop health, and optimizing farming practices for increased yield and efficiency.

System Architecture Overview

Frontend (React.js):

- User authentication (Login, Register)
- Dashboard for real-time sensor data display (soil moisture, temperature, pH, etc.)
- Weather forecast display
- Crop health monitoring section (e.g., visual indicators, growth stages)
- Irrigation and fertilizer scheduling interface
- Crop recommendation display
- Alerts and notifications view
- Data visualization charts (yield trends, historical sensor data, rainfall)
- Admin/Expert panel for managing users, crop types, and recommendation rules.

Backend (ASP.NET Core Web API):

- REST APIs for user management, sensor data ingestion, weather data, crop information, recommendations, alerts, and scheduling.
- Business logic for processing sensor data, generating recommendations, and managing schedules.
- Integration with external APIs (Weather, potentially soil data if external sensors are used).
- Role-based access control (Farmer, Admin, Expert).

Database (SQL Server):

- Tables for Users, Farms, Fields, Sensors, SensorData, Crops, FieldCrops, Recommendations, Schedules, WeatherData, Alerts.

User Roles

Role	Description
Farmer	Access their farm's dashboard, view data, receive recommendations and alerts, manage schedules for their fields.
Expert	Access aggregated data across multiple farms (if permitted), refine recommendation algorithms, provide advanced insights.
Admin	Manage users, roles, crop types, sensor configurations, system settings, and moderate content/data if needed.

Core Features

1. User Authentication & Role Management

- Secure user registration and login using JWT tokens.
- Role-based dashboards and feature access (Farmer, Expert, Admin).

2. Sensor Data Monitoring & Visualization

- API endpoints for receiving sensor data (simulated or real).
- Real-time display of soil moisture, temperature, pH, nutrient levels, etc.
- Historical data charts and trends (e.g., daily/weekly/monthly averages).

3. Weather Forecast Integration

- Integration with a third-party Weather API (e.g., OpenWeatherMap, AccuWeather).
- Display of current weather, hourly, and multi-day forecasts relevant to the farm's location.

4. Crop Health Monitoring

- Input fields for farmers to log observations (e.g., pest sightings, growth stage).
- Visual indicators of crop health based on sensor data and observations.
- Potentially integrate with image recognition (future enhancement) or simple health status (good, moderate, poor).

5. Crop Recommendation System

- Based on soil type, weather patterns, historical yield data, and farmer input.
- Suggest optimal crops for specific fields and seasons.
- Consider factors like water requirements, nutrient needs, and local climate.

6. Intelligent Alerting & Notifications

- Real-time alerts for critical conditions (e.g., low soil moisture, sudden temperature drops, pest warnings).
- Notifications for recommended actions (e.g., "Time to irrigate Field A," "Apply fertilizer to Field B").

7. Irrigation & Fertilizer Scheduling

- Farmers can view and manage automated or recommended irrigation schedules.
- Dashboard to display upcoming tasks and track completed activities.

8. Data Visualizations

- Interactive charts (line, bar, pie) to show yield trends, sensor data fluctuations, rainfall patterns, and resource consumption.

9. Admin/Expert Panel

- Manage user accounts and roles.
- Define and update crop types and their characteristics.
- Monitor system performance and data integrity.
- (Expert Role): Analyze aggregated data, fine-tune recommendation parameters.

Database Schema Suggestion (SQL Server)

1. Users Table

Column Name	Data Type	Description
UserId	INT (PK, IDENTITY)	Unique identifier for each user
FullName	NVARCHAR(100)	User's full name
Email	NVARCHAR(100)	User's email address (must be unique)

Column Name	Data Type	Description
PasswordHash	NVARCHAR(255)	Hashed password for secure login
Role	NVARCHAR(20)	User role: 'Farmer', 'Expert', or 'Admin'
Phone	NVARCHAR(15)	Contact number
Address	NVARCHAR(255)	Physical address (e.g., farm location)
CreatedAt	DATETIME	Date and time of user account creation

2. Farms Table

Column Name	Data Type	Description
FarmId	INT (PK, IDENTITY)	Unique ID for each farm
FarmName	NVARCHAR(100)	Name of the farm
Location	NVARCHAR(255)	Geographical location (e.g., coordinates or address)
UserId	INT (FK)	Reference to the farmer who owns this farm
CreatedAt	DATETIME	Date and time of farm registration

3. Fields Table

Column Name	Data Type	Description
FieldId	INT (PK, IDENTITY)	Unique ID for each field within a farm
FieldName	NVARCHAR(100)	Name/Identifier of the field (e.g., "North Plot")
SizeAcres	DECIMAL(10,2)	Size of the field in acres
FarmId	INT (FK)	Reference to the farm it belongs to
CreatedAt	DATETIME	Date and time of field creation

4. Sensors Table

Column Name	Data Type	Description
SensorId	INT (PK, IDENTITY)	Unique ID for each sensor
SensorType	NVARCHAR(50)	Type of sensor (e.g., 'Soil Moisture', 'Temperature', 'pH')
Manufacturer	NVARCHAR(100)	Sensor manufacturer

Column Name	Data Type	Description
Model	NVARCHAR(100)	Sensor model
FieldId	INT (FK)	Reference to the field where sensor is placed
LastCalibrated	DATETIME	Last calibration date
IsActive	BIT	Is sensor currently active?

5. SensorData Table

Column Name	Data Type	Description
DataId	BIGINT (PK, IDENTITY)	Unique ID for each sensor reading
SensorId	INT (FK)	Reference to the sensor that took the reading
Value	DECIMAL(10,2)	The sensor reading value
Unit	NVARCHAR(20)	Unit of measurement (e.g., '%', '°C', 'pH')
Timestamp	DATETIME	Date and time of the reading

6. Crops Table (Master data for crop types)

Column Name	Data Type	Description
CropId	INT (PK, IDENTITY)	Unique ID for each crop type
CropName	NVARCHAR(100)	Name of the crop (e.g., 'Wheat', 'Paddy', 'Cotton')
OptimalSoilpHMin	DECIMAL(4,2)	Minimum optimal pH for this crop
OptimalSoilpHMax	DECIMAL(4,2)	Maximum optimal pH for this crop
AvgWaterReqmm	DECIMAL(10,2)	Average water requirement (mm/day or week)
GrowthDurationDays	INT	Average growth duration in days
Description	NVARCHAR(MAX)	General description or growing tips

7. FieldCrops Table (What crop is planted in which field)

Column Name	Data Type	Description
FieldCropId	INT (PK, IDENTITY)	Unique ID for each field-crop instance
FieldId	INT (FK)	Reference to the field

Column Name	Data Type	Description
CropId	INT (FK)	Reference to the crop planted
PlantedDate	DATE	Date crop was planted
HarvestDate	DATE	Expected or actual harvest date
CurrentGrowthStage	NVARCHAR(50)	e.g., 'Germination', 'Vegetative', 'Flowering'
CreatedAt	DATETIME	Timestamp of record creation

8. Recommendations Table

Column Name	Data Type	Description
RecId	INT (PK, IDENTITY)	Unique ID for each recommendation
FieldId	INT (FK)	Reference to the field this recommendation is for
RecommendationType	NVARCHAR(50)	'Crop', 'Irrigation', 'Fertilizer', 'Pesticide'
Description	NVARCHAR(MAX)	Detailed recommendation text
SuggestedAction	NVARCHAR(255)	e.g., 'Apply NPK 20:20:20', 'Irrigate for 2 hours'
ConfidenceScore	DECIMAL(5,2)	Confidence of the recommendation (0-100)
GeneratedAt	DATETIME	Timestamp when recommendation was generated
IsDismissed	BIT	Has the farmer dismissed this?
DismissedAt	DATETIME	Timestamp of dismissal

9. Schedules Table (for irrigation, fertilizer, etc.)

Column Name	Data Type	Description
ScheduleId	INT (PK, IDENTITY)	Unique ID for each schedule
FieldId	INT (FK)	Reference to the field
ScheduleType	NVARCHAR(50)	'Irrigation', 'Fertilizer', 'Pesticide', 'Harvest'
ScheduledDate	DATETIME	Date and time for the scheduled action
DurationHours	DECIMAL(5,2)	Duration for irrigation (if applicable)
QuantityUnit	DECIMAL(10,2)	Quantity of fertilizer/pesticide
Unit	NVARCHAR(20)	Unit for quantity (e.g., 'kg', 'liters')
IsCompleted	BIT	Has this schedule been completed?

Column Name	Data Type	Description
CompletedAt	DATETIME	Timestamp of completion
CreatedBy	INT (FK)	User who created the schedule

10. WeatherData Table (Cache for API calls, by location)

Column Name	Data Type	Description
WeatherId	BIGINT (PK, IDENTITY)	Unique ID for each weather record
Location	NVARCHAR(255)	Geographical location (e.g., city, lat/long for weather forecast)
Latitude	DECIMAL(9,6)	Latitude of the location
Longitude	DECIMAL(9,6)	Longitude of the location
Temperature	DECIMAL(5,2)	Temperature in Celsius
Humidity	DECIMAL(5,2)	Humidity in percentage
Precipitation	DECIMAL(5,2)	Precipitation amount (mm)
WindSpeed	DECIMAL(5,2)	Wind speed (m/s or km/h)
WeatherCondition	NVARCHAR(100)	e.g., 'Clear', 'Clouds', 'Rain'
ForecastDate	DATETIME	Date and time for which forecast applies
RetrievedAt	DATETIME	Timestamp when data was retrieved from API

11. Alerts Table

Column Name	Data Type	Description
AlertId	INT (PK, IDENTITY)	Unique ID for each alert
UserId	INT (FK)	User who receives the alert
FieldId	INT (FK)	Field associated with the alert (nullable)
AlertType	NVARCHAR(50)	'Critical', 'Warning', 'Info'
Message	NVARCHAR(MAX)	Detailed alert message
IsRead	BIT	Has the alert been read by the user?
GeneratedAt	DATETIME	Timestamp when alert was generated

Summary of Primary & Foreign Keys

Table	Primary Key	Foreign Keys
Users	UserId	—
Farms	FarmId	UserId → Users
Fields	FieldId	FarmId → Farms
Sensors	SensorId	FieldId → Fields
SensorData	DataId	SensorId → Sensors
Crops	CropId	—
FieldCrops	FieldCropId	FieldId → Fields CropId → Crops
Recommendations	RecId	FieldId → Fields
Schedules	ScheduleId	FieldId → Fields CreatedBy → Users
WeatherData	WeatherId	—
Alerts	AlertId	UserId → Users FieldId → Fields

UI/UX Suggestions

- **Farmer Dashboard:**
  - **Overview:** Widgets for overall farm health (e.g., average soil moisture, pending alerts, upcoming tasks).
  - **Field-specific View:** Click on a field to see detailed sensor data charts, current crop status, and recommendations.
  - **Alerts & Notifications:** A dedicated section or pop-ups for real-time alerts.
  - **Scheduling Calendar:** Visual calendar for irrigation, fertilizer, and harvest schedules.
- **Expert/Admin Panel:**
  - User Management table with search and filter.
  - Crop Master Data management.
  - System performance metrics.
- **Data Visualizations:** Use clear, intuitive charts (line graphs for trends, bar charts for comparisons, gauge charts for current sensor values).
- **Form Design:** Clean and simple forms for adding farms, fields, sensors, and managing crop details.

- **Responsive Design:** Essential for farmers accessing the dashboard on mobile devices in the field.
- **Component Libraries:** Leverage React components for data tables, date pickers, and modals.

Tech Stack Summary

Layer	Technology
Frontend	React.js, Tailwind CSS / Bootstrap, Chart.js / Recharts
Backend	ASP.NET Core Web API (C#)
Database	SQL Server
Auth	JWT + Role-Based Auth
External APIs	Weather API (e.g., OpenWeatherMap, AccuWeather)
Hosting	Azure / IIS / Vercel (for frontend)
Dev Tools	Postman, GitHub, VS Code, SSMS

Optional Enhancements

- **SMS/Email Notifications:** For critical alerts.
- **GIS Integration:** Visualize farm fields on a map with overlaid sensor data.
- **Image Processing:** Use AI/ML for pest detection or crop disease diagnosis from uploaded photos.
- **IoT Device Integration:** Direct integration with actual farm sensors (e.g., using Azure IoT Hub).
- **Yield Prediction Model:** Machine learning model to predict yield based on historical data and current conditions.
- **Pest and Disease Database:** Comprehensive database with visual aids and recommended treatments.
- **Cost/Benefit Analysis:** Track input costs vs. yield for profitability insights.