

# Project Planning Document — Smart Agriculture Management System (SAMS)

---

## 1. Project Overview

The **Smart Agriculture Management System (SAMS)** is a modern digital solution designed to simplify and automate key agricultural processes.

The system brings all farm operations into one unified platform where administrators can monitor farms, zones, and crops, track environmental conditions, detect problems early, and make faster, data-driven decisions.

### Key Objectives

- Centralize all information related to farms, zones, and crops.
  - Monitor crop growth stages and seasonal needs.
  - Provide real-time alerts whenever unusual conditions appear.
  - Help supervisors and farmers take timely corrective actions.
  - Improve farm productivity and operational efficiency.
- 

## 2. Project Scope

### In Scope

- User management and assigning users to farms.
- Creating farms and organizing them into zones.
- Assigning crops to zones.
- Tracking crop growth stages.
- Real-time alerts and notifications.
- Dashboards and analytics.

### Out of Scope

- Automated irrigation systems.
- Automated fertilization.

- Weather API integration.
  - AI-based prediction models.
- 

### **3. Project Phases & Timeline (10 Weeks)**

1. **Requirements Analysis** – 1 week
2. **System Architecture & Design** – 2 weeks
3. **Backend Development** – 3 weeks
4. **Mobile / Frontend Development** – 2 weeks
5. **Alerts & Notifications Engine** – 1 week
6. **Testing & Deployment** – 1 week

**Total Duration: 10 Weeks**

---

### **4. Workflow Mapping to System Features**

- User registration and role management
  - Farm creation and assignment
  - Zone creation within farms
  - Crop assignment per zone
  - Growth stage tracking and environmental requirements
  - Real-time monitoring and insights
  - Automatic alert generation
  - Notifications for users
  - Dashboards and analytical reports
- 

### **5. Project Deliverables**

- SRS (System Requirements Specification)
  - Use Case Diagrams
-

- ERD (Entity Relationship Diagram)
  - API Documentation
  - Backend Implementation
  - Frontend or Mobile App
  - Testing Reports
  - Deployment Scripts
  - User & Admin Manuals
- 

## **6. Risk Management**

### **Major Risks**

- Performance problems
- Inaccurate sensor data
- Scalability limitations
- Low user adoption

### **Mitigation Strategies**

- Optimize APIs and use caching
  - Validate sensor data before processing
  - Adopt scalable cloud infrastructure
  - Provide training and proper onboarding
- 

## **7. Key Performance Indicators (KPIs)**

- Dashboard loading time  $\leq 2$  seconds
- Alert generation time  $\leq 10$  minutes
- Data accuracy  $\geq 95\%$
- Reduction in manual inspections  $\geq 90\%$
- Yield accuracy  $\geq 98\%$

---

## 8. Tools & Technologies

- **Backend:** ASP.NET Core, EF Core
- **Frontend / Mobile:** React.js or Flutter
- **Database:** SQL Server
- **Cloud:** Azure / AWS
- **Version Control:** Git & GitHub
- **Testing:** Postman, Swagger