Additional Documents for Real-Time Smart Farming System

---

1. Requirement Specification Document (RSD)

Purpose: Defines functional and non-functional requirements for real-time system.

Functional Requirements

1. Accept soil test data from farmers (manual upload or app input).
2. Real-time crop recommendation based on soil, weather, location, and market data.
3. Generate day-wise crop management plan with fertilizer & irrigation schedule.
4. Upload crop images for disease detection.
5. Send alerts and notifications for disease, irrigation, and fertilizer.
6. Provide market price prediction and selling suggestions.

Non-Functional Requirements

1. System should handle 1000+ farmers concurrently.
2. Real-time alerts must reach farmer within 5 minutes.
3. Secure data storage with encrypted sensitive information.
4. User-friendly interface for low literacy farmers.
5. Multi-language support (local languages).

---

2. Deployment & Infrastructure Document

Purpose: Defines how the system will be deployed in real-time.

Components

- Backend server: Flask/FastAPI hosted on cloud (AWS, GCP, or Render).
- Database: PostgreSQL or MySQL for structured data.
- ML Models: Hosted on server with REST API.
- Frontend: Web or mobile app with offline caching for farmers.
- Notifications: SMS/WhatsApp API integration for real-time alerts.

Deployment Architecture

```
Farmer Device (App/Web)
        |
        ▼
   API Gateway (HTTPS)
        |
```

```
      ▼
Backend Server (Flask/FastAPI) ──> ML Models (Crop/Disease/Market)
      │
      ▼
 Database (PostgreSQL/MySQL)
      │
      ▼
 Notification Service (SMS/WhatsApp)
```

Scalability Considerations

- Auto-scaling backend servers using cloud services.
- Caching frequent data to reduce latency.
- Load balancing for multiple farmer requests.

---

3. Test Plan Document

Purpose: Ensure system works correctly in real-time conditions.

Test Types

1. **Unit Testing**: Test each module (soil input parser, ML models, crop plan generator).
2. **Integration Testing**: Check end-to-end flow (soil input → crop recommendation → crop plan → alerts).
3. **User Acceptance Testing (UAT)**: Field test with 10–20 farmers.
4. **Performance Testing**: Check system handles concurrent users with low latency.
5. **Security Testing**: Ensure data encryption, secure file uploads.

Test Data

- Sample soil reports
- Historical crop and price data
- Sample crop images

---

4. Maintenance & Monitoring Document

Purpose: Define how real-time system will be maintained.

Monitoring

- Track system uptime and response time.
- Monitor ML model accuracy regularly.
- Collect farmer feedback for UI/UX improvements.

Maintenance

- Update ML models with new data every season.
- Apply security patches regularly.
- Backup database and logs daily.

End of Additional Documents