**Software Requirements Specification (SRS) for AgriMitra**

**1. Introduction**

**1.1 Purpose**

AgriMitra is an agriculture-based web application that provides farmers with an integrated platform to access a marketplace, weather forecasts, and news updates. The system will enhance the efficiency of agricultural practices by providing real-time information and trading opportunities.

**1.2 Scope**

AgriMitra consists of the following core modules:

1. **Login/Signup** - User authentication and authorization.
2. **Marketplace** - A platform for buying and selling agricultural products.
3. **News** - Latest agricultural news and updates.
4. **Weather** - Real-time weather updates and forecasts.

**1.3 Definitions, Acronyms, and Abbreviations**

* **User:** A farmer, seller, or buyer using the platform.
* **Admin:** Platform administrator with privileges to manage users and content.
* **UI:** User Interface.
* **DBMS:** Database Management System.

**1.4 References**

* IEEE Standard for Software Requirements Specifications (IEEE 830-1998)

**1.5 Overview**

The document provides a comprehensive description of the system’s functionalities, user requirements, system architecture, and necessary diagrams.

**2. Overall Description**

**2.1 Product Perspective**

AgriMitra will be a web-based system accessible from mobile and desktop devices. It will be developed using the MERN stack.

**2.2 User Classes and Characteristics**

* **Farmers:** Buy/sell products, check weather, read news.
* **Buyers:** Purchase agricultural products.
* **Sellers:** List products for sale.
* **Admin:** Manage users, products, and content.

**2.3 Operating Environment**

* **Frontend:** React.js with Tailwind CSS
* **Backend:** Node.js with Express.js
* **Database:** MongoDB
* **Hosting:** AWS / Firebase

**2.4 Design and Implementation Constraints**

* Secure authentication with JWT.
* Scalability to support thousands of users.
* High availability for real-time data access.

**2.5 Assumptions and Dependencies**

* Users should have internet access.
* Weather API and news feeds must be accessible.

**3. Functional Requirements**

**3.1 User Authentication (Login/Signup)**

* Users can register using an email and password.
* Passwords should be stored securely with hashing.
* Users should be able to reset passwords.

**3.2 Marketplace**

* Users can list agricultural products.
* Buyers can browse and filter products.
* Sellers can update or remove listings.
* Transactions can be conducted securely.

**3.3 News**

* Fetch and display the latest agricultural news.
* Admin can add or remove news articles.

**3.4 Weather**

* Display real-time weather updates for the user’s location.
* Provide weekly weather forecasts.
* Integration with third-party weather APIs.

**4. Non-Functional Requirements**

**4.1 Performance**

* The system should handle 500+ concurrent users.
* Page load time should not exceed 2 seconds.

**4.2 Security**

* HTTPS encryption for secure communication.
* Role-based access control for admin functionalities.

**4.3 Usability**

* Mobile-responsive UI.
* Intuitive design with easy navigation.

**4.4 Availability**

* 99.9% uptime with cloud hosting solutions.

**5. System Architecture**

**5.1 High-Level Architecture Diagram**

[User] → [Frontend (React.js)] → [Backend (Node.js, Express)] → [Database (MongoDB)]

**5.2 Use Case Diagrams**

**5.2.1 Login/Signup**

* User registers or logs in.
* Admin manages user accounts.

**5.2.2 Marketplace**

* Sellers list products.
* Buyers browse and purchase products.

**5.2.3 News**

* Users view the latest agricultural news.

**5.2.4 Weather**

* Users check real-time weather updates.

**6. Interface Requirements**

**6.1 User Interfaces**

* **Login Page:** Email/password authentication.
* **Marketplace Page:** Product listings, filtering options.
* **News Page:** Display articles in a card-based format.
* **Weather Page:** Display forecasts in a graphical interface.

**6.2 Hardware Interfaces**

* Compatible with mobile and desktop devices.

**6.3 Software Interfaces**

* Uses REST APIs for data communication.
* Integrates with third-party APIs for news and weather.

**6.4 Communications Interfaces**

* WebSocket for real-time updates.

**7. Other Requirements**

* Multi-language support (English & Hindi).
* Dark mode for better user experience.

**8. System Diagrams**

**8.1 Context Diagram**

* Depicts the interaction between the system and external entities.

**8.2 Level Data Flow Diagram (DFD)**

* Shows the flow of data between different components of AgriMitra.

**8.3 Entity-Relationship (ER) Diagram**

* Defines the relationship between different data entities within the system.

**8.4 Data Dictionary**

* A structured repository of all database attributes, tables, and relationships.

**9. Appendices**

* API documentation
* Wireframes
* Database schema

**10. Conclusion**

AgriMitra aims to be a one-stop solution for farmers and buyers by providing essential agricultural resources in an easy-to-use platform. This SRS outlines all necessary details to ensure a smooth development process.