**Transforming Agriculture Through Innovation**

**Project Title: AgriSmart – Transforming Agriculture Through Innovation**

**1. Project Overview**

AgriSmart is a full-stack MERN application designed to digitize and modernize agricultural practices by providing smallholder farmers with real-time access to farming advice, market prices, weather updates, and farm data analytics. The platform promotes innovation, efficiency, and data-driven decision-making in agriculture.

**2. Project Modules & Features**

**A. Backend Development (Node.js + Express.js + MongoDB)**

**Key Features:**

* **User Management**: Registration, login, role-based access (farmer, agronomist, admin)
* **Crop Advisory Service**: AI/ML API integration for smart farming tips
* **Market Linkage**: CRUD APIs for listing and viewing market produce and prices
* **Weather Forecast**: Integrate with external weather APIs
* **Farm Monitoring**: Store and retrieve data on farm activities
* **Notifications**: SMS/email alerts for tips, weather, and market updates

**Backend Structure:**

* Models: User, Farm, Crop, Market, Advisory, Alert
* RESTful API with JWT-based authentication
* Input validation with Joi or express-validator
* Centralized error handling

**B. Frontend Development (React.js)**

**Key Features:**

* **Dashboard**: Personalized insights, alerts, and quick access to key features
* **Farm Data Entry**: Forms for logging planting, irrigation, pesticide usage, etc.
* **Market Section**: View current crop prices, demand trends
* **Advisory Portal**: View AI-generated tips and agronomist recommendations
* **Weather Updates**: Weekly forecasts and real-time updates
* **Responsive UI**: Mobile-first design

**Frontend Stack:**

* React (Functional Components & Hooks)
* Redux Toolkit or Context API for state management
* Axios for API calls
* Tailwind CSS or Material UI for design

**C. Testing Strategy**

**Backend:**

* Unit Tests: Jest + Supertest (e.g., user registration, crop record creation)
* Integration Tests: Ensure API endpoints work as expected

**Frontend:**

* Component Testing: React Testing Library
* E2E Testing: Cypress or Playwright for simulating user workflows

**D. Deployment**

**Infrastructure:**

* **Backend**: Hosted on [Render](https://render.com/)
* **Frontend**: Deployed via Vercel
* **Database**: MongoDB Atlas

**CI/CD:**

* GitHub Actions for automated testing & deployment

**3. Timeline & Milestones**

| **Week** | **Tasks** |
| --- | --- |
| 1 | Requirements gathering, wireframes, setup repo & environments |
| 2 | Backend setup: DB schema, API routes, user auth |
| 3 | Frontend: UI setup, login/register, dashboard |
| 4 | Core features: market module, weather API, advisory logic |
| 5 | Testing implementation, error handling |
| 6 | Deployment, CI/CD setup, documentation |

**4. Technologies Stack Summary**

| **Area** | **Tech Stack** |
| --- | --- |
| Frontend | React.js, Tailwind CSS, Axios |
| Backend | Node.js, Express.js |
| Database | MongoDB Atlas |
| Testing | Jest, Supertest, React Testing Library, Cypress |
| Deployment | Vercel/Netlify (Frontend), Render/Railway (Backend) |
| CI/CD | GitHub Actions |