**Trader Web Application – Project Documentation**

**1. Title and Scope**

**Title:**  
AI-Powered Agricultural Market Intelligence – Trader Web Portal

**Scope:**  
A Single Page Application (SPA) designed exclusively for verified traders to explore available crops, interact with farmers, place crop purchase orders, track order and logistics status, and post demand requests in real time. This platform connects directly with the mobile app used by farmers and central backend APIs.

**2. Purpose / Objective**

* Enable verified traders to buy crops from farmers directly
* Access real-time crop availability, pricing, and regional insights
* Post demand for crops with quantity, region, and preferred price
* Place and track orders with delivery updates
* View order history, download invoices
* Maintain secure profile with role-based access
* Offer multilingual FAQ/help support

**3. Technology Stack**

| **Component** | **Technology** |
| --- | --- |
| Frontend | React.js (SPA), Tailwind CSS |
| Backend | Node.js, Express.js |
| Authentication | JWT (role-based access) |
| Database | MongoDB / MySQL |
| Charts | Recharts / Chart.js |
| Hosting | Vercel / Netlify |
| DevOps | GitHub Actions, Docker |

**4. Trader Web Portal – Key Modules**

| **Module** | **Description** |
| --- | --- |
| Login/Register | Mobile number with OTP verification for secure access |
| Dashboard | Overview of orders, crop availability, and current market trends |
| Crop Marketplace | List of crops available from farmers including price, quantity, region |
| Place Order | Place orders directly for selected crops |
| Order Tracking | Track delivery and status of ongoing and completed orders |
| Post Demand | Create demand entries for future crop requirements |
| FAQs / Support | View frequently asked questions or raise support queries |
| Profile Management | Edit trader details and change login credentials |

**5. API Endpoints (Sample)**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /api/trader/login | Authenticate trader via OTP |
| GET | /api/crops/available | Get available crops list |
| POST | /api/order/place | Place a new crop order |
| GET | /api/order/history | View trader’s previous orders |
| POST | /api/demand/create | Submit crop demand |
| GET | /api/faqs | Retrieve multilingual FAQs |

**6. Database Tables (Essential)**

| **Table Name** | **Key Fields Included** |
| --- | --- |
| users | id, name, email, phone, role (‘trader’), region |
| crops | id, name, region, unit, price, availability |
| inventory | crop\_id, farmer\_id, quantity, available\_date |
| orders | id, trader\_id, crop\_id, quantity, price, status |
| transactions | order\_id, amount, payment\_status, timestamp |
| demand\_posts | id, trader\_id, crop\_name, qty, preferred\_price |

**7. Dashboard Features**

* **Live Crop Listings**
* **My Orders**: Ongoing & Completed
* **Order Location Tracking**
* **Notifications**: Delivery, New Stock
* **Help Center** with FAQs

**8. Security Features**

* Role-based JWT authentication
* Password hashing using bcrypt
* Input validation using middleware
* Rate limiting & CORS enabled
* Action logs for each trader activity

**9. Deployment Setup**

* **Frontend**: React app deployed on Vercel
* **Backend**: Node.js API on Render / AWS
* **Database**: MongoDB Atlas / PlanetScale
* **DevOps**: CI/CD with GitHub Actions
* **Process Manager**: PM2 or Docker

**10. Advanced (Optional) Features**

* Real-time updates via WebSocket
* Invoice PDF download from order history
* WhatsApp notifications for delivery
* Dark/light mode UI toggle
* Multi-language support (English, Hindi, Marathi)