**Admin Panel – Project Documentation**

**1. Title and Scope**

**Title:**  
AI-Powered Agricultural Market Intelligence – Admin Panel

**Scope:**  
A central web-based platform for admins/project handlers to oversee users, crops, pricing, predictions, FAQs, and system logs. This panel enhances operational control, real-time monitoring, and system-wide configuration of the agricultural ecosystem.

**2. Purpose / Objective**

* Manage all platform users (Farmers, Traders)
* Maintain an updated crop inventory with price tracking
* Visualize AI-based crop price forecasts
* Manage multilingual FAQs and chatbot knowledge base
* Ensure secure and audited system activity
* Provide region-wise operational insights
* Support announcement broadcast to users
* Set automated alerts for price thresholds and risks

**3. Technologies Used**

| **Component** | **Technology** |
| --- | --- |
| Frontend | React.js, Tailwind CSS |
| Backend | Node.js, Express.js |
| Database | MySQL / MongoDB |
| Charts | Chart.js, Recharts |
| Authentication | JWT (role-based) |
| Hosting | Vercel / Netlify (frontend), AWS / Render |

**4. Admin Panel Screens & Modules**

**A. Dashboard**

* Total Users, Crops, FAQs
* Real-time price movement graph
* Active logistics stats
* Region-wise performance indicators
* Recent login/session activity

**B. User Management**

* List users by role and region
* Activate/Suspend users
* Email + Phone verification status
* Export user list as CSV

**C. Crop Management**

* Add/update crop details (base price, unit, season, category)
* Toggle availability
* Attach reference price data (e.g., Govt. API source)

**D. Price Prediction**

* View prediction graphs by crop + region
* Upload new model predictions (CSV/Excel/JSON)
* Model versioning support
* Accuracy display (AI confidence level)

**E. FAQ / Chatbot**

* Add/edit multilingual questions
* Link FAQs to voice or text assistants (Speech-to-Text)
* Analytics for most asked questions

**F. Transaction Logs**

* View crop sell history (crop, farmer, price, date)
* Monitor abnormal activities
* Export logs for audit

**G. Alert Management**

* Define price change thresholds
* Notify farmers/logistics via SMS/Push/Email
* Schedule warning or harvest reminders

**H. Broadcast System**

* Push announcements to all users
* Filter by language, location, or user role
* View engagement stats

**5. API Endpoints**

GET /api/dashboard/stats - System overview

POST /api/users - Create user

GET /api/users?role=farmers - Filtered user list

POST /api/crops - Add new crop

PUT /api/crops/:id - Edit crop info

POST /api/predictions/upload - Upload prediction data

GET /api/faqs?lang=mr - FAQs by language

POST /api/alerts - Create a new alert

GET /api/transactions - List all sell transactions

POST /api/broadcast - Push announcements

**6. Database Tables**

* users – Manage all users
* crops – Crop name, base price, season, region
* predictions – Crop-wise price forecasts
* faq – Language-wise Q&A for chatbot
* transactions – Sell transactions by farmers
* alerts – Threshold and notification rules
* broadcasts – Admin-pushed announcements
* logs – System action history with timestamps

**7. Security & Reliability**

* Role-based JWT tokens
* Passwords hashed using bcrypt
* Rate limiting + IP blacklisting
* Admin-only permission for critical operations
* Logs for all changes in crops, predictions, and users

**8. Advanced Features**

* Dark Mode Toggle
* Notification Badge for New Logs/Alerts
* Regional Analytics Heatmap
* AI Model Status with Upload Date
* Multilingual Admin Panel Support (EN, MR, HI)
* WebSocket Alerts for Realtime Updates
* Search + Filters on every table

**9. Dev & Deployment Notes**

* CI/CD with GitHub Actions
* PM2 for backend process management
* Docker setup for local dev and containerized deployment
* Nginx reverse proxy
* Automated backups for database (Daily via cron)

**1. users Table**

| **Field Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| id | INT | PK, AUTO\_INCREMENT |
| name | VARCHAR(100) | NOT NULL |
| email | VARCHAR(100) | NOT NULL, UNIQUE |
| phone | VARCHAR(15) | NOT NULL |
| password | VARCHAR(255) | NOT NULL (hashed) |
| role | ENUM | 'farmer', 'trader', 'admin' |
| region | VARCHAR(100) | NOT NULL |
| language | VARCHAR(10) | DEFAULT 'en' |
| is\_verified | BOOLEAN | DEFAULT FALSE |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP |

**2. crops Table**

| **Field Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| id | INT | PK, AUTO\_INCREMENT |
| name | VARCHAR(100) | NOT NULL |
| category | VARCHAR(50) | NOT NULL |
| unit | VARCHAR(50) | NOT NULL (e.g. kg, ton) |
| region | VARCHAR(100) | NOT NULL |
| season | VARCHAR(50) | NOT NULL |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP |

**3. price Table**

| **Field Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| id | INT | PK, AUTO\_INCREMENT |
| crop\_id | INT | FK → crops(id) |
| region | VARCHAR(100) | NOT NULL |
| price | DECIMAL(10,2) | NOT NULL |
| date | DATE | NOT NULL |
| source | VARCHAR(100) | NULL |

**4. sell\_crops Table**

| **Field Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| id | INT | PK, AUTO\_INCREMENT |
| farmer\_id | INT | FK → users(id) |
| crop\_id | INT | FK → crops(id) |
| quantity | DECIMAL(10,2) | NOT NULL |
| price\_per\_unit | DECIMAL(10,2) | NOT NULL |
| location | VARCHAR(100) | NOT NULL |
| available\_till | DATE | NOT NULL |

**5. inventory Table**

| **Field Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| id | INT | PK, AUTO\_INCREMENT |
| user\_id | INT | FK → users(id) |
| crop\_id | INT | FK → crops(id) |
| quantity | DECIMAL(10,2) | NOT NULL |
| unit | VARCHAR(20) | NOT NULL |
| last\_updated | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP |

**6. predictions Table**

| **Field Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| id | INT | PK, AUTO\_INCREMENT |
| crop\_id | INT | FK → crops(id) |
| forecast\_price | DECIMAL(10,2) | NOT NULL |
| predicted\_on | DATE | DEFAULT CURRENT\_DATE |
| accuracy | DECIMAL(5,2) | NULL |
| model\_version | VARCHAR(50) | NULL |