🏭 FactoryOS – Hybrid Production Monitoring Platform

(SaaS + On-Prem Enterprise Product Master Document)

This document is designed to be: - Copy-paste ready - Proposal ready - Investor ready - Client pitch ready - Developer roadmap ready - Enterprise sales ready

You can use this as: - Business document - Product document - Architecture document - Sales document

---

Product Vision

FactoryOS is a hybrid digital production monitoring platform that enables factories to: - Digitize production operations - Track machine-wise, shift-wise, and operator-wise data - Monitor real-time production - Improve efficiency and quality - Generate automated MIS - Run either in **Cloud (SaaS)** or **Factory Intranet (On-Prem)**

> "One platform. Two deployment models. Unlimited scalability."

Deployment Models

A) SaaS Mode (Cloud)

- Multi-tenant
- Subscription-based
- Cloud hosted
- Auto updates
- API integrations

B) On-Prem Mode (Intranet)

- Single-tenant
- One-time license
- Local deployment
- No internet dependency
- Factory server hosting
- LAN access

Target Market

Primary Focus (Phase 1)

🎯 **On-Prem Clients** - Traditional factories - MSMEs - Auto component units - Security-sensitive plants - Plants without stable internet - Government suppliers

Reason: Faster trust, faster deals, larger ticket size, less SaaS resistance

---

Business Strategy (Start with On-Prem)

Why On-Prem First?

- Factories trust local systems
- Easier to close deals
- One-time revenue
- Customization allowed
- Long-term AMC revenue
- Reference clients

This builds: - Credibility - Case studies - Cash flow - Industry trust

Then expand into SaaS.

Technical Stack (Your Skill-Based Strategy)

Frontend

- Expo
- React Native
- React Native Web
- Single codebase
- No separate web/mobile folders

Backend

- Node.js
- Express/NestJS

Database

- PostgreSQL (production data)
- MongoDB (logs, events)

Infra

- Docker
- Nginx
- Local server deployment
- Cloud deployment ready

Phase-Wise Development Plan

💈 Phase 0 – Sales Demo Prototype (2–3 Weeks)

🎯 Goal: Visual demo to close orders

Build:

- Login screen
- Machine list
- Production planning screen
- Operator input screen
- Live dashboard
- Basic reports

Features:

- Mock data
- Local storage
- Fake real-time updates
- Hardcoded machines

Tech:

- Expo
- React Native Web
- Node.js mock API

🎯 Output: Sales demo app

💈 Phase 1 – MVP On-Prem System (6–8 Weeks)

🎯 Goal: Real deployable system

Core Features:

- User roles
- Machine management
- Production planning
- Operator updates
- Shift management
- Rejection tracking
- Downtime tracking
- Live dashboard
- Basic reports

Deployment:

- Docker
- Local PostgreSQL
- Local Node server
- LAN access

🎯 Output: First real client deployment

💈 Phase 2 – Enterprise On-Prem Product (8–12 Weeks)

🎯 Goal: Sellable enterprise product

Add:

- License system
- Role permissions
- Audit logs
- Backup system
- Offline support
- Local admin panel
- Export reports

🎯 Output: Commercial-grade product

💈 Phase 3 – SaaS Conversion (Cloud) (6–8 Weeks)

🎯 Goal: SaaS product launch

Add:

- Multi-tenancy
- Subscription billing

- Cloud auth
- Tenant isolation
- SaaS dashboards
- Self onboarding

🎯 Output: SaaS platform

Demo App Strategy (For Fast Sales)

Demo Must Show:

- Production planning
- Operator input
- Live update
- Dashboard
- Reports

Not full backend logic – only flow proof.

Purpose:

> "Show capability, not completeness"

On-Prem Sales Strategy

Sales Pitch

> "We deploy a private production monitoring system inside your factory network — no cloud, no internet dependency, full control."

Offer Model

Package Example:

- Software license: ₹8–15 Lakhs
- Deployment: ₹1–2 Lakhs
- Training: ₹50k
- AMC: ₹1–2 Lakhs/year

SaaS Selling Model

Cloud Pricing Example:

- ₹999/machine/month
- ₹299/user/month
- Annual plans

🚚 Selling Criteria

Factories buy when you show:

Production visibility Accountability Live data Efficiency improvement Cost reduction Quality control Compliance Reporting

Decision makers care about: - Money - Control - Risk - Data - Security

---

Cloud Deployment Strategy

Infrastructure

- AWS/GCP
- Docker
- Kubernetes
- PostgreSQL
- Redis

Architecture

- API gateway
- Auth service
- Production service
- Reporting service

Go-To-Market Plan

Step-by-step

1. Build demo app
2. Visit factories

3. Show demo
4. Offer on-prem deployment
5. Close 1st client
6. Deploy
7. Create case study
8. Repeat
9. Build SaaS
10. Launch cloud

Long-Term Vision

FactoryOS = Digital Factory Platform

Future modules: - Inventory - Maintenance - Quality - HR - ERP-lite - SCM - AI - IoT

Positioning Statement

> "FactoryOS is a private digital operating system for factories – deployable in cloud or inside your intranet."

Business Model

Revenue Streams: - On-prem license - AMC - SaaS subscriptions - Customization - Integrations - Data services

---

Why This Will Sell

Because it solves: - Manual tracking - No visibility - No accountability - No real-time data - No digital control

---

Execution Philosophy

Start small → Sell early → Deploy early → Learn → Improve → Scale → SaaS → Platform

---

Your Advantage

You already have: - Node.js - React Native - Expo - SQL/NoSQL - System thinking

This is founder-level technical capacity.

Ready-to-Use Sales Line

"We build private digital factory systems that run inside your factory network and scale to cloud when you're ready."

If needed next, we can generate:

Pitch Deck Proposal Template Client Contract Template License Agreement AMC Template UI Wireframes API Design DB Schema Architecture Diagram Installer Script Design

---

This is not a project idea. This is a **software product company blueprint**.