

Build a full-stack **Smart Energy Platform as a Service (PaaS)** that includes:

1. **End-User Web App (Consumer Dashboard)**
2. **Admin / OEM Service Portal**
3. **Multi-tenant SaaS Architecture**

The platform should simulate a real-world smart metering ecosystem where multiple residential or commercial projects can be managed centrally.

Use:

- Frontend: Next.js (React) + Tailwind CSS or Material UI
- Backend: Node.js (Express) or FastAPI (Python)
- Database: PostgreSQL + Time-series DB simulation
- Authentication: JWT + Role-Based Access Control
- Real-time updates: WebSockets

The system must be production-structured, scalable, and modular.

SYSTEM REQUIREMENTS

1 Multi-Tenant SaaS Architecture

Implement:

- Organization (Tenant) Model
- Each tenant has:
 - Multiple users
 - Multiple smart meters
 - Multiple appliances
- Data isolation between tenants
- Admin role can view all tenants
- Tenant Admin can view only their organization

2 End-User Web App (Consumer Dashboard)

Features:

Authentication

- Login / Signup
- Role: User

Dashboard

- Total real-time energy usage (kWh)
- Live energy graph (hourly)
- Current estimated bill
- Carbon footprint tracker
- Savings indicator

Appliance Management

- Appliance list (AC, EV Charger, Washing Machine, etc.)
- Real-time usage per appliance
- ON/OFF toggle
- Schedule appliance
- View predicted cost for running appliance

AI Optimization Panel

- Detect peak vs off-peak hours (ToD tariff logic)
- Recommend optimal run time
- Show projected savings
- Highlight expensive usage patterns

Bill & Insights

- Monthly forecast
- Budget alert
- Usage comparison (This month vs last month)
- Downloadable report (CSV)

3 Admin / OEM Service Portal

Role: Super Admin / Tenant Admin

Organization Management

- Create new project (Apartment / City / Utility)
- Assign tariff plans
- Add smart meters
- View tenant-level consumption

Energy Analytics Dashboard

- Total city/project energy usage
- Peak load analysis
- Revenue analytics
- Demand response analytics

Device Monitoring

- Meter health status
- Last communication timestamp
- Error logs
- Remote device control simulation

Tariff Management

- Configure Time-of-Day tariffs
- Define peak / off-peak pricing
- Apply tariff to tenant

User Management

- Add/remove users
- Assign roles
- View usage history

4 Backend Logic Requirements

Create APIs for:

- Authentication
- Tenant creation
- Smart meter data ingestion (simulated)
- Appliance control
- Scheduling logic
- Tariff-based billing calculation
- AI-based recommendation engine

5 AI & Optimization Logic

Implement:

- Time-of-Day pricing model:
 - Peak hours → higher rate
 - Off-peak hours → lower rate
 - Predict bill based on historical usage
 - Recommend scheduling appliances in cheaper slots
 - Calculate:
 - Monthly savings
 - Carbon footprint reduction
-

6 IoT Simulation Layer

Simulate:

- Smart meter real-time data (random but realistic)
 - Appliance energy consumption
 - Device online/offline state
 - WebSocket live updates
-

7 Database Schema

Tables:

- Organizations (Tenants)
- Users
- Smart Meters
- Appliances
- Energy Readings (time-series)
- Tariff Plans
- Usage Logs
- Recommendations

Ensure tenant_id isolation everywhere.

8 Security Requirements

- JWT authentication
- Role-based access control
- Encrypted API responses
- Input validation
- Protected routes

9 UI/UX Design Guidelines

- Clean energy-themed UI (blue/green gradients)
 - Card-based dashboard
 - Modern charts (Recharts / Chart.js)
 - Responsive layout
 - Dark mode toggle
 - Professional SaaS feel
-

10 Expected Output

Generate:

1. Complete folder structure
2. Frontend pages
3. Backend APIs
4. Database schema
5. Sample seed data
6. AI logic
7. WebSocket setup
8. README with setup instructions

Build this like a production-ready energy SaaS platform, not a simple demo.

🔥 EXTRA (For Even Better Generation)

Add this line at end of prompt:

Generate the system step-by-step:

1. Database schema
2. Backend APIs
3. Authentication system
4. End-user frontend
5. Admin portal frontend
6. AI optimization module
7. Real-time simulation
8. Deployment instructions