**Training Need Analytic (TNA) System Development Plan**

**1. Define the Purpose & Scope**

**Objective:**

Develop an **AI-powered Training Needs Analysis (TNA) System** with **Big Data Analytics** to:

* Identify training needs at **Organization, Department, and Individual Levels**
* Predict skill gaps using **SBERT (Skill Embeddings)**
* Track **Training ROI** (Return on Investment) **6 months post-training**
* Fetch **Industry Trends** (WEF, McKinsey, Google Trends, etc.)

**Current System Status (Stage 1a - Organization Level)**

* **Frontend:** React.js (OrgForm.js, App.js)
* **Backend:** FastAPI (training\_analytics\_api.py)
* **Storage:** organization.json (Stores company details)
* **Industry Trends API Placeholder**
* **WEF Skill Extraction API Placeholder**

**2. Research & Plan the Architecture**

**Updated System Architecture**

+-------------------------------+

| User Interface (React.js) | 👤 HR, Managers, Employees

+-------------------------------+

| API Layer (FastAPI) | 🚀 RESTful API for frontend-backend communication

+-------------------------------+

| AI & Analytics Layer | 🧠 SBERT Skill Matching, AI Predictions, ROI calculations

+-------------------------------+

| Data Processing Layer | 🔄 Data Cleaning, Transformation, Preprocessing

+-------------------------------+

| Storage & Database Layer | 📊 PostgreSQL, NoSQL (MongoDB for AI models)

+-------------------------------+

| External Data Integration | 🌍 Industry Trends, WEF Skills, Benchmarking

+-------------------------------+

**Key API Endpoints (Existing & Planned)**

| **Endpoint** | **Function** |
| --- | --- |
| /organization/ | Manage & analyze company-level training needs |
| /department/ | Fetch department-specific training recommendations |
| /individual/ | Provide personalized training plans |
| /roi\_analysis/ | Calculate Training ROI after 6 months |
| /wef\_skills/ | Fetch skills from WEF Report |
| /industry\_trends/{industry}/ | Fetch latest industry trends |

**3. Create a Prototype (Wireframe & MVP Design)**

**React.js UI Components**

* **Dashboard:** Displays Org-Level, Department-Level, Individual Training Needs
* **Company Form (OrgForm.js)** ✔️ Already implemented
* **Industry Trends Page:** Fetches real-time industry skill demand
* **Training ROI Analysis Page:** Tracks training impact over time

**4. Start Development (Iterative Approach - Adjusted for React & FastAPI)**

**✅ Step 1: Backend Development (FastAPI)**

* ✔️ **Organization-Level API** (Already implemented)
* 🔹 **Training Recommendation AI Model (SBERT) – Planned**
* 🔹 **Department & Individual API Extensions – Next Steps**

**✅ Step 2: Frontend Development (React.js)**

* ✔️ **Company Form (OrgForm.js)** (Already working)
* 🔹 **New Components:**
  + **Industry Trends View** (Connects /industry\_trends/{industry}/)
  + **Training ROI View** (Displays 6-month performance data)

**✅ Step 3: AI Integration**

* 🔹 **Skill Embeddings Model (SBERT) for AI-powered skill matching**
* 🔹 **ROI Prediction Model (Calculates training impact)**

**5. Test & Debug**

* ✅ **Unit Testing:** API responses for accuracy
* ✅ **Integration Testing:** React.js components fetching data correctly
* ✅ **Security Testing:** HR/Admin role-based access
* ✅ **Performance Testing:** AI model efficiency

**6. Deploy & Monitor**

**✅ Deployment Plan**

* ✔️ **Frontend:** React.js hosted on Vercel/Netlify
* ✔️ **Backend:** FastAPI deployed on **AWS/GCP/Azure** or Docker-based on-premise
* ✔️ **Database:** PostgreSQL (TNA data) + MongoDB (AI models)
* ✔️ **Logging & Monitoring:** Track API performance & errors

**7. Iterate & Improve**

🚀 **Next Steps (Based on User Feedback):**

* 🔹 Improve AI accuracy for skill matching
* 🔹 Add more industry integrations for real-time updates
* 🔹 Optimize React.js dashboard & training visualizations

**🛠 Next Actions – What Do You Need First?**

1️⃣ **React.js UI Wireframe (Mockup of how it looks)?** 2️⃣ **Develop Department-Level Training API?** 3️⃣ **Start AI Model for Training Recommendations?**

Let me know what you’d like to prioritize first! 🎯🚀