AI Chatbot & Resume Evaluation System - Product Documentation

# Product Overview

This system combines a Resume Evaluation API suite with an AI-powered chatbot integrated with a product knowledge base.

## Modules

1. **Resume Evaluation APIs**:

   - `/job-embed`: Accepts job descriptions, returns semantic embeddings.

   - `/match-resumes`: Compares parsed resumes with job embeddings, returns relevance score.

   - Uses OpenAI for relevance computation and PDF/DOCX parsing for candidate info.

2. **LinkedIn Posting APIs**:

   - `/linkedin/profile-post`: Post to user profile using access token.

   - `/linkedin/company-post`: Post to company page using organization URN and token.

3. **AI Chatbot**:

   - Uses OpenAI + RAG to answer product-related queries.

   - Three endpoints:

     - `/chatbot/upload-kb`: Uploads documents (PDF) and stores embeddings.

     - `/chatbot/add-problem`: Manually adds known issues and resolutions.

     - `/chatbot/query`: Accepts a question, returns a response using RAG or fallback.

## Architecture

- FastAPI backend.

- MongoDB for persistence (if needed).

- In-memory store (Python dict) for chatbot knowledge base.

- Embeddings from `text-embedding-3-small` or latest OpenAI model.

## Setup Instructions

1. **Environment Setup**

   - Install dependencies:

     ```bash
     pip install fastapi uvicorn python-multipart openai pypdf langchain
     ```

   - Set environment variables:

     ```env
     OPENAI_API_KEY=<your_key>
     ```

2. **Run the Server**

   ```bash
   uvicorn demo:app --reload
   ```

## Usage Flow

- **Resume Evaluation**:

  - Call `/job-embed` -> get embeddings.

  - Upload resume(s) -> `/match-resumes` -> get scores.

- **Chatbot**:

  - Upload docs (PDF) via `/chatbot/upload-kb`.

  - Add fallback via `/chatbot/add-problem`.

  - Ask query via `/chatbot/query`.

## Common Errors

- 403 ACCESS_DENIED on LinkedIn: Check scopes.

- "Knowledge base is empty": Upload PDFs first.

- "User ID not found": Access token invalid or expired.

## Notes

- Uses OpenAI v1 SDK. Avoid deprecated `.Embedding.create`.

- Uses in-memory store. For production, use a database (MongoDB/Redis).

- LinkedIn `userinfo` gives member ID -> build `urn:li:person:{id}`.