Odens AI - Hiring Test: Development of Quote Automation System
Deadline: This system need to be finished before Wednesday 16.00 CET

Project: Development of AI-Assisted Quote Automation System

Purpose: This project tests your ability to design, develop, and implement a prototype for a system that automates and streamlines the quoting process for a company selling custom aluminum profiles based on customer orders. The goal is to deliver a functional prototype, ready for testing. The evaluation will focus on your technical approach, problem-solving skills, architectural design, execution, and how well the prototype meets the requirements. We want to see how you tackle the challenge of enabling a user to input information and then using the system to automatically generate a complete quote proposal based on this data.

Background: Many manufacturing companies in the aluminum industry spend significant time manually calculating and creating quotes. This system aims to automate the process: a user (customer or salesperson) inputs all relevant information (customer data, specifications, communication, etc.), the system calculates a price based on an AI model trained on historical quotes, and then the system uses this information, including the calculated price, to generate a complete quote document. The system should be viewed as a potential module within a larger business system (ERP-like structure).

Overall System Description: You are to build a web-based application/system where users input information for a quote request. The system should:

1. Collect and structure the user's input.
2. Calculate a price using an AI model trained on the company's historical, manually created quotes.
3. Generate (Output) a complete quote document that the aluminum company can review and send to its customer, minimizing manual calculation and writing.

Core Functionality to Include:

1. **Customer Management (Basic):**
   - Ability to select an existing customer.
     - *Example Implementation:* A simple dropdown list fetching names from a hardcoded list or a minimal database table.
   - Ability to add basic customer details for the quote.
     - *Example Implementation:* Text fields for Company Name, Contact Person, Email.
2. **Quote Details:**
   - Input fields for basic quote information.
     - *Example Implementation:* Fields for Quote Title (e.g., "Delivery of Profile System XYZ"), Reference, Validity Date.
3. **Product Specification:**
   - Ability to enter specifications as free text.

- - *Example Implementation:* A text area (`<textarea>`) where the user describes profile type, alloy, dimensions, quantity, surface treatment, tolerances, etc.
  - Ability to upload relevant files (specifically PDF, e.g., drawings).
    - *Example Implementation:* A file upload button. The backend stores the file and associates it with the quote.

4. **Customer Communication/Context:**
   - A field/mechanism to paste or upload text-based communication (e.g., email excerpts, meeting notes).
     - *Example Implementation:* Another text area or file upload.
     - *Example AI Integration:* AI could analyze this text to identify key factors ("needs fast delivery," "specific packaging," "previously discussed discount") that might influence the quote text or be used as input for the pricing model.

5. **Pricing Engine:**
   - **AI-based Learning (Sole method in this test):**
     - The system should be trainable by an administrator uploading historical quotes (at least 150 examples recommended). (In PDF format)
       - *Example variables for the pricing engine:* (e.g., `ProfileType`, `Alloy`, `Weight_per_meter`, `Total_length`, `SurfaceTreatment`, `MachiningComplexity`, `FinalPrice`). Your backend trains a prediction model (e.g., regression, gradient boosting) on this data. The model should then be able to predict a price for the *new* quote based on its specifications and potentially context.
   - **Result:** The engine must return a *calculated price* to be used in the next step.

6. **AI-Driven Quote Generation (Core of the test):**
   - The system must use AI (specifically a Large Language Model, LLM) to create the actual quote text.
   - **Input to AI:** All collected data: Customer details, quote details, text/extracted data from product specifications, context from customer communication, **and the calculated price**.
   - **Output from AI:** A complete, coherent, and professionally formulated quote text.
     - *Example Implementation:* The backend constructs a prompt for a language model (via API or locally). The prompt includes all information: "Generate a quote for [Company Name] regarding [Quote Title]. Customer contact: [Contact Person]. Specifications: [Text from spec field, possibly key info from PDF]. Important points from communication: [Summary/text from communication field]. **The calculated price is [Price] SEK excl. VAT.** The quote is valid until [Date]. Include details about [specific parameters like profile type, quantity]. Use a formal and business-like tone. Include standard terms for payment and delivery."

Odens AI - Hiring Test: Development of Quote Automation System
Deadline: This system need to be finished before Wednesday 16.00 CET

> ■ *Example Output Format:* The AI's response (the generated text) is put into a pdf file.

Suggested Workflow (Example - adapt as needed):

1. Start New Quote.
2. Select/Enter Customer Details.
3. Enter Quote Details (Title, etc.).
4. Upload/Enter Product Specifications (Text + PDF).
5. Upload/Enter Customer Communication.
6. System calculates price (via AI prediction based on history).
7. **AI Generates Quote:** The system sends all collected data (incl. price) to the AI module, which generates a complete text proposal for the quote.
8. **Review Quote:** The user views the AI-generated quote in the interface (editing capability is good but not a strict requirement).

Technical Aspects and Freedom:

- **Technology Stack:** Free choice, justify your decisions.
- **AI Implementation:** Focus on how you use AI (specifically an LLM) to create the quote text based on structured and unstructured input, and how you implement the price calculation (AI prediction based on history).
- **Security:** Critical! Ensure secure data handling, file uploads, and any API calls (e.g., through anonymization if using external LLMs).
- **Architecture:** Design as a potential module for a complete ERP system.

Evaluation Criteria:

- Understanding of Requirements (especially AI's role in pricing and text generation).
- Technical Solution & Code Quality (structure, readability, robustness).
- Problem-Solving & Approach (data handling, AI model selection, and prompting).
- Effectiveness of AI Generation: How well does the AI succeed in creating a relevant, coherent, and accurate quote based on the input data?
- Architecture & Design (modularity, scalability considerations).
- Security Awareness.
- Creativity & Own Initiatives.

Deliverables:

- Source code (e.g., via a link to a Git repository).
- A brief description (solution, design/technology choices, assumptions, how AI prompting was handled).
- Instructions on how to run/test the prototype (must be a **web application** or similar).
- (Optional) Short demo/screen recording.

Odens AI - Hiring Test: Development of Quote Automation System

Deadline: This system need to be finished before Wednesday 16.00 CET

Good luck! We look forward to seeing your solution and your way of thinking about this problem.