.QIT AI Engineer Assessment Task

Invoice Data Extraction System

Overview

This assessment is designed to evaluate your approach to solving real-world AI engineering challenges. We are looking for a Full Stack AI Engineer with strong expertise in computer science fundamentals and software development processes. The focus is on **how you approach and solve problems**, not just the final accuracy of your solution.

Task Description

Design and implement a system that extracts structured data from invoices using AI/ML techniques. The system should process both text-based invoices and OCR-processed invoice images, extracting key information in a structured format.

Input: Invoice documents (PDF, images, or text)

Output: Structured JSON response

Sample Output Structure:

```
InvoiceResponse {
    general_fields: {
        invoice_number, date, supplier_name, total_amount, currency
    },
    items: [
        { product_name, quantity, unit_price, total_price, description }
    ]
}
```

Technical Requirements

• Docker containerization with FastAPI backend

• Celery + Redis for asynchronous processing

• LLM-based approach with prompt engineering

• Agent-based or agentic design patterns

• Support for both local LLM and API-based LLM

• All technology stack must be open source

• GDPR/KVKK compliance considerations

Solution Approach

The solution method is entirely up to you. We expect an LLM and prompt engineering focused approach with agentic design patterns. If you don't have sufficient hardware to run local LLMs, you may use APIs for demonstration, but **local LLM support should also be included** in your implementation.

Data Intelligence (Bonus)

Beyond simple extraction, implement intelligent analysis of the extracted data. Examples include:

- **Tax Validation:** Verify VAT calculations (18% tax rate verification)
- **Arithmetic Validation:** Ensure quantity × unit_price = total_price for each item

Evaluation Criteria

Category	Weight	Focus Areas
AI Approach	50%	LLM selection reasoning, prompt engineering creativity, agent design patterns, local vs API integration, problem-solving methodology, AI pipeline error handling
Technical Implementation	25%	Code architecture & design patterns, Docker quality, API design (FastAPI + Celery), database/cache integration, performance optimization
Software Engineering	15%	Documentation quality, code readability & structure, testing approach, deployment readiness
Delivery & Presentation	10%	Timeline management, commit history quality, live demo presentation, problem understanding

Our Primary Focus

We want to see **HOW** you approach this problem, not just whether you achieve 100% accuracy. We're evaluating your problem-solving methodology, architectural decisions, and development practices. A well-reasoned approach with 70% accuracy is more valuable than a hacky solution with 95% accuracy.

Bonus Points

- **Turkish Character Support:** High accuracy OCR for ç, ö, ü
- **Prompt Engineering Skills:** Creative and effective prompt design
- **Local LLM Familiarity:** Demonstrated experience with local model deployment
- **Agentic Approach:** Not just framework usage, but thoughtful design patterns
- **Problem Clarity:** Clear understanding and articulation of the problem space
- **Solution Architecture:** Clean, scalable, and maintainable code structure

Delivery Requirements

- **Public GitHub Repository:** Clean commit history preferred
- **Documentation:** Clear README with setup and usage instructions
- **Live Demo:** 30-minute online presentation + Q&A; session
- **Timeline:** Maximum 10 days
- **Sample Data:** You can source your own sample invoices (any language)

Sample Evaluation Questions

During the presentation, expect questions like:

- Why did you choose this specific LLM for the task?
- How would you handle memory limitations in a production environment?
- If your error rate was 20%, what would be your first optimization step?
- How would you scale this solution to handle 1000x more requests?
- What edge cases did you consider and how did you handle them?

Important Notes

- Focus on your approach and methodology rather than perfect accuracy
- Clean, readable code is highly valued
- Document your design decisions and trade-offs
- Consider scalability and maintainability in your solution
- Feel free to be creative with your AI implementation
- Production-ready thinking is appreciated but not required for this assessment

Alternative Approaches

While we encourage an LLM-based approach for this challenge, if you believe a different solution would be more effective for invoice data extraction, you are welcome to pursue that path. However, you must be prepared to explain and justify your approach during the Q&A; session, demonstrating why your chosen method is superior for this specific problem domain.

Good luck! We're excited to see your unique approach to this challenge. Remember, this is about demonstrating your problem-solving skills and technical thinking process.