Talent Pulse 360: Multi-Agent Work Experience Evaluator Task

# Introduction to Talent Pulse 360

Talent Pulse 360 (TP360) is a revolutionary SaaS ecosystem designed to provide real-time talent intelligence to organizational leadership. The platform transforms how businesses hire, retain, develop, and appraise talent by leveraging data across multiple touchpoints of an employee's journey. TP360 brings intelligence to each stage of talent management, from recruitment to performance evaluation, enabling data-driven decision-making for human resource professionals and business leaders.

# Task Objective

You are tasked with designing and implementing a multi-agent system that extracts work experience requirements from job descriptions and evaluates candidate resumes against these criteria. Your solution must use **Node.js and Python** as the primary development languages.

# Detailed Deliverables

## 1. Agent Prompts and Definitions

* Required Output:
  + System prompts for each specialized agent in your system
  + Task definitions that outline each agent's responsibilities
  + Input/output specifications for each agent
  + Error handling protocols

## 2. Architecture Diagram

* Required Elements:
  + Visual representation of all agents in the system
  + Data flow between agents
  + Processing sequence and decision points
  + Integration points with external systems
  + User interaction touchpoints

## 3. Working Code Implementation

* Technology Stack:
  + Backend: Node.js and Python
  + Database: Your choice (justify selection)
  + API integration: REST or GraphQL
  + Documentation: Markdown or similar

# Detailed Task Requirements

## A. Job Description Analysis

* Extraction Requirements:
  + Create agents to identify discrete work experience requirements from any job description
  + Design a parsing system that recognizes explicit and implicit experience criteria
  + Develop a classification system for different types of work experience
  + Implement entity recognition for skills, roles, and responsibilities
  + Create a standardization mechanism for varied terminology

## B. Resume Evaluation System

* Comparison Mechanics:
  + Design agents that map resume content to identified JD requirements
  + Create evaluation algorithms for two key metrics:
    - **Relevance (Scale 1-100):** How closely the experience matches the requirement
    - **Significance (Scale 1-100):** The importance/impact of the experience
  + Develop evidence extraction that identifies supporting information from resumes
  + Implement confidence scoring for all evaluations

## C. Multi-Agent Coordination

* System Architecture:
  + Design communication protocols between agents
  + Create arbitration mechanisms for conflicting assessments
  + Implement orchestration logic for sequential and parallel processing
  + Develop a central coordination mechanism or agent
  + Include feedback loops for continuous improvement

## D. Output Generation

* Results Formatting:
  + Design structured output format (JSON recommended)
  + Create visualization components for results
  + Implement detailed explanation generation for scores
  + Develop summary statistics across multiple candidates
  + Include confidence metrics with all outputs

# Rules for Task Completion

## 1. Timeline

* 48-Hour Deadline:
  + Clock starts upon receipt of this document
  + All deliverables must be submitted before the deadline
  + Time management will be considered in evaluation
  + Extensions will not be granted
  + Incomplete submissions will not be considered

## 2. Original Work Requirement

* Authenticity Guidelines:
  + All agent prompts must be 100% your original writing
  + LLM/Gen AI tools are prohibited for prompt creation
  + Your writing style, grammar, and voice will be analyzed
  + Citations required for any referenced frameworks or methodologies
  + Code may leverage libraries but core logic must be original

### 3. Quality Assessment

* Evaluation Criteria:
  + Clarity and precision of instructions to agents
  + Logical structure and flow of the multi-agent system
  + Technical feasibility of the proposed architecture
  + Scalability considerations for large volumes of data
  + Edge case handling and error recovery mechanisms
  + Documentation quality and completeness

# Interview Preparation

During the subsequent interview, be prepared to:

* Code Demonstration:
  + Run your implementation with provided JDs and resumes
  + Explain how scores are calculated
  + Demonstrate how agents communicate
  + Show how the system handles edge cases
* Architectural Discussion:
  + Justify your choice of agents
  + Explain communication patterns
  + Discuss alternative approaches considered
  + Address scaling challenges
* Design Decisions:
  + Explain your prompt engineering approach
  + Discuss why certain experience factors were prioritized
  + Rationalize evaluation algorithms
  + Present improvement opportunities

Submission Instructions

Your final submission package should include:

1. Complete source code (Node.js and Python)
2. Agent prompt documents (one per agent)
3. Agent task definition documents (one per agent)
4. Architecture diagram (high resolution)
5. Setup and execution instructions
6. Brief technical documentation
7. Sample outputs from test runs

Submit all materials as a compressed archive to email address: info@talentpulse360.com

We look forward to reviewing your innovative approach to this multi-agent system challenge and seeing how you leverage advanced NLP techniques to enhance the talent evaluation process at Talent Pulse 360.