

# Project Setup & Architecture: TalentFlow AI

## 1. Defined Problem Statement

Recruitment is one of the most resource-intensive processes in human resource management. Traditional hiring involves manually screening hundreds of resumes, writing job descriptions, and conducting repetitive first-round interviews all of which consume significant recruiter time and introduce potential bias.

Problem:

Current recruitment systems lack automation and intelligence in screening, evaluating, and shortlisting candidates efficiently. Recruiters struggle to:

- Quickly identify the best-fit candidates from large resume pools
- Generate tailored job descriptions that align with company tone and needs
- Conduct unbiased, scalable preliminary interviews

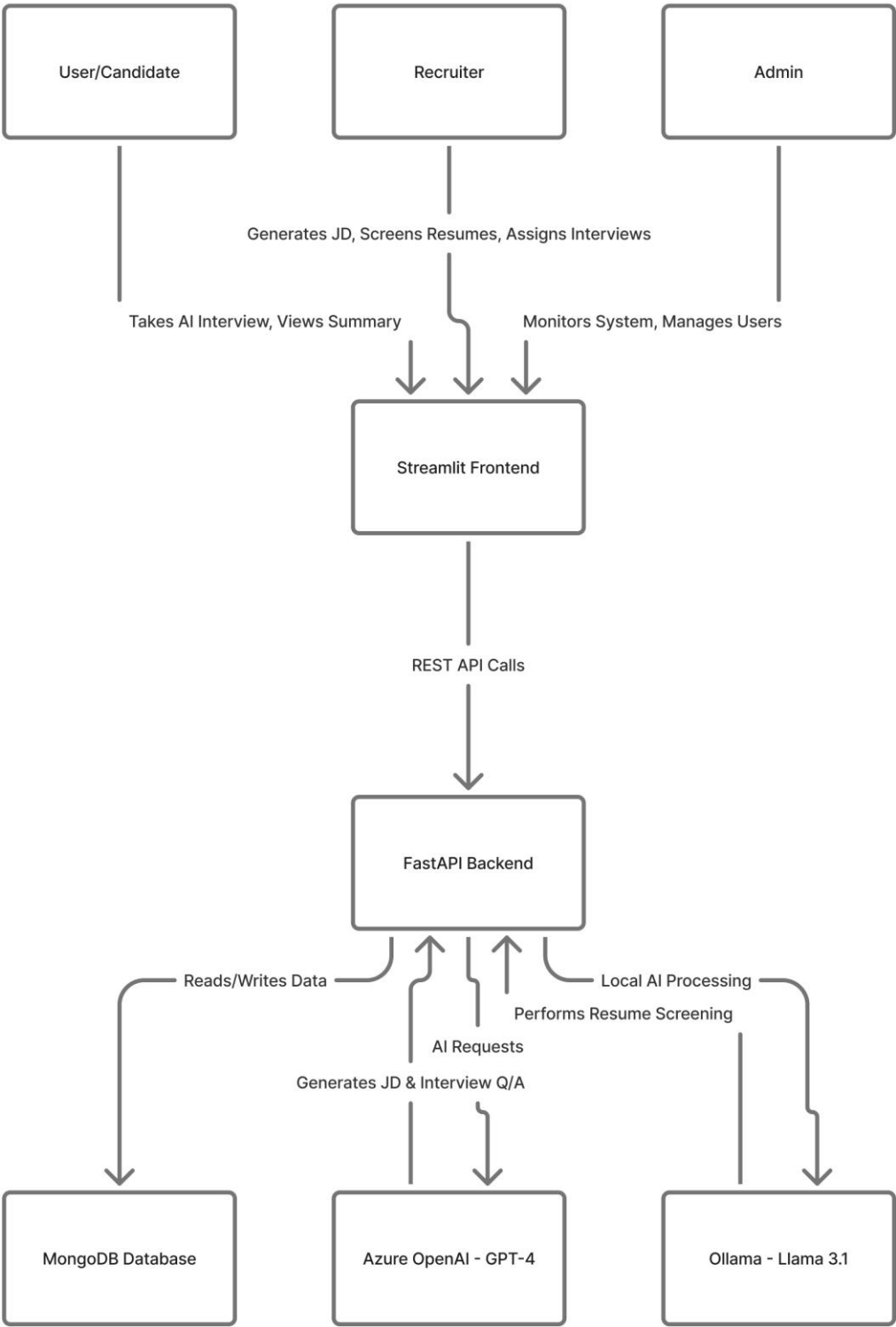
Proposed Solution – TalentFlow AI:

TalentFlow AI is an AI-powered recruitment automation system designed to optimize the hiring process by integrating intelligent modules for:

- Automated resume screening using AI-based matching
- AI-generated job descriptions using GPT models
- AI-conducted interviews for objective evaluation
- Role-based dashboards for recruiters, admins, and candidates
- Secure MongoDB-based authentication with both persistent and temporary accounts

This solution aims to significantly reduce manual effort, improve hiring accuracy, and enhance candidate experience.

## 2. Draft Architecture Diagram



### 3. Finalized Technology Stack

Component	Technology	Justification
Frontend	Streamlit	Provides an intuitive and rapid way to build interactive dashboards and web apps for recruiters, admins, and candidates without heavy frontend frameworks.
Backend Framework	FastAPI	Modern, high-performance Python web framework ideal for asynchronous I/O, scalable APIs, and built-in automatic API documentation.
Database	MongoDB	NoSQL database suitable for flexible document-based storage of resumes, JDs, and interview records; handles dynamic schema well.
AI Model – JD & Interview	Azure OpenAI (GPT-4)	Used for generating job descriptions, interview questions, and evaluating responses, ensuring enterprise-grade reliability and compliance.
AI Model – Resume Screening	Ollama (Llama 3.1)	Local LLM for privacy-preserving resume analysis and matching, reducing dependency on cloud APIs.
AI Orchestration	LangChain & LangGraph	Simplifies building modular and stateful AI workflows, especially for interviews and AI-based reasoning.
Authentication & Security	Bcrypt + MongoDB Auth System	Provides encrypted password storage and supports temporary candidate accounts for secure and role-based access.
Speech Services	Azure Whisper + Azure Speech Services	Adds voice-based question-answer capability for AI interviews.
Containerization	Docker	Encapsulates the backend, frontend, and dependencies into isolated containers for consistent deployment across environments.
Orchestration	Docker Compose	Simplifies multi-container deployment (FastAPI, Streamlit, MongoDB) with a single configuration file, enabling easy scalability and local setup.