# **AI Developer Assignment: Debt Collection Intelligence System**

# **Role Description**

This is a full-time on-site role for an AI Developer - Python/Fast API/Django at Noida or a Remote location.

As an AI Developer, you will be responsible for day-to-day tasks such as pattern recognition, computer science research, neural network development, software development, and natural language processing (NLP).

You will work closely with the AI team, contributing to the development and enhancement of AI solutions.

To apply for this position, please go through the job description and complete the assignmen.

**Salary Details**

* Remote ASE - 16K INR in Hand
* Remote SDE 1 - 22K INR in hand
* Office ASE Role - INR 20K
* Office SDE 1 Role - INR 30K [ 4 LPA ]

# **Must-Have Skills**

* Proficient with database systems (e.g., PostgreSQL, MySQL, MongoDB)
* Proficient in version control systems such as Git.
* Proficient with cloud platforms (e.g., AWS, Azure, Google Cloud) and containerization technologies (e.g., Docker, Kubernetes)
* Develop and maintain backend services using Python FastAPI and Django.
* Design and implement RESTful APIs and ensure their performance, reliability, and scalability.
* Collaborate with front-end developers to integrate user-facing elements with server-side logic.
* Write clean, maintainable, and testable code following best practices.
* Optimise applications for maximum speed and scalability.
* Troubleshoot and debug applications to ensure smooth functionality.
* Participate in code reviews and contribute to the continuous improvement of our development processes.

# **Job Details**

1. **The position is based in Noida - Office is located in Sector 136, Noida**
2. **For Remote Roles - you will be notified.**
3. Candidates are required to have their laptops while they are working with us
4. We are hiring for a Full-time Job where the Candidate will be required to work Monday - Saturday (6-day-a-week job. Saturdays are usually for Planning and Work from home. We have alternate Saturdays off)
5. We allow 2 days a week to work from home for Regular Employees
6. Experience: 1+ years
7. Immediate Joiners Only - The person should be willing to start in 2 Weeks max

If you agree with the above terms, then please complete the assignment below and submit it to start the review process

## 

## **Assignment Tasks**

## **Problem (pick FastAPI or Django)**

Build a **production-ish “Contract Intelligence API”** that ingests PDFs, extracts structured fields, answers questions over the contract, and runs clause-risk checks. It must run locally via Docker.

### **High-level scope**

1. **Ingest**: POST /ingest — upload 1..n PDFs; store metadata + text; return document\_ids.
2. **Extract**: POST /extract — given document\_id, return JSON fields:  
   * parties[], effective\_date, term, governing\_law, payment\_terms, termination, auto\_renewal, confidentiality, indemnity, liability\_cap (number + currency), signatories[] (name, title).
3. **Ask (RAG)**: POST /ask — question answering grounded **only** in uploaded docs; return answer + citations (document\_id + page/char ranges).
4. **Audit**: POST /audit — detect risky clauses (e.g., auto-renewal w/ <30d notice, unlimited liability, broad indemnity). Return list of findings w/ severity, evidence spans.
5. **Stream**: GET /ask/stream — SSE or WebSocket streaming tokens for the same question.
6. **Webhook**: optional POST /webhook/events (candidate implements emitter on server side that would POST to a provided URL when long tasks finish).
7. **Admin**: GET /healthz, GET /metrics (basic counters), GET /docs (OpenAPI/Swagger).

Use **any 3–5 public contract PDFs** of your choice (NDA/MSA/ToS). Do not include proprietary data. Document links in README.

## **What to submit**

1. **Repo** (GitHub/GitLab) with:  
   * Source, tests, Docker, compose, migrations
   * **README**: setup, env vars, endpoints, example curls, trade-offs
   * **Design doc (≤2 pages)**: architecture diagram, data model, chunking rationale, fallback behavior, security notes
   * **prompts/** folder: if you used any LLM prompts, include them verbatim + short rationale
   * **eval/** folder: your Q&A eval set + script + a one-line score summary
2. **Loom (8–10 min)**:  
   * Run make up; show Swagger docs
   * Ingest 2–3 PDFs live, call /extract, /ask, /audit
   * Toggle **rule engine fallback** and show different outputs
   * Show logs redacting PII + metrics endpoint
   * Open tests, run them, and explain one edge case you handled
3. **Commit history** demonstrating incremental work (not one mega-commit).

# **How to Submit**

Before the interview, please complete the above assignment and upload your assignment in the Google Drive Link -https://forms.gle/Wdn86VmtvT5XKjDFA

* Submit the assignment through a GitHub repository. Zip files will not be evaluated.
* Record a [Loom video](https://www.loom.com/looms/videos) of the application showing the features and share the link. Talk about your thought process and various points that you might have considered while designing the application