**Mini-Project 2 Requirements**

**Project Title: AI-Powered Resume Analyzer**

**1. Project Overview**

An **AI-powered resume analysis platform** that allows users to upload their resumes and get **automated feedback on skills, formatting, and job compatibility**. Companies can also search and filter resumes to find suitable candidates. The project features a **Django backend** with **Django Rest Framework (DRF)** and a **Vue.js / React / Next.js frontend**. The system integrates multiple databases to optimize performance.

**2. Functional Requirements**

**2.1 Authentication & User Management**

* **User registration & login** (JWT-based authentication).
* **Roles:** **Job Seeker, Recruiter, Admin**.
* **Email verification & password reset**.

**2.2 Resume Upload & Parsing**

* Users can **upload resumes (PDF, DOCX)**.
* AI extracts key details (skills, experience, education).
* Automated **resume rating and recommendations** based on job descriptions.

**2.3 Job Matching System**

* AI matches resumes with **job descriptions** using NLP.
* Recruiters can **filter candidates** based on skills, experience, and location.
* **Scoring system** for resume-job compatibility.

**2.4 Resume Feedback & Suggestions**

* Users receive feedback on:
  + **Skill gaps** (based on job trends).
  + **Resume formatting** suggestions.
  + **Keyword optimization** for ATS (Applicant Tracking Systems).

**2.5 Multi-Database Connection**

* **PostgreSQL** for user authentication and job listings.
* **MongoDB** for resume storage & AI analysis results.
* **MySQL** for logging & analytics.

**2.6 API & Serializers**

* **DRF-based API** for frontend communication.
* **Serializers:**
  + **Pydantic / Marshmallow** for AI-based resume validation.
  + **ModelSerializer** for database interactions.
* API rate limiting & caching for performance.

**2.7 Admin Panel**

* Django admin panel for managing users, resumes, and job listings.
* Role-based permissions with Django authentication.

**3. Tech Stack**

**Backend (Django & DRF)**

* **Django 4+** with **Django Rest Framework**.
* **Django ORM + multiple database configurations** (PostgreSQL, MongoDB, MySQL).
* **Celery** for handling background tasks (e.g., processing resumes).
* **Redis** for caching and session management.
* **Pydantic / Marshmallow** for validation.
* **NLTK / spaCy** for AI-driven resume parsing.

**Frontend (Choose one)**

* **Vue.js + Vuetify** for a clean UI.
* **React + Tailwind CSS** for a modern, fast frontend.
* **Next.js + Chakra UI** for SEO-optimized job listings.

**Other Integrations**

* **Swagger / drf-yasg** for API documentation.
* **JWT authentication** for secure API access.
* **Logging & Monitoring** with Django Logging and Sentry.