**Hiring App Documentation**

**1. Introduction**

The Hiring App is a recruitment platform built using **React** (frontend) and **Django** with **Django Rest Framework (DRF)** (backend). It integrates **AI-based resume screening, candidate ranking, and interview scheduling** to streamline the hiring process.

**2. Tech Stack**

* **Frontend:** React (with Tailwind CSS / Material UI)
* **Backend:** Django & Django Rest Framework (DRF)
* **Database:** PostgreSQL / MySQL
* **AI Tools:** OpenAI (for NLP), Scikit-learn (for ranking), NLTK / Spacy (for text processing)

**3. Features**

**3.1 User Authentication**

* Role-based authentication (Employer & Job Seeker)
* JWT authentication (Simple JWT for Django)

**3.2 Job Posting & Application System**

* Employers can post job listings
* Job seekers can apply with resumes

**3.3 AI-Powered Resume Screening**

* Extract key skills from resumes using **NLTK/Spacy**
* Match resumes with job descriptions using **cosine similarity (TF-IDF)**

**3.4 Candidate Ranking System**

* Machine learning model (Scikit-learn) ranks candidates based on experience & skills

**3.5 AI Chatbot for FAQs**

* OpenAI API or Dialogflow-based chatbot for common queries

**3.6 Automated Interview Scheduling**

* AI-based recommendations for interview slots
* Google Calendar API integration

**3.7 Sentiment Analysis on Candidate Responses**

* Use TextBlob / VADER to analyze candidate sentiment during interviews

**4. System Architecture**

**4.1 Backend (Django DRF)**

* User authentication (JWT-based)
* Job & application management
* AI models for resume screening & ranking

**4.2 Frontend (React)**

* User dashboard for job posting & applications
* Resume upload & AI feedback
* Chatbot integration

**4.3 Database (PostgreSQL/MySQL)**

* Users (Employers, Job Seekers)
* Jobs & Applications
* AI-generated ranking & feedback

**5. API Endpoints**

**5.1 Authentication**

* POST /api/auth/register/ – User registration
* POST /api/auth/login/ – Login
* GET /api/auth/user/ – Get logged-in user data

**5.2 Job Management**

* POST /api/jobs/ – Create a job post
* GET /api/jobs/ – List jobs
* GET /api/jobs/{id}/ – Retrieve job details

**5.3 Application Management**

* POST /api/apply/ – Apply for a job
* GET /api/applications/{job\_id}/ – Get applicants for a job

**5.4 AI Resume Screening & Ranking**

* POST /api/resume/upload/ – Upload & analyze resume
* GET /api/candidates/rank/ – Get ranked candidates for a job

**5.5 AI Chatbot**

* POST /api/chatbot/query/ – Ask a question

**5.6 Interview Scheduling**

* POST /api/interviews/schedule/ – Schedule an interview
* GET /api/interviews/{user\_id}/ – Get interview details

**6. AI Implementation**

**6.1 Resume Parsing & Screening**

* Use **NLTK/Spacy** to extract skills & experience
* Apply **TF-IDF & cosine similarity** to match with job descriptions

**6.2 Candidate Ranking**

* Train a **Scikit-learn** model using historical hiring data
* Rank candidates based on job fit score

**6.3 Sentiment Analysis**

* Analyze candidate responses using **TextBlob/VADER**

**7. Deployment**

**7.1 Backend Deployment (Django DRF)**

* Use **AWS EC2 / Digital Ocean** for hosting
* Set up **PostgreSQL** as the production database

**7.2 Frontend Deployment (React)**

* Deploy using **Vercel / Netlify**

**7.3 AI Model Deployment**

* Host ML models on **Flask/FastAPI** for AI processing

**8. Future Enhancements**

* Implement **voice-based screening interviews**
* Advanced **predictive analytics** for hiring trends
* AI-based **resume improvement suggestions**