## **Project Overview**

The AI-Driven Resume Filtering App is designed to help corporations streamline the hiring process by leveraging AI to generate job profiles, filter resumes, and rank candidates efficiently. Recruiters can input rough job requirements, and the system will generate an ideal job profile using industry trends, skill graphs, and AI analysis. Submitted resumes are analyzed, ranked, and categorized based on their alignment with the refined job profile.

## **Core Features**

1. **Job Profile Generation:**
   * Recruiters input a job title and rough requirements.
   * AI (Google Gemini) generates a structured job profile including required skills, soft skills, experience, qualifications, and achievements.
2. **Resume Parsing & Extraction:**
   * Resumes (PDF/DOCX) are uploaded.
   * AI extracts key information like skills, experience, education, and projects.
3. **Resume Comparison & Ranking:**
   * AI compares extracted resume data against the refined job profile.
   * Candidates are categorized as **Best**, **Almost Best**, **Good**, and **Bad** based on match percentage.
4. **Adjustable Tag-Based Ranking System:**
   * Recruiters can modify rankings by adding or removing tags.
   * Changes are reflected dynamically in the ranking.
5. **Final List Generation:**
   * A "Final List" button generates a shortlist of **Best** and **Good** candidates for interviews.
   * Recruiters can set constraints on the number of candidates.

## **Technology Stack**

### **Frontend:**

* **Framework:** React.js (Vite setup for optimization)
* **UI Library:** Tailwind CSS + DaisyUI
* **State Management:** Zustand
* **Routing:** React Router DOM
* **Icons & Animations:** Lucide React
* **Authentication:** Firebase Auth/Auth0

### **Backend:**

* **Server:** Node.js with Express.js
* **Database:** MongoDB with Mongoose
* **Authentication:** JSON Web Tokens (JWT) & bcrypt.js for password hashing
* **File Storage:** Cloudinary (for resume uploads & management)
* **Environment Management:** dotenv
* **Middleware:** Cookie-parser
* **Task Management:** Nodemon for development

### **AI & NLP Components:**

* **Resume Parsing:** PyMuPDF (PDFs), docx2txt (DOCX)
* **Text Analysis:** Google Gemini AI (for extracting key information from resumes)
* **Ranking Algorithm:** Cosine similarity & weighted scoring using AI embeddings

### **Deployment & Infrastructure:**

* **Frontend Hosting:** Vercel
* **Backend Hosting:** Render
* **CI/CD:** GitHub Actions
* **Monitoring:** Sentry (for error tracking), Prometheus + Grafana (for metrics)

## **Workflow**

1. **Recruiter posts a job opening** with rough requirements.
2. **AI generates an ideal job profile** based on the job title and trends.
3. **Resumes are uploaded** and parsed using AI.
4. **AI compares resumes** with the job profile and assigns rankings.
5. **Recruiter adjusts tags** to refine rankings.
6. **Final List is generated**, showing top candidates for interviews.

## **Hackathon MVP Scope**

For the 24-hour hackathon, we aim to deliver:

* Basic **job posting & profile generation** (Gemini API)
* Resume **upload & parsing** (text extraction via AI)
* **Basic ranking system** using keyword-based matching
* **Simple UI** for recruiters to post jobs, upload resumes, and view rankings
* **Tag adjustment feature** (basic add/remove functionality)
* **Final List generation** with two categories: Best & Good

## **Future Enhancements**

* **Advanced AI Matching:** Use deep learning models for better resume-job profile matching.
* **Multi-Tenancy:** Support multiple companies with role-based access.
* **Interview Scheduler:** Automate interview invitations based on shortlisted candidates.
* **SaaS Implementation:** Subscription-based model with corporate integration.

## **Conclusion**

This project demonstrates how AI can streamline the hiring process by automating job profile creation, resume filtering, and ranking. The hackathon MVP will showcase a functional prototype with scalable potential for a full-fledged SaaS product.