AI-Powered Resume & Job Description Analyzer

1. Introduction

In today's competitive job market, many candidates struggle to understand whether their resume truly matches a job description. Recruiters use Applicant Tracking Systems (ATS) and skill-based filtering, which often leads to rejection due to missing or misaligned skills rather than lack of potential.

This project, **AI-Powered Resume & Job Description Analyzer**, is a full-stack web application designed to bridge this gap. It uses Artificial Intelligence to analyze resumes against job descriptions and provides intelligent, explainable feedback to help users improve their job readiness.

The system integrates modern frontend technologies, a secure Java Spring Boot backend, local AI processing using Ollama, and containerized deployment using Docker.

---

2. Problem Statement

During the development of this project, several real-world technical challenges were identified and addressed, which further strengthened the system design and implementation.

Challenge 1: AI Model Performance and Memory Constraints

Initially, the project used the **Gemma 3n E2B LLM model** for resume and job description analysis. However, during runtime testing, the following issues were encountered:

- High memory consumption when processing large resume prompts
- Increased response time due to memory overload
- In some cases, the application returned memory usage errors instead of valid AI responses

Additionally, the Gemma 3n E2B model supported **text-to-text input**, but it did not effectively analyze resume content, especially when resumes were provided as structured or image-based data.

To resolve this issue:

- The AI architecture was redesigned
- The **Granite 3.2 Vision model** was integrated via Ollama
- Granite supports **text and image input** with reliable text output
- This change significantly reduced memory issues and improved resume analysis accuracy

Challenge 2: Authentication, OAuth, and JWT Security Implementation

Another major challenge was implementing **secure authentication and authorization**:

- Integrating OAuth login with Google and GitHub

- Generating JWT tokens after OAuth authentication
- Handling token encryption, decryption, and validation
- Securing backend APIs using Spring Security

During this phase, multiple issues occurred such as token validation failures and unauthorized access errors. These challenges were resolved through a combination of:

- Deep understanding of Spring Security flow
- Proper JWT filter configuration
- Collaboration between manual debugging and AI code assistance

By resolving these challenges, the project achieved a secure and scalable authentication mechanism suitable for real-world applications.

---

3. Objectives of the Project

- Analyze resume and job description alignment using AI
- Identify missing skills and gaps
- Provide AI-generated suggestions and rewritten resume bullet points
- Ensure secure access using JWT and OAuth authentication
- Store analysis results for future reference
- Demonstrate real-world AI system design for academic evaluation

---

4. Proposed Solution

The proposed system allows users to upload resume content and job descriptions through a web interface. The backend securely processes the request and interacts with a local AI model to generate insights.

Depending on the analysis:

- If the resume is weak, the system provides missing skills, suggestions, and rewritten bullet points
- If the resume is strong, the system confirms successful alignment

All interactions are secured and stored in the backend database.

---

5. System Architecture

High-Level Architecture

- Frontend: Vite + React + TypeScript
- Backend: Java Spring Boot (REST APIs)
- Database: H2
- AI Engine: Ollama (Granite 3.2 Vision Model)

- Deployment: Docker

Architecture Flow

1. User logs in using JWT or OAuth
2. Frontend sends resume and job description via secured API
3. Backend validates request and constructs AI prompt
4. Ollama AI analyzes input and returns response
5. Backend stores data and sends response to frontend

---

6. Technology Stack

Frontend

- Vite
- React
- TypeScript
- Tailwind CSS
- CSS
- React Router
- Axios

Backend

- Java
- Spring Boot
- Spring Security
- JWT Authentication
- OAuth (Google & GitHub)
- Spring Data JPA
- H2 Database

AI & Tools

- Ollama (Local AI Runtime)
- Granite3.2-Vision:2B Model
- Trae AI Code Assistant IDE

DevOps

- Docker
- GitHub

---

7. Functional Requirements

- User registration and login
- Secure authentication and authorization
- Resume and job description submission
- AI-based analysis
- Display of missing skills and suggestions
- Storage of analysis history

---

8. Non-Functional Requirements

- Security
- Scalability
- Performance
- Data privacy
- Maintainability

---

9. Security Design

- JWT-based stateless authentication
- OAuth2 login support
- Protected backend routes
- Token-based API access

---

10. AI Module Description

The AI module uses Ollama to run the Granite 3.2 Vision model locally. The backend constructs structured prompts combining resume and job description data.

AI outputs include:

- Skill gap analysis
- Resume improvement suggestions
- Rewritten bullet points
- Success confirmation for good matches

---

11. Database Design (Overview)

Tables

- User

- Resume
- JobDescription
- AiAnalysisResult

The database stores both user input and AI-generated outputs securely.

---

12. Dockerization

The application is containerized using Docker to ensure:

- Consistent environment
- Easy deployment
- Platform independence

---

13. Advantages of the System

- AI-driven decision support
- Privacy-focused local AI
- Secure authentication
- Real-world applicability
- Modular and scalable design

---

14. Limitations

- Depends on user-provided data quality
- Not a replacement for human recruiters
- H2 database not suitable for production scale

---

15. Future Enhancements

- PostgreSQL or Neon database integration
- Resume scoring system
- ATS compatibility checks
- PDF resume parsing
- Cloud deployment

---

16. Conclusion

This project demonstrates the integration of Artificial Intelligence with secure full-stack web development to solve a real-world career problem. It showcases system design, backend security, AI reasoning, and modern deployment practices, making it suitable for academic evaluation and portfolio presentation.

---

17. Author

Faleel
Final Year Student
AI Java Full Stack Developer

---

18. License