AI Resume Analyzer

Saul Juarez, RongXin Liu, Johnathan Gallimore

Table of Contents

1. Project Definition

1.1 Purpose

Our system was created to help users improve their resumes and find jobs that match their skills. It gives users feedback on their resumes to make them stronger and easier for employers to notice. It also finds job listings that fit the information in their resumes, so users don’t have to deal with unrelated job recommendations. This tool is useful for both job seekers

and employers. Job seekers can improve their chances of getting hired, and employers can find the right candidates faster. By combining resume feedback with smart job matching, our system makes the job search process easier and more effective for everyone.

1.2 Project Goals

The goal of our project is to help users find jobs that match their resumes and improve their chances of success. The resume analyzer reviews important areas such as education, skills, grammar, experience, and any included portfolio. The system provides users with a grade for their resume along with helpful feedback. If a user receives a low score, the feedback suggests ways to improve their resume. This ensures users can make updates to increase their chances of landing a job. Additionally, the system can assist HR teams in companies by identifying the best candidates for specific positions. By focusing on resumes with the highest grades, HR can efficiently choose the most qualified applicants.

1.3 Development

For this project, we developed a system that integrates multiple components to provide users with valuable feedback on their resumes and job matches. Here's an overview of the development work:

Resume Analysis and Feedback Generation:

We implemented an AI-powered resume analyzer using Python. This system evaluates resumes based on skills, education, grammar, and experience. It provides users with a grade and detailed feedback to help them improve their resumes.

Job Matching:

We integrated external job search APIs, such as Adzuna, to fetch relevant job listings based on the user's resume content. This ensures that users receive personalized and accurate job recommendations.

Database Management:

We used PostgreSQL to store user information, uploaded resumes, and feedback history securely. This allows users to access and track their feedback over time.

Frontend Development:

We designed a user-friendly interface with HTML, CSS, and JavaScript. The frontend includes features like a responsive layout, modals for user interaction, and an intuitive dashboard for job search results.

Backend Development:

The backend was developed using Flask, handling user authentication, API integration, and communication between the database and frontend.

Version Control:

All development work was managed collaboratively using GitHub for version control, ensuring efficient teamwork and code organization.

1.3.1 Teamwork Division

Adam: Frontend development, login page, landing page UI, documentation
Leo: Backend development, Database development, job filtering, API, and backend code connecting,
John: Testing, job filtering connection and implementing code, project documentation

1.3.2 Source Code

Here is a link to the project's GitHub Repository

1.3.3 Demo

490 Demo

2. Project Requirements Analysis

2.1 Functionality

The main functionality of our project is to analyze user-uploaded resumes and provide useful outputs based on the analysis. The key functionalities include:

Resume Analysis:
Users can upload their resumes, which are then processed by the system. The analysis includes grading the resume and providing detailed feedback on areas such as formatting, grammar, experience, education, skills, and any portfolio details included.

Feedback Delivery:
The system generates specific suggestions to help users improve their resumes. This feedback aims to enhance the resume's quality, ensuring it meets professional standards and aligns with job requirements.

Job Recommendations:
After analyzing the resume, the system retrieves a list of jobs that best match the user's skills, experience, and qualifications. These job recommendations are tailored to the content of the resume, ensuring users only see relevant opportunities.

These functionalities combine to provide a seamless and efficient system that helps users improve their resumes and discover jobs suited to their profiles.

2.2 Usability

2.2.1 User Interface

Our application is designed to provide the best user experience when accessed via a web browser. Users begin by either logging in to an existing account or creating a new one. New users are guided through a straightforward registration process, after which they can log in to access the application. Once logged in, users are directed to the AI Analyzer page, where they can upload their resumes. After uploading, the AI processes the resume and provides a grade. Users can view detailed feedback through the "Feedback History" tab, which allows them to revisit the analysis for further improvements. The user interface is intentionally designed to be intuitive and user-friendly while maintaining a visually appealing layout.

2.2.2 Performance

The application is optimized to provide a simple and reliable user experience. The AI's resume analysis is designed to be efficient, with fast processing times to ensure users do not face delays. Since this is a desktop-based web application, users are encouraged to access it via a laptop or desktop for the best performance and usability. The system is built to handle multiple requests while maintaining consistent functionality and responsiveness.

2.3 System

2.3.1 Hardware

For the best performance to run the application, a desktop or laptop is required. This system must have the power to access a web browser, as well as having multiple I/O devices such as mouse, keyboard and a monitor.

2.3.2 Software

The application requires a stable internet connection, as well as access to a web browser.

2.3.3 Database

The system uses PostgreSQL for managing user data, resumes, feedback, and job listing. The database can support large-scale data storage and quick retrieval.

2.4 Security

The users accounts have been password protected, as well as their email addresses for password recovery.

3. Project Specification

3.1 Focus / Domain / Area

Domain: The domain of our project is job seeking and career development support. Our AI Resume Analyzer leverages the power of AI to provide detailed feedback for resume improvement and personalized job recommendations based on the content of a user's resume.

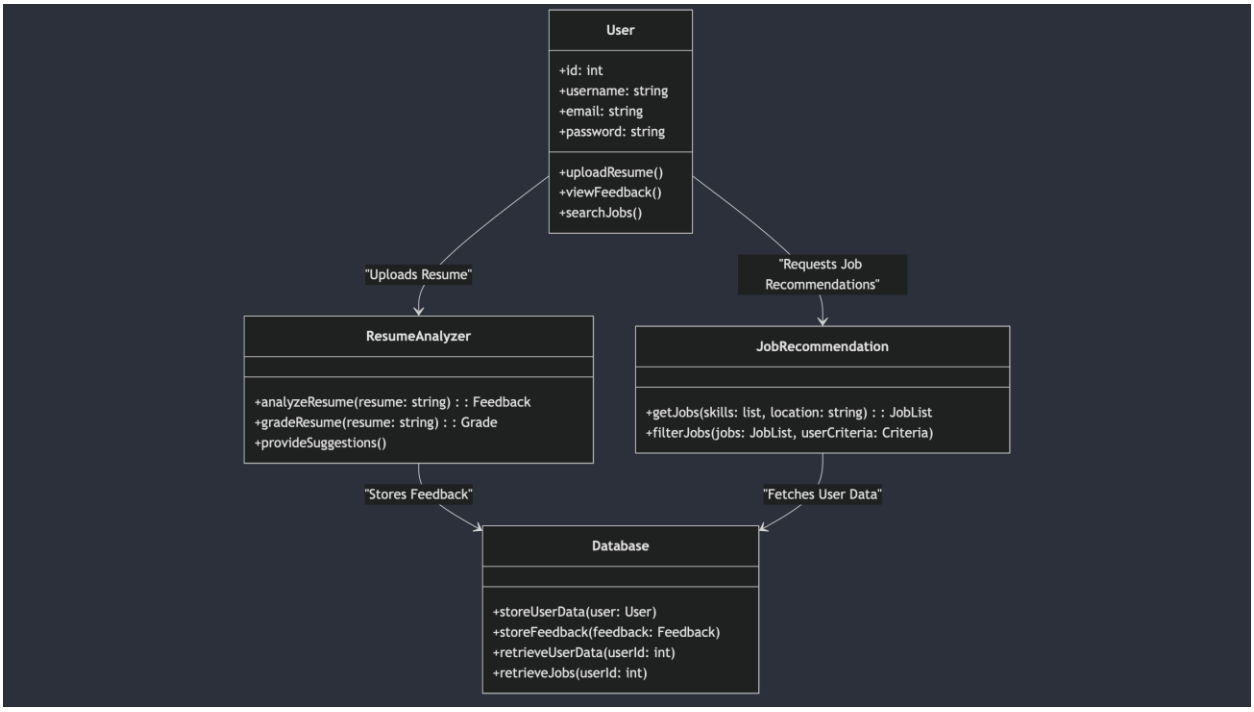
Our primary focus is to assist job seekers in improving their resumes and enhancing their chances of finding the right job opportunities. As college graduates, we understand the challenges of entering the job market, such as meeting experience requirements, identifying suitable roles, and competing with other applicants. Our application is designed to reduce the stress and frustration of job searching by offering personalized feedback and tailored job recommendations. The AI Resume Analyzer aims to instill confidence in users—whether recent graduates or experienced professionals—by helping them refine their resumes and connect with opportunities that align with their skills and aspirations. This tool not only simplifies job searching but also empowers users to showcase their qualifications effectively.

3.2 Libraries / Development Environment

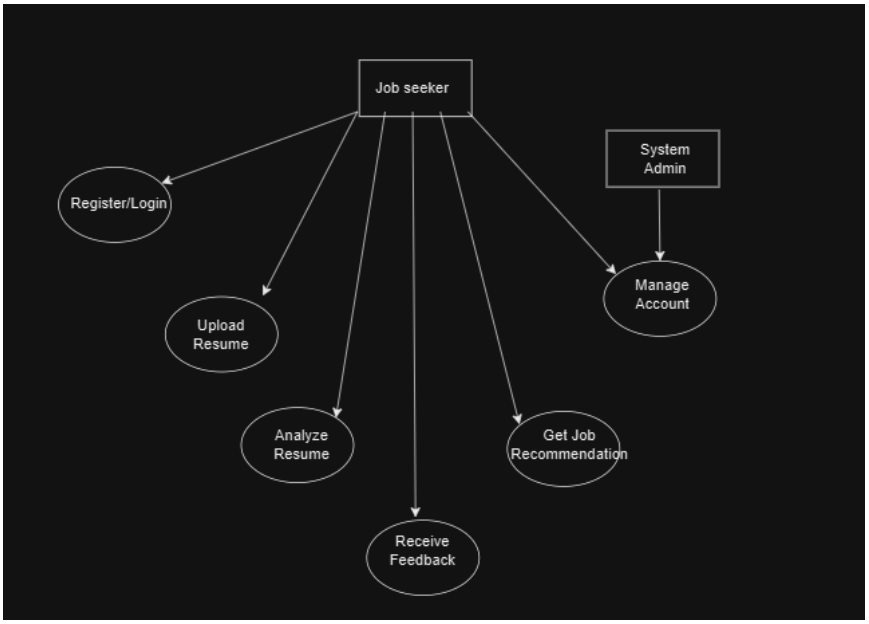
Libraries: SpaCy, PdfPlumber, Docx, OpenAi, Flask
Development Environments: Visual Studio Code, GitHub, PostgreSQL, Google Chrome
Platform: Desktop
Genre: Business or personal use because users could apply for jobs on their own. Businesses can use it instead of searching through hundreds of emails for the right candidates.
Dataset / Data Resources: Resume Data from users

4. System Design

4.1 Overall System Architecture



4.2 Feature Diagram



4.3 Subsystems

4.3.1 Subsystem 1: Resume Analysis

- **Functionality**: Analyzes uploaded resumes for grading and feedback.
- **Flow**:
    - Input: User uploads a resume in PDF or DOCX format.
    - Process:
        - File is parsed using **PdfPlumber** or **Docx**.
        - Text analysis is performed using OpenAI to extract keywords, skills, and grammar.
        - Resume is sent to **OpenAI API** for feedback and grading.
    - Output: Grading and detailed feedback stored in the database and displayed to the user

4.3.2 Subsystem 2: Job Recommendation

- **Functionality**: Matches the user's resume with relevant job opportunities.
- **Flow**:
    - Input: User requests job recommendations.
    - Process:
        - Skills and experience extracted from the resume using **NLP tools**.

        - Keywords generated from the analysis are used in APIs like **Adzuna** or **GitHub Jobs** to search for jobs.
        - Filters out irrelevant jobs based on user-specific criteria.
    - Output: A curated list of job postings returned to the user.

5. Evaluation

5.1 Typical Usage Examples:

- Resume Upload
    - Users upload their resume, and the system provides a grade with detailed feedback to help them improve it.
- Job Recommendations
    - The system suggests jobs that match the user's resume based on skills, education, and experience.
- Feedback History
    - Users can view and track previous resume grades and feedback in the "Feedback History" section.
- Account Management
    - Users can create an account, log in, and manage their profile easily

5.2 System Performance:

- Speed
    - Resume analysis is completed within seconds.
    - Job recommendations load quickly after analysis.
- User Experience
    - The system is easy to navigate, with simple and clear instructions.
- Reliability
    - The application handles errors gracefully and ensures users can retry actions if needed.
- Accessibility
    - The system works best on desktops or laptops, where users can easily upload resumes.

6. Discussion

6.1 Limitations

- Resume Format
    - The system may struggle with parsing non-standard resume formats
- Limited Data

    - The AI's feedback relies heavily on the resume's content, and if the resume lacks details, the grading and recommendations may be less accurate.
- Job Recommendations
    - The job recommendations are limited by the quality and availability of job APIs, which may not cover all industries or locations.

6.2 Future Work

- Expanded Job APIs
    - Integrate additional APIs to provide a wider range of job opportunities, including niche industries.
- Machine Learning Improvement
    - Continuously refine the AI model using user feedback and anonymized data to improve accuracy and relevance.