National Institute of Technology Karnataka, Surathkal
Indian Society For Technical Education
NITK Chapter

AI Resume Insights

SIG	Crypt
Project 3rd years (number of project heads)	4
Project Juniors (number of juniors required)	4 - 5

ELI5

Imagine you're playing a game where you show a robot your toy collection (your resume), and the robot gives you tips on how to make your collection better so other robots (employers) like it more. The robot checks each toy (your resume points) and gives you a score, telling you which toys are cool and which ones need more work.

Explain in Detail

The primary goal of the project is to create an intelligent, automated system that not only evaluates resumes based on key parameters (like brevity, verb usage, structure, and impact) but also provides personalized suggestions for improvement so that it make it through an ATS System. ATS software is used by many companies to sort through job applications by automatically scanning resumes for keywords, formatting, and other relevant details. Additionally, it maintains a history of users' resumes to track progress and inform them if their revisions have led to meaningful improvements.

Key Technologies:

LangChain: The core of our backend. For creating agents that perform specific tasks such as analyzing different aspects of resumes (e.g., brevity, impact).

Gemin API: Agents use Gemini API to analyze the natural language used int the resume. It provides core natural language processing capabilities, ensuring nuanced feedback.

Django (Backend): Backend framework to create and manage the REST API. Store resumes and keep track of a particular user's scores.

ReactJS / NextJS (Frontend): This is for building an interactive and responsive user interface. It'll power features like uploading resumes, displaying feedback and tracking progress.

TypeScript: Ensures robust and error-free frontend code with strong typing.

Project breakdown

1. **Initial Setup**

Set up Django for handling user accounts and backend logic.
Build a basic React frontend with the ability to upload resumes.

2. **Design a multi-agent system with LangGraph and LangChain**

One agent might check formatting, another focuses on keyword matching, while a third looks for readability and clarity. Each of this agent will be using Gemini API to analyze the natural language in the resume.

3. **Feedback Generation**

Based on analysis, generate suggestions for improvement: rephrasing sentences, adding missing keywords, improving formatting, etc.

Provide a score that reflects how well the resume is optimized for ATS and human readability.

4. **User Interface**

Design the React/NextJS frontend to display resume analysis in a user-friendly way, allowing users to view scores and recommendations. Add features like tracking resume versions and progress over time.

5. **Testing and Deployment**

Test the system with real resumes to ensure accurate keyword detection, scoring, and feedback. Deploy the application in a suitable platform.

References

LangChain Documentation: LangChain Docs
Gemini API: Gemin Docs

Timeline 1 (This timeline pertains to the learning process for your project's student members (juniors).)

Checkpoint Number	Deadline and expected time to completion	**Learning Goal-** If any. Towards the end you might find learning is not that intense so it can be blank for the last few checkpoints	**Implementation Goal -** It must be there for **every checkpoint,** even if it's tiny.	Comments if any
1.LLMs/ MAS	3-4 weeks		Getting acquainted with these topics at the end of the	

			deadline	
2.React/ Django	2-3 weeks		Getting acquainted with these topics at the end of the deadline	

Timeline 2 (The complete project - actual implementation timeline with every stage.)

Checkpoint Number	Deadline and expected time to completion	**Task Goal-** The exact task that needs to be completed for the set deadline.	**Implementation Goal -** It must be there for **every checkpoint,** even if it's tiny. Remember: Facebook was not built in the last week before the launch	Comments if any
1.Setting up Django & React	2-3 weeks		Layout of how backend and frontend should look like.	
2. Designing multi agent system.	1-2 weeks		The system should provide accurate, scalable, and efficient resume scoring.	
3.LLMs/ Feedback	2-3 weeks		Resume summary and	

generation			improvements.	
4.UI/Testing	1-2 weeks		Improving user interface and testing part	

Future Scope:

The system could integrate with job portals to match resumes with relevant job openings, assisting users with automatic job applications based on their qualifications and preferences.