AISynapse

Confidential

Take-Home Task: Build a Consumer-Facing Resume Fit Analyzer

Objective:
Create a web application that allows users to upload their resume, provide a target job description (via text or a link to a job posting), and receive an AI-generated fit score (0-100) assessing their suitability for the job, along with actionable insights to improve their score and stand out. The tool must also extract job and company details from the provided link to inform the analysis.

Requirements:
1. Frontend (React)
- Build a responsive, user-friendly interface where users can:
  - Upload their resume (PDF format).
  - Input a job description manually or provide a link to a job posting (e.g., from LinkedIn, Indeed).
  - View an AI-generated fit score (0-100) and actionable insights (e.g., "Add experience with AWS" or "Highlight your teamwork skills").
- Ensure the design is intuitive and works seamlessly on desktop and mobile devices.

2. Backend (Python with FastAPI)
- Develop API endpoints to:
  - Handle resume uploads and job description inputs.
  - Extract job details (e.g., title, required skills, qualifications) and company information (e.g., name, industry) from the provided job posting link using web scraping (e.g., BeautifulSoup, Scrapy) or an API (if available).
  - Store user inputs and analysis results in a database (e.g., PostgreSQL or MongoDB).
- Integrate an AI agent to process the resume and job data (see AI Agent section below).

3. AI Agent
- Use a pre-trained language model (Claude API given in resources provided section) to:
  - Extract key skills, experiences, and qualifications from the resume.
  - Analyze the job description and extracted job/company details to identify requirements and preferences.
  - Calculate a fit score (0-100) based on how well the resume matches the job.
  - Generate 3-5 actionable insights to help the user improve their score and stand out (e.g., "Include a project using React" or "Mention your familiarity with Agile methodologies").
- Ensure the score and insights are meaningful and tailored to the specific job and company.

4. Data Privacy & Security
- Securely handle resume data with basic encryption and access controls.

- Follow simple data privacy best practices (e.g., no unnecessary retention of user data).

5. Deployment
   - Containerize the application using Docker for easy setup and testing.
   - Include clear instructions for running the app locally.

Deliverables:
- Source Code: Submit via a GitHub repository with well-organized, documented code.
- Demo Video: A 5-minute Loom video demonstrating the app's functionality, explaining your approach, and providing setup instructions.
- Write-Up: A short document (1 page max) covering:
  - How you'd scale the app for broader use.
  - Key decisions or trade-offs made during development.
  - Ideas for future enhancements.

Time Limit:
72 hours from assignment receipt.
Evaluation Criteria:
- Functionality: Does the app correctly process resumes, extract job/company details, and deliver a fit score and insights?
- Code Quality: Is the code clean, modular, and maintainable? Are best practices followed?
- AI Integration: Does the AI provide accurate, job-specific scores and useful insights?
- User Experience: Is the interface clear and easy to use?
- Efficiency: Was the task completed on time with an optimized solution?
- Innovation: Are there creative features or thoughtful touches beyond the basics?

Resources Provided:
- Temporary Anthropic API key ($6.50 limit): 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

Notes:
- Use React for the frontend and Python (FastAPI) for the backend, but you're free to choose additional libraries.
- If web scraping is blocked (e.g., by anti-scraping measures), you may mock the job data for this task—just note it in your write-up.

Instructions for Candidates
1. You'll receive an email with this task, including any API keys or sample data.
2. Complete and submit within 48 hours via GitHub (code), Loom (video), and email (write-up).
3. Prioritize a working solution, then refine the UI or add extras if time allows.
4. In your demo video:
   - Show the app working with a sample resume and job link.

- Explain your AI scoring logic and how you handled job extraction.
- Mention any challenges and how you addressed them.

5. This is your shot to shine—demonstrate your full-stack and AI skills under pressure!