Project Proposal: Utilizing AI for better productivity in the job search space - Generate portfolios, prepare for interviews, and explore hot job openings.

1. Introduction

Background

Job seekers today face a scattered and time-consuming process when it comes to applying for relevant jobs. Manually filtering through job listings and building portfolios that reflect their skills is not only tedious but inefficient. Additionally, many users miss out on great opportunities due to lack of personalization in job search tools.

Objective

The goal of this project is to build an intelligent end-to-end system that:

- Suggests top 10 job opportunities to a user based on their resume.
- Automatically generates a customizable personal portfolio.
- Provides filtered interview questions tailored to the specific job role.
- Makes use of GitHub Pages for free hosting of portfolio websites.

2. Project Overview

Scope:

- This project aims to build an integrated system for automating the resume pre-processing, LinkedIn job scraping, and personalized portfolio creation pipelines.
- The system will begin by ingesting and parsing user-uploaded resumes, using large language models (LLMs) for data cleaning and standardization, storing structured resume data in Snowflake for easy retrieval.

- It will then scrape LinkedIn job listings, applying keyword-based filtering to focus on specific job categories and storing the data in Snowflake for incremental processing.

- Finally, the system will generate personalized portfolio templates based on user resumes, automatically create a GitHub repository for each user, and deploy the portfolio to GitHub Pages via a CI/CD pipeline

- This end-to-end solution ensures seamless processing, personalized job matching, and automated portfolio generation and hosting.

Stakeholders

- **Job Seekers**: Primary users who will benefit from automated job matching and portfolio creation.

- **Recruiters**: Indirect stakeholders who may benefit from well-structured portfolios and relevant candidates.

3. Problem Statement

Current Challenges

- Difficulty in identifying jobs that are a strong match for user skills.

- Lack of easy-to-use tools for building professional portfolios.

- No personalization in most interview preparation tools.

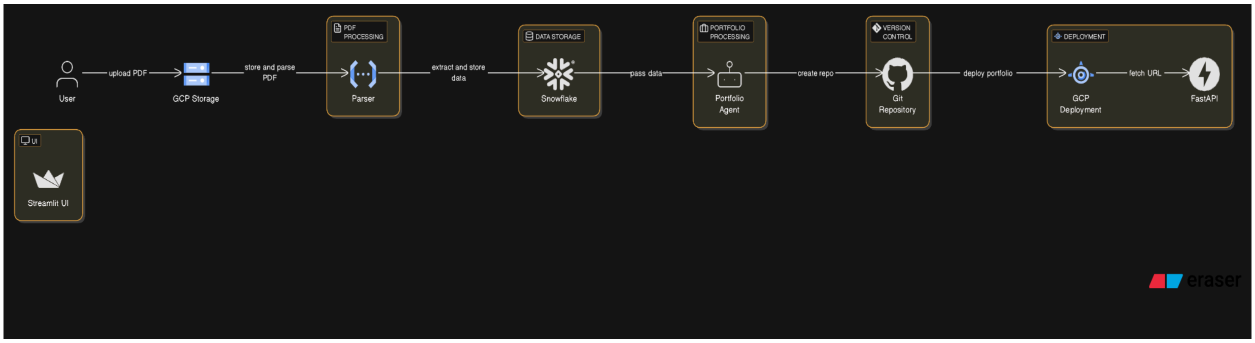
Opportunities

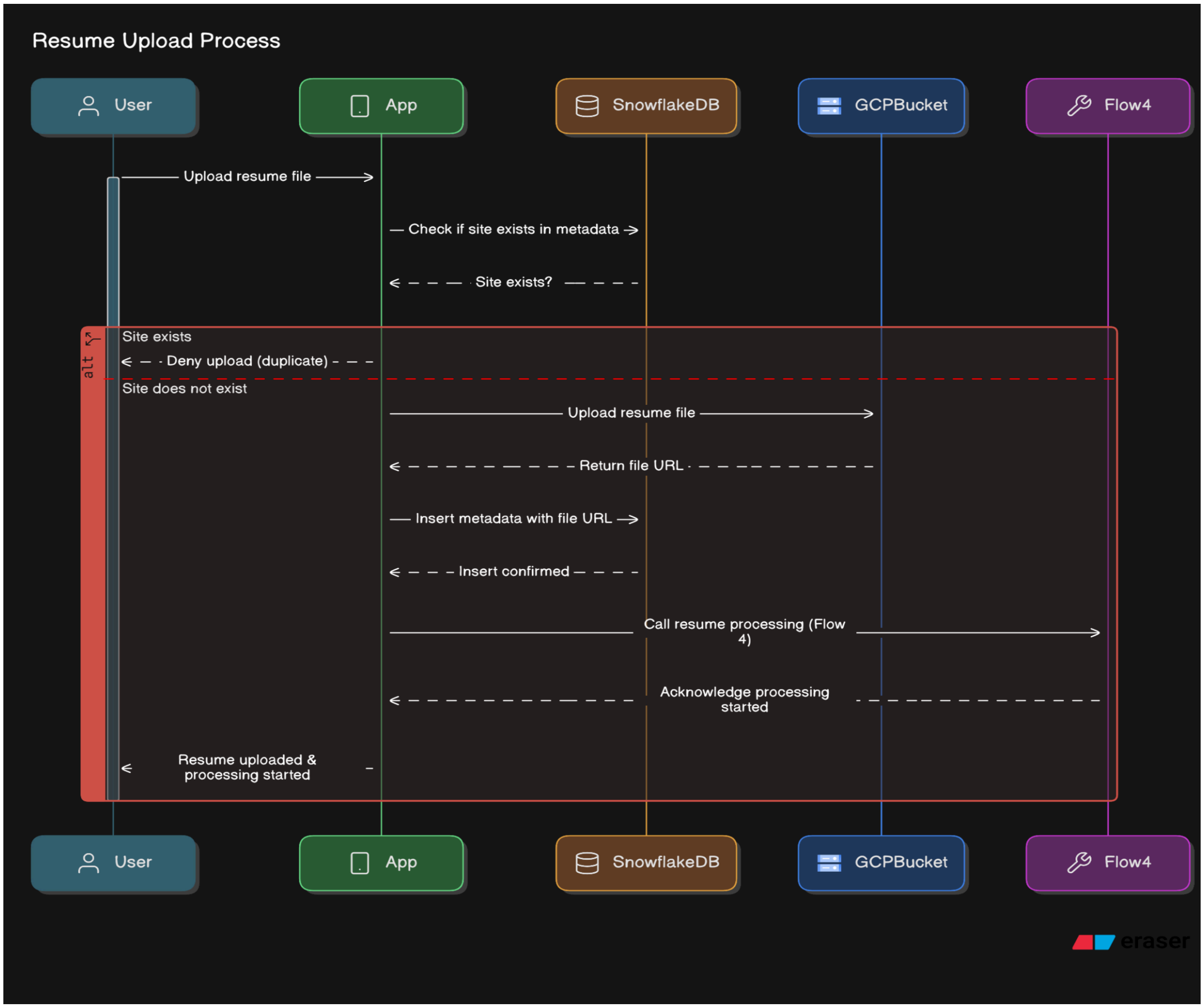
- Automating the resume parsing and job filtering process with LLMs.

- Offering portfolio creation with customizable themes.

- Enhancing interview readiness with frequently asked questions available on the web.

Architecture Diagram




Resume Upload Process

4. Methodology

Data Sources

- **Resume**: Uploaded by the user.
- **Job Listings**: Scraped from LinkedIn.
- **Interview Questions**: DataCamp/GeeksForGeeks 50+ sources.

Technologies and Tools

- **FastAPI**: Backend logic and API endpoints.
- **Streamlit**: Frontend UI for resume upload and template selection.
- **GitHub API**: To create repositories and push portfolio websites.
- **LLM**: Resume parsing, job matching, and generating interview questions.
- **Agentic Framework** : Langgraph
- **Jinja**: For creating customizable HTML templates for portfolios.
- **Snowflake**: Application Database
- **Tavily and bs4**: Interview Data Collection and standardization

Data Pipeline Design

- **Resume Pre-Processing Data Pipeline**
  - **Resume Ingestion** : Uploaded by user.
  - **Parsing with LLM for data cleaning and standardization** : Structured JSON output with skills, experience, education, etc.
  - **Staging :** divide user resume and store as json in snowflake for easy content retrieval

- **LinkedIn Job Scraping Data Pipeline**
  - **LinkedIn Scraping** : Job details like title, company, description, link, etc.

    - **Keyword based filtering :** specific to 3-4 job categories (Eg: Data Scientist, Data Engineer etc.)
    - **Stage to S3 :** backed up snowflake external tables and streams for incremental data processing

- **Portfolio Creation**
    - Generate template files using LLM personalized to user resume
    - Create git repository for users (on-demand) and store the updated files
    - Trigger deployment with CI/CD enabled (github pages)

Data Processing and Transformation

- Resume text converted to structured JSON using LLM.

- Job postings cleaned and filtered based on JSON data.

- Jinja templates dynamically filled using JSON data.

- Portfolio pushed to user-specific GitHub repo using GitHub API.

5. Risks and Mitigation Strategies

Identify Risks

- LinkedIn anti-scraping policies.

- LLM inaccuracies in parsing or job matching.

- GitHub Pages limitations (static only).

Mitigation Strategies

- Use proxy rotation and rate limiting for scraping.

- Implement validation layers for LLM output.

- GitHub Pages, explore netlify or google bucket static hosting.

7. Expected Outcomes and Benefits

Measurable Goals

- Recent job matching.
- Portfolio created within 1 minute of resume upload.
- Interview questions aligned with job description and skills.

Timeline

Week 1 (April 6 – April 12)

Date	Task	Assigned To
Apr 6	Finalize system design, APIs, data flows	Tanmay + Sathya
Apr 7	Build resume upload + parser using LLM (convert to JSON)	Tanmay
Apr 8	Scrape job listings from LinkedIn (build reusable scraping script)	Ankit
Apr 9	Create matching logic (match % based on skill + experience overlap)	Tanmay + Ankit

Apr 10	Snowflake architecture design for incremental data Processing, prototyping	Sathya
Apr 11	Integrate job suggestions logic with FastAPI	Tanmay
Apr 12	Test JSON output → job filtering pipeline	Tanmay + Ankit

Week 2 (April 13 – April 18)

Date	Task	Assigned To
Apr 13	Build portfolio selection UI in Streamlit (theme picker)	Sathya
Apr 14	Setup Jinja templates for 2 sample portfolio themes	Ankit
Apr 15	Integrate JSON data into Jinja and auto-generate HTML	Tanmay
Apr 16	Add Agentic LLM-based interview question generation module	Sathya
Apr 17	Validation of Q&A, testing	Ankit

Apr 18	Final testing, deploy and demo-ready MVP	All Team

Expected Benefits

- The data-as-a-service feature will help users to get quality job data rather than stale job posts.
- Simplifies the job search and portfolio building process.
- Personalizes the experience for each user.
- Helps users stand out with well-matched opportunities and polished profiles.

8. Conclusion

This project addresses a real gap in the job-hunting process by merging resume intelligence, job relevance scoring, and portfolio automation with popular interview questions and critiquing user’s response. It not only saves time but helps users present themselves professionally and prepare better for interviews. With LLMs, GitHub Pages, and a clean user interface, with adding productivity while job search