Final Project Documentation: AI Career Redirection Mentor

- **Project Title:** AI Career Redirection Mentor
- **Participant:** Sanjay A
- **Event:** Agentic AI Hackathon
- **Submission Date:** June 20, 2025

1. Executive Summary

The AI Career Redirection Mentor is a full-stack, AI-powered web application designed to assist students facing repeated job rejections. The platform provides personalized, actionable guidance by leveraging a sophisticated multi-agent system to analyze a user's professional history, identify core strengths and weaknesses, and generate tailored career pivots. The system suggests realistic alternative careers, creates detailed upskilling roadmaps, and connects students to a relevant support network. This project successfully implements a complex, five-agent workflow as required by the hackathon, delivering a polished, end-to-end solution that addresses a significant real-world challenge for aspiring professionals.

2. Project Architecture

The application is built on a modern, decoupled, multi-layer architecture designed for scalability and clear separation of concerns.

2.1. Architectural Layers

- **Frontend (Presentation Layer):**
  - **Technology:** React.js
  - **Responsibility:** Provides the complete user interface for student data upload (resume, rejection feedback, etc.) and a dynamic, tabbed results dashboard for visualizing the final analysis.
- **Backend (Orchestration Layer):**
  - **Technology:** Python with the FastAPI framework.
  - **Responsibility:** Manages the API for data handling, validates inputs, and orchestrates the multi-agent workflow using LangGraph to manage state and trigger agents sequentially.
- **AI Integration (Intelligence Layer):**
  - **Technology:** Google Gemini, LangChain, LangGraph.
  - **Responsibility:** Leverages Large Language Models (LLMs) for the core analytical tasks of rejection analysis, strength identification, and roadmap generation. It also implements Retrieval-Augmented Generation (RAG) for suggesting career alternatives.

- **Database (Data & RAG Layer):**
    - **Technology:** Weaviate (Vector Database).
    - **Responsibility:** Stores knowledge bases for careers and support networks. It supports both vector search for RAG and keyword filtering, and is designed to persist final analyses for each student.

2.2. Architecture Diagram

This diagram illustrates the high-level system architecture and the flow of information between the different layers.

```
graph TD
  subgraph "User Layer"
    User_Start("User Uploads Documents")
    User_End("Views Personalized Dashboard")
  end

  subgraph "Frontend Layer (React.js)"
    UI_Form["Data Upload UI"]
    UI_Dashboard["Results Dashboard (Tabs)"]
  end

  subgraph "Backend Layer (FastAPI & LangGraph)"
    API["API Endpoint (/analyze)"]
    Workflow["[Agentic Workflow Engine]"]
  end

  subgraph "AI & Data Services"
    Gemini["Google Gemini (LLM)"]
```

```
        Weaviate["Weaviate Database (RAG)"]

    end


    %% Define the flow of events

    User_Start --> UI_Form;

    UI_Form -->|1. POST Request with Files| API;

    API -->|2. Invokes Workflow| Workflow;

    Workflow -->|3. Agents perform Generative Tasks| Gemini;

    Workflow -->|4. Agents perform RAG & Keyword Search| Weaviate;

    Gemini --> Workflow;

    Weaviate --> Workflow;

    Workflow -->|5. Returns Final Aggregated JSON| API;

    API -->|6. Sends API Response| UI_Dashboard;

    UI_Dashboard -->|7. Renders All Results| User_End;
```

3. Folder Structure

The project is organized into distinct `backend` and `frontend` directories, with a clear and modular structure within each.

```
/ai_career_mentor/

|

|--- backend/

|    |--- app/

|    |    |--- __init__.py

|    |    |--- agents/

|    |    |    |--- __init__.py
|    |    |    |--- rejection_analyzer_agent.py
|    |    |    |--- strength_finder_agent.py
|    |    |    |--- career_generator_agent.py
|    |    |    |--- upskill_planner_agent.py
|    |    |    |--- support_matcher_agent.py
|    |    |--- graph/
|    |    |    |--- __init__.py
|    |    |    |--- agent_graph.py
|    |    |--- services/
|    |    |    |--- __init__.py
|    |    |    |--- weaviate_service.py
|    |    |--- main.py
|    |
|    |--- data/
|    |    |--- career_database.csv
|    |    |--- support_network_database.csv
|    |
|    |--- .env
|    |--- config.py
|    |--- requirements.txt
|    |--- setup_database.py
|
|--- frontend/
|    |--- public/
|    |    |--- index.html

|    |--- src/

|    |    |--- components/

|    |    |    |--- FileUpload.js

|    |    |    |--- ResultsDashboard.js

|    |    |--- App.css

|    |    |--- App.js

|    |    |--- index.js

|    |

|    |--- .env

|    |--- package.json

|

|--- README.md
```

4. Agentic Workflow

The core of the application is a sequential pipeline of five specialized AI agents.

1. **Rejection Pattern Analyzer Agent**

    - **Function:** Analyzes the student's rejection history by processing interview feedback, resume critiques, and coding test results.
    - **Input:** Rejection feedback (text), resume (PDF/DOCX), coding test results (text/CSV).
    - **Output:** A structured JSON object identifying and labeling key rejection patterns (e.g., "coding: low accuracy").

2. **Strength Finder Agent**

    - **Function:** Identifies the student's strong areas by analyzing their resume (projects, skills) and past performance data (hackathon wins, achievements).
    - **Input:** Resume (PDF/DOCX), performance data (text/CSV).
    - **Output:** A structured JSON object listing specific, evidence-backed strengths (e.g., "frontend: React proficiency").

3. **Career Alternative Generator Agent (RAG-enabled)**

   - **Function:** Takes the identified strengths and weaknesses and uses Retrieval-Augmented Generation (RAG) to query a career database, suggesting viable and realistic alternative career paths.
   - **Input:** The JSON outputs from the Strength Finder and Rejection Pattern Analyzer agents, plus a career database.
   - **Output:** A structured JSON object with at least three career recommendations, complete with descriptions and rationale.

4. **Upskill Planner Agent**

   - **Function:** For each recommended career, this agent designs a personalized upskilling roadmap with at least five milestones, including specific resources like online courses and project ideas.
   - **Input:** The JSON outputs containing the career recommendations and the student's strengths.
   - **Output:** A structured JSON object detailing the upskilling plan for each career path.

5. **Support Matchmaking Agent**

   - **Function:** Connects the student with relevant support networks by performing a keyword search on a database of industry mentors, bootcamps, and peer groups.
   - **Input:** The JSON outputs containing the career recommendations and the student's strengths.
   - **Output:** A structured JSON with actionable connection details for at least one relevant support option per student.

5. Conclusion

The AI Career Redirection Mentor project successfully meets all the objectives outlined in the project statement. It provides a complete, end-to-end solution featuring a polished user interface and a sophisticated multi-agent AI backend. The system delivers personalized, data-driven, and actionable career guidance, demonstrating a powerful and practical application of Agentic AI principles.