AI Research Assistant (AIRA): Search, Summarize & Cite

CSCI 6180 Software Design and Development Project Vision Document

Team Members:

Dimitri Nanmejo
Ahmeed Yinusa
Shang Chen

Team Number: Team 3

September 12, 2025

A.I. Disclaimer: All work for this assignment was completed by ourselves and entirely without the use of artificial intelligence tools such as ChatGPT, MS Copilot, other LLMs, etc.

1 Problem Statement

Students, researchers, and professors expend much time searching, filtering, and synthesizing academic sources. The PDFs are challenging to query, and the process of getting methods, datasets, and results is manual and prone to errors. Numerous AI-powered research assistant software or web apps can offer useful prose. However, they are unreliable when writing an academic paper, because they do not include grounded in-text citations and references.

2 Proposed Solution

AIRA is a web app that combines paper discovery (Semantic Scholar), AI summaries, and "Chat with my PDF". The React frontend talks to a Node/Express AI gateway, which can route to DeepSeek over OpenRouter (default), others (OpenAI, Claude, Gemini), and even open-source LLMs hosted on Hugging Face. For documents, AIRA parses PDF

documents, chunks + embeds text, stores vectors in Supabase pgvector, answers questions with inline citations to the exact snippets that are sources of the answer.

3 Target Users

Primary users: undergraduate and graduate students, research assistants, and faculty conducting a literature review, class reading, or report.
Secondary users: users in industries who need to get fast and negotiated information from PDFs and saved papers.

4 Key Features

- **Paper Discovery (Semantic Scholar):** Users can search various topics and view titles, authors, year, venue, abstract, and links.
- **Abstract Summaries:** The abstracts will be one-click bullet points with a one-line takeaway.
- **PDF Chat with Citations:** The flow of the PDF will be to upload a file → parse → index → answers cite snippets.
- **Writer Workspace:** Notes — Editor — AI Assistant ("insert evidence") with sources.
- **Save and History:** Save papers, recent activity, and chats (according to the user).
- **Pluggable AI Providers:** The default third-party AI API will be DeepSeek (from OpenRouter). Optionally, other general foundational LLMs such as OpenAI/Claude/Gemini or open-source LLMs can be employed.
- **Security/Accessibility:** The backend will have server secrets, Supabase Auth, chat stream available to users, and an interactive UI.

5 Feasibility Check

5.1 Timeline (12 weeks)

We are projecting approximately 12 weeks for the execution of the project for the semester. The outlined weeks and steps that will be followed are listed below:

- **Week 1-2:** The user interface (UI), including the scaffolding of the various pages such as the Home, Dashboard, Writer, Recent, and more if needed, and the integration of the search API using (Semantic Scholar), the data, and login authentication using (Supabase/Auth configuration).
- **Week 3-4:** We will implement the AI integration for these weeks, using DeepSeek with OpenRouter through AI Gateway, for chat, summarization, and batch abstract summaries.

- **Week 5-6:** We will implement the PDF parsing endpoint (LlamaParse or Unstructured or Adobe Extract or any other PDF parsing resource), and Supabase Storage.
- **Week 7-8:** We will implement RAG with text chunks for paper citations.
- **Week 9-10:** We will polish the UI and accessibility of the web app and check other parts of the app.
- **Week 11-12:** Small fixes will be implemented, afterwards, we will proceed to the documentation and demo.

5.2 Team Skills

Each team member is equipped with the basic skills required for the project. Moreover, we will further learn more of the following required skills, which are more advanced than the basic ones mentioned for the completion of the project. The required skills are listed below:

- React and Tailwind for the frontend development.
- Node & Express for the backend development.
- Semantic Scholar API is used to retrieve research topics.
- Supabase (built on PostgreSQL) for the database.
- Embeddings and simple RAG environment/security (dotenv, Cors) for AI API calling and information retrieval.

5.3 Available Resources

We are employing several required and relevant resources for the project. Below are the resources available for the project:

- **APIs/Platforms:** Semantic Scholar (Free) — Supabase free tier (Postgres + Auth + Storage + pgvector) — OpenRouter low cost (DeepSeek default) — parser API free tiers (LlamaParse/Unstructured/Adobe Extract) — Hugging Face Spaces - Free (OSS Models/Embeddings).
- **Project Tooling:** VS Code, GitHub, Vercel/Netlify, Render/Fly/Heroku (gateway).

6 GitHub

This is the GitHub link for the project: https://github.com/dimsonMba/AIRA