Project Proposal

Research Assistant Agent

Group 6

Team Members:
Jesus Chavez Espino
Mohammad Uraib
Mahdi Mohammadkhani

1. Problem Statement

Students and professionals often struggle to efficiently locate reliable academic sources, evaluate credibility, extract key information, organize findings, and produce properly formatted citations. Research is a multi-step process that can be overwhelming, especially when dealing with large volumes of information from various sources. This frequently results in inconsistent research quality and time-consuming manual tasks. The proposed Research Assistant Agent aims to streamline this workflow by automating information retrieval, summarization, relevance evaluation, and citation generation. The goal is to create an assistant that enhances productivity, reduces cognitive load, and supports users in producing high-quality academic or professional research.

2. Project Option Selected

The team has selected Option 1: Research Assistant Agent. This agent will assist users in performing academic and professional research tasks by retrieving information, evaluating source credibility, generating structured summaries, organizing findings, and creating accurate citations.

3. Agent Design

The agent architecture will consist of the following components: Input Processing: Interprets user prompts and research questions using natural language understanding. Memory System: Stores retrieved documents, user preferences, session history, and extracted summaries for long-term or short-term use. Reasoning Component: Implements the ReAct (Reasoning and Acting) framework to determine when to search, summarize, evaluate credibility, or produce output. Output Generation: Produces organized research reports, summaries, and citation lists in APA, MLA, or Chicago format. This design ensures structured decision-making and consistent performance across complex tasks.

4. Tool Selection

The Research Assistant Agent will integrate at least two external tools: 1. Web Search API – Used to retrieve credible information, articles, academic content, and verified sources. 2. Document Processing or PDF Reader – Allows the agent to parse uploaded documents, extract text, and analyze PDFs. 3. Citation Generator API – Automatically formats references into standard citation styles. The agent will include error handling for API failures, timeouts, and missing data. It will also interpret outputs to ensure accuracy and relevance.

5. Development Plan and Timeline

Week 1: Architecture design, environment setup, selecting APIs and tools. Week 2: Implement core agent logic including the memory system and reasoning loop. Week 3: Add reinforcement learning

feedback features and credibility scoring. Week 4: Perform testing, debugging, documentation writing, and prepare the demonstration video.

6. Evaluation Strategy

Evaluation will focus on accuracy and relevance of retrieved information, quality of summaries, correctness of citations, ability to evaluate source credibility, user satisfaction, and agent robustness under diverse research scenarios.

7. Resource Requirements

Azure AI Studio or Google Colab LangChain or similar orchestration library Python 3.10 or higher Vector database used for information retrieval APIs for web search and citation generation

8. Risk Assessment

API failures can be handled using fallback mechanisms and retries. Hallucinations will be reduced by implementing credibility scoring and verification layers. Token and cost limits can be managed with optimized prompts and caching. Time constraints can be addressed by building a minimal viable product first. Low quality sources will be minimized through filtering and sorting mechanisms.