CAPSTONE PROJECT: AI AGENT CREATION

SELECTED OPTION: RESEARCH ASSISTANT AGENT

Name: Faiza Abdullah

Professor: Patricia Mcmanus

Course: ITAI2376 Deep Learning AI

Deliverable: Project Proposal

PROBLEM STATEMENT

Researching topics efficiently remains a significant challenge for students, academics, and professionals. The process involves sifting through vast amounts of information, evaluating source credibility, summarizing findings, and organizing them into coherent reports—tasks that are time-consuming and prone to error when done manually. This problem is critical because it impacts academic success, productivity, and the quality of research outputs in an era where information overload is common. For instance, students often struggle to find reliable sources or format citations correctly, leading to wasted time or lower grades.

By streamlining these tasks, this virtual research agent ***"Resea"*** will address a pressing need for automation in research workflows and empower users to focus on analysis and creativity rather than tedious data management.

PROJECT OPTION

Research Assistant Agent - "Resea":

Justification: This option was selected due to its broad applicability and potential to transform how individuals conduct research. Unlike narrower tools (e.g., citation generators), this agent offers an end-to-end solution—retrieval, summarization, evaluation, and reporting—making it a versatile assistant. It aligns with the mission to accelerate human scientific discovery by automating foundational research tasks.

Focus Areas: The project will emphasize (1) topic-based information retrieval from diverse sources, (2) accurate summarization using NLP, (3) credibility assessment of sources, and (4) structured report generation with citation support. These aspects ensure the agent meets academic and professional standards while remaining user-friendly.

Limitations: As a capstone project, we were given four options out of which this one looked feasible keeping time, my limited coding aptitude and not much paid subscription of tools at hand. I prefer using Colab however thanks to Professor and Azure we have $100 student

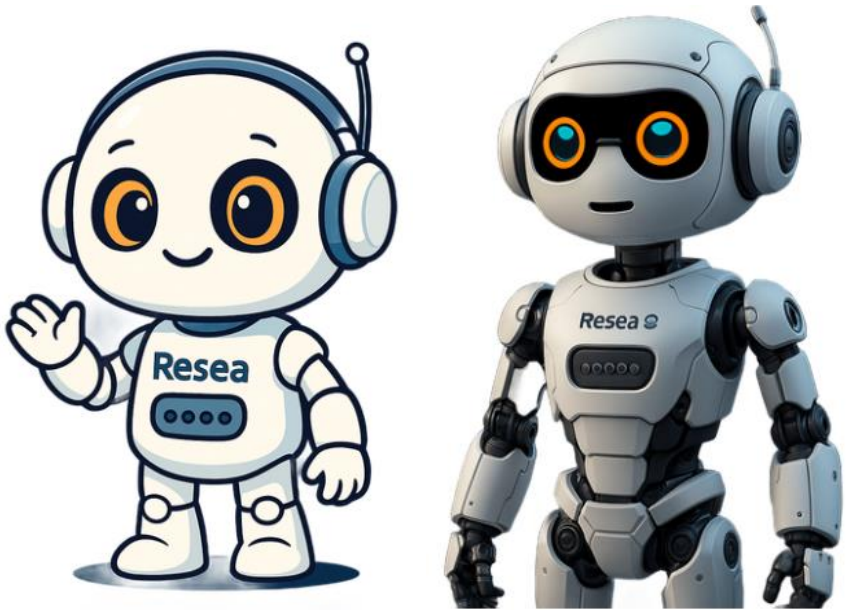
credit which can also be used. My proposal will show pathways through both. The sequence and quality of deliverables might be different once the work will start.

BRANDING

Name: Resea (pronounced "Re-see")

- *Short-form of Research and Re-see illustrates the meaning.*

Brand Persona: *Have created one **animated** and one real-life **Robo** mascot to relate to the right target audience*



AGENT DESIGN

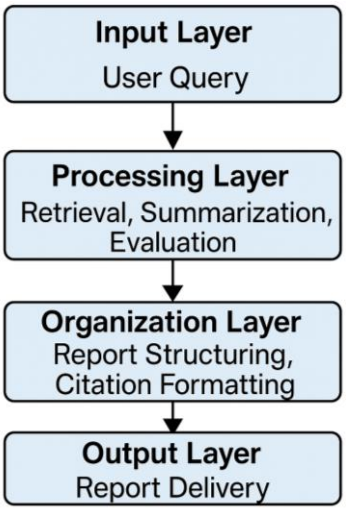
Main Components:

- Input Layer: Accepts a topic via text input (user query)
- Processing Layer:
    - Retrieval Module: Gathers data from web APIs and user uploads.
    - Summarization Module: Condenses content using NLP models.
    - Evaluation Module: Scores source credibility and relevance.
- Reasoning Component:

- Report Generation through the structures findings and citations.

- User Interface / Output Layer: Accepts inputs and delivers outputs.

Architecture Diagram:



Resea Pattern: It follows ReAct (Reasoning + Acting) pattern with memory, responding to user queries while storing past interactions to refine future outputs (e.g., learning preferred citation styles). It will reason source quality and acts by generating a report.

Processing: Inputs (e.g., topic, preferences) are processed by fetching data, summarizing it, evaluating sources, and organizing results into a report. Outputs are delivered as text or interactive displays, with citations formatted per user specification.

TOOL SELECTION

Google Colab:

Google Colab: Free Python environment for running scripts.

Hugging Face (Free Tier): Pre-trained summarization model (e.g., BART).

Why Chosen: Both are free, accessible, and support NLP without subscriptions.

Integration: Colab hosts the script; Hugging Face provides summarization, called via free libraries. Manual web search supplements data.

Microsoft Azure ($100 Credit only):

Azure Cognitive Services (Text Analytics): Summarizes text, extracts key phrases.

Azure Logic Apps: Automates workflow (e.g., process input, call APIs).

Why Chosen: Pre-built, low-code, and within $100 credit limit for prototyping.

Integration: Logic Apps connects input to Text Analytics, outputs to storage/email. Manual search supplements data.

If possible:

Google Scholar API: For academic source retrieval.

BeautifulSoup: For web scraping non-API sources.

citeproc-py: For citation formatting.

SQLite: For lightweight source storage.

Why Chosen: These tools are open-source, widely supported, and optimized for research tasks—Transformers for NLP accuracy, Google Scholar for credibility, and citeproc-py for citation standards.

Integration: APIs and libraries will be called via Python scripts, with SQLite managing metadata and Transformers running locally or on cloud GPUs for efficiency.

DEVELOPMENT PLAN

Realistic Timeline: 10 weeks

Week 1-2: Requirements gathering, initial retrieval module.

Week 3-4: Summarization and evaluation modules.

Week 5-6: Report generation and citation integration.

Week 7-8: UI development and testing.

Week 9-10: Finalize demo, report, video.

Methodology: Waterfall (linear, beginner-friendly).

EVALUATION STRATEGY

Success Measurement:

Accuracy: Summaries match source content (manual review).

Outputs: Generation in <5 minutes.

Credibility: 90% of top sources rated reliable by experts.

Usability: User satisfaction score > 4/5 (survey).

Colab Metrics: Summary length, source retrieval time, citation error rate.

Azure Metrics: API call success rate, runtime, user satisfaction.

Testing: *Test on 3 topics, compare to manual outputs.*

RESOURCE REQUIREMENTS

Computational:

- Laptop + Colab (free GPU) AND / OR
- Azure credits ($100) for Cognitive Services, Logic Apps.

APIs/Services:

- None (manual search, free Hugging Face) OR
- Text Analytics (~$0.50/1000 calls, ~200 calls budgeted).

In case of Azure:

Credits: *$100 covers prototype (e.g., $50 compute, $50 APIs).*

RISK ASSESSMENT

Challenges:

- Learning Curve: Slow progress without coding skills.
- Tool Limits: Free NLP may underperform.
- Manual Effort: Retrieval not fully automated.
- API Limits: Rate limits on Google Scholar.
- Azure Learning: Steep curve for beginners.
- Credit Limits: Overuse depletes $100 early.

Mitigation: Use tutorials, pre-built scripts, manual backups, monitor Azure usage, reliance on Colab would be more.

CONCLUSION:

This still seems wishful thinking at the moment: picking up a project solo and for the first time doing hands-on such a comprehensive project, having midterms, finals and capstone projects in other courses, no coding aptitude or paid subscriptions to appropriate tools, and most importantly, having a super-active toddler by my side.

Additionally, that much time is not available; at the maximum, this needs to be developed within a month of project proposal submission, however, I am up for this challenge.