

# SynthesisTalk Report

## Conversation

User: Project.pdf

User: Summarize key points

Assistant: ■ **\*\*Key Points from Project.pdf:\*\***

Let's think step by step.

### Summary of Key Points from "Project.pdf"

### #### Overview

- **\*\*Course\*\***: CSAI 422: Advanced Topics in Generative AI
- **\*\*Project Name\*\***: SynthesisTalk - Collaborative Research Assistant
- **\*\*Objective\*\***: Build a sophisticated LLM-powered application for interactive research assistance.
- **\*\*Context\*\***: Combines document analysis, web search capabilities, and customizable outputs.

### #### Learning Objectives

1. Implement conversational agency concepts including tool usage and reasoning techniques.
2. Design and build effective LLM workflows for complex tasks.
3. Create a responsive frontend that effectively communicates with LLM-powered backends.
4. Develop practical experience with modern AI application development.
5. Apply collaborative software development practices in an AI context.

### #### Project Requirements

#### **\*\*Core Features\*\***

1. **\*\*Contextual Research Conversation\*\***:
  - Multi-turn conversations maintaining context about research topics.
  - Document uploads (PDFs, text files) with content extraction and analysis.
  - Web search integration for supplementary user knowledge.
  - Maintaining coherent conversation across multiple information sources.
2. **\*\*Intelligent Synthesis Engine\*\***:
  - Extract key information from documents and web searches.
  - Connect related concepts across different sources.
  - Generate insights based on patterns in collected information.
  - Implement advanced reasoning techniques (Chain of Thought or ReAct).
3. **\*\*Flexible Output Generation\*\***:
  - Generate structured summaries in at least two formats.
  - Create simple visualizations based on research findings.
  - Support exporting findings in standard document formats.
4. **\*\*Tool-Enhanced Experience\*\***:
  - Document analysis tools (extraction, summarization).
  - Web search tools for fact-finding and verification.
  - Note-taking and organization tools.
  - Explanation and clarification tools.

#### **\*\*Technical Requirements\*\***

1. **\*\*Frontend\*\***:

- React-based web application with responsive design.
- Chat interface supporting text and document uploads.
- Components for displaying research insights and visualizations.
- Context management UI showing current research topics and sources.
- Real-time feedback for lengthy operations.

## 2. **Backend**:

- FastAPI or Flask backend serving LLM integration.
- Tool management system for handling different LLM tool calls.
- Document processing pipeline for extracting and indexing content.
- Conversation history management system.
- Implementation of advanced reasoning techniques (Chain of Thought, ReAct).

## 3. **LLM Integration**:

- Integrate with an LLM API (Gemini, Groq, NGU LLM).
- Define and implement tools for various research functions.
- Create a workflow that effectively orchestrates multiple tools.
- Implement self-correction mechanism.

## #### Team Structure and Roles

- **Frontend Development and UX Design**
- **Backend API and Tool Integration**
- **LLM Workflows and Reasoning Implementation**

## #### Development Approach

- Encouraged to use Aider (or similar AI-assisted development tools).
- Hands-on experience with building AI systems and using AI tools in the development process.

## #### Deliverables

1. **Working Prototype**: Functional SynthesisTalk application.
2. **Source Code**: Well-organized GitHub repository, regular commits, README, comments.
3. **Technical Report (5-8 pages)**: System architecture, implementation details, tool integration, challenges.
4. **Presentation**: 15-minute team presentation with live demonstration and reflection on development process.

## #### Evaluation Criteria

- Functionality (40%)
- Usability (20%)
- Scalability (15%)
- Efficiency (10%)
- Creativity (15%)

## ### The Answer is:

The SynthesisTalk project requires teams of three students to build an intelligent research assistant using