SynthesisTalk Report

Conversation

User: Summarize the key points from the following documents:

--- Content of Project.pdf ---
CSAI 422: Advanced Topics in Generative AI
Group Project:
SynthesisTalk - Collaborative Research
Assistant
Overview
In this group project, you will apply the concepts learned throughout this
course to build a sophisticated LLM-powered application that demonstrates
conversational agency, tool usage, and advanced reasoning techniques. Working
in teams of three, you will create "SynthesisTalk," an intelligent research assistant
that helps users explore complex topics through an interactive, conversational
interface.
SynthesisTalk combines document analysis, web search capabilities, and cus-
tomizable outputs to support deep research on complex topics. The system will
demonstrate how LLM-based conversational agents can effectively utilize tools,
maintain context, apply reasoning techniques, and generate useful outputs in
various formats.
Learning Objectives
• Implement conversational agency concepts including tool usage and rea-
soning techniques
• Design and build effective LLM workflows for complex tasks
• Create a responsive frontend that effectively communicates with LLM-
powered backends
• Develop practical experience with modern AI application development
• Apply collaborative software development practices in an AI context
Project Requirements
Core Features
1. Contextual Research Conversation
• Implement multi-turn conversations that maintain context about the
research topic
• Support document uploads (PDFs, text files) with content extraction
and analysis
• Integrate web search to supplement user knowledge and document
content
• Maintain 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 at least one advanced reasoning technique (Chain of
Thought or ReAct)
1

3. Flexible Output Generation

- Generate structured summaries in at least two different formats
- Create simple visualizations based on research findings
- Support exporting findings in a standard document format

4. Tool-Enhanced Experience
- Implement at least four different tool types:
  – 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
- Build a React-based web application with responsive design
- Implement a chat interface supporting text and document uploads
- Create components for displaying research insights and visualizations
- Design a context management UI showing current research topics and sources
- Provide real-time feedback for lengthy operations

2. Backend
- Develop a FastAPI or Flask backend serving LLM integration
- Implement a tool management system for handling different LLM tool calls
- Create a document processing pipeline for extracting and indexing content
- Design a conversation history management system
- Implement at least one advanced reasoning technique (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 some form of self-correction mechanism

Team Structure and Roles
Each team of three students should distribute responsibilities across these core areas:
- Frontend development and UX design
- Backend API and tool integration
- LLM workflows and reasoning implementation

While each team member will have primary responsibility for one area, all members should collaborate across components to ensure effective integration.
2

Development Approach
You are encouraged to use Aider (or similar AI-assisted development tools) to help with implementation. This project is designed to give you hands-on experience with both building AI systems and using AI tools in the development process.

Deliverables

1. Working Prototype
- Functional SynthesisTalk application with all core features
- Deployed version or clear instructions for local setup

2. Source Code
- Well-organized GitHub repository with documentation
- You must have REGULAR commits throughout the project duration,

showing the contribution of different team members, no "big bangs"!
• README with setup instructions
• Comments explaining key implementation decisions
3. Technical Report (5-8 pages)
• System architecture description
• Implementation details for reasoning techniques
• Description of tool integration approach
• Discussion of challenges encountered and solutions
• Evaluation of system performance
4. Presentation
• 15-minute team presentation
• Live demonstration of key features
• Discussion of technical implementation
• Reflection on development process and lessons learned
Evaluation Criteria
Your project will be evaluated based on the following criteria:
1. Functionality (40%)
• Successful implementation of al
Assistant: ### Key Points from Project.pdf

- **Course Overview**: CSAI 422: Advanced Topics in Generative AI, focusing on building a sophisticat
- **Project Objective**: Develop an intelligent research assistant that demonstrates conversational ager
- **Core Features**:
  - Contextual Research Conversation
  - Intelligent Synthesis Engine
  - Flexible Output Generation
  - Tool-Enhanced Experience
- **Technical Requirements**:
  - Frontend: React-based web application with a chat interface and responsive design.
  - Backend: FastAPI or Flask for LLM integration, document processing pipeline, conversation history r
  - LLM Integration: Integration with an LLM API (Gemini, Groq, NGU LLM), tool management system, a
- **Team Structure**:
  - Frontend development and UX design
  - Backend API and tool integration
  - LLM workflows and reasoning implementation
- **Development Approach**: Use Aider or similar AI-assisted tools for hands-on experience with AI sys
- **Deliverables**:
  - Working Prototype
  - Source Code
  - Technical Report (5-8 pages)
  - Presentation
- **Evaluation Criteria**:
  - Functionality
  - User Experience
  - Tool Integration
  - Reasoning Techniques
  - Scalability and Performance

Summarized Key Points and Extracted Document

http://localhost:8000/uploads/80a0650d-cdd5-4da6-bfe3-2b3c477a9d71.pdf