Agentic AI Research & Writing Tool

By: Dhruv

Mentor: [Mentor’s Name]

Organization: [Company/Institution]

Date: [Today’s Date]

# 1. Introduction

A Streamlit-based web application that allows users to perform AI-powered research and content generation based on user-selected topics and preferences such as language, depth, style, and search engine.

# 2. Objective

- Automate research across multiple sources like Wikipedia and DuckDuckGo.

- Allow multilingual content generation (Hindi, Bengali, Tamil, Marathi).

- Summarize and structure content into subtopics.

- Provide downloadable results.

# 3. Tech Stack

Frontend: Streamlit  
AI/LLM: Google Gemini API (gemini-pro)  
Language Handling: Python dotenv  
Data Sources: Wikipedia, DuckDuckGo API  
File Handling: Python, Markdown

# 4. Project Setup Instructions

1. Clone the repository:  
 git clone <your repo link>  
  
2. Set up virtual environment:  
 python -m venv venv  
 venv\Scripts\activate  
  
3. Install requirements:  
 pip install -r requirements.txt  
  
4. Add `.env` file:  
 GEMINI\_API\_KEY=your-api-key  
  
5. Run the app:  
 streamlit run app.py

# 5. How It Works

- User Inputs: Topic, Language, Depth, Style, Source.  
- Agent Flow:  
 1. Planning Agent: Breaks topic into subtopics using Gemini.  
 2. Research Agent: Fetches info from selected engine.  
 3. Writing Agent: Generates article.  
 4. Review Agent: Checks article quality.  
 5. Summary Agent: Provides TL;DR.  
- Output: Full article + summary + download option.

# 6. Features

- Multilingual output  
- Plug-and-play API support  
- Real-time article writing  
- Streamlit UI  
- Downloadable markdown file

# 7. Challenges Faced

- Gemini API syntax issues (`generate\_text` vs `generate\_content`)  
- Environment variable handling  
- Wikipedia disambiguation errors  
- Output formatting for different languages

# 8. Future Improvements

- Add support for Google Search or Serper properly  
- Error-handling UI feedback  
- Save history of generated articles  
- Add voice input option

# 9. Screenshots

Insert the following screenshots manually:  
- Streamlit UI screenshot  
- Output article  
- Error handling  
- `.env` sample (blur API key)

# 10. Sample Output

Paste one complete run of your article and summary output here.

# 11. Conclusion

This project demonstrates how agentic AI systems can simplify research and writing tasks. By integrating Streamlit, Gemini AI, and real-time APIs, users can generate content tailored to their needs. Future updates can enhance reliability and add richer features.