Smart AI Communication Agent using n8n & Ollama

Chat & Image Analysis with Automated Email Response System

Project Overview

• This project is an AI-powered automation workflow built using n8n and Ollama.

• It accepts both text and image inputs from users through a webhook API.

• Based on the input type, it performs intelligent processing and returns meaningful responses.

• If an email address is provided, the system automatically sends AI-generated content via Gmail.

• The entire system runs using Docker containers for easy deployment and scalability.

Problem Statement

• Manual handling of user queries and image analysis is time-consuming.

• Integrating AI responses with communication tools like email is complex.

• There is a need for an automated system that can understand multimodal input and respond intelligently.

• Businesses require agentic AI workflows without heavy backend development.

Objectives

• Build an AI workflow that handles both text and image inputs.

• Use open-source local LLM models via Ollama.

• Automate response generation using AI.

• Send responses to users through email automatically.

• Deploy the entire system using Docker containers.

Technologies Used

• n8n – Workflow automation and orchestration.

• Ollama – Local LLM and vision model hosting.

• Granite 3.2 Vision Model – For chat and image understanding.

• Docker & Docker Compose – Containerized deployment.

• Postman – API testing and webhook triggering.

• Google Cloud Console – Gmail OAuth and API configuration.

• Trae IDE – Development environment for Docker and workflow setup.

System Architecture

• User sends request via Postman or frontend.

• Request hits n8n webhook endpoint.

• Workflow checks whether input contains image or only text.

• Image is converted to Base64 if present.

• Prompt and image are sent to Ollama model using HTTP Request node.

• AI generates response based on input.

• If email is provided, Gmail node sends response to user.

• Respond to Webhook node sends final response back to user.

Workflow Logic

• Webhook receives JSON or multipart form data.

• If Node checks for presence of binary image data.

• Text-only path goes directly to Ollama prompt.

• Image path includes Base64 conversion before model call.

• Merge node combines final AI output for response handling.

• Conditional Gmail node sends email if email field exists.

Docker Deployment

• Two containers are used: Ollama and n8n.

• Docker Compose manages both services.

• Ollama exposes model API to n8n via internal network.

• Volumes are used to persist model and workflow data.

• Docker Desktop is used to monitor and manage containers.

API Testing with Postman

• Text requests are sent as raw JSON.

• Image requests are sent using form-data.

• Fields used: file (image), message (text), email (optional).

• Postman verifies webhook responses and AI output.

• Helps debug workflow and HTTP request formats.

Gmail Integration

• Google Cloud Console used to create OAuth credentials.

• Gmail API enabled in Google project.

• Client ID and Client Secret configured in n8n credentials.

• Gmail node sends AI-generated content to provided email.

• Supports automated communication use cases.

Challenges Faced

• Handling binary image data and Base64 conversion.

• Configuring conditional workflows for multimodal input.

• Ollama model memory limits and process crashes.

• Webhook payload formatting issues.

• OAuth configuration errors in Gmail API setup.

Solutions Implemented

• Used Move Binary Data node for Base64 conversion.

• Applied If node logic to detect image presence.

• Optimized prompts and model selection.

• Corrected Postman request formats.

• Reconfigured OAuth redirect URLs and scopes.

Project Outcome

• Successfully built an agentic AI workflow.

• System handles chat and image-based queries.

• Automatic email delivery works reliably.

• Fully containerized and portable deployment.

• Demonstrates real-world AI automation skills.

Learning Outcomes

• Hands-on experience with n8n automation.

• Understanding of multimodal AI workflows.

• Docker-based microservice deployment.

• API integration and debugging skills.

• OAuth and Gmail API configuration knowledge.

Use Cases

• Customer support automation.

• Image-based query handling systems.

• AI-powered email assistants.

• Internal business automation agents.

• Educational AI tools.

Future Enhancements

• Add frontend UI for users.

• Support multiple AI models dynamically.

• Add database for chat history.

• Enable voice input and output.

• Deploy on cloud infrastructure.

Conclusion

• This project demonstrates strong understanding of AI automation pipelines.

• It shows integration of LLMs, workflows, APIs, and communication tools.

• The system reflects practical agentic AI implementation skills.

• It is suitable as a portfolio and showcase project.