Isolated LangGraph Node: Unregistered User Handler

This project is an isolated LangGraph node implementation designed to manage unregistered users in a conversational AI system. Built using LangGraph, Gemini 1.5 Flash, and FastAPI, the system handles registration flow, access control, and user interaction states. It is packaged and executed using the blazing-fast uv Python runner.

# 🧠 Overview

When a user begins interacting with the agent, the system starts by checking if the user is registered or not. Based on this condition:  
  
- If registered, the conversation is routed to the `registered` node.  
- If not registered, the conversation is handled by the `unregistered` node, which supports basic interactions and initiates the registration process if needed.

# 🚦 Agent Flow Logic

1. User sends a message.  
2. The system checks if the user is registered.  
3. If unregistered:  
 - Handles general queries.  
 - Blocks restricted requests (e.g., booking, leads).  
 - Prompts for registration.  
 - Calls FastAPI to register and confirm user data.  
 - If confirmed, reroutes to registered node.  
4. If already registered, continues to the registered node.

# ⚙️ Tech Stack

- LangGraph – for stateful agent workflows  
- Gemini 1.5 Flash – as the foundation LLM  
- FastAPI – lightweight and efficient backend  
- CSV – for file-based user data storage  
- uv – a next-gen Python package manager and runner

# 🗂️ Project Structure

main\_agent.py # LangGraph logic (Unregistered Node)  
tools\_file.py # Tool functions for register and confirm  
prompts.py # Prompt templates used in node  
fastapi\_app.py # FastAPI backend to handle user data  
leads.csv # File storing registered user data  
pyproject.toml # uv configuration and script definition  
.env # Gemini API Key  
README.md # Project documentation

# 🚀 Getting Started

1. Install uv:

pip install uv

3. Install Dependencies:

uv pip install -r requirements.txt

4. Set API Key in `.env`:

GOOGLE\_API\_KEY=your\_gemini\_api\_key\_here

# ▶️ Running the Agent

Run the isolated node from pyproject.toml script:

uv run isolated-node-unregister

# 🌐 Running FastAPI Backend

Start the FastAPI backend to manage user data:  
uvicorn fastapi\_app:app --reload