🔧 Vega-Agent – Project Setup & Handover Documentation

📌 Overview

Vega-Agent is a Streamlit-based AI career assistant platform tailored to help users (especially women restarters) with personalized job search, career guidance, resume building, and roadmap planning. The system integrates chatbots (LLM agents), HerKey dynamic data fetching, and external job search using Firecrawl, Tavily, and SERP.

🚀 Project Setup Instructions

1️⃣ Clone the Repository

```
git clone https://github.com/nandini-kuppala/Vega-Agent.git
cd Vega-Agent
```

2️⃣ Install Dependencies

Install all required packages from the `requirements.txt` file:

```
pip install -r requirements.txt
```

> ✅ Make sure you have Python 3.9 or above installed.

📁 Folder Structure

Refer to the screenshots for visual understanding. Here's a detailed breakdown:

Folder/File	Description
`Agentic_ai/`	Main chatbot logic and agent code. Includes job recommendations, external job search, and RAG (retrieval-augmented generation) integration.
`assets/`	Frontend animation files used across the platform.
`backend/`	MongoDB connection handlers and data operations.

Folder/File	Description
`Herkey_MCP_servers/`	MCP (multi-channel processing) servers to dynamically fetch job, event, and community data from HerKey.
`Knowledge/`	Contains daily career knowledge/insight snippets.
`Resume/`	ATS-friendly Resume builder code.
`Roadmap/`	Code to generate customized learning/career roadmaps.
`Screens/`	Streamlit UI pages for the app.
`session_context/`	Handles multi-turn conversations (refer to session flow diagram in `README.md`).
`skill_assessment/`	Skill assessment related modules (optional integration).
`user_profile/`	Sign-up, login, and user profile creation logic.
`utils/`	Common UI styles and utility functions.
`app.py`	**Main Streamlit app launcher** integrating all pages and logic.
`.gitignore`	Standard git ignore file.
`README.md`	Project summary, diagram, and instructions.
`requirements.txt`	List of dependencies.

💡 Details in `Agentic_ai/`

This is the core logic folder. Important files include:

- `chatbot.py` : Main chatbot interface and logic.
- `career_guide.py` : Classifies user queries and maps them to job roles.
- `external_job_search.py` : Uses **Firecrawl** and **Tavily** for fetching jobs externally.
- `herkey_rag.py` : Pulls job, event, and session recommendations from scraped HerKey JSONs using RAG.
- `Herkey_data/` : Contains scraped JSON files from the HerKey website.
- `Scraping_herkey.ipynb` : Code to scrape HerKey using Selenium + BeautifulSoup.
- `recommendations.json` : Static sample recommendations (for fallback).

🌐 Running the Application Locally

Step-by-Step:

1. **Ensure all dependencies are installed.**
2. **Create a** `.env` **file** in the root folder:

```
MONGO_URI = your_mongo_uri
GEMINI_API_KEY=your_gemini_key
GROQ_API_KEY=your_groq_key
TAVILY_API_KEY=your_tavily_key
FIRECRAWL_API_KEY=your_firecrawl_key
SERP_API_KEY=your_serp_key
```

> These keys were previously fetched from `st.secrets` during Streamlit deployment. In local setup, use the `.env` file and load it using `dotenv`.

3. **Run the app:**

```
streamlit run app.py
```

This opens the application in your browser.

🤖 LLM Customization

You can easily swap the LLM by changing the model in `agent_def()` function:

```python
def general_purpose_agent():
    llm = ChatLiteLLM(
        model="gemini/gemini-2.0-flash-lite",
        api_key=GEMINI_API_KEY,
        temperature=0.2
    )
```

Replace with any OpenAI, Claude, or custom LLMs as per need.

🔑 API Key Management

If you deploy on Streamlit Cloud:

- Use `st.secrets` for key storage.

For **local setup**:

- Create a `.env` file and load keys using:

```
from dotenv import load_dotenv
import os

load_dotenv()
GEMINI_API_KEY = os.getenv("GEMINI_API_KEY")
```

✅ Final Checklist

- Clone the repo
- Install requirements
- Create `.env` with API keys
- Run with `streamlit run app.py`
- Use `README.md` and internal documentation for flow understanding