Project Proposal: World Explorer Agent Crew (Open Source AI Travel Assistant)

Overview:
World Explorer Agent Crew is a fully open-source AI-powered virtual travel assistant that helps users explore global destinations, build itineraries, estimate budgets, and simulate immersive travel experiences. Using a team of collaborative agents managed by CrewAI (or Agno), the project aims to deliver a personalized travel planner experience entirely for free.

Project Goals:

1. Recommend travel destinations based on interests and constraints.
2. Create detailed, day-wise travel itineraries.
3. Estimate budgets based on destination, duration, and preferences.
4. Provide weather forecasts and safety insights.
5. Translate local language phrases.
6. Simulate a virtual travel experience.

How It Differs from Traditional Projects:
Unlike static travel apps or simple itinerary generators, World Explorer leverages multi-agent AI orchestration to divide responsibilities across specialized autonomous agents. Traditional tools often rely on hardcoded APIs and rule-based logic. In contrast, this system adapts in real-time, can simulate conversations, and generate contextual narratives using LLMs running locally.

Key Differences:

- **Local-first, open-source AI models** via Ollama (no OpenAI/GPT dependencies)
- **Modular and agent-based**, making tasks independently executable and composable
- **Immersive storytelling** through a Narrator Agent, simulating real-world travel days
- **No vendor lock-in** – all APIs, models, and tools are fully open and self-hostable
- **Community extensible** – developers can plug in new agents or modify behaviors easily
- **Real-time collaboration between agents** – unlike monolithic apps, each agent performs a focused role
- **Scalability and transparency** – each component is observable, testable, and replaceable by contributors

---

Tech Stack (Fully Open Source):

- **Agent Framework**: CrewAI or Agno
- **LLM Backend**: Ollama (models like LLaMA3, Mistral, Phi3)
- **Frontend**: Streamlit (or Gradio)
- **Mapping/Location Data**: OpenStreetMap, OpenTripMap

- **Weather Data**: Open-Meteo API
- **Currency Exchange**: ExchangeRate.host
- **Time Zone Info**: TimeZoneDB
- **Knowledge Base**: Wikipedia, Wikivoyage
- **Vector Store (optional)**: ChromaDB or Weaviate
- **Containerization**: Docker

Agent Roles & Responsibilities:

1. **Destination Scout Agent**
    - Inputs: Interests, budget, travel dates
    - Sources: OpenTripMap, Wikipedia
    - Outputs: Top 3-5 suggested destinations
2. **Itinerary Planner Agent**
    - Inputs: Selected destination, trip length
    - Logic: POI grouping, proximity logic, OpenTripMap data
    - Output: Day-by-day activity plan
3. **Budget Manager Agent**
    - Inputs: Destination, accommodation level, meals, travel duration
    - Logic: Local cost tables (mock or scraped)
    - Output: Estimated trip cost breakdown
4. **Cultural Guide Agent**
    - Data: Wikivoyage, prebuilt JSON guides
    - Output: Cultural norms, do's and don'ts, etiquette
5. **Weather & Safety Agent**
    - API: Open-Meteo, REST Countries
    - Output: Weather forecast + safety tips
6. **Translator Agent**
    - Model: Multilingual model from Hugging Face (NLLB or bge-m3)
    - Function: Phrasebook for local language + sample dialogues
7. **Experience Narrator Agent**
    - Model: Local LLM (LLaMA3/Mistral)
    - Output: Immersive "day in the life" storytelling

8. **Visual Generator Agent (optional)**
    - Tool: Stable Diffusion WebUI
    - Output: AI-generated images of locations/experiences

UI Flow (Streamlit):

- User selects travel preferences (nature, food, history, budget)
- Destination Scout returns location choices
- User selects one and inputs duration/budget
- Itinerary, budget, weather, and culture agents activate
- Final page: Full plan + immersive narrative + optional image generation

---

Folder Structure Suggestion:

```
world-explorer/
├── agents/
│   ├── destination_scout.py
│   ├── itinerary_planner.py
│   ├── ...
├── tools/
│   ├── opentripmap.py
│   ├── weather.py
├── ui/
│   └── streamlit_app.py
├── models/
│   └── ollama_config.yaml
├── requirements.txt
├── README.md
└── docker-compose.yaml
```

License:

- MIT or Apache 2.0

Future Enhancements:

- Real-time booking integrations (with mock APIs)
- Personalized memory (via vector store)
- Offline desktop version using Electron + Streamlit
- Integration with a voice assistant for hands-free planning

Conclusion:

World Explorer Agent Crew combines AI, open data, and an extensible agent framework to offer a free, rich, and engaging travel planning experience. Its modularity ensures easy customization and contribution from the open-source community while significantly advancing beyond traditional rule-based travel apps.