**Smart City Associate Gen AI & Agentic AI - Intern Assessment**

**🏙️ Project Overview: Smart City Information Assistant**

**Duration:** 2 Days  
**Use Case:** Build an intelligent assistant for citizens to query information about city services, policies, and facilities using our provided Smart City Knowledge Base.

**🎯 Main Challenge**

Create a **Smart City Information Assistant** that helps citizens get instant answers about:

* City services (waste collection, permits, utilities)
* Public facilities (parks, libraries, hospitals)
* Transportation (bus routes, parking, traffic)
* City policies and regulations
* Emergency contacts and procedures

**📋 Technical Requirements**

**Required Technologies**

* **Ollama** - Local LLM deployment
* **LangChain** - LLM orchestration and RAG pipeline
* **Vector Database** - Milvus (preferred) or alternatives (ChromaDB, Pinecone, FAISS)
* **FastAPI** - Backend API service
* **Streamlit** - Web interface
* **LLM** – any model

**Optional (Major Bonus)**

* **CrewAI** - Multi-agent system ⭐ *25% bonus points*

**📊 Provided Smart City Knowledge Base**

**What We Provide:**

* **City Services Documentation** (permits, utilities, waste management)
* **Public Facilities Information** (parks, libraries, hospitals, schools)
* **Transportation Data** (bus routes, schedules, parking)
* **Policy Documents** (regulations, procedures, guidelines)
* **Emergency Information** (contacts, procedures, safety protocols)
* **Test Queries** for validation

**Format:** JSON, PDF, and text files ready for processing

**🛠️ Implementation Requirements**

**Core Implementation**

1. **Setup Environment** - Install required tools, configure Ollama
2. **Process Knowledge Base** - Load, chunk, and embed provided data
3. **Build RAG Pipeline** - Implement retrieval system with vector database
4. **Create FastAPI Backend** - Essential endpoints:
   * /query - Main chat endpoint
   * /health - Health check
   * /search - Vector search
5. **Build Streamlit Frontend** - User interface consuming API
6. **Integration & Testing** - Connect all components, test with provided queries

**Optional CrewAI Enhancement**

* **Information Retriever Agent** - Searches city data
* **Policy Expert Agent** - Handles regulations
* **Service Coordinator Agent** - Provides guidance

**🎯 Key Features**

**Must Have**

* [ ] FastAPI backend with proper endpoints
* [ ] RAG system using provided knowledge base
* [ ] Vector database integration
* [ ] Clean Streamlit interface
* [ ] Conversation history
* [ ] Error handling

**Bonus Features**

* [ ] **CrewAI Multi-Agent System** ⭐ *Major advantage*
* [ ] Advanced API endpoints
* [ ] Response confidence scoring
* [ ] Additional external data sources

**📝 Example Test Cases**

1. **"How do I apply for a building permit?"**
2. **"What are the library hours near downtown?"**
3. **"When is garbage collection in Sector 5?"**
4. **"What's the process for starting a small business?"**

**🏆 Evaluation Criteria (100%)**

**Technical Implementation (70%)**

* **FastAPI Backend** (20%) - Well-designed endpoints, error handling
* **RAG System** (20%) - Effective use of provided knowledge base
* **LLM Integration** (15%) - Ollama + LangChain implementation
* **Frontend Integration** (10%) - Streamlit consuming API
* **Code Quality** (5%) - Clean, documented code

**User Experience (20%)**

* **Interface Design** (10%) - Intuitive Streamlit UI
* **Response Quality** (10%) - Accurate answers from knowledge base

**Documentation (10%)**

* **Setup Instructions** (5%)
* **Demo Video** (5%)

**Bonus Points (Up to 30% extra)**

* **CrewAI Implementation** (25%)
* **Advanced Features** (5%)

**📋 Deliverables**

1. **GitHub Repository** with complete code
2. **FastAPI Documentation** - auto-generated endpoints docs
3. **Demo Video** (2-3 minutes):
   * 30s: Architecture overview
   * 30s: API demonstration
   * 90s: Frontend demo with queries
   * 30s: Special features highlight
4. **README** with setup instructions
5. **Knowledge Base Usage Report** - how you processed our data

**🚀 Success Metrics**

**Minimum Viable Product:**

* Working FastAPI + Streamlit integration
* 70% accuracy on provided test queries
* Proper use of provided knowledge base
* Complete documentation

**High Performance:**

* CrewAI multi-agent implementation
* Advanced API features
* 85%+ accuracy on test queries
* Professional code quality

**💡 Key Tips**

* **Start with our knowledge base** - this is your foundation
* **Test RAG pipeline first** - ensure retrieval works with provided data
* **Focus on core features** before attempting CrewAI
* **Use provided test queries** to validate your system
* **Document your approach** to handling the knowledge base

**Focus:** Build a working system that effectively uses our Smart City Knowledge Base to help citizens get the information they need quickly and accurately.