

# DataSmith AI – GenAI Intern Assignment

---

Role: GenAI Intern (Generative AI)

Assignment: Post Discharge Medical AI Assistant (POC)

Duration: 2-3 Days

Company: DataSmith AI

## Overview

This assignment evaluates your ability to build a Proof of Concept (POC) multi-agent AI system for post-discharge patient care. You'll demonstrate core GenAI skills including RAG implementation, multi-agent orchestration, and medical data processing in a simplified but functional system.

## Assignment Objective: Post Discharge Medical AI Assistant POC

Build a chatbot system with multi-agent architecture that:

- Manages 25+ dummy post-discharge patient reports
- Uses RAG with nephrology reference materials
- Implements two specialized AI agents with clear workflows
- Provides simple web interface for patient interactions

## Core Requirements

### 1. Data Setup

- Create 25+ dummy post-discharge reports (JSON/CSV format is fine)
- Integrate nephrology reference book (PDF or text format)
- Simple database storage (SQLite or JSON files acceptable)
- Vector embeddings for semantic search

### 2. Multi-Agent System

#### *Receptionist Agent*

- Asks patient for their name
- Uses explicit database retrieval tool to fetch patient's discharge report
- Retrieves patient's discharge report from database
- Asks follow-up questions based on the discharge information
- Routes medical queries to Clinical Agent

### **Clinical AI Agent**

- Handles medical questions and clinical advice
- Uses RAG over nephrology reference book for answers
- Uses web search tool for queries outside reference materials
- Provides citations from reference materials
- Logs patient interactions

### **3. RAG Implementation**

- Process and chunk nephrology reference materials
- Create vector embeddings for semantic search
- Implement retrieval and answer generation
- Include source citations in responses

### **4. Web Search Tool**

- Implement web search capability for queries outside reference materials
- Integrate with Clinical Agent workflow
- Clearly indicate when information comes from web search vs. reference materials
- Handle fallback when specialized information is needed

### **5. Logging System**

- Implement comprehensive logging throughout the system
- Log all interactions between agents and users
- Log agent handoffs and decision processes
- Maintain log file with timestamps showing complete system flow
- Include information retrieval attempts and results

### **6. Patient Data Retrieval Tool**

- Create dedicated tool for database interaction
- Implement patient lookup by name
- Return structured discharge report data
- Handle error cases (patient not found, multiple patients with same name)
- Log all database access attempts

## **Technical Specifications**

### **Frontend (Choose One)**

- Streamlit (Recommended for simplicity)
- React.js
- Basic HTML/CSS/JS

## Backend

- FastAPI (Recommended)
- Flask (Alternative)

## Multi-Agent Framework (Choose One)

- CrewAI
- LangGraph
- Custom implementation with LangChain

## Databases & Storage

- Vector DB: ChromaDB, FAISS, or Qdrant
- Data Storage: SQLite, JSON files, or PostgreSQL
- Embeddings: OpenAI, HuggingFace, or Sentence-Transformers

## Sample Patient Reports Structure

```
{
  "patient_name": "John Smith",
  "discharge_date": "2024-01-15",
  "primary_diagnosis": "Chronic Kidney Disease Stage 3",
  "medications": ["Lisinopril 10mg daily", "Furosemide 20mg twice
daily"],
  "dietary_restrictions": "Low sodium (2g/day), fluid restriction
(1.5L/day)",
  "follow_up": "Nephrology clinic in 2 weeks",
  "warning_signs": "Swelling, shortness of breath, decreased urine
output",
  "discharge_instructions": "Monitor blood pressure daily, weigh yourself
daily"
}
```

## Expected System Workflow

### Initial Interaction

System: "Hello! I'm your post-discharge care assistant. What's your name?"

Patient: "John Smith"

Receptionist Agent: [Uses patient data retrieval tool to fetch discharge report]

Receptionist Agent: "Hi John! I found your discharge report from January 15th for Chronic Kidney Disease. How are you feeling today? Are you following your medication schedule?"

### Medical Query Routing

Patient: "I'm having swelling in my legs. Should I be worried?"

Receptionist Agent: "This sounds like a medical concern. Let me connect you with our Clinical AI Agent."

Clinical Agent: "Based on your CKD diagnosis and nephrology guidelines, leg swelling can indicate fluid retention... [RAG response with citations]"

### Web Search Fallback Example

Patient: "What's the latest research on SGLT2 inhibitors for kidney disease?"

Clinical Agent: "This requires recent information. Let me search for you... According to recent medical literature [Web search results with source]..."

### Architecture Justification (Required)

- LLM Selection
- Vector Database
- RAG Implementation
- Multi-Agent Framework
- Web Search Integration
- Patient Data Retrieval
- Logging Implementation

### Deliverables

- Working POC Application
- GitHub Repository
- Brief Report (2-3 pages)
- Demo Video (5 minutes)

### Important Notes

- Keep It Simple
- Use dummy data - no real patient information
- Basic UI is perfectly acceptable
- Prioritize working features over visual design
- Medical Disclaimers - add: 'This is an AI assistant for educational purposes only' and 'Always consult healthcare professionals for medical advice'

## Final Checklist

- 25+ dummy patient reports created
- Nephrology reference materials processed
- Receptionist Agent implemented
- Clinical AI Agent with RAG implemented
- Patient data retrieval tool implemented
- Web search tool integration
- Comprehensive logging system
- Simple web interface working
- Agent handoff mechanism functional
- GitHub repo with clean code
- Brief report with architecture justification
- Demo video recorded
- All code commented and documented