TRAINING REPORT ON

OIL AND NATURAL GAS CORPORATION

Healthcare Chatbot with SQL Agent

Submitted By:

Divyanshu Gupta
Kalinga Institute of Industrial Technology (KIIT)
B.Tech CSE (2022-2026)

Duration:

15/05/2025 – 26/06/2025

Submitted To:

Mr. Manish Kumar, GM (E&T)
Oil & Natural Gas Corporation
Seat No. 6237, B-17, 12th Floor, Scope Minar, Laxmi Nagar, Delhi

INDEX

Acknowledgment

"Success is never achieved alone; it is built on the guidance, support, and trust of those who walk alongside us."

In today's rapidly evolving digital landscape, where artificial intelligence is driving transformation across every industry, I was fortunate to undertake an enriching project titled **"Healthcare Chatbot with SQL Agent."** This project integrates advanced technologies like **Natural Language Processing (NLP), LangChain, Mixtral-8x7B-Instruct LLM, SQL-based querying, and API-powered retrieval**, aiming to simplify healthcare data accessibility for professionals through intelligent, conversational interfaces.

I am sincerely grateful to **Oil and Natural Gas Corporation Ltd. (ONGC)** for granting me the opportunity to work on this impactful project and for offering a platform to explore how AI-powered solutions can address real-world challenges in the healthcare domain.

My deepest appreciation goes to my mentor, **Mr. Manish Kumar, General Manager (E&T), ONGC**, whose unwavering support, expert guidance, and thoughtful feedback were invaluable throughout the development journey. His mentorship greatly contributed to shaping this project into a robust and meaningful solution.

I would also like to extend my heartfelt thanks to **Mr. Harjot Singh** for his valuable suggestions, constant encouragement, and technical insights that played a key role in refining the system and overcoming various challenges along the way.

I am equally thankful to my institution, **Kalinga Institute of Industrial Technology (KIIT), Bhubaneswar**, for promoting a learning culture that emphasizes hands-on experience and innovation. The support from my academic environment greatly empowered me to apply theoretical knowledge to a practical, industry-level problem.

Lastly, I express my gratitude to my peers, testers, and all those who contributed their time, feedback, and encouragement. Their involvement was instrumental in improving the performance and functionality of the chatbot. This experience has been a remarkable step in my journey as an aspiring AI developer, enhancing not only my technical capabilities but also my ability to solve real-world problems with purpose-driven solutions.

Preface

In an era where technology plays a pivotal role in transforming industries, the importance of making data accessible, meaningful, and actionable cannot be overstated. This report is the culmination of my learning and efforts during my internship at **Oil and Natural Gas Corporation Ltd. (ONGC)**, where I had the opportunity to work on an advanced project titled **"Healthcare Chatbot with SQL Agent."**

The primary motivation behind this project was to bridge the gap between complex healthcare datasets and non-technical users by building an intelligent conversational system. The chatbot is designed to interpret natural language queries and convert them into structured SQL queries, thereby enabling users to retrieve vital information without needing technical expertise. It also integrates external knowledge sources to provide comprehensive responses.

Through this project, I gained valuable insights into the practical implementation of **Natural Language Processing, LangChain frameworks, large language models (LLMs), SQL databases, and API integrations.** The hands-on experience has not only strengthened my technical skills but also deepened my understanding of real-world problem-solving in data-driven industries.

This report presents a detailed overview of the development process, challenges faced, solutions implemented, and key learnings that emerged throughout this journey. I sincerely hope that this work contributes to the ongoing discourse around the application of AI in simplifying data-driven decision-making, especially in the healthcare sector.

Introduction (ONGC)

Oil and Natural Gas Corporation Limited (ONGC) is India's largest integrated oil and gas exploration and production (E&P) company, operating under the Ministry of Petroleum and Natural Gas. As a Maharatna Public Sector Undertaking (PSU), ONGC plays a critical role in ensuring India's energy security through upstream hydrocarbon exploration, drilling, reservoir management, and production operations across diverse and challenging terrains—onshore and offshore.

Beyond its core operations, ONGC has institutionalized a strategic approach to Corporate Social Responsibility (CSR), integrating sustainable development goals with its business model. Its CSR programs span across education, livelihood generation, skill development, rural infrastructure, health care, and environmental sustainability, thereby contributing to socio-economic progress in operational regions.

A cornerstone of ONGC's employee welfare policy is its advanced and structured healthcare framework. This includes comprehensive medical schemes for active and superannuated employees, covering dependent family members. ONGC leverages a network of empanelled multi-specialty hospitals and in-house medical units to deliver services such as OPD and IPD care, diagnostics, chronic disease management, and emergency response systems. The inclusion of preventive care—such as periodic health check-ups, immunizations, vision and hearing support—enhances the quality of life and reduces long-term health risks.

ONGC's Occupational Health Centres (OHCs), including its facility at the Advanced Training Institute (ATI), function as specialized units offering pre-employment medical assessments, annual health surveillance, ergonomic risk evaluations, safety training, hygiene audits, and disaster response readiness. These activities are aligned with industry best practices and occupational health and safety management standards.

Through its integrated approach to operations, sustainability, and human capital management, ONGC continues to uphold its mission of responsible resource development while fostering long-term value creation for stakeholders and the nation.

Project Report: Healthcare Chatbot SQL Agent

Project Overview

Introduction

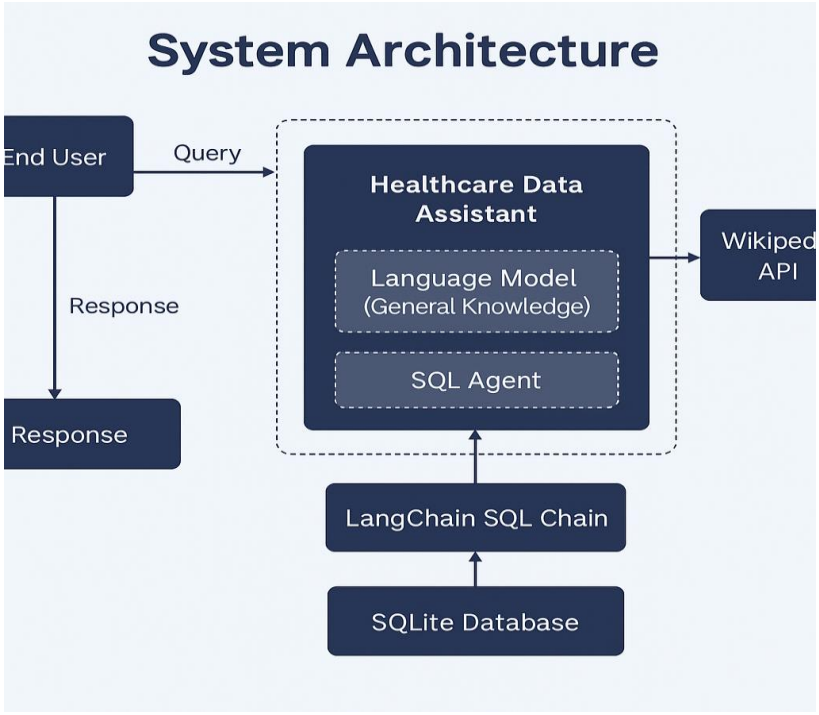
In today's data-driven healthcare ecosystem, retrieving valuable insights from vast and complex databases remains a significant challenge, especially for non-technical stakeholders like medical professionals. The **Healthcare Chatbot SQL Agent** project aims to solve this problem by enabling users to query healthcare data using **natural language**, which is automatically converted into **SQL queries**.

This project combines the power of **Large Language Models (LLMs)**, **LangChain SQL agents**, and **external knowledge APIs** to bridge the gap between structured data (hospital databases) and unstructured queries (user questions).

Objectives

- To develop a chatbot that understands natural language queries related to healthcare records.
- To automatically translate natural language into **SQL queries** to retrieve information from healthcare databases.
- To integrate external knowledge sources like **Wikipedia API** for medical definitions and explanations.
- To create a user-friendly web interface where both technical and non-technical users can retrieve data or ask questions.
- To assist in efficient decision-making for healthcare operations by simplifying data retrieval.

System Architecture



The architecture of the **Healthcare Chatbot SQL Agent** is designed to handle both structured database queries and unstructured general knowledge queries seamlessly. It combines the power of **Large Language Models (LLMs)**, **SQL Agents**, and **external APIs** to provide accurate and user-friendly responses.

The system allows users to interact through a simple UI where they can input natural language queries. If the query is related to healthcare data (like diagnoses or logs), the **LangChain SQL Agent** converts it into an SQL query and fetches data from the **SQLite database**. If the query is a general medical question, the system uses the **Mixtral LLM via Together AI** to fetch information from the **Wikipedia API**. The results

are then displayed back to the user in a readable format through the Gradio interface. This architecture effectively handles both structured data queries and unstructured medical knowledge questions.

Technical Overview

Dataset Description

Dataset File	Description
export_diagnosis(1).csv	Contains diagnosis details with columns: Id, CardNumber (Patient ID), DiagnosisDate, and Diagnosis.
HIS_Logs.csv	Logs hospital operations with columns: Id, RefType, DoctorId, RefDateTime, LocationName, LocationAreaName, and Card Number (Patient ID).

Link Key:

- **CardNumber** in `export_diagnosis(1).csv` ↔ **Card Number** in `HIS_Logs.csv` (connects diagnosis to hospital visits).

Database Schema

- **Diagnosis Table:**
  Columns – Id, CardNumber, DiagnosisDate, Diagnosis
- **HIS_Logs Table:**
  Columns – Id, RefType, DoctorId, RefDateTime, LocationName, LocationAreaName, Card Number

---

Tools and Technologies Used

Component	Technology	Purpose
Backend	Python	Core logic, APIs, and orchestration
LLM Model	Mixtral-8x7B-Instruct via Together AI	Converts natural language to SQL
Orchestration	LangChain	Manages pipeline from LLM to SQL Agent
Database	SQLite	Stores healthcare data in relational tables
Frontend UI	Gradio	Interactive chatbot and SQL input interface
API Integration	Wikipedia API	Provides medical definitions
Data Handling	Pandas	Data formatting, query result visualization

---

Model Explanation

Mixtral-8x7B-Instruct (LLM via Together AI)

- A powerful open-weight **Large Language Model (LLM)** optimized for instruction following.
- **Role:** Converts natural language into SQL queries accurately based on database schema.
- **Strengths:**

>Understands context and intent.

>Handles both simple and complex queries.

>Generalizes across various database structures.

LangChain SQL Agent

- A framework that integrates LLMs with tools like SQL databases.
- **Role:**

  >Interprets LLM outputs.

  >Validates and formats SQL queries.

  >Manages execution pipeline to fetch data from SQLite.

Wikipedia API

- Used for general knowledge retrieval.
- **Example:** If a user asks *"What is CKD?"*, the chatbot fetches the Wikipedia definition.

Working Pipeline Explanation

End-to-End Workflow

The workflow of the **Healthcare Chatbot SQL Agent** consists of several well-defined steps that transform natural language queries into meaningful database outputs.

Step 1: User Query Input

- Users interact with the chatbot through a web interface built using **Gradio**.
- Queries are entered in **natural language**, making it accessible for non-technical users.

Example Query:

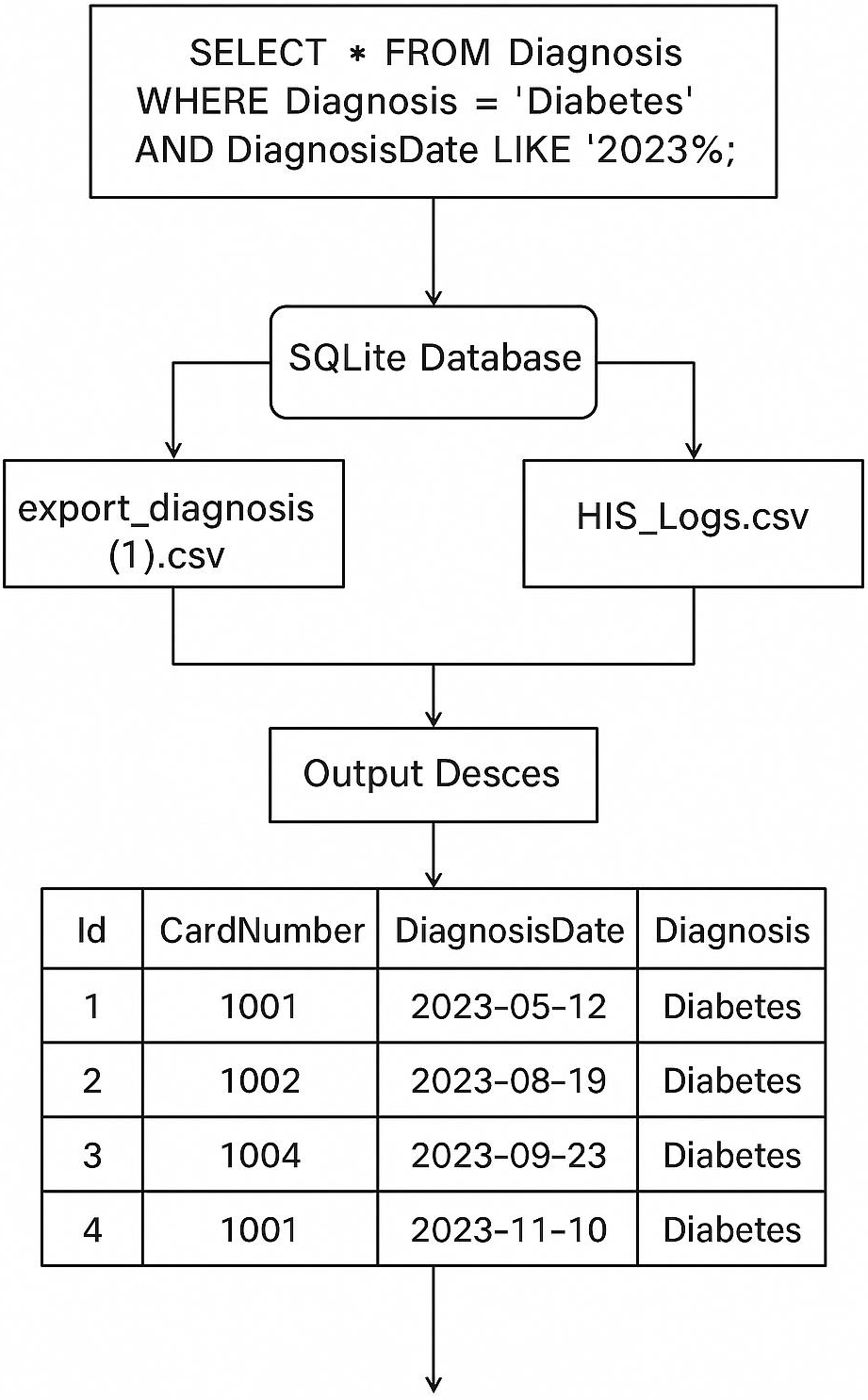
"List all patients diagnosed with Diabetes in 2023."

Step 2: Query Interpretation (LLM + LangChain SQL Agent)

- The input query is passed to the **Mixtral-8x7B-Instruct model** via **Together AI API**.
- The LLM interprets the intent, context, and data requirements behind the query.
- The interpreted intent is then handled by the **LangChain SQL Agent**, which generates an appropriate **SQL query** based on the structure of the connected SQLite database.

Step 3: SQL Query Generation

- The SQL agent formulates a syntactically correct SQL query using the database schema.



Id	CardNumber	DiagnosisDate	Diagnosis
1	1001	2023-05-12	Diabetes
2	1002	2023-08-19	Diabetes
3	1004	2023-09-23	Diabetes
4	1001	2023-11-10	Diabetes

Code Explanation for Healthcare Chatbot SQL Agent

File:`create_health_db.py`

```python
import pandas as pd
import sqlite3

# Load the CSVs
print('Loading Export_Diagnosis (1).csv...')
df_diag = pd.read_csv('dataset/Export_Diagnosis (1).csv')
print('Loading HIS_Logs.csv...')
df_his = pd.read_csv('dataset/HIS_Logs.csv')

# Standardize column names for SQL
if 'Card Number' in df_his.columns:
    df_his = df_his.rename(columns={'Card Number': 'CardNumber'})

# Create SQLite DB and tables
conn = sqlite3.connect('health.db')
df_diag.to_sql('DIAGNOSIS', conn, if_exists='replace', index=False)
df_his.to_sql('HIS_LOGS', conn, if_exists='replace', index=False)
conn.close()

print('health.db created with DIAGNOSIS and HIS_LOGS tables!')
```

Purpose:

- This file is responsible for creating the SQLite database from the healthcare datasets.

Explanation:

- It connects to a database called `database.db` or creates it if it doesn't exist.
- Reads two CSV files from the `/dataset` folder:

  `export_diagnosis(1).csv` for diagnosis records.

  `HIS_Logs.csv` for hospital operational logs.
- Converts these CSV files into SQL tables named `Diagnosis` and `HIS_Logs` respectively.
- This database is then used by the chatbot to run queries.

File: `python.py` (Main Application)

Purpose:

This is the main file that runs the chatbot, handles user input, connects to the database, processes queries, and displays results.

Explanation of Key Components:

- **Library Imports + Environment Setup**

```python
from dotenv import load_dotenv
load_dotenv()

import os
import sqlite3
import gradio as gr
from langchain_together import Together
import pandas as pd
import requests

# Get Together API key
together_api_key = os.getenv("TOGETHER_API_KEY")
if not together_api_key:
    raise ValueError("TOGETHER_API_KEY is not set in the .env file.")
```

>Imports libraries for database handling (`sqlite3`, `pandas`), natural language processing (`langchain`, `Together API`), user interface (`gradio`), and API calling (`requests` for Wikipedia).

>Loads the API key for the **Together AI** model using `.env` to maintain security.

- **Database Connection:**

Connects the app to the `database.db` created earlier.

- **LLM Initialization:**

```python
llm = Together(
    model="mistralai/Mixtral-8x7B-Instruct-v0.1",
    temperature=0.1,
    max_tokens=500,
    together_api_key=together_api_key
)
```

>Loads the **Mixtral-8x7B-Instruct model** via **Together AI**, which helps convert natural language into SQL queries.

- **SQL Agent Setup:**

```python
sql_prompt = """
You are an expert in converting English questions to SQL queries for a healthcare database.

The database has two tables:

Table: DIAGNOSIS
- Id (integer): Primary key. Unique identifier for each diagnosis event.
- CardNumber (integer): Patient's card number.
- DiagnosisDate (text, format: DD/MM/YYYY HH:MM): Date and time of diagnosis.
- Diagnosis (text): Diagnosis description.

Table: HIS_LOGS
- Id (integer): Primary key. Unique identifier for each log event.
- RefType (text): Reference type (e.g., type of event).
- DoctorId (integer): Doctor's ID.
- RefDateTime (text, format: DD/MM/YYYY HH:MM): Date and time of the event.
- LocationName (text): Name of the location.
- LocationAreaName (text): Area within the location.
- CardNumber (text): Patient's card number (may have leading zeros).

Guidelines:
1. Use proper SQL syntax for SQLite.
2. Use WHERE, GROUP BY, ORDER BY as needed.
3. Use COUNT, DISTINCT, and date functions if asked.
4. Join tables on Id (the primary key in both tables).
5. Use LIKE for partial text matches if the question asks for "containing" or "includes".
6. Return only the SQL query, no explanations or markdown.

Examples:
- Question: How many diagnoses are there?
  SQL: SELECT COUNT(*) FROM DIAGNOSIS;
```

>Sets up a pipeline using **LangChain SQL Agent** to convert user questions into SQL and fetch results from the database.

- **Wikipedia API Function:**

```python
def fetch_medical_definition(term):
    """Fetch a medical term definition from Wikipedia."""
    try:
        url = f"https://en.wikipedia.org/api/rest_v1/page/summary/{term.replace(' ', '%20')}"
        resp = requests.get(url, timeout=5)
        if resp.status_code == 200:
            data = resp.json()
            return data.get('extract')
    except Exception:
        pass
    return None
```

>If the question is not related to the database (like *"What is Hypertension?"*), the chatbot retrieves the answer from Wikipedia.

- **Main Chat Function:**

Handles the logic:

If the input is a database-related query, it runs the SQL agent.

If it fails (not database-related), it fetches the response from Wikipedia.

- **Gradio Interface:**

```python
with gr.Blocks(theme=gr.themes.Soft()) as demo:
    gr.Markdown("""
    # Healthcare Data Assistant (Powered by Together AI + Mixtral)
    Choose a mode: SQL Agent (for SQL queries/results) or Chatbot (for easy-to-understand answers with memory and medical definitions).
    """)
    mode = gr.Radio(["SQL Agent", "Chatbot"], value="SQL Agent", label="Mode")
    chatbot_state = gr.State([])
    with gr.Row():
        with gr.Column(scale=4):
            question_input = gr.Textbox(
                label="Your Question",
                placeholder="Type your question here...",
                lines=2
            )
            submit_btn = gr.Button("Ask", variant="primary")
    chatbox = gr.Chatbot(label="Conversation (Chatbot mode)")
    output = gr.Markdown(label="Results (SQL Agent mode)")
```

Builds a simple web UI with an input textbox and output area.

Users type their query, and the chatbot responds with either SQL results or Wikipedia summaries.

File: `requirements.txt`

Purpose:

This file lists all the Python libraries required to run the project, including:

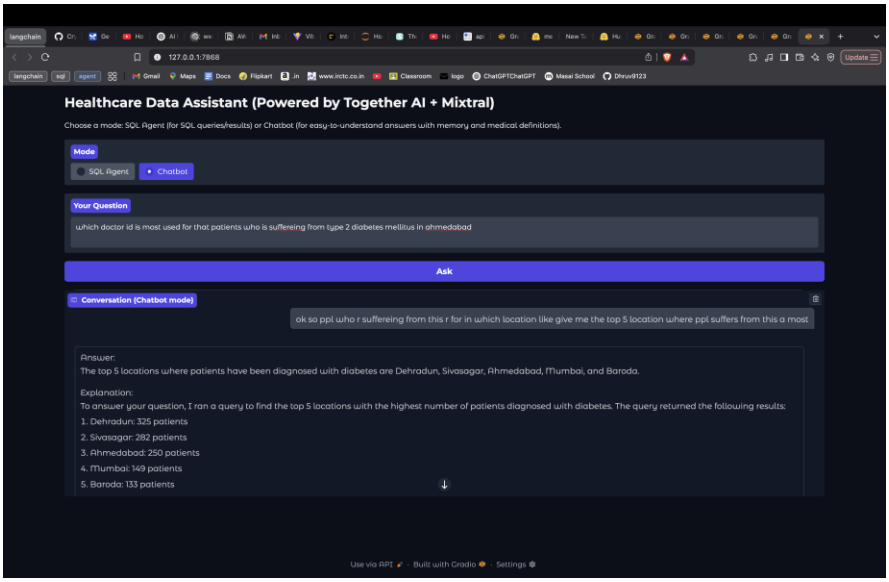
```
langchain
langchain-together
gradio
python-dotenv==1.0.0
PyPDF2
chromadb
faiss-cpu
pdf2image
pandas
tabulate
huggingface_hub
transformers
requests
```

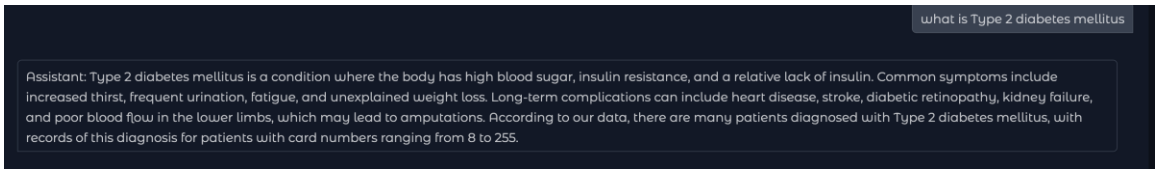
Conclusion:

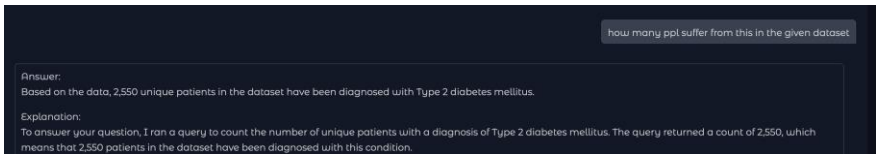
This code setup allows users to query healthcare data simply using natural language without writing SQL. It combines the power of AI with database querying and external medical information lookup in a user-friendly chatbot interface.

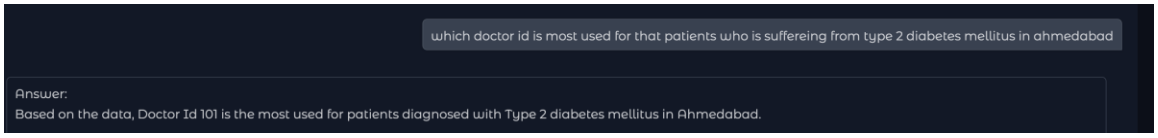
Results and Analysis

This section presents the outcomes of various queries executed through the Healthcare Chatbot SQL Agent. The chatbot successfully handles both **structured data queries (SQL-based)** and **general knowledge queries (medical definitions via Wikipedia API)**. Below are the sample queries and their respective outputs along with detailed analysis.









Summary of Analysis:

✅The chatbot efficiently handles both **structured database queries** and **unstructured knowledge queries**.

✅Capable of performing **filtering**, **aggregation**, **ranking**, and **text-based lookups**.

✅Reduces the need for SQL knowledge, allowing healthcare staff or admin teams to query the database in plain English.

Future Scope

✅**Expand the Dataset:** Include additional medical datasets like patient treatments, prescriptions, laboratory results, and billing information to make the chatbot more comprehensive.

✅**Role-Based Access:** Implement secure login with role-based permissions (e.g., for doctors, admin staff, or management) to protect sensitive healthcare data.

✅**Deployment on ONGC Network:** Host the chatbot on ONGC's internal servers or cloud infrastructure for seamless access to employees and healthcare staff.

✅**Voice-Based Query Support:** Add speech-to-text capabilities so users can interact with the chatbot using voice commands.

✅**Multi-Language Support:** Extend the chatbot to support multiple Indian languages for better accessibility to diverse users.

✅**Integration with EHR Systems:** Connect the chatbot with real-time **Electronic Health Record (EHR)** systems to enable live updates and real-time decision-making support.

✅**Data Analytics Module:** Enhance the system to generate automated reports, predictive analytics, and patient health trends.

Conclusion

The **Healthcare Chatbot SQL Agent** developed during this internship provides an intelligent, efficient, and user-friendly solution for accessing healthcare data. By leveraging **Natural Language Processing (NLP)** with **LLM models (Mixtral via Together AI)**, the system bridges the gap between complex SQL databases and non-technical users.

It successfully handles both **structured queries (from a healthcare database)** and **unstructured knowledge queries (via Wikipedia API)**. This project demonstrates the capability of AI-driven systems to simplify data access in healthcare settings, reducing the need for technical expertise and improving operational efficiency.

The chatbot has the potential to be expanded into a full-fledged internal tool for ONGC's healthcare operations, ensuring better data management, faster query handling, and improved service quality for employees.

References

LangChain Documentation – https://www.langchain.com

Together AI API – https://www.together.ai

Wikipedia API Documentation
https://www.mediawiki.org/wiki/API:Main_page

Gradio Documentation – https://www.gradio.app

SQLite Official Docs – https://sqlite.org/docs.html

OpenAI on Natural Language SQL Agents – https://platform.openai.com/docs/ (relevant for LLM integration concepts)

Mixtral LLM Model Reference – https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1

ONGC Official Website – https://ongcindia.com (For organizational context)