

PROJECT REPORT

**🟦 Title:**

**QueryCraft: Text-to-SQL using Gemini**

**👥 Team Members:**

| **Name** | **Role** |
| --- | --- |
| KETHIREDDY MOHAN LAKSHMI REDDY | Frontend & API Integration |
| PANIDAPU MEGHANA | Prompt Engineering & Testing |
| BANDREDDI YASASRI MADHAVI | Documentation & Report |
| MADDIKAYALA LAKSHMI REDDY | UI & Output Screenshots |
|  |  |

**🎓 College Name:**

VIT-AP University, Amaravati

**🧪 Internship Track:**

SmartBridge – Generative AI in Partnership with Google

**📅 Project Duration:**

May 2025 – June 2025

**DEFINING:**

**❗ Problem Statement:**

**Many users—including managers, analysts, and students—struggle to access data from databases due to their lack of SQL knowledge. This creates a dependency on developers or database administrators, causing delays and limiting direct data exploration. There is a need for a natural and user-friendly solution that allows users to query databases using plain English.**

**Scenario 1: Lack of SQL Knowledge Among Non-Technical Users**

**Many users in business, education, and research domains work with structured data stored in relational databases. However, they often lack the knowledge of SQL (Structured Query Language), making it difficult to extract meaningful insights or reports. Writing correct SQL queries requires understanding database schemas, syntax rules, and logic construction, which is a major barrier for non-technical users.**

**Scenario 2: Time-Consuming and Error-Prone SQL Querying**

**Even experienced developers spend significant time writing and debugging complex SQL queries, especially those involving joins, aggregations, or filtering. This process can be repetitive and error-prone, reducing productivity and increasing the chances of incorrect data retrieval.**

**🎯 Objective:**

**To develop an intelligent web application that allows users to input natural language queries and automatically converts them into SQL queries using Google's Gemini Pro model. This tool helps users generate database queries without knowing SQL, increasing accessibility and productivity.**

**📚 Project Description:**

**QueryCraft is a lightweight, user-friendly Streamlit web app powered by Gemini Pro. It enables users to type database-related questions in plain English and receive valid SQL queries in response. The Gemini API processes the language input and returns the appropriate SQL code. This makes interacting with databases easier, especially for those with no prior SQL knowledge.**

**The app ensures:**

* **Seamless integration with the Gemini model**
* **Real-time SQL query generation**
* **Clean and responsive user interface**
* **Secure API key handling using environment variables**

**🛠️ Tools & Technologies Used:**

| **Tool/Technology** | **Purpose** |
| --- | --- |
| **Python** | **Backend logic** |
| **Streamlit** | **Frontend/UI framework** |
| **Gemini API (Google)** | **Natural Language Processing** |
| **dotenv** | **Environment variable management** |
| **Git & GitHub** | **Version control and project hosting** |

**✨ Key Features:**

* **Convert plain English to SQL using Gemini**
* **Secure .env for managing API keys**
* **Real-time output display via Streamlit**
* **Clean UI and user-friendly layout**
* **Organized folder structure for easy deployment**

**🏗️ System Architecture:**

User Input

↓

Streamlit Frontend

↓

Gemini API (Text-to-SQL)

↓

Generated SQL Response

↓

Displayed to User

**📁 Project Folder Structure:**

**querycraft-text2sql-using-gemini/**

**├── app.py # Main app code**

**├── requirements.txt # Python dependencies**

**├── image/ #images used in the interface**

**├── .env # API Key (excluded via .gitignore)**

**├── .gitignore # Hides .env and system files**

**├── README.md # Project overview**

**├── project\_report.doc # Final documentation**

**├── demo.mp4 / README link # Demo video**

**├── screenshots.doc # i/p and o/p screenshots**

**PROJECT PLANING and SCHEDULING**

**1. Introduction:**

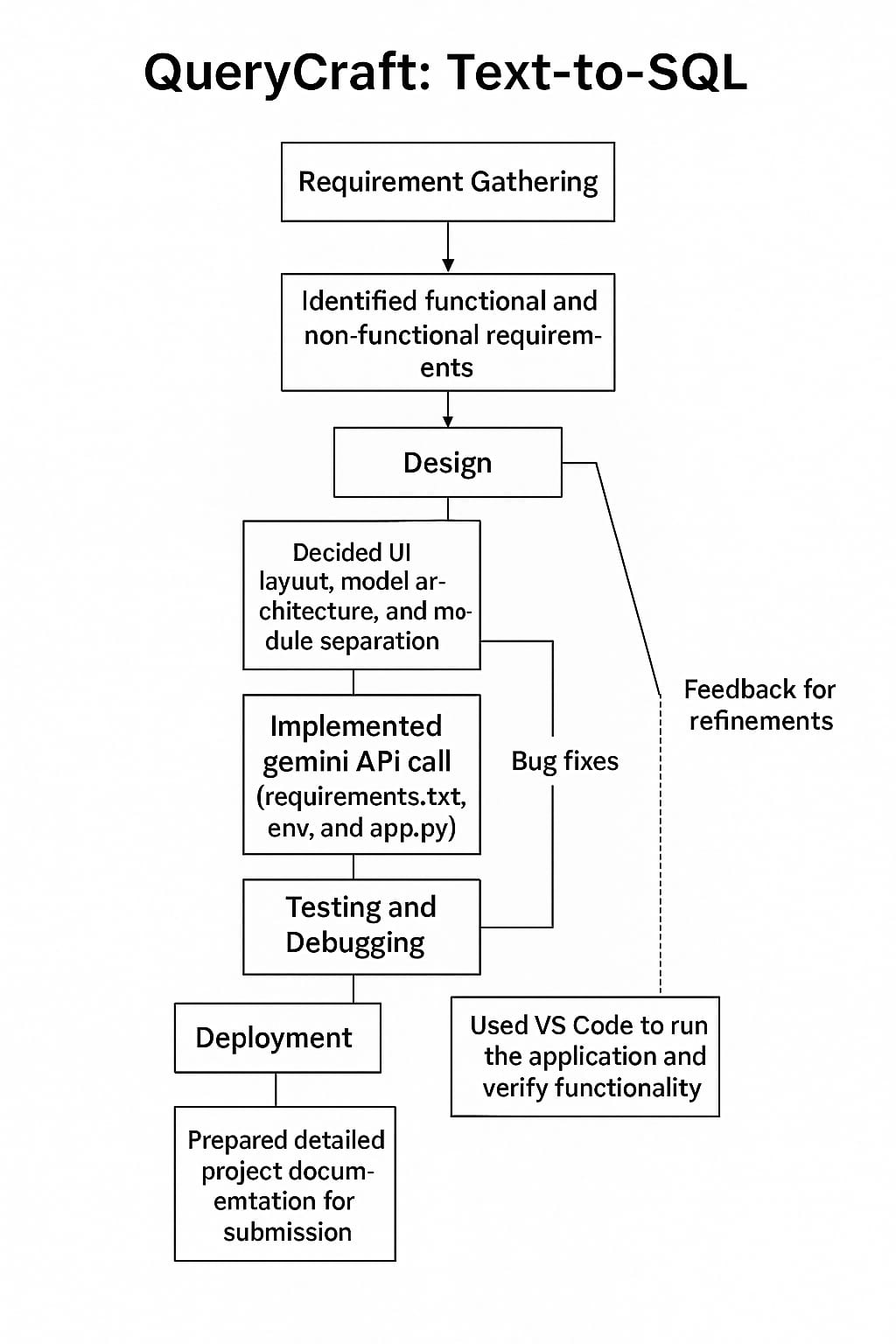
**Project planning and scheduling are critical to the success of software development projects. For the QueryCraft project, a structured and iterative planning approach was adopted to ensure the solution was developed efficiently and delivered on time. The planning focused on integrating Google’s Generative AI (Gemini) with a user-friendly interface, ensuring secure API usage, and validating accurate SQL output.**

**2. Planning Objectives**

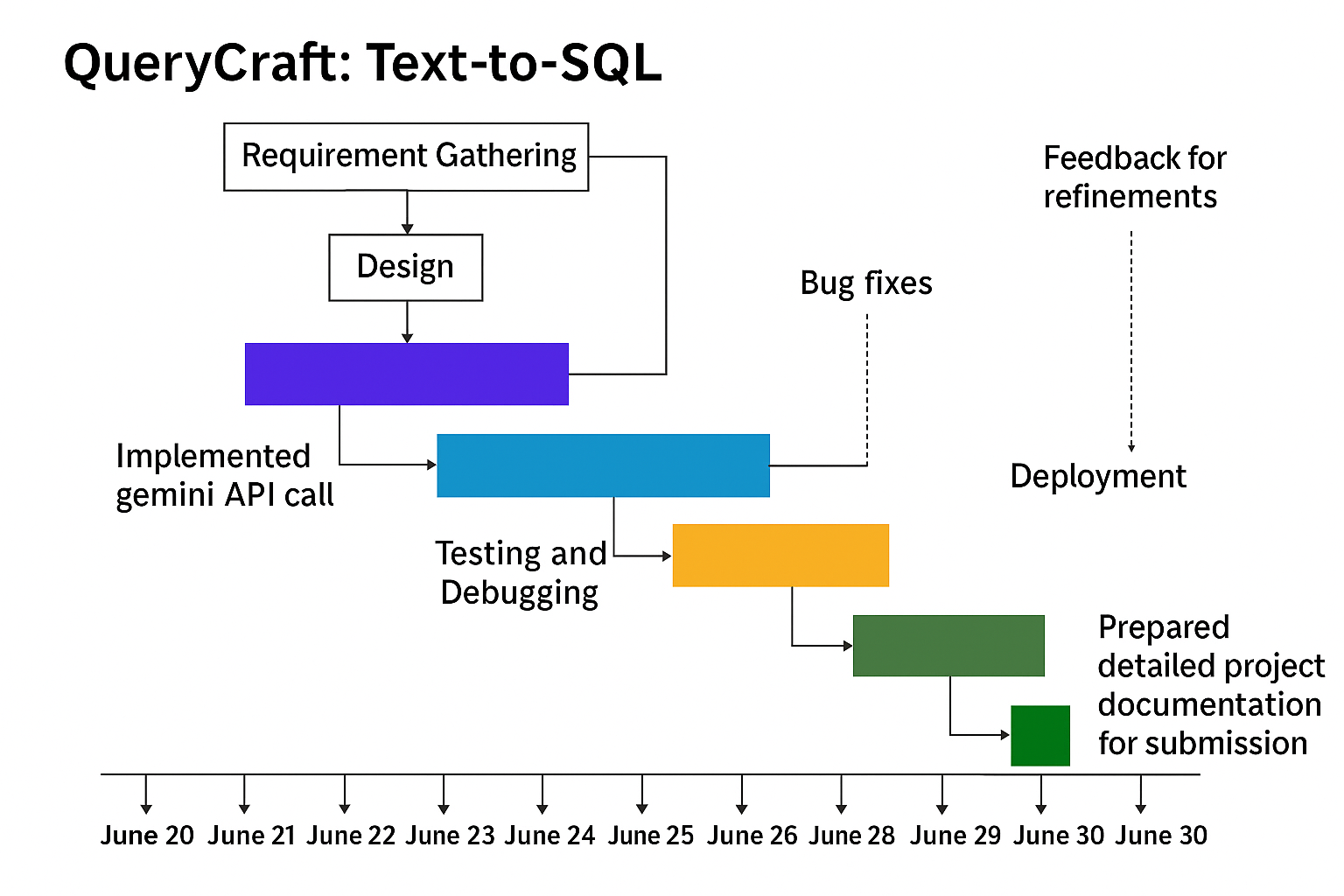
**The main goals of project planning for QueryCraft were:**

1. **Define the scope and deliverables of the QueryCraft project clearly, including natural language processing, SQL generation, and user interface components.**
2. **Identify major development milestones such as prompt design, Gemini API integration, UI development, and testing.**
3. **Allocate tasks and time effectively among team members to ensure a balanced and productive workflow.**
4. **Minimize risks and ensure quality, such as managing API key security, handling unexpected model responses, and UI usability.**

**3.Project Phases**



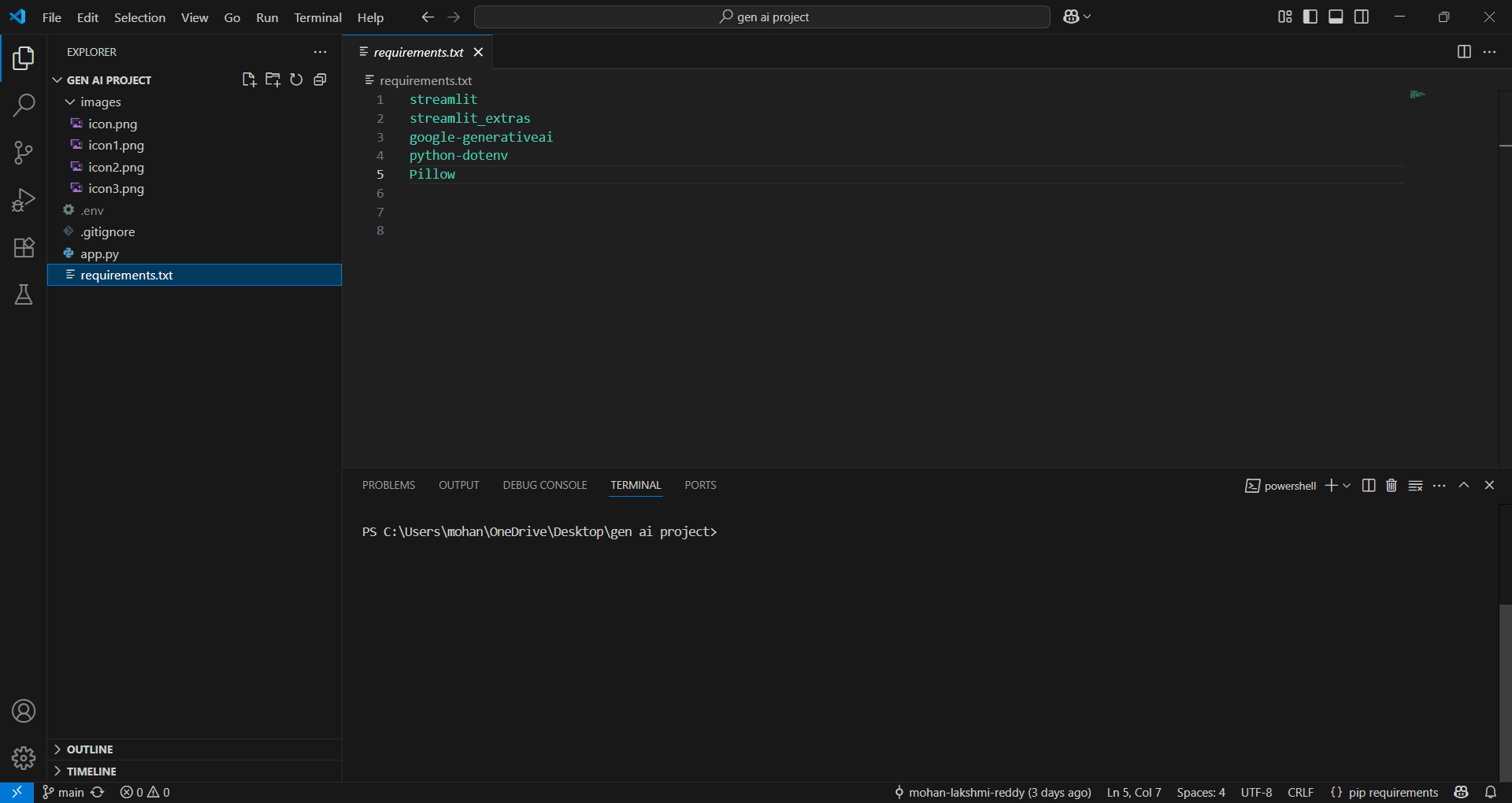
**SCHEDULING BREAKING**



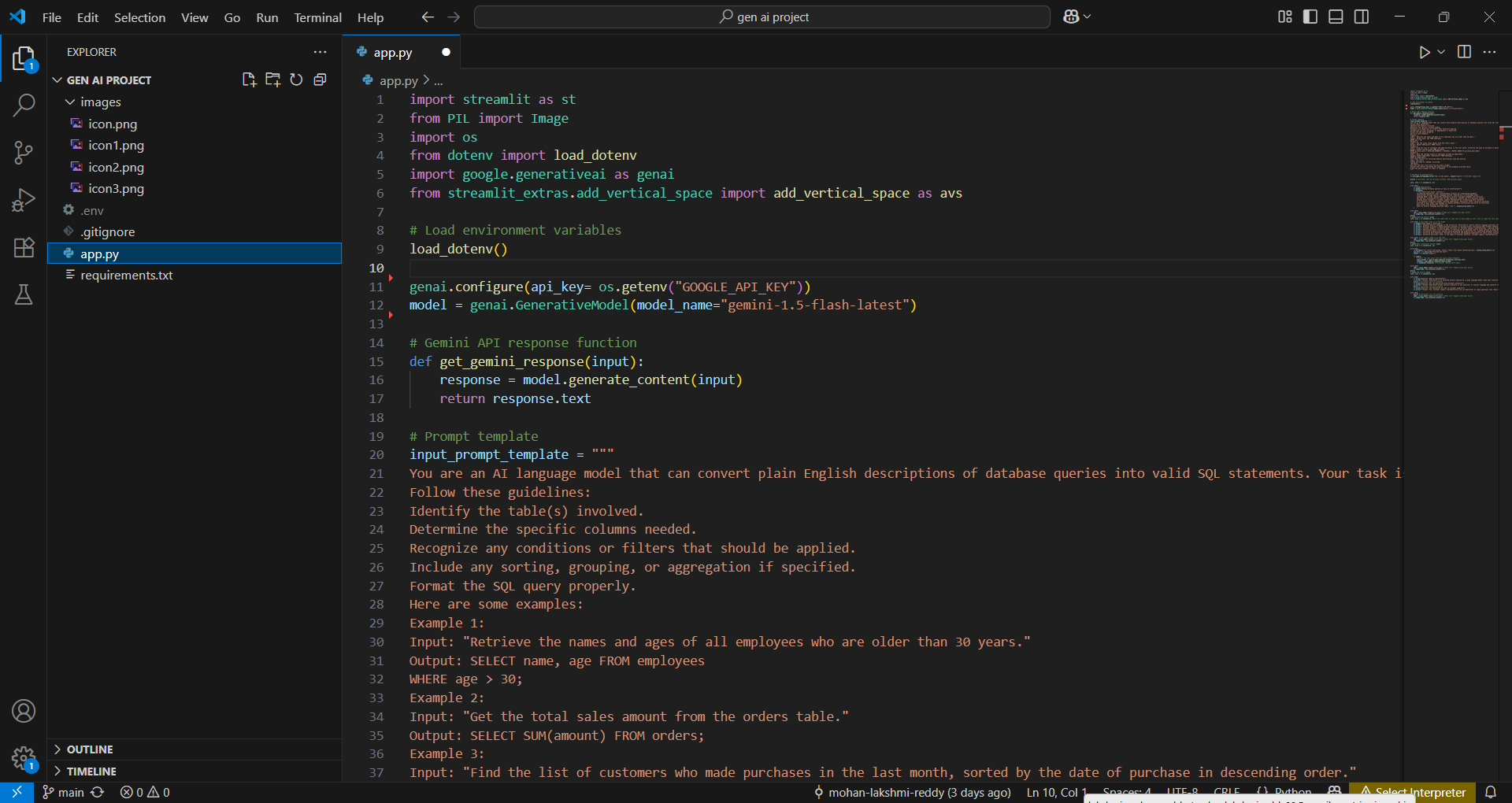
**Developing**

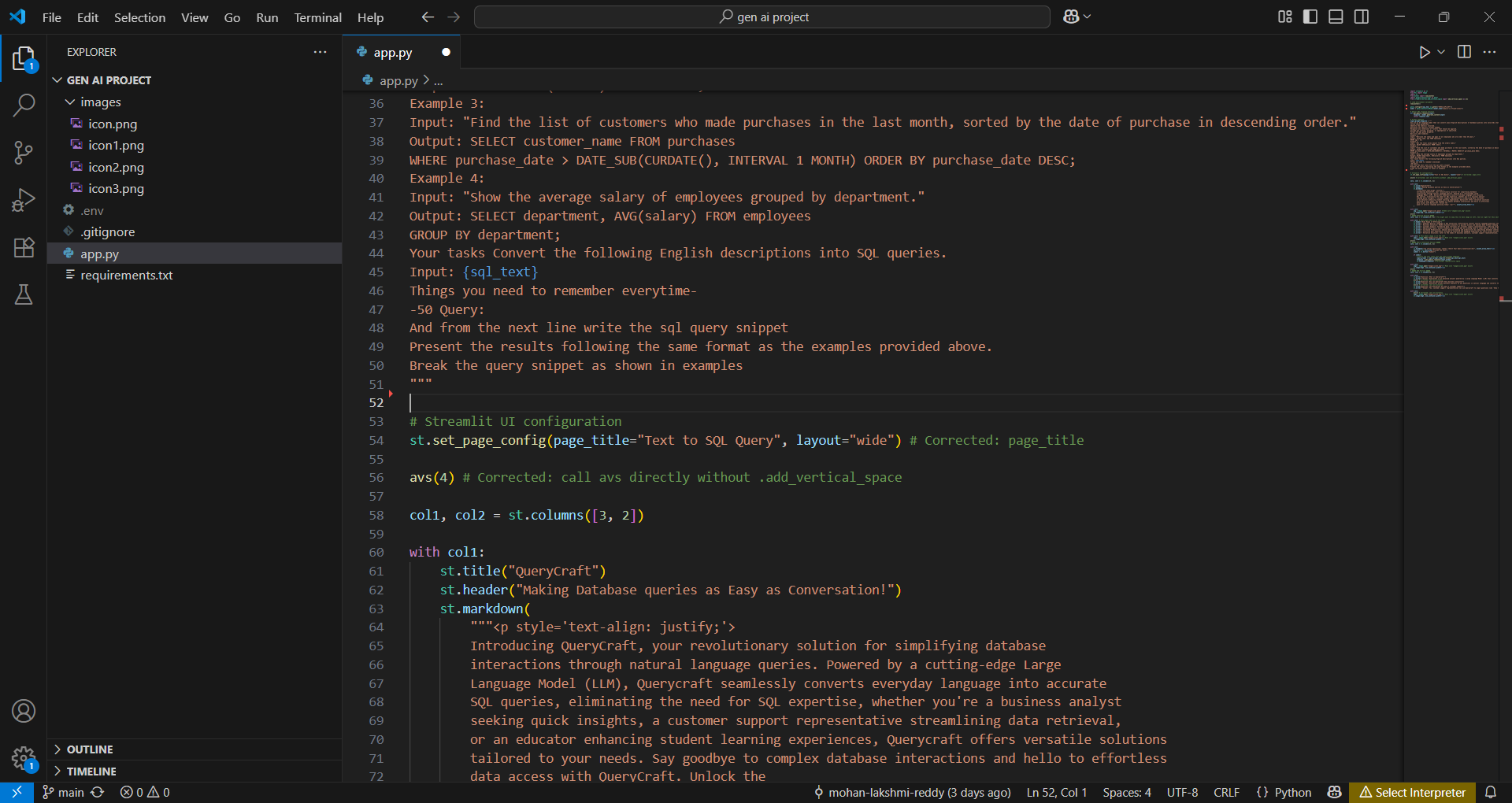
**🧠 Key Code Snippets**

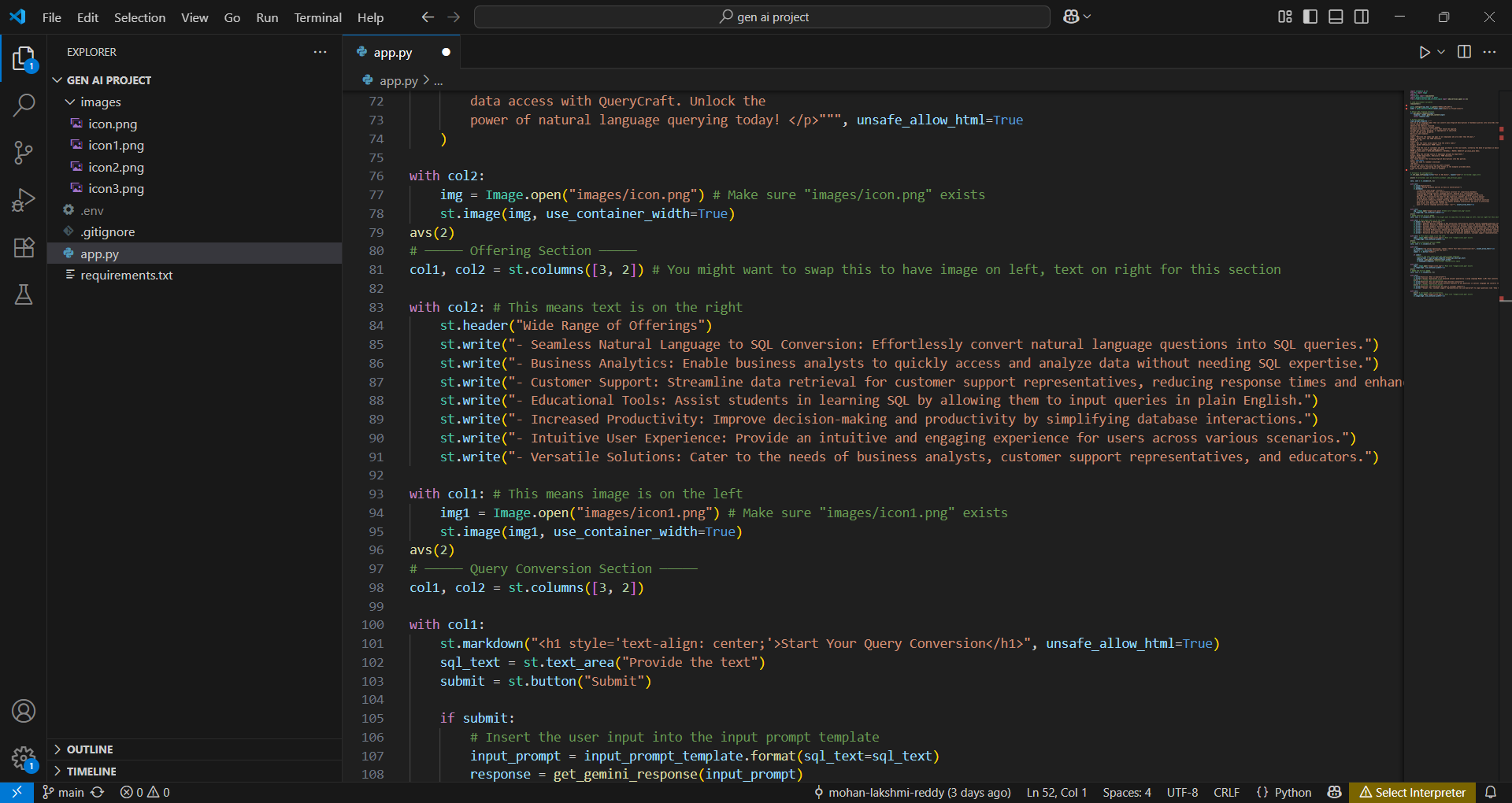
**REQUIREMENTS.TXT**

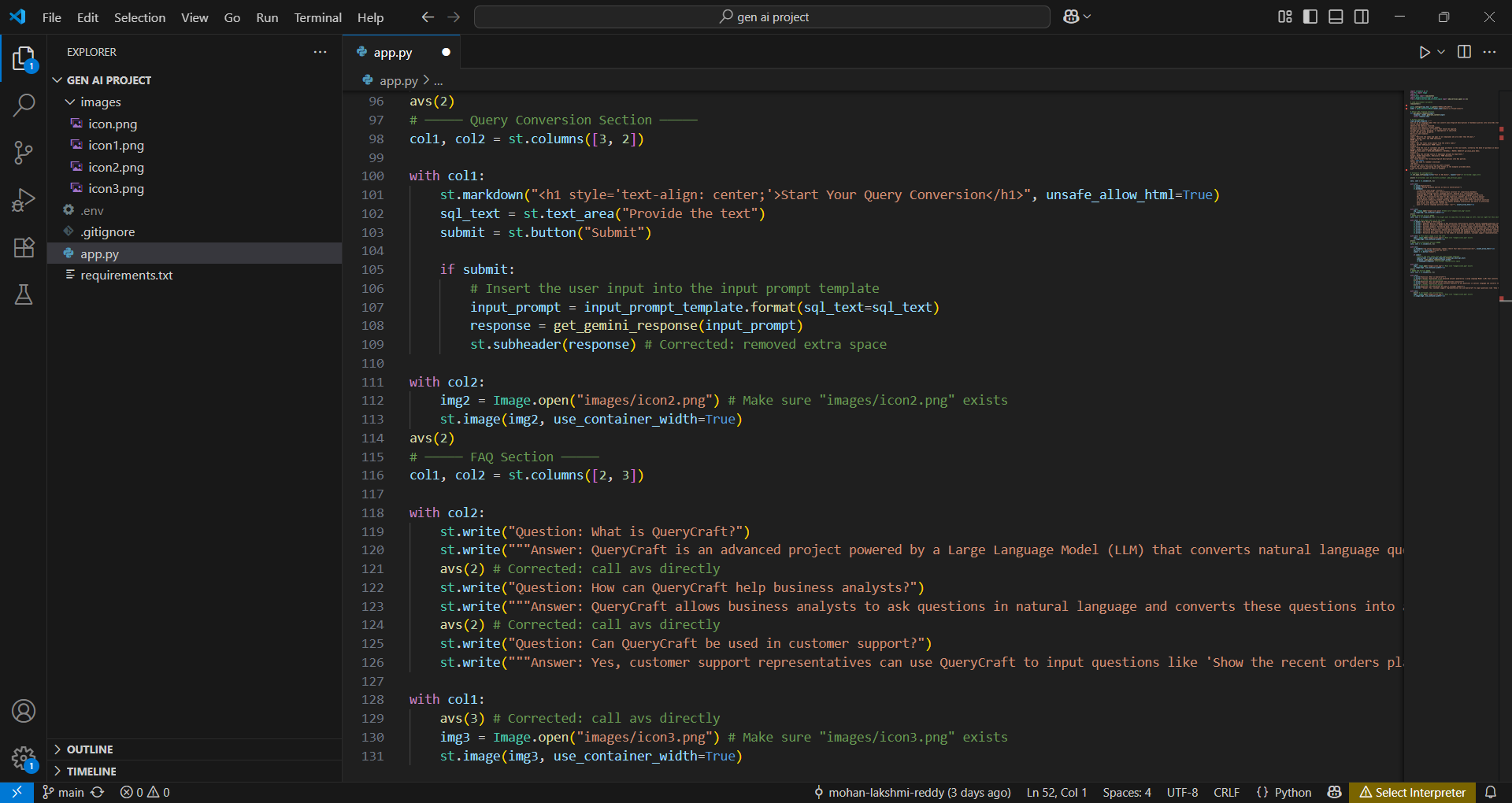
****

**APP.PY**

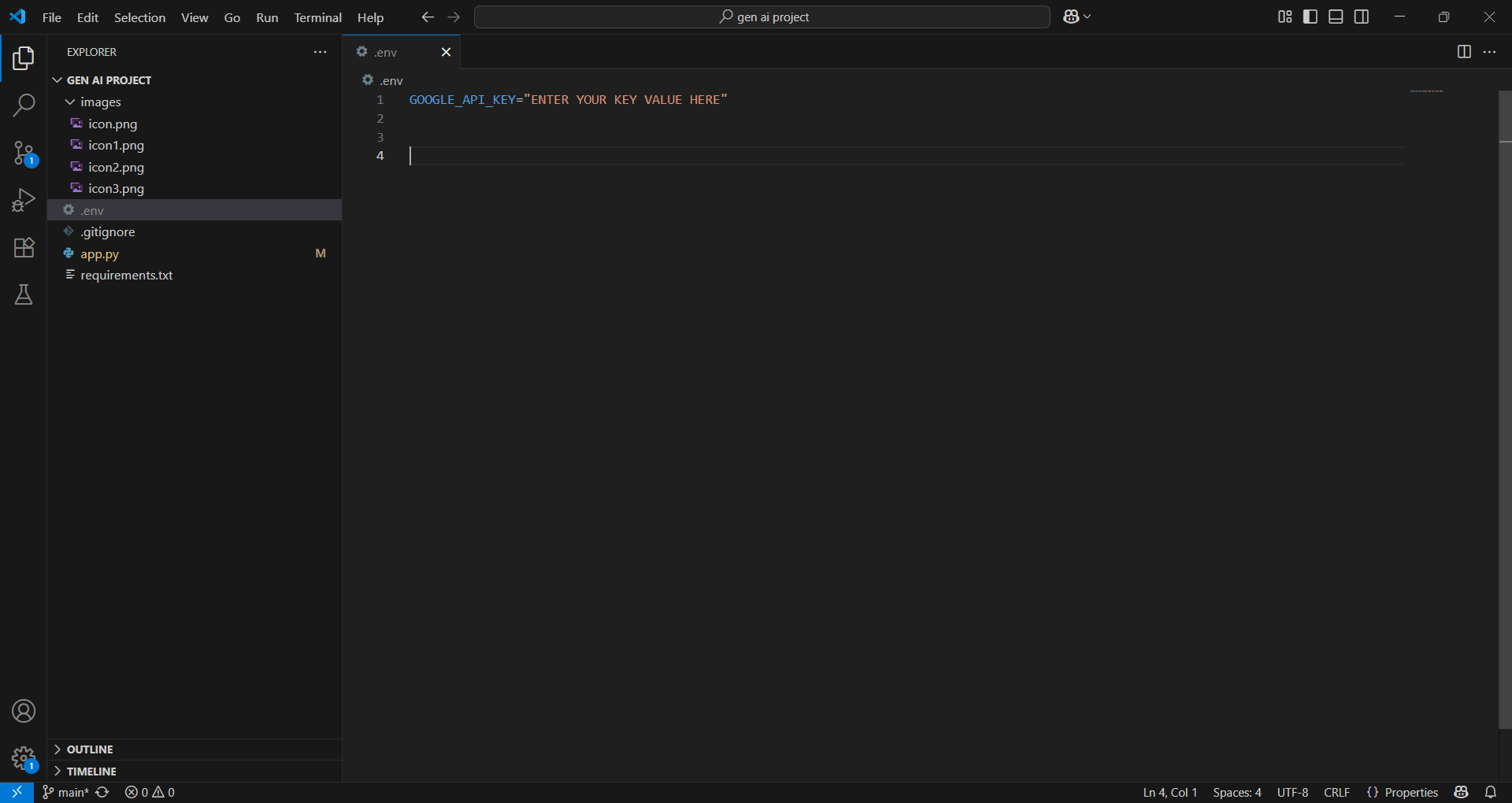
****

****

****

****

**.ENV**

****

**🧪 Setup & Execution Instructions:**

1. **Clone the repo:**

**git clone https://github.com/mohan-lakshmi-reddy/querycraft-text2sql-using-gemini.git**

**cd querycraft-text2sql-using-gemini**

1. **Create a .env file:**

**env**

**GOOGLE\_API\_KEY=your\_api\_key\_here**

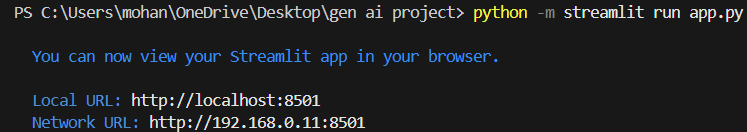
1. **Install dependencies:**

**pip install -r requirements.txt**

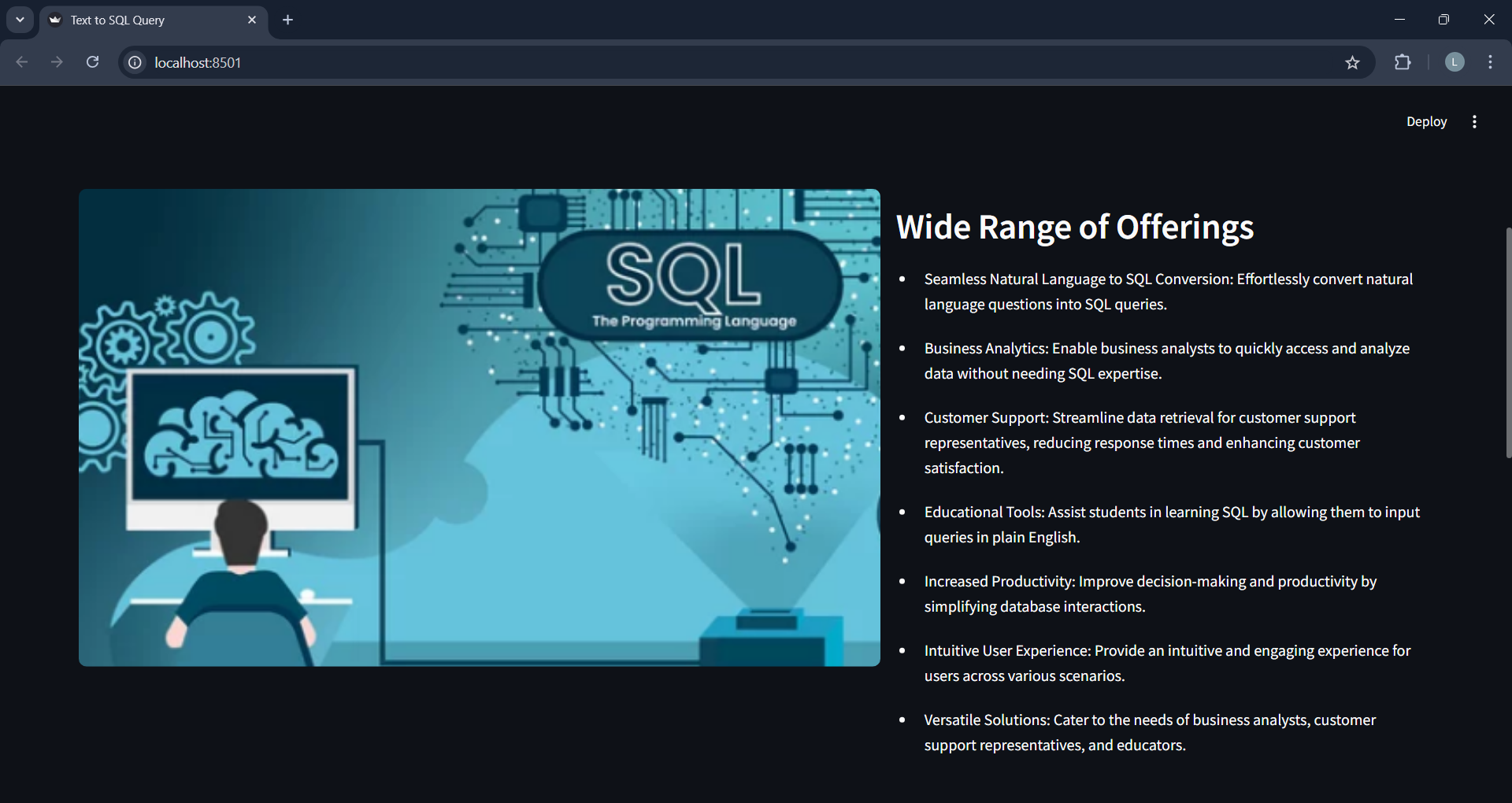
1. **Run the app:**

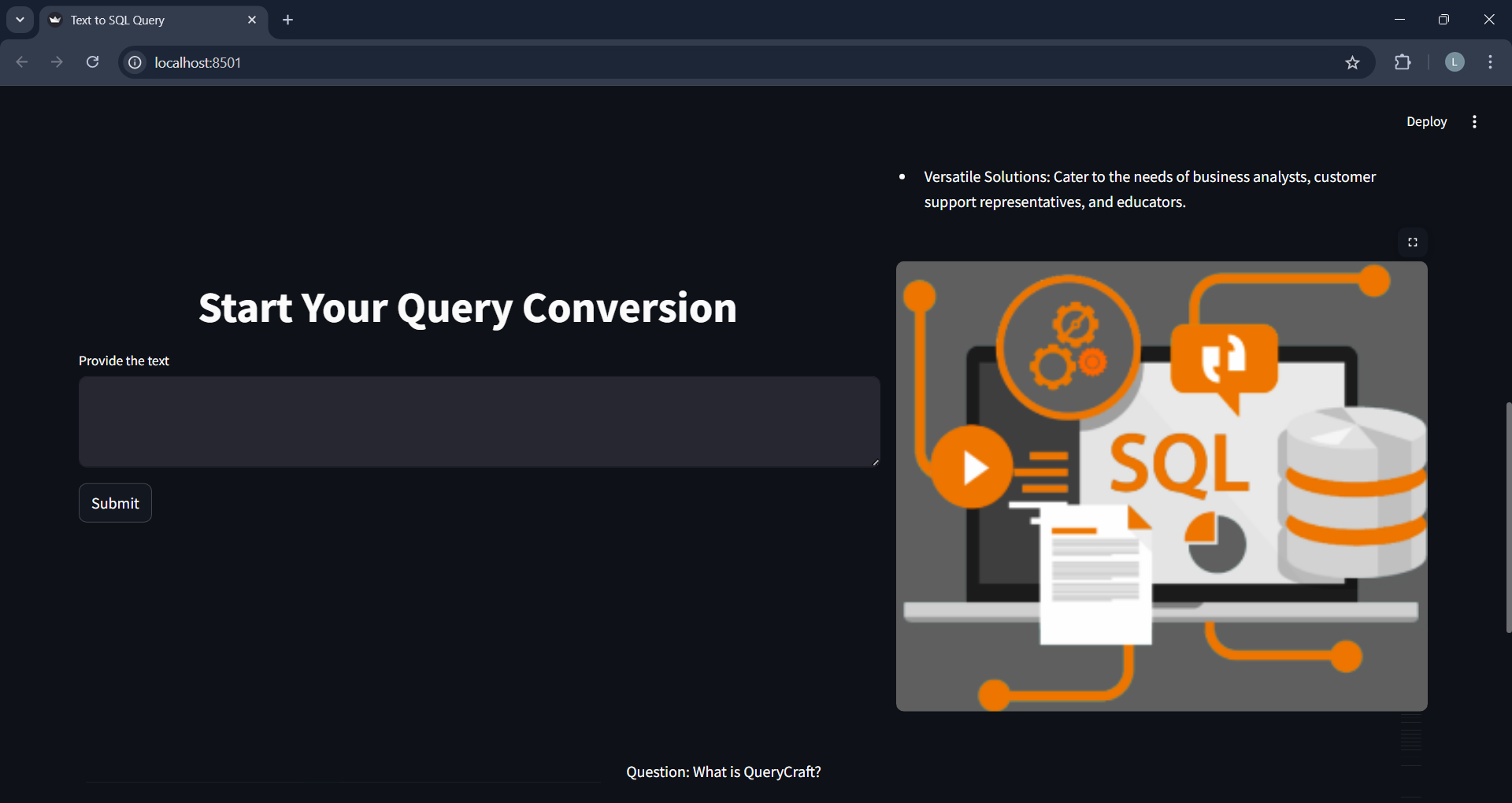
**streamlit run app.py**

**🖼️ Screenshots:**

****

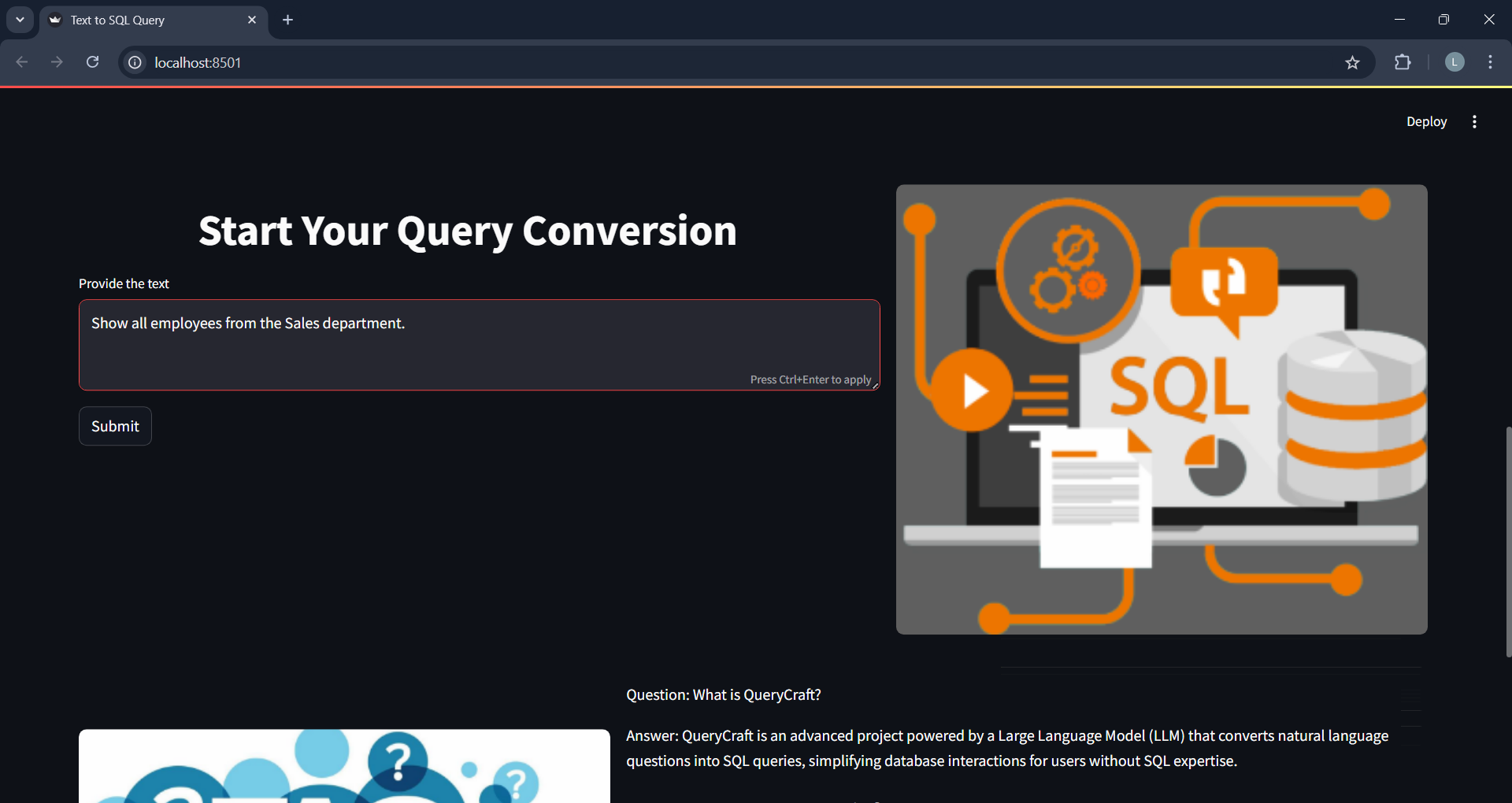


****

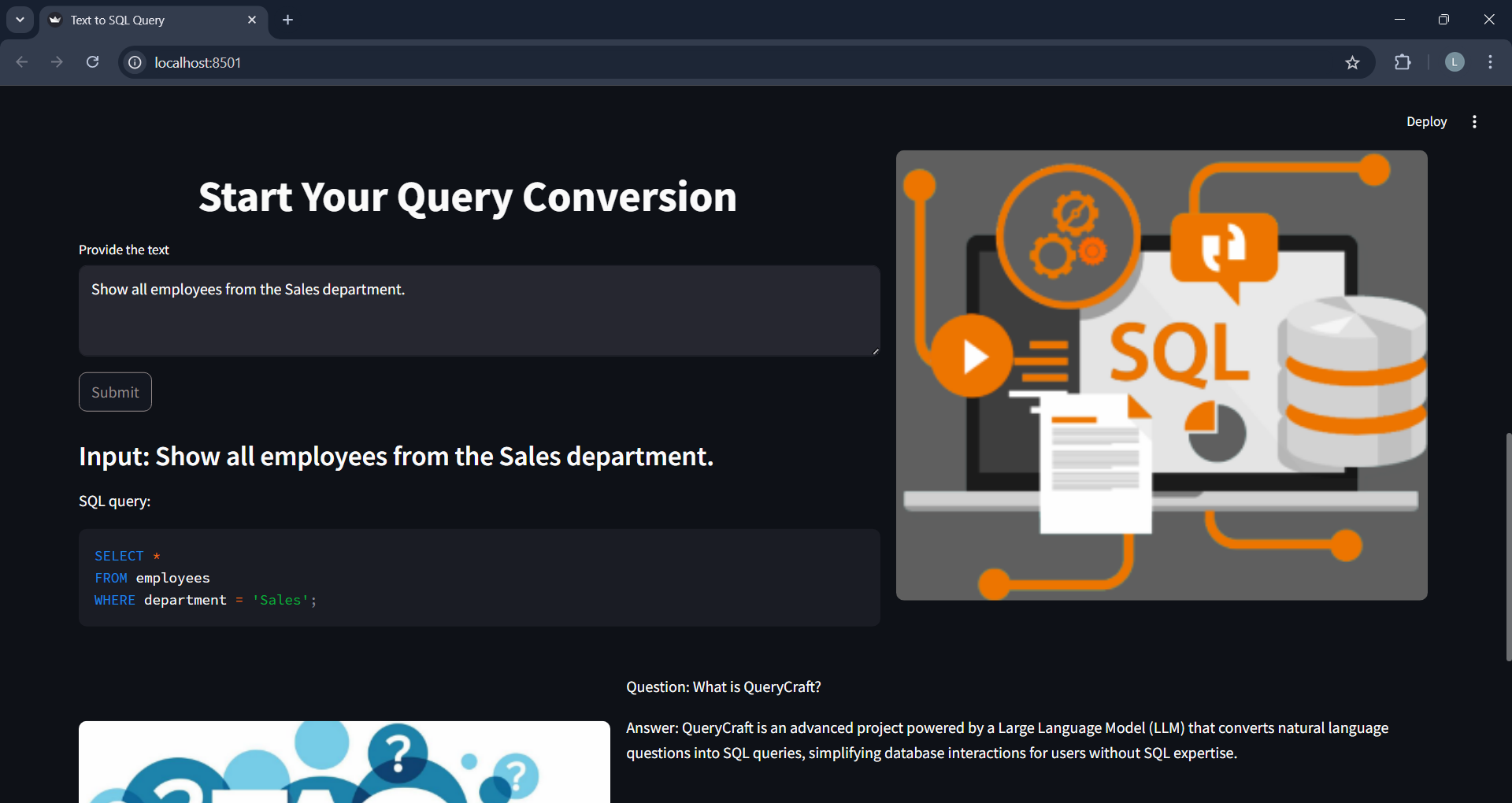
****

**TESTING**

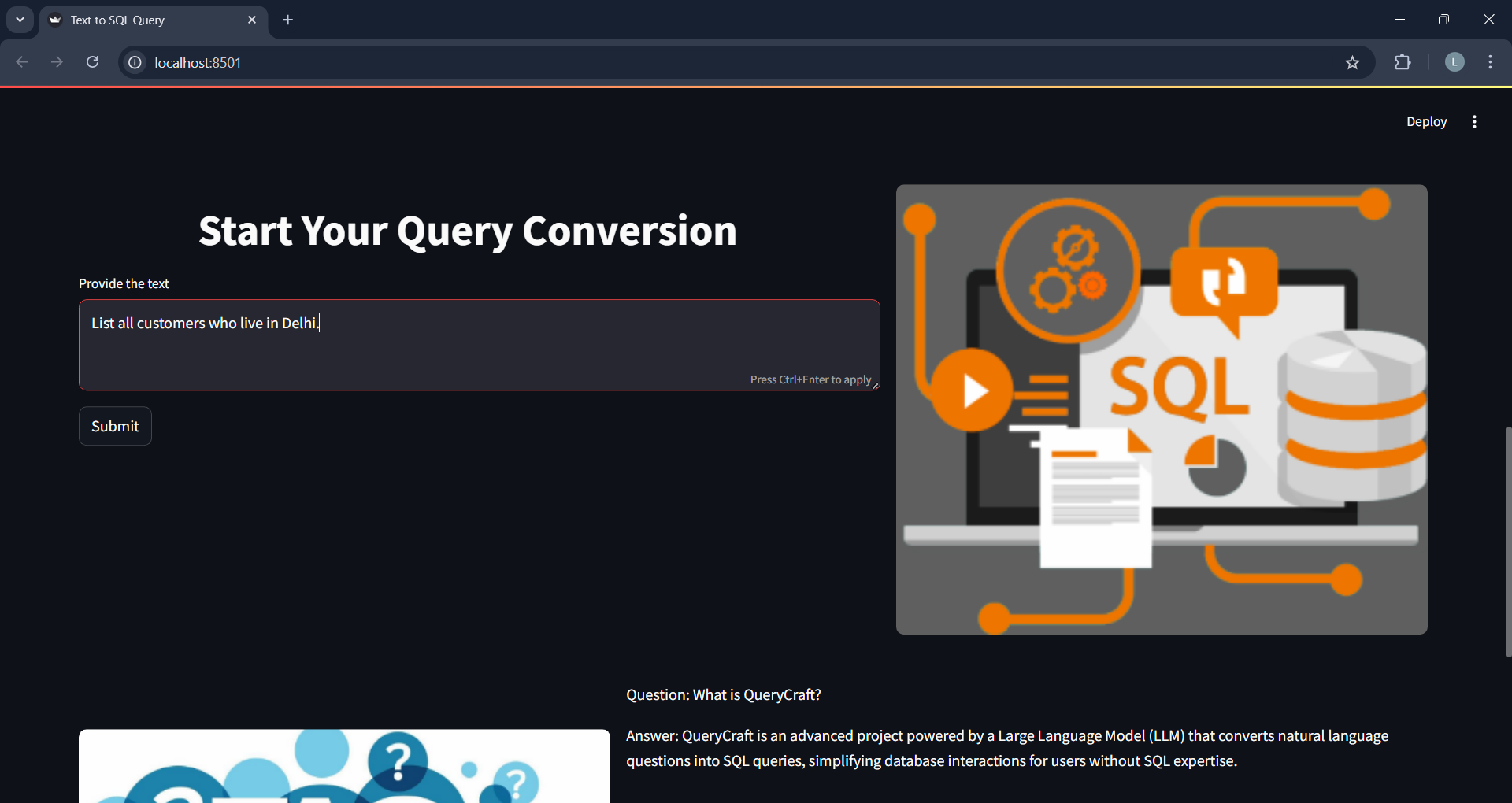
**Input1**

****

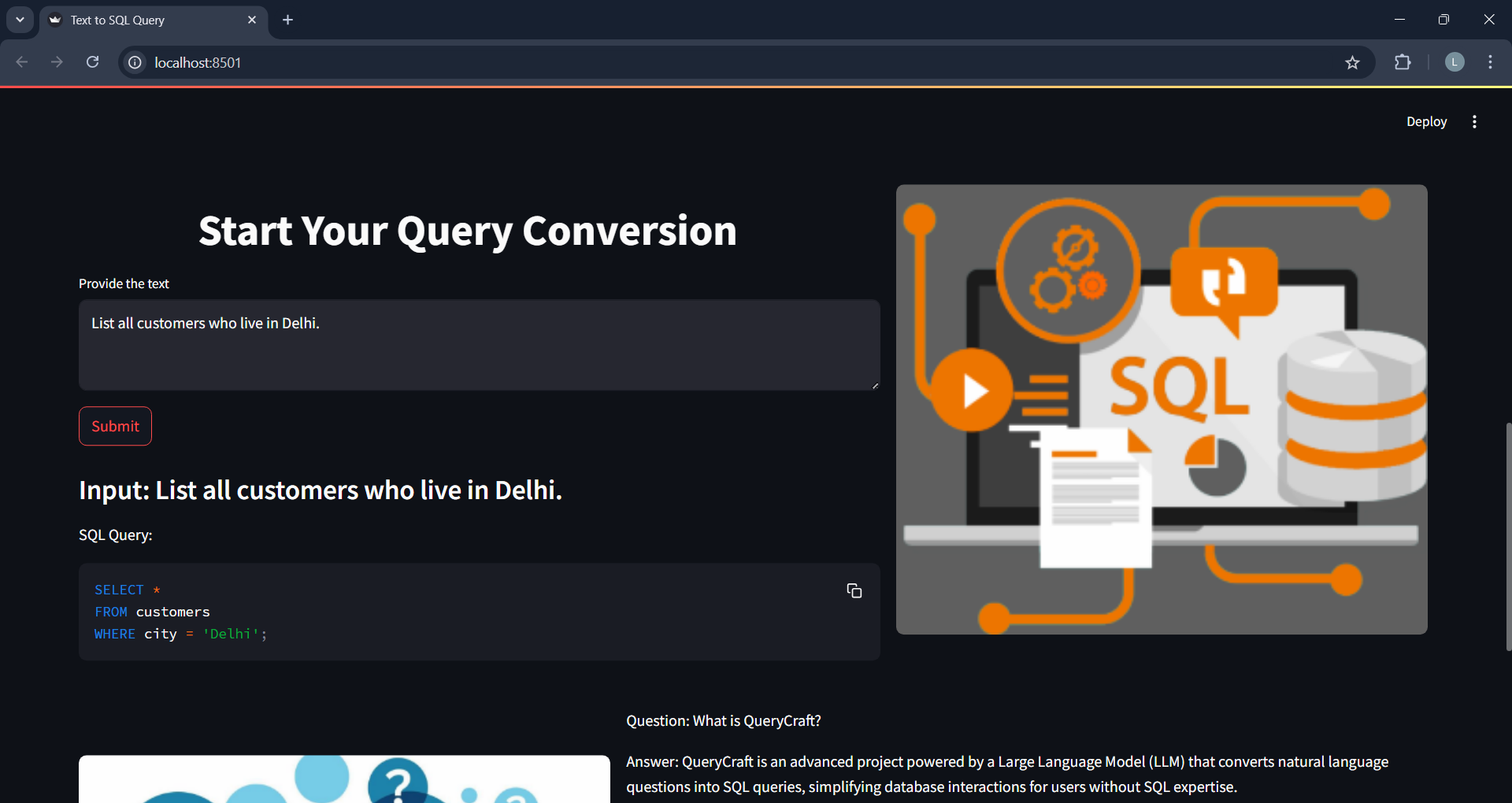
**Output1**

****

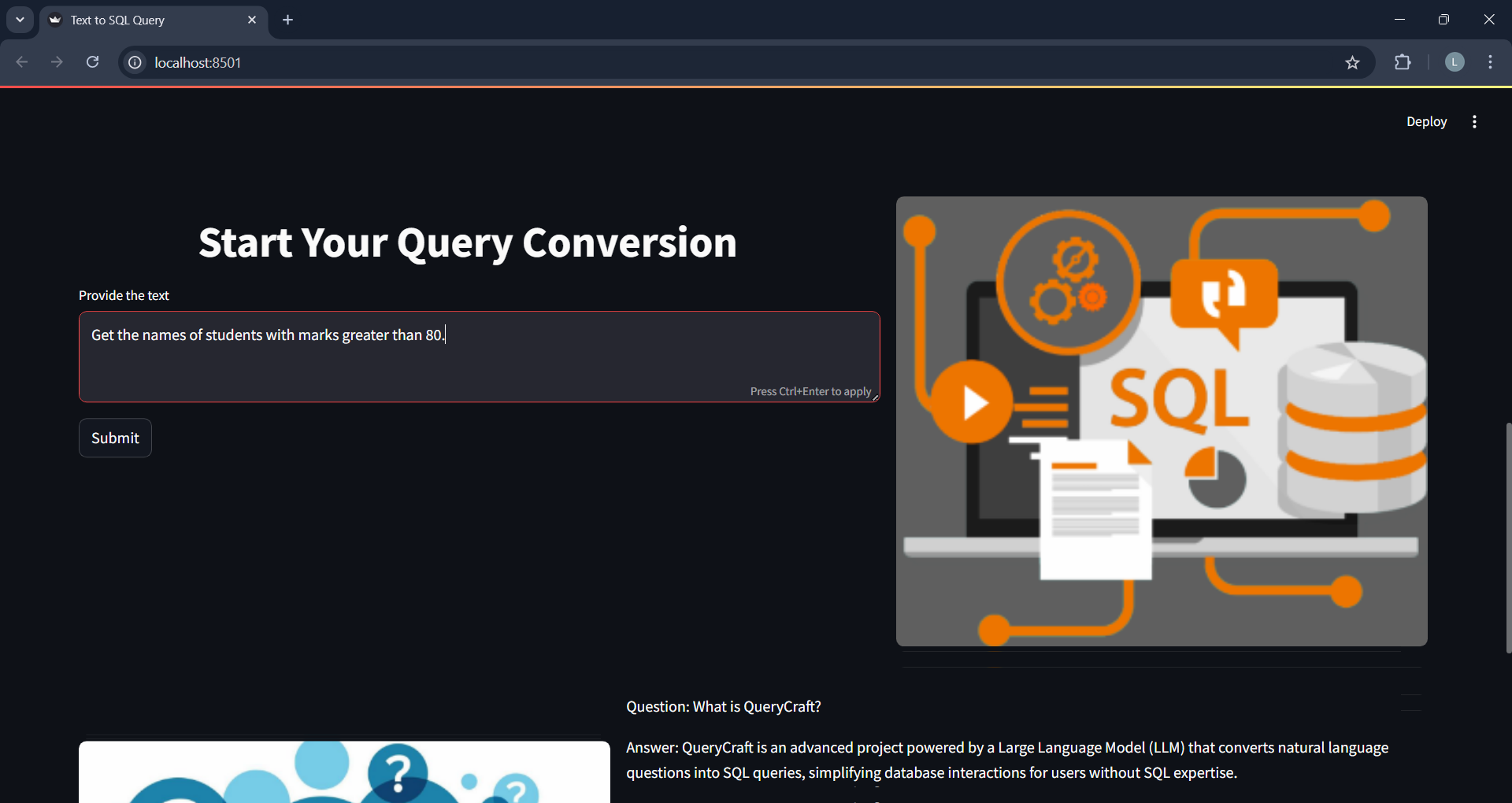
**Input2**

****

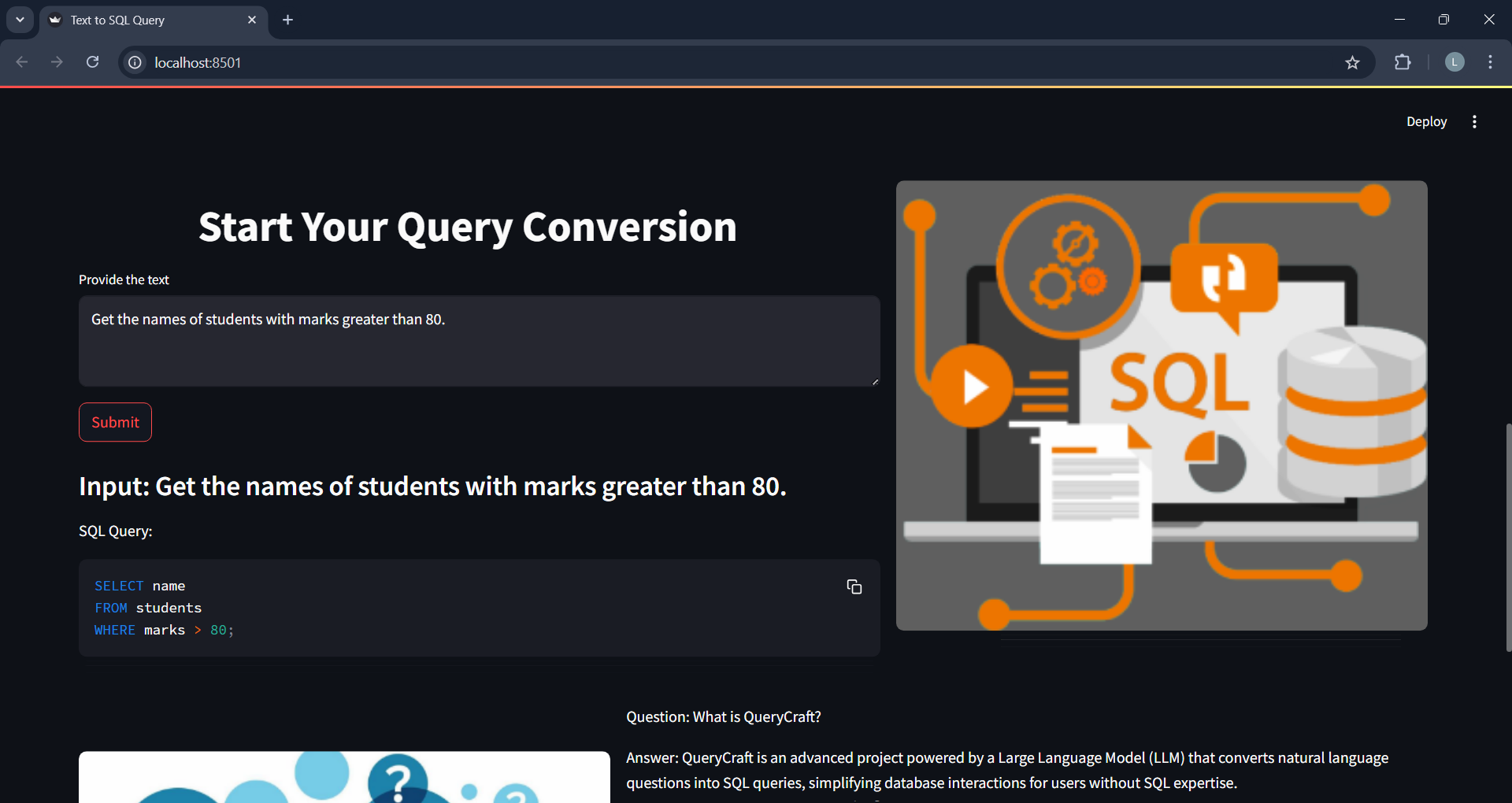
**Output2**

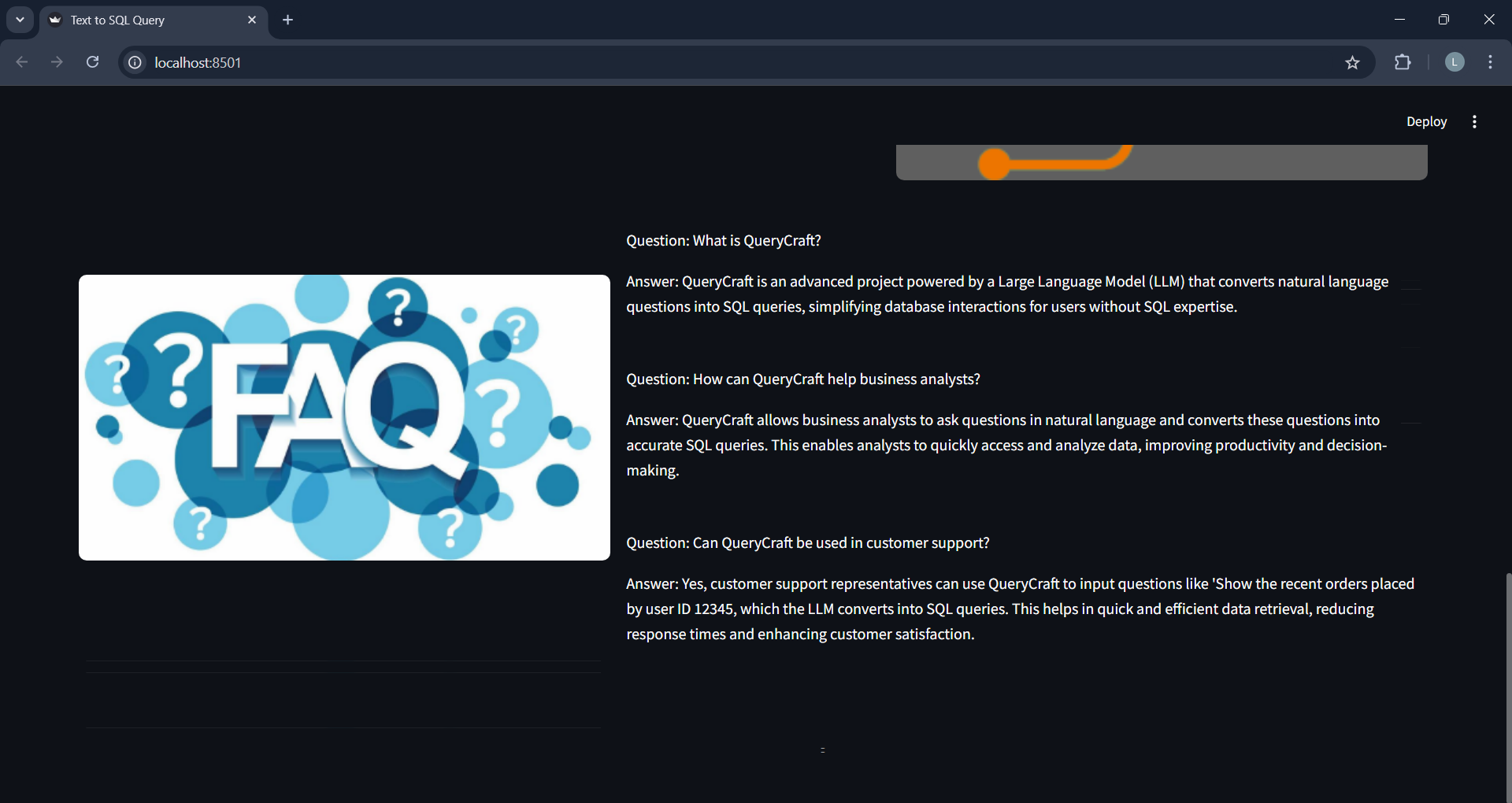
****

**Input3**

****

**Output3**

****

****

**📹 Demo Video:**[***https://drive.google.com/file/d/1whm4\_FVTxsM49NsMDaf7JwwaFN8MJ-tX/view?usp=sharing***](https://drive.google.com/file/d/1whm4_FVTxsM49NsMDaf7JwwaFN8MJ-tX/view?usp=sharing)

**🔗 GitHub Repository:**

**🔗** [**https://github.com/mohan-lakshmi-reddy/querycraft-text2sql-using-gemini**](https://github.com/mohan-lakshmi-reddy/querycraft-text2sql-using-gemini)

**Advantages of Text-to-SQL using Gemini :**

* **No SQL Required: Anyone can query databases without writing SQL.**
* **Time-Saving: Rapid generation of complex queries.**
* **Educational Value: Helps students understand SQL structure.**
* **User-Friendly UI: Web interface via Streamlit.**
* ***Cloud/Local Flexibility: Can be run on Colab or local machine.***
* ***Real-Time Output: Instant SQL generation.***
* **Supports Multiple Domains: Sales, HR, Education, Marketing, etc.**

**Disadvantage of Text-to-SQL using Gemini :**

* **Model Limitations: Accuracy depends on Gemini’s understanding.**
* **No Execution (Optional): Without linking to a real database, execution is not possible.**
* **Data Schema Dependency: Model may require schema hints for complex queries.**
* **Internet Requirement: Gemini API requires connectivity.**
* **Security Risks: Input validation needed to avoid injection issues.**
* **Edge Cases May Fail: Ambiguous queries might return incorrect SQL.**

**✅ Conclusion:**

**This project enhanced our understanding of how Generative AI models like Gemini can solve real-world problems such as translating human language into structured queries. It gave our team hands-on experience with prompt engineering, API handling, and frontend development. This internship allowed us to explore the intersection of AI and databases in a practical way.**

**📚 References:**

* [**Google Gemini Developer Docs**](https://ai.google.dev/)
* **Streamlit Documentation**
* [**SmartBridge Internship Resources**](https://smartinternz.com/)
* [**Python dotenv Package**](https://pypi.org/project/python-dotenv/)