**📘 SQL Query & Visualization AI System Documentation**

**🚀 Overview**

**Welcome to the SQL Query & Visualization AI System! This powerful system is designed to:**

**✅ Generate optimized SQL queries based on natural language input. ✅ Execute and validate SQL queries safely. ✅ Visualize query results using Matplotlib, Plotly, and Pandas. ✅ Leverage AI agents to automate SQL generation & visualization.**

**Built on CrewAI, this system employs a structured workflow where AI-driven agents collaborate seamlessly to generate SQL queries and visualize results—all while keeping performance and security in check.**

**🏗️ Tech Stack & Frameworks Used**

**🔹 CrewAI**

**CrewAI is a powerful framework for orchestrating multiple AI agents to work collaboratively on tasks. In this project, CrewAI is used to:**

* **Define SQL Query Agent and Visualization Agent.**
* **Handle task delegation and sequential execution.**
* **Maintain clear goal-driven processes for query execution and visualization.**

**🔹 LangChain & LangChain Community Tools**

**LangChain is used for:**

* **SQL Database tools for interacting with MySQL.**
* **Python REPL Tool for executing visualization scripts.**

**🔹 MySQL + SQLAlchemy**

**MySQL serves as the database, while SQLAlchemy manages database connections.**

**🔹 Data Visualization Libraries**

* **Matplotlib & Seaborn → Static charts.**
* **Plotly → Interactive graphs.**
* **Pandas → Data handling.**

**🛠️ System Workflow**

**1️ User Input**

**The user enters a natural language query or an SQL statement.**

**2️ SQL Query Agent (AI-Powered)**

* **Uses AI to convert user input into an optimized SQL query.**
* **Checks available tables and schema.**
* **Validates the query before execution.**
* **Executes the query securely.**

**3️ Data Extraction**

* **The extracted data is stored in memory.**
* **Transformed into a Pandas DataFrame.**

**4️ Visualization Agent (AI-Powered)**

* **Analyzes the query results.**
* **Decides the best visualization method.**
* **Uses Matplotlib, Seaborn, and Plotly to create clear insights.**
* **Runs code in a Python REPL environment to render the visualization.**

**5️ Display Results**

* **The SQL result is printed.**
* **The visualization appears.**

**🛠️ Tools Used in the System**

**🔹 SQL Query Tools**

| **Tool Name** | **Function** |
| --- | --- |
| **list\_tables** | **Lists all available tables in the database.** |
| **tables\_schema** | **Retrieves schema details of selected tables.** |
| **execute\_sql** | **Executes the SQL query and fetches results.** |
| **check\_sql** | **Validates the correctness of an SQL query.** |

**🔹 Python REPL Tool**

| **Tool Name** | **Function** |
| --- | --- |
| **python\_repl** | **Executes Python code for visualizations and data transformations in a secure REPL environment.** |

**🖼️ Using the Python REPL Tool for Visualization**

**🚀 The Python REPL Tool allows you to execute Python scripts dynamically. You can visualize SQL query results using Pandas, Matplotlib, and Plotly.**

**🔥 Example: Running a Bar Chart Visualization**

**Step 1: Execute a SQL Query**

**query\_result = execute\_sql("SELECT country, population FROM world ORDER BY population DESC LIMIT 10;")**

**Step 2: Convert to a DataFrame**

**import pandas as pd**

**from io import StringIO**

**data = StringIO(query\_result)**

**df = pd.read\_csv(data)**

**Step 3: Generate a Bar Chart**

**import matplotlib.pyplot as plt**

**plt.figure(figsize=(10,6))**

**plt.bar(df["country"], df["population"], color='skyblue')**

**plt.xlabel("Country")**

**plt.ylabel("Population")**

**plt.title("Top 10 Most Populated Countries")**

**plt.xticks(rotation=45)**

**plt.show()**

**🎨 Boom! The visualization appears instantly! You can modify the script to generate line graphs, scatter plots, and more.**

**🖥️ CLI Usage Guide**

**🚀 Running the SQL Query & Visualization System**

**Run the script directly in a terminal:**

**python your\_script.py**

**📌 Enter a SQL Query in Natural Language**

**Enter your SQL Query in Natural Language: Show me the top 10 most populated countries.**

**📌 System Response**

**Query Result:**

**+------------+------------+**

**| Country | Population |**

**+------------+------------+**

**| China | 1.4B |**

**| India | 1.3B |**

**...**

**📊 Generating visualization...**

**A bar chart pops up showing the top 10 most populated countries! 🎉**

**🔴 Exit**

**Type exit to quit.**

**🔥 Key Features**

**✅ AI-powered SQL writing - No need to be an expert in SQL! ✅ Real-time SQL validation - Ensure queries are safe & correct. ✅ Seamless data visualization - Convert query results into insights. ✅ Python REPL execution - Run Python code dynamically. ✅ Interactive CLI interface - User-friendly experience.**

**🎯 Final Thoughts**

**This AI-driven SQL & Visualization system automates SQL query generation and visualization—turning raw data into actionable insights instantly! 🚀💡**