

Project Documentation: QueryCraft

- **Samhitaa Saravanakumar**
- **Aayush Emmanuel**
- **Dhyan Salavi**
- **Bharati Sameera**

1. Problem Statement

In many organizations, users who need to interact with databases often lack proficiency in SQL. This creates a dependency on technical personnel, delays data retrieval, and limits decision-making efficiency. QueryCraft addresses this challenge by enabling users to input queries in natural language, which are then converted into SQL using a large language model (LLM), specifically Google's Gemini Pro. The goal is to make database querying accessible to non-technical users such as analysts, support representatives, and students.

2. Project Requirements

2.1 Functional Requirements

- Accept natural language input through a user interface.
- Process the input using the Gemini Pro LLM.
- Generate accurate SQL queries based on user input.
- Display the generated SQL and query output in the frontend.

2.2 Non-Functional Requirements

- Ensure secure handling of API keys and sensitive data.
 - Maintain a responsive user interface with minimal latency.
 - Provide a scalable backend capable of handling multiple requests.
 - Implement error handling and logging mechanisms.
-

3. User Stories

User Role	User Story
Business Analyst	As a business analyst, I want to ask questions in plain English and receive SQL queries to retrieve relevant data.
Customer Support Agent	As a support agent, I want to access user-related data by typing natural language queries.
Student	As a student, I want to understand how natural language maps to SQL for educational purposes.

4. Project Planning and Scheduling

Timeline Overview

Task	Duration	Status
Requirements Gathering	Day 1	Completed
API Setup and Configuration	Day 2	Completed
Frontend Design and Development	Day 3–4	Completed
Backend Development and Integration	Day 5–6	Completed
Testing and Debugging	Day 7	In Progress
Hosting and Deployment	Day 8	Pending

5. System Architecture and Workflow

High-Level Workflow:

1. User submits a natural language query via the frontend interface.
2. The frontend transmits the input to the backend using the Google API key.
3. The backend invokes Gemini Pro to process the input.
4. The model returns a SQL query.
5. The frontend receives and displays the result.

6. Technical Setup and Requirements

Required Libraries

List of libraries included in `requirements.txt`:

```
nginx
CopyEdit
flask
google-generativeai
python-dotenv
```

Environment Setup

- Store the Google API Key securely in a `.env` file:

```
ini
CopyEdit
GOOGLE_API_KEY=your_api_key_here
```

- Initialize the key in your Python code using `dotenv`.
-

7. Codebase Structure

- `app.py` – Main application backend using Flask.
 - `templates/index.html` – User interface.
 - `utils.py` – Contains the Gemini Pro model interaction function.
 - `.env` – Secure storage for API key.
 - `requirements.txt` – Python dependency file.
-

8. Integration with Gemini Pro

Steps to implement:

- Configure the API using `google.generativeai`.
- Write a prompt template to instruct the model.
- Implement `get_gemini_response()` to process user input and return SQL.

Example usage:

```
python
CopyEdit
import google.generativeai as genai
genai.configure(api_key=os.getenv("GOOGLE_API_KEY"))
```

9. Use Case Scenarios

Business Analytics

Input: “What were the total sales last quarter?”

Generated SQL:

```
sql
CopyEdit
SELECT SUM(sales) FROM orders WHERE order_date BETWEEN '2024-01-01' AND
'2024-03-31';
```

Customer Support

Input: “Show the recent orders placed by user ID 12345”

Generated SQL:

```
sql
CopyEdit
SELECT * FROM orders WHERE user_id = 12345 ORDER BY order_date DESC;
```

Educational Tools

Input: “List all students who scored above 90”

Generated SQL:

```
sql
CopyEdit
SELECT name FROM students WHERE score > 90;
```

10. Testing

Unit Testing

- Validate input formats and edge cases.
- Simulate API errors and check responses.

Integration Testing

- Perform end-to-end flow testing from UI input to final SQL output.
 - Test multiple use-case scenarios for accuracy.
-

11. Advantages

- Reduces dependency on SQL expertise.
 - Increases productivity by speeding up data retrieval.
 - Adaptable across multiple domains.
 - Encourages learning by demonstrating SQL equivalents of natural language queries.
-

12. Limitations

- Accuracy is dependent on the LLM’s understanding and prompt effectiveness.
 - Requires internet access and an active Google API key.
 - May not fully support very complex or ambiguous queries without fine-tuning.
-

13. Conclusion

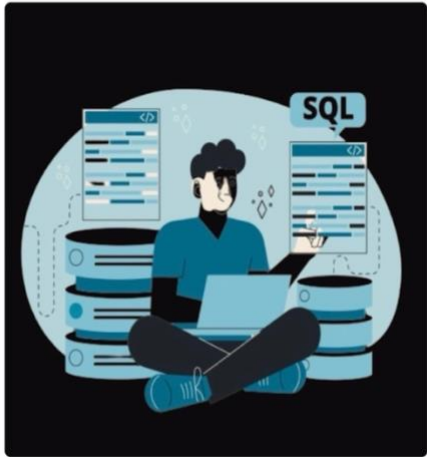
QueryCraft offers an innovative solution for bridging the gap between natural language understanding and structured database querying. By leveraging the Gemini Pro large language model, the application empowers users across various domains to interact with databases efficiently and intuitively. With further development, this project can evolve to support more advanced use cases, including voice-based queries, multilingual support, and integration with diverse database systems.

QueryCraft


Making Database Queries as Easy as Conversation!

Introducing QueryCraft, your revolutionary solution for simplifying database interactions through natural language queries. Powered by a cutting-edge Large Language Model (LLM), QueryCraft seamlessly converts everyday language into accurate SQL queries, eliminating the need for SQL expertise. Whether you're a business analyst seeking quick insights, a customer support representative streamlining data retrieval, or an educator enhancing student learning experiences, QueryCraft offers versatile solutions tailored to your needs. Say goodbye to complex database interactions and hello to effortless data access with QueryCraft. Unlock the power of natural language querying today!

The use_column_width parameter has been deprecated and will be removed in a future release. Please utilize the use_container_width parameter instead.



The use_column_width parameter has been deprecated and will be removed in a future release. Please utilize the use_container_width parameter instead.



Wide Range of Offerings

- Seamless Natural Language to SQL Conversion: Effortlessly convert natural language questions into SQL queries.
- Business Analytics: Enable business analysts to quickly access and analyse data without needing SQL expertise.
- Customer Support: Streamline data retrieval for customer support representatives, reducing response times and enhancing customer satisfaction.
- Educational Tools: Assist students in learning SQL by allowing them to input queries in simple English.
- Increased Productivity: Improve decision-making and productivity by simplifying database interactions.
- Intuitive User Experience: Provide an intuitive and engaging experience for users across various scenarios.
- Versatile Solutions: Cater to the needs of business analytics, customer support representatives, and educators.

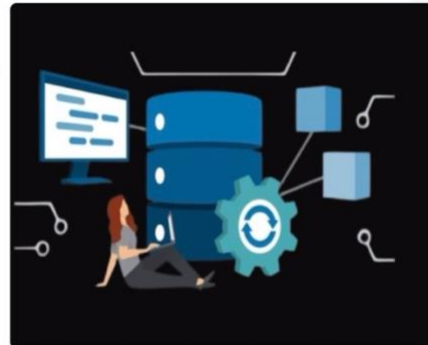


Start Your Query Conversation

Provide the text

Submit

The `use_column_width` parameter has been deprecated and will be removed in a future release. Please utilize the `use_container_width` parameter instead.



Question: What is QueryCraft?

Answer: QueryCraft is an advanced project powered by a Large Language Model (LLM) that converts natural language questions into SQL queries, simplifying database interactions for users without SQL expertise.

The `use_column_width` parameter has been deprecated and will be removed in a future release. Please utilize the `use_container_width` parameter instead.

Question: How can QueryCraft help business analysts?

Answer: QueryCraft allows business analysts to ask questions in natural language and these questions into accurate SQL queries. This enables analysts to quickly access and analyse improving productivity and decision-making.



Question: Can QueryCraft be used in customer support?

Answer: Yes, customer support representatives can use QueryCraft to input questions like 'Show the recent orders placed by user ID 12345', which the LLM converts into SQL queries. This helps in quick and efficient data retrieval, reducing response times and enhancing customer satisfaction.