Getting Started with the Wikipedia Question Answering System

# 1. Introduction

The **Wikipedia Question Answering System** is designed to help users get accurate and relevant information directly from Wikipedia based on their questions. The system leverages the Wikipedia API to search and retrieve summaries of Wikipedia articles that best match the user's input. It is especially useful for quickly retrieving concise, trustworthy answers without needing to browse through multiple Wikipedia pages.

This system includes several features to enhance the user experience:

* **Direct Wikipedia Search**: Simply input a question, and the system will extract the key concept and search Wikipedia for relevant articles.
* **Disambiguation Handling**: If a query could refer to multiple topics (e.g., "Java" could mean the programming language or the island), the system will present a list of possible options for you to choose from, ensuring you get the most accurate result.
* **User Feedback**: After receiving an answer, you can provide feedback on whether the information was helpful. This feedback helps the system improve over time.
* **Performance Tracking**: The system tracks its performance using metrics like Precision, Recall, and F1 Score, all of which are displayed based on user feedback.

By offering a streamlined interface and leveraging Wikipedia's vast database, this system provides quick and relevant answers, whether you're looking for information on historical events, scientific concepts, or general knowledge.

# 2. System Requirements

* **Required Python**:

Python 3.12.5 (What I used)

* **Required Python Packages**:

Openai 1.47.0

Wikipedia 1.4.0

Sentence-transformers 3.1.1

Scikit-learn 1.5.2

Streamlit 1.38.0

Pandas 2.2.3

* **Installation Instructions**:

(Optional) If possible, it is better to create a virtual environment so that you can use a clean environment to run the program. You can do it as below:

python3 -m venv virtualenv

source virtualenv/bin/activate

Install dependencies:

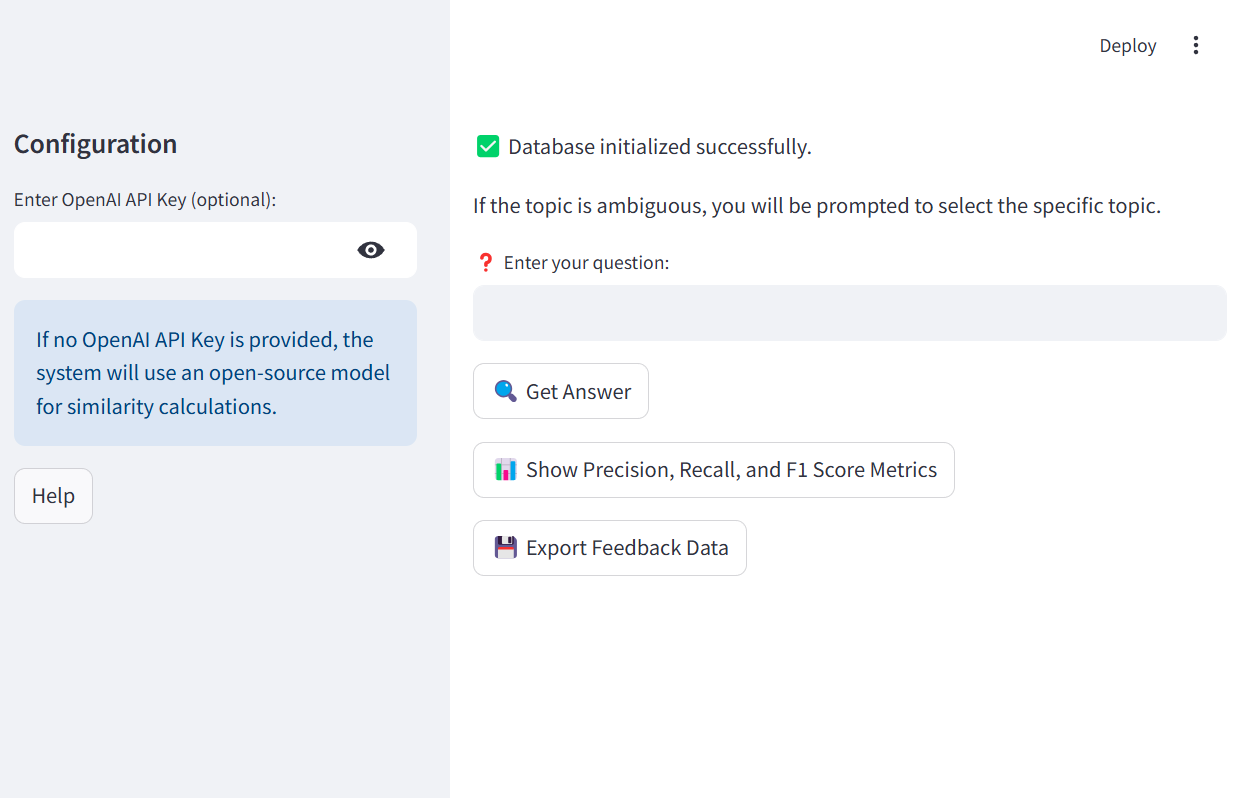
pip install -r requirements.txt

# 3. How to Launch the System

* **Launching the Application**:

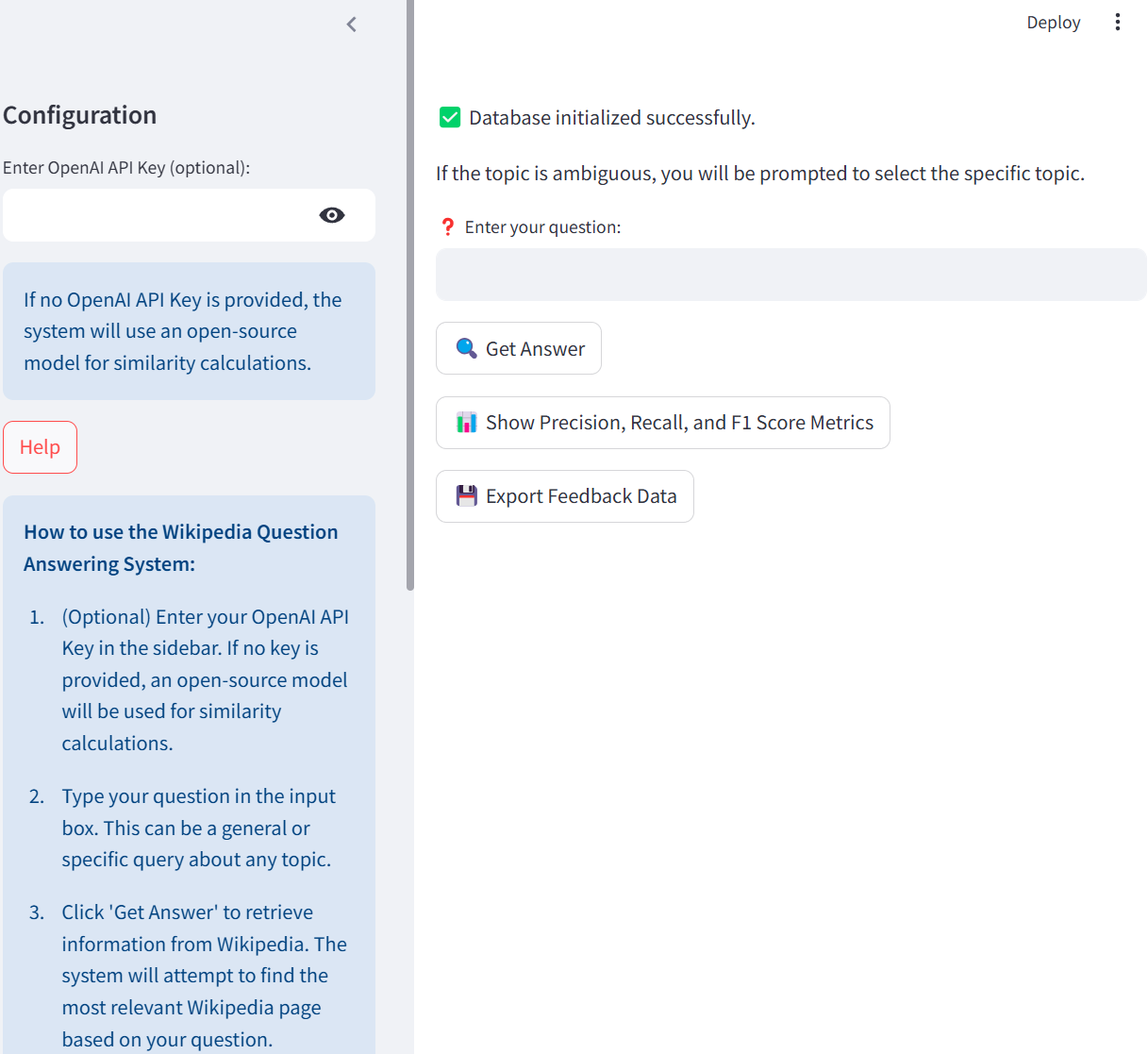
streamlit run app.py

#### Since the system may need to download or load the sentence-transformers model, it could take a little time, especially for the first time. Once it's ready, you should see an interface like the one below:



On the left, you can enter an OpenAI API key if you wish. This will allow the system to use "GPT-4o-mini" to enhance the results. However, it will also work with the open-source transformer model if you choose not to provide a key.

Additionally, you can click the "Help" button for instructions on how to use the system.

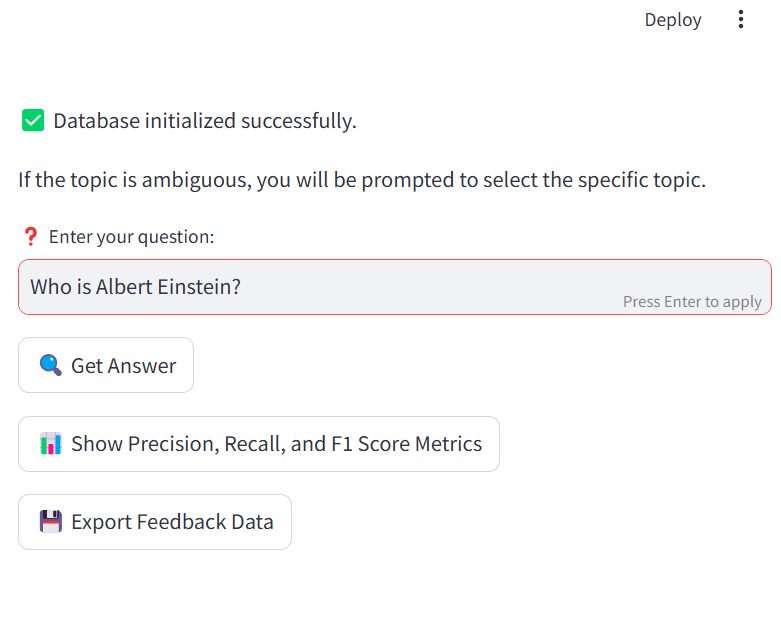


# 4. How to Ask a Question

Asking a question in the Wikipedia Question Answering System is simple and intuitive. Follow these steps to get an answer from Wikipedia:

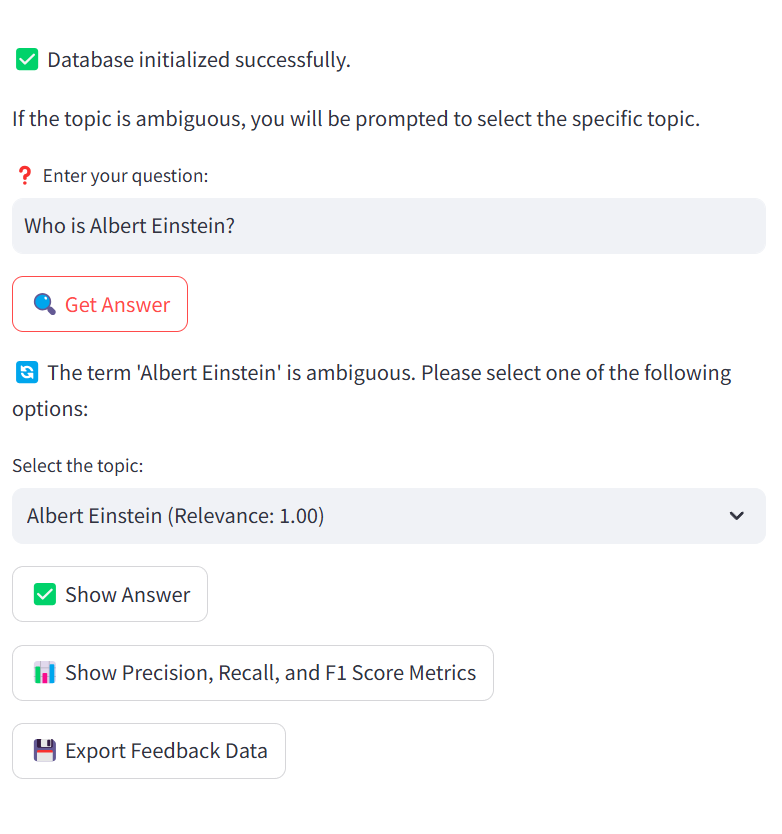
**Enter Your Query:**

In the input box provided on the main interface, type your question. It can be a simple question like "Who is Albert Einstein?" or a more complex one such as "What is the theory of relativity?"



**Processing the Query:**

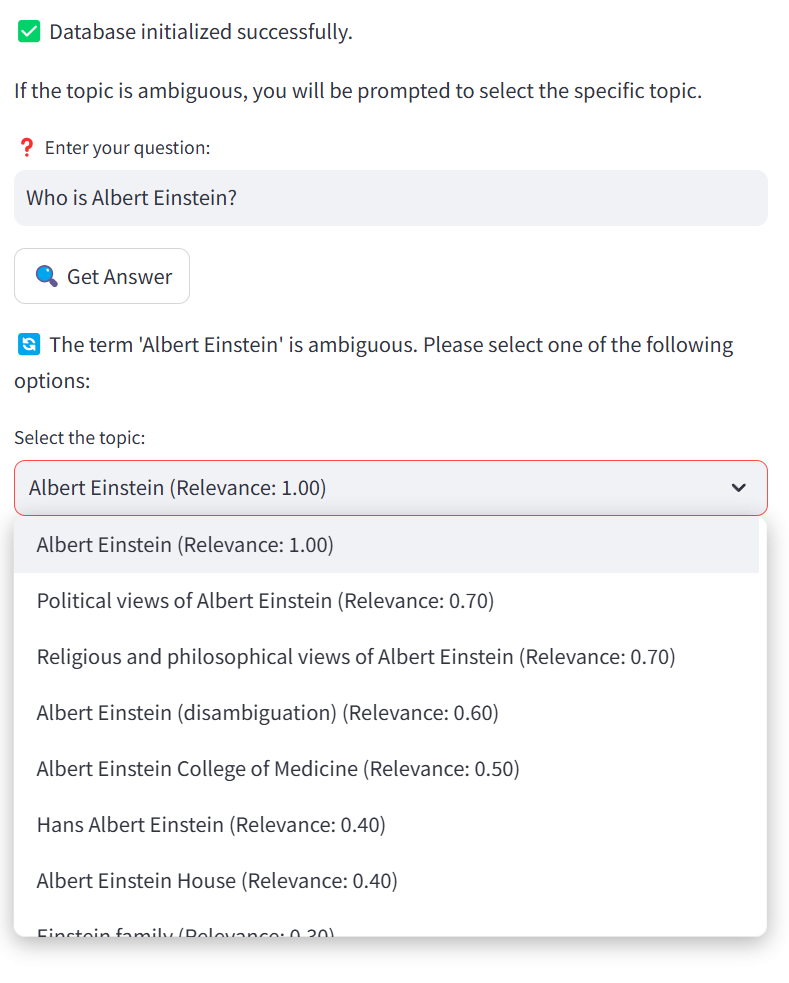
Once you click the "Get Answer" button, the system processes your query by extracting the main concept. This concept is then used to search for relevant information from Wikipedia.



**Handling Ambiguity:**

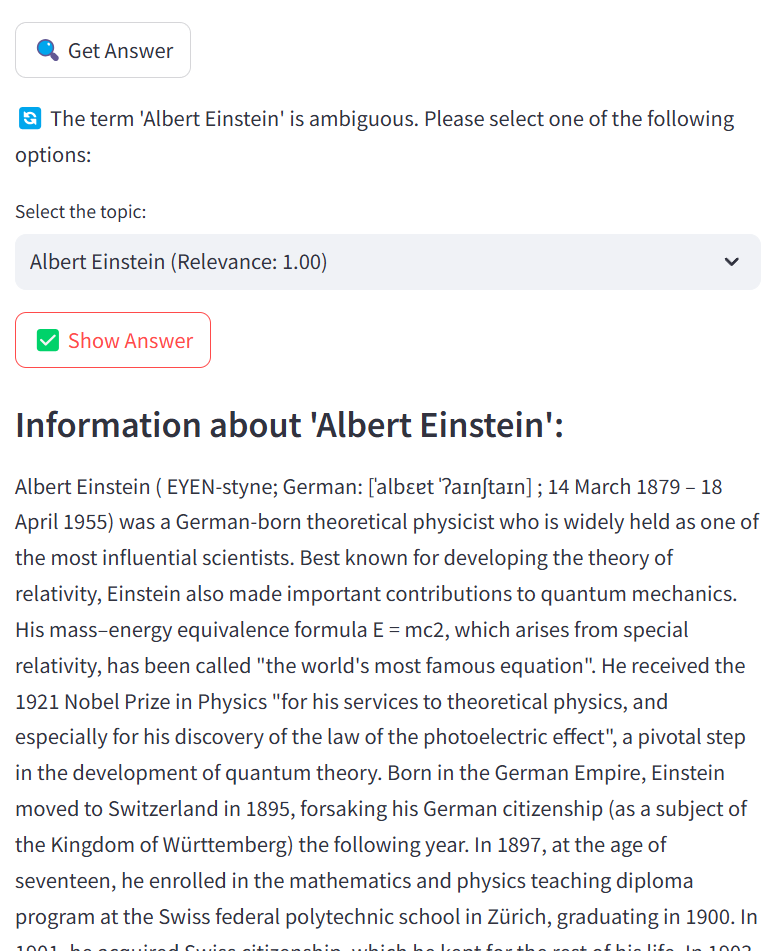
If your query could refer to multiple topics (for example, "Java" could refer to the programming language, the coffee, or the Indonesian island), the system will display a list of possible interpretations.

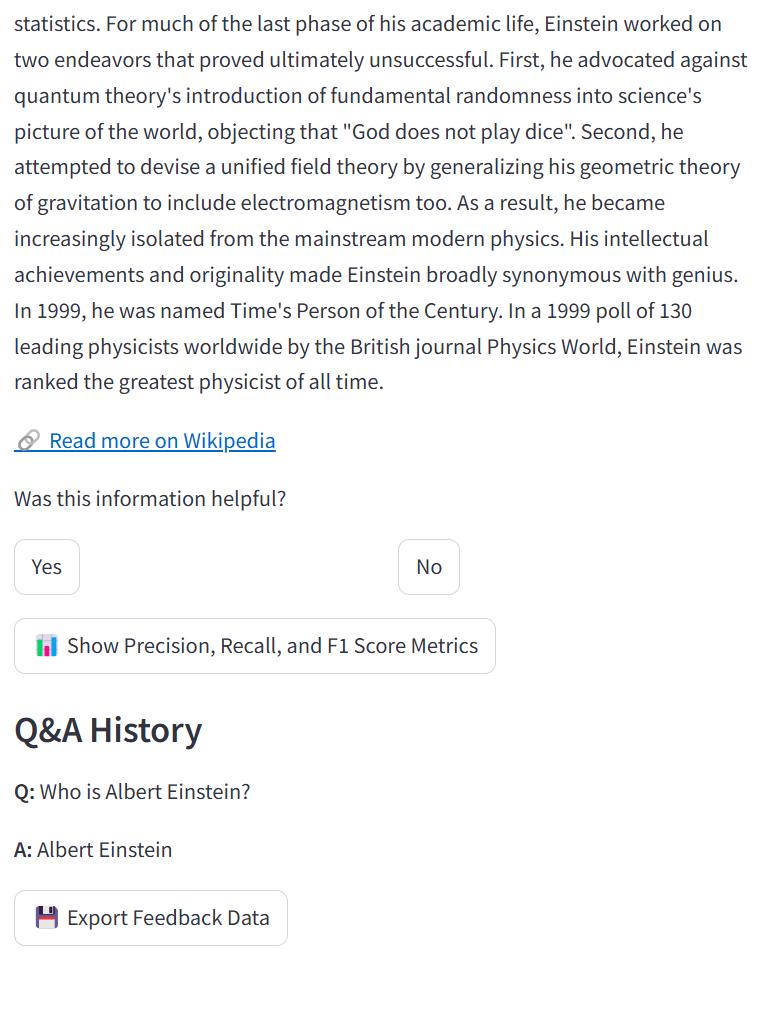
You will be asked to choose the correct option from a dropdown list or selection panel.



**Viewing the Answer:**

After selecting the relevant topic (if needed), the system will display a summary of the corresponding Wikipedia page. You will see a brief overview of the topic along with a link to the full Wikipedia article for further reading.





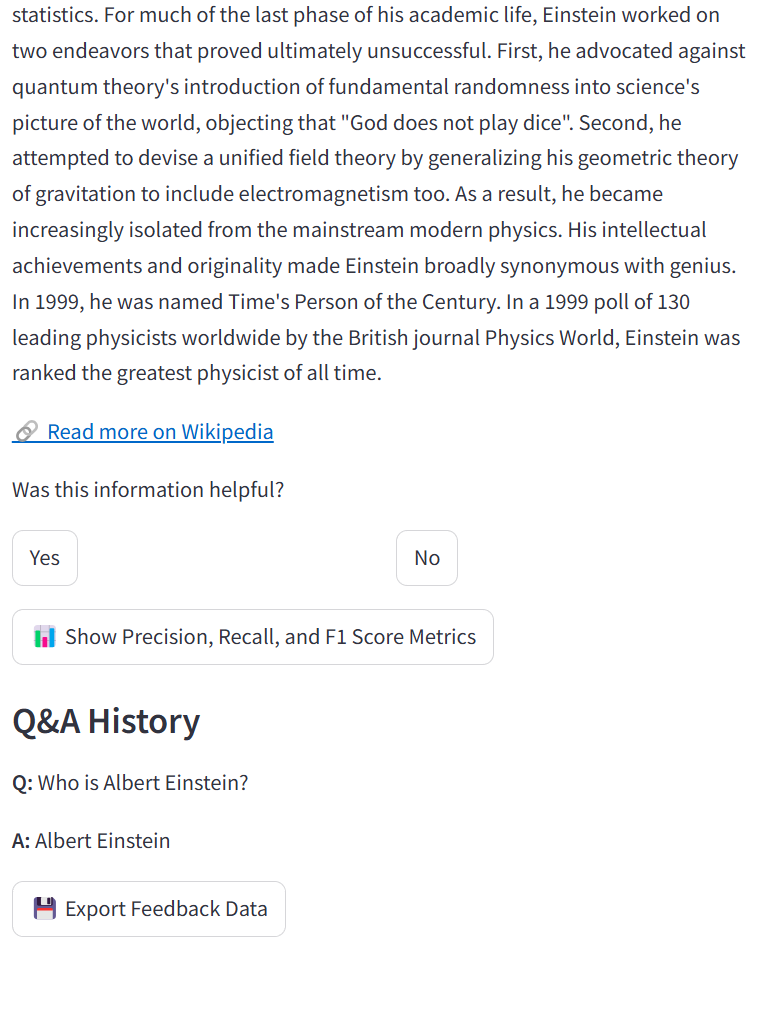
By following these simple steps, you can easily ask questions and get quick, reliable information directly from Wikipedia.

# 5. Providing Feedback

After receiving an answer, the system allows you to provide feedback on whether the information was helpful. This feedback helps the system improve over time by adjusting its responses based on user input. Here’s how you can provide feedback:

**Feedback Prompt:**

After the system displays the answer, you will see a prompt asking whether the information provided was helpful. This will appear directly below the summary.



**The options typically are:**

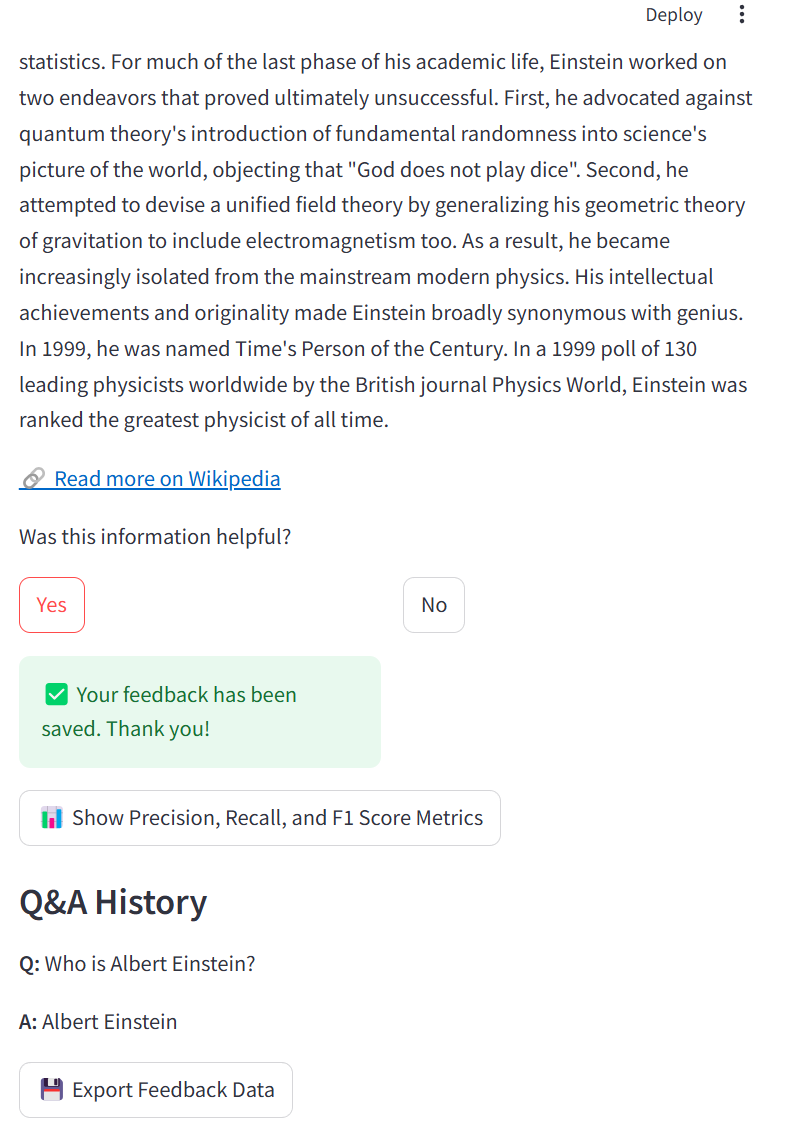
**Yes**: If the information was accurate and helpful.

**No**: If the information was not relevant or didn’t answer your question.

**Submitting Feedback:**

Simply click on either the Yes or No button to submit your feedback.

Once submitted, you will see a confirmation message indicating that your feedback has been recorded.



**Why Feedback Matters:**

Your feedback plays a crucial role in helping the system improve its performance. The more feedback the system receives, the better it can adjust its responses to provide more accurate answers in the future.

Feedback data is stored in the system’s database and used to calculate key performance metrics like Precision, Recall, and F1 Score, which reflect the system's accuracy and effectiveness.

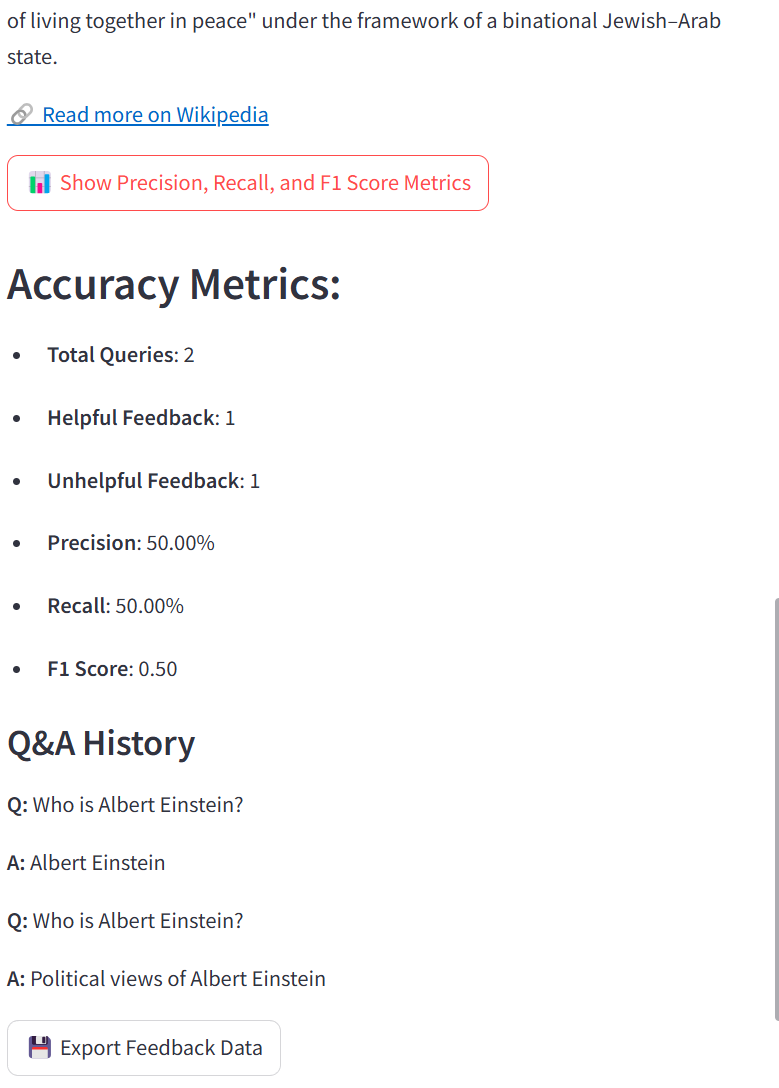
# 6. Tracking System Performance

The Wikipedia Question Answering System includes a feature that tracks its performance based on user feedback. This feature allows you to see how well the system is performing in terms of accuracy and relevance. Performance is measured using three key metrics: Precision, Recall, and F1 Score.

Here’s how you can view and interpret the system’s performance:

**Viewing Performance Metrics:**

On the interface, you’ll find a button labeled "Show Precision, Recall, and F1 Score Metrics". Click this button to display the system’s performance data, which is calculated based on the feedback provided by users.



**Understanding Precision:**

Precision measures the proportion of correct (helpful) answers out of all the answers provided by the system. A higher Precision indicates that the system is accurately identifying the correct answer for most queries where feedback was submitted.

Formula: Precision = (Helpful Answers) / (Total Answers with Feedback)

**Understanding Recall:**

Recall measures how well the system identifies the correct answer relative to all user queries. A higher Recall means the system is effective at covering a wide range of queries accurately.

Formula: Recall = (Helpful Answers) / (Total Queries)

**Understanding F1 Score:**

F1 Score is the harmonic mean of Precision and Recall, providing a balanced view of the system’s overall performance. It’s particularly useful when both Precision and Recall are equally important, offering a single measure of performance.

Formula: F1 Score = 2 \* (Precision \* Recall) / (Precision + Recall)

**Interpreting the Results:**

If the Precision is high but Recall is low, this means the system is accurate when it provides answers but may miss the correct answer in some queries.

If the Recall is high but Precision is low, the system is good at covering a wide range of queries but may not always provide the most accurate results.

A high F1 Score indicates a well-balanced system that provides accurate answers consistently across a variety of queries.

**Continuous Improvement:**

The metrics are recalculated every time feedback is provided, ensuring that the system is always adapting and improving based on real user interactions.

Regularly checking the performance metrics allows you to track how the system is evolving over time and identify areas for further optimization.

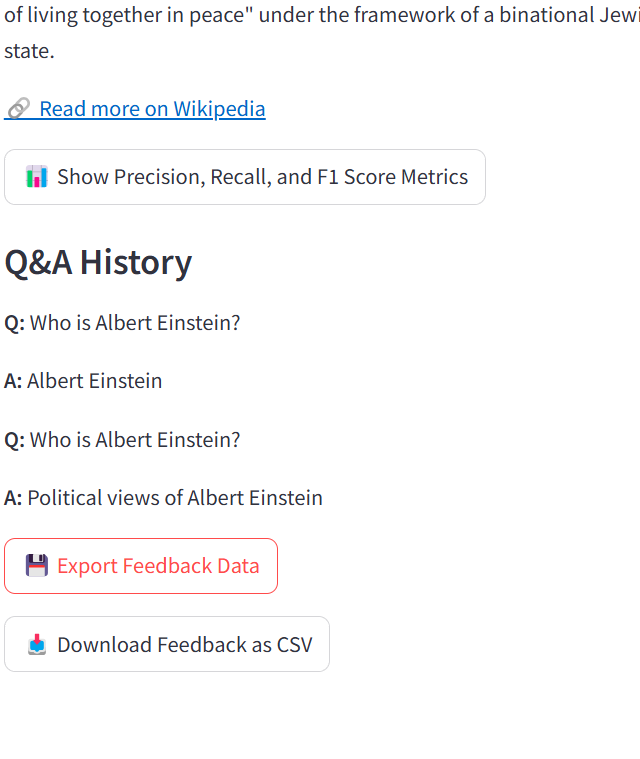
By viewing these performance metrics, you can get a clear understanding of how well the system is meeting user needs, and how it improves based on your feedback.

# 7. Exporting Feedback Data

The Wikipedia Question Answering System allows you to export all stored user feedback data for external analysis. This feature is useful if you want to track the system’s performance over time, especially if multiple users or long-term feedback collection is involved. Here's how you can export the feedback data:

**Export Button:**

On the main interface, you’ll find a button labeled "Export Feedback Data". Clicking this button allows you to download all the feedback data stored in the system in a CSV format.



**What Data is Exported:**

The exported file will include key details for each query submitted to the system:

**Query ID**: A unique identifier for each query.

**Question**: The question asked by the user.

**Answer**: The answer selected by the user from the provided options (in cases where the system offered multiple disambiguation options).

**User Feedback**: Whether the user found the selected answer helpful (Yes or No).

**Feedback Submitted**: A flag indicating whether feedback was provided.

**Timestamp**: The date and time when the query was submitted.

Note: It is important to emphasize that the "Answer" field reflects the specific option chosen by the user, particularly when multiple disambiguation options were presented. This ensures that the feedback corresponds to the user's choice rather than an automatically generated response.

**Downloading the CSV File:**

Once you click the "Export Feedback Data" button, the system will generate a CSV file containing all feedback data and prompt you to download it.

The file can be saved on your computer for further analysis, such as performance tracking, identifying trends in user queries, or even improving the system based on detailed feedback.

**Using the Exported Data:**

You can open the CSV file in any spreadsheet application (e.g., Excel, Google Sheets) or import it into data analysis tools like Python’s pandas library or R for more advanced analysis.

This feature is particularly useful for administrators or developers who want to dive deeper into the feedback patterns and make improvements to the system based on long-term feedback.

# 8. Troubleshooting

In this section, we address common issues users might encounter while using the Wikipedia Question Answering System and provide solutions.

**No Results Returned:**

Problem: After submitting a query, the system doesn't return any relevant Wikipedia pages or results.

Solution: This might happen if the query is too vague or the topic isn't well-covered on Wikipedia. Try refining your query by:

* Using more specific terms.
* Adding additional context to the question (e.g., “Albert Einstein biography” instead of just “Einstein”).
* Checking for spelling mistakes or uncommon terms that might not match Wikipedia’s content.

**Disambiguation Errors:**

Problem: The system presents a disambiguation error or returns multiple irrelevant options.

Solution: This occurs when the query is too broad and can refer to multiple unrelated topics. For example, “Python” could refer to the programming language or the snake. To resolve this:

* Select the correct option from the disambiguation list that best matches your intended topic.
* If none of the options are relevant, refine your query to be more specific (e.g., “Python programming language” instead of just “Python”).

**Slow Response or Timeout:**

Problem: The system takes too long to return a result or times out during a query.

Solution: Slow response time can be caused by network issues or Wikipedia API rate limiting. In such cases:

* Try refreshing the page and submitting the query again.
* Ensure you have a stable internet connection.
* If the issue persists, wait a few minutes and retry the query.

**Answer Not Relevant to the Question:**

Problem: The returned answer is not relevant to the query.

Solution: This can happen if the query is too ambiguous, or if the extracted concept doesn't match the intended topic. To improve relevance:

* Provide more detailed or specific questions.
* Use the disambiguation list to refine your search.
* Submit feedback so the system can learn and improve over time.