Email - hr.xygenai.com
Address - Northampton , United Kingdom,
Address - Colombo, Sril Lanka
Contact Us - 94 74 236 5804

XYGen.ai - ML/AI Assignment

Introduction

You can use Python and Flask

Duration - 2 Weeks

Task Overview Task - Knowledge-Based Search Retrieval System Development

Your task is to develop an AI-based Knowledge-Based Search Retrieval System for our company using RAG (Retrieval-Augmented Generation) architecture. This system should integrate Large Language Models (LLMs) with vector databases to provide accurate, context-aware responses to complex any domain queries.

Technologies to Use

- **Frontend: HTML/CSS/JavaScript**
- **Backend: Python & Flask**
- **Database: MongoDB or MSSQL**
- **AI/ML Tools: Large Language Models (LLMs), Vector Databases**

Task Description

Develop the System

1 - Document Upload

- **Create a section where users can securely upload various types of legal documents.**
- **The system should support multiple document categories, including.**

1. Name of the Report: (e.g., abc.pdf)
2. Type of Report: Add a drop-down menu to select the type of report (e.g., Insurance, Medical, Legal Document).
3. Additional Notes: Allow users to add any notes about the document.
4. Pages of Document: Automatically capture the number of pages in the document during the upload process.

- **Store the following information in MongoDB or MSSQL databases**

1. Name of the Report
2. Type of Report
3. Additional Notes
4. Pages of Document
5. PDF Location: Store the location of the PDFs within the local system

2 - Search Functionality

- **Implement a search section where users can search for specific words or sentences within all uploaded documents.**
- **The search results should display**

  1. The number of occurrences of the search term.
  2. The page numbers where the term is found.
  3. The count of occurrences on each page.

Architecture Requirements

- Use RAG (Retrieval-Augmented Generation) to enhance the accuracy of search results.
- Integrate Large Language Models (LLMs) to understand and process the queries.
- Implement a vector database to store document embeddings for efficient retrieval.

Database Schema

You are required to share the schema of your MongoDB or MSSQL database used to store the document metadata and PDF locations.

Deliverables

- Code Submission - Upload your complete code to a GitHub repository and ensure it is public.
- Documentation - Provide clear and detailed documentation within the repository, explaining how your system works, how to run it, and any dependencies required.
- Demonstration - Take a screenshot or record a short video demonstrating the functionality of your system.
- Submission - Upload the screenshot or video along with your GitHub repository link to the provided Google Form.

Use Case

- Scenario - Users need to upload 10 PDF files to the system.
- Functionality - The system should allow users to search for multiple queries. For each query, the system should retrieve and display only the relevant PDFs and the specific content within those PDFs that match the query.
- Search Results - The system should display the number of occurrences, page numbers, and term counts within a page for the search term.

Evaluation Criteria

- Accuracy and Relevance: How well the system retrieves and displays relevant PDFs and content.
- Efficiency: The performance of the system in handling multiple queries and large documents.
- Code Quality: Adherence to best practices in coding, clear documentation, and maintainability.
- UI/UX Design: The usability and design of the web interface.
- Innovation: Creative and efficient use of RAG architecture, LLMs, vector databases, and database management.

Deadline

- **Please complete the task and submit your work via the Google Form by 27 th August by 5.00 PM**

Submission Link

- Add your Link in to Google Form - Link

Feel free to ask any questions if you need further clarification or assistance. Happy coding!

Contact Us For More queries - 94 74 236 5804

Admin - XYGen . ai

13/08/2024