Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Title of the Project

DocuCore : Text & Code Summariser In Cloud

Abstract of the project

This project introduces a cloud-based system designed for document summarization and code analysis, utilizing advanced machine learning technologies, particularly Large Language Models (LLMs). The system aims to deliver an efficient, scalable solution for handling extensive volumes of text and complex codebases, providing users with actionable insights in real-time.

The Text Summarization Module automatically extracts key insights from lengthy documents, distilling the most critical information into concise, coherent summaries. This feature saves users significant time and effort, ensuring they quickly grasp the essential points of any document.

The Code-Based Analysis Module streamlines the code review process by identifying code complexity, performance issues, and potential bugs. Supporting multiple programming languages, this module automates the assessment of code quality and provides targeted recommendations for improvement, enhancing the software development lifecycle with increased efficiency and accuracy.

By deploying the system on the cloud, it ensures scalability, real-time processing, and efficient collaboration. The integration of cloud technology allows the system to dynamically scale resources based on user demands, making it a flexible and powerful tool for modern data and software analysis needs.

Ultimately, this project offers a powerful, user-friendly solution for modern text and code analysis, combining the benefits of AI-driven summarization with automated code evaluation, all within a flexible and accessible cloud environment.

Keywords

Generic Keywords

Machine Learning, NLP, Cloud Infrastructure, Code Analysis.

Specific Technology Keywords

Python, Azure, Databricks.

Project Type keywords

Analysis, Design, Implementation, Testing, Cloud Deployment.

Functional components of the project

Users of the system:

- Admin
- Developer
- End User

Functional Requirements of the System:

- **Text Summarization Module:**

Utilizes LLMs for generating summaries from large text datasets.
Users can input long documents and receive concise, contextually accurate summaries.

- **Cloud Infrastructure Module:**

Deploys the summarization and code analysis system on the cloud using Azure for scalability and availability.
Provides secure and real-time access to the services.

- **Code-Based Analysis Module:**

Uses code analysis tools to perform static and dynamic code examination.
Identifies code complexity, bugs, and provides recommendations for code optimization.

- **Web-Based User Interface**

The system should provide a web-based user interface for interacting with the document summarization and code analysis tools.

Non-Functional Requirements:

- **Performance:**

The system should perform efficiently, handling multiple document and code submissions concurrently**.**

- **Scalability:**

The system should scale to accommodate an increasing number of users and data.

- **Reliability:**

The system should be highly available and reliable, with minimal downtime.

- **Security:**

The system must ensure secure access to data and prevent unauthorized access.

- **User Experience:**

The system should provide an intuitive, user-friendly interface with an emphasis on ease of use.

- **Maintainability:**

The system should be easy to maintain and update.

- **Compliance:**

The system should comply with relevant data privacy and protection regulations.

Steps to start-off the project:

The project involves several steps and considerations to ensure its functionality, security, and user-friendliness. Here's an outline of the process:

1. **Technology Familiarization:** Gain a strong understanding of the core technologies required for the project.
   - Study and explore Large Language Models (LLMs) for document summarization.
   - Familiarize yourself with code analysis tools.
   - Learn cloud services for deploying scalable applications.

2. **Team Role Allocation:** Clearly define the responsibilities of each team member.
   - Assign roles for Text Summarization, Code Analysis, and Cloud Integration.
   - Allocate individual milestones (research, development, testing) for each module.
   - Make sure all team members have a thorough understanding of their respective modules.

3. **Requirements Gathering and Specifications:** Collect and define the functional and non-functional requirements of the project.
   - Identify all user requirements, such as web-based access, document uploading, code submission, etc.
   - Define system requirements for summarization accuracy, code analysis metrics, and performance benchmarks.
   - Document these requirements and map them to the project's core modules (Summarization, Code Analysis, Cloud Integration).

4. **Cloud Setup:** Configure the cloud environment for deploying and running the system.
   - Select the appropriate cloud platform (AWS, Azure, or GCP)..
   - Set up core services such as compute instances (e.g., EC2), storage solutions (e.g., S3), and databases.
   - Configure auto-scaling and load balancing for high availability.

5. **Web Interface Design:** Create a user-friendly web-based interface for accessing the system.
   - Design the front-end for document uploads, code submissions, and displaying summarization/analysis results
   - Ensure the web interface is intuitive, visually appealing, and responsive across devices.
   - Integrate the front-end with backend APIs to allow seamless communication between modules.

6. **Initial Testing:** Test the individual modules before integration
   -Test the summarization model to ensure it generates accurate summaries from documents.
   - Run code analysis on sample codebases to verify the tool's functionality.
   -Test the cloud deployment to ensure the system is scalable and accessible.

7. **Integration of Modules:** Combine all project modules (Text Summarization, Code Analysis, Web UI, Cloud) into a fully functional system.
   - Integrate the summarization and code analysis functionalities with the web interface.
   - Ensure the backend APIs communicate effectively with the cloud services.
   - Perform integration tests to ensure all modules work together seamlessly.

8. **Final Testing and Optimization:** Conduct comprehensive system testing and optimize performance.
   - Perform load testing on the system to ensure it handles multiple requests simultaneously.
   - Optimize the summarization model for speed and accuracy.
   - Fine-tune cloud resource allocation to optimize cost and performance.
   - Ensure all security measures are fully implemented.

Requirements

Hardware requirements

Number	Description	Alternatives (If available)
1	PC with 5 GB hard-disk and 512 MB RAM	Not-Applicable

Software requirements

Number	Description	Alternatives (If available)
1	Windows 10/11	Linux
2	Python	HuggingFace Transformers
3	Azure for Cloud Infrastructure	Google Cloud, AWS
4	Visual Studio Code	PyCharm

Manpower requirements

3 students are sufficient to complete the project in 6-8 months with part-time involvement..

Milestones and Timelines

Number	Milestone Name	Milestone Description	Timeline (In Weeks)	Remarks

1	Requirements Specification	Complete system specifications, including document/text processing and code analysis requirements.	2-3	Include all functional and non-functional requirements for the system.
2	Technology Familiarization	Learn the necessary technologies (Azure, LLMs, Python) to implement the project.	4-5	Practical understanding of tools and frameworks needed for the project.
3	Cloud and Database Setup	Configure Azure infrastructure, create a database for text storage and processing	6-7	Finalize cloud platform setup and database requirements.
4	Text Summarization Model Design	Design and implement the model for summarizing text documents using LLMs.	8-10	Develop model and initial testing with small datasets.
5	Code-Based Analysis Tool Setup	Implement the code analysis tool and integrate with code repositories.	11-12	Test and run the analysis tool on sample codebases.
6	Cloud Integration and Model Deployment	Deploy the summarization model and code analysis tool in the cloud for real-time use.	13-14	Ensure successful cloud integration and deployment.
7	Testing and Validation	Run test cases on the deployed modules, validate the results, and ensure all components work smoothly.	15-16	Perform extensive tests for all functionalities.
8	Final Review	Prepare final documentation, review results, and present the complete project.	17-18	Ensure all objectives have been achieved before the final submission.

Guidelines and References

https://learn.microsoft.com/en-us/azure/?product=popular (Azure Documentation)

https://huggingface.co/ (NLP Resources)