Constitutional Archive Data Management and Search Interface

Introduction

This project aims to develop a full-stack web application for managing and querying historical constitutional data. Instead of automatically ingesting data, the application will feature an admin portal where authorized users can sign up, log in, and upload archive files. The data should be organized in hierarchical directory structures (mirroring the “Access to Memory” archive) with accompanying metadata. A public-facing search interface—reminiscent of Perplexity—will allow users to perform natural language queries and retrieve relevant content (text snippets, PDFs, multimedia).

Objectives

The team is required to use Agile methodology, incorporate CI/CD principles, and take a test-driven approach to design, develop, and operationalize a publicly available web-based application that meets the following requirements:

- Implement secure authentication for admin users.
- Develop an admin interface for uploading and managing data in a structured directory format with metadata.
- Build a public search interface that processes natural language queries and displays relevant results.
- **Design a general RESTful API** that not only supports the web search interface but can also be extended in the future to integrate other interfaces (e.g., WhatsApp chatbots).
- Deploy the application end-to-end on Microsoft Azure.

Overview of Features

- **Admin Portal:**
  - **Secure Authentication:** Login/signup functionality (using a 3rd party identity provider if desired) to restrict data management to admins.
  - **Data Upload Interface:** An interface for admins to upload files and associated metadata, organizing them into hierarchical directory structures similar to “Access to Memory.”
  - **Data Management:** Capabilities to edit, delete, and reorganize uploaded files.
- **Public Search Interface & API:**
  - **User-Friendly Search:** A responsive search interface inspired by Perplexity, featuring a central search bar and dynamic results display.

    - **Generalized API:** A RESTful API that handles natural language queries, processes and ranks the uploaded data, and returns a mix of text snippets, PDF links, and multimedia content. The API should be designed modularly to allow future integration with additional interfaces, such as WhatsApp chatbots.
- **Deployment & CI/CD:**
    - **Cloud Deployment:** Host the complete application (front end & back end) on Microsoft Azure.
    - **Automated Pipelines:** Utilize CI/CD pipelines (e.g., via Azure DevOps) for automated testing, integration, and deployment.
- **Bonus:**
    - **Enhanced Search:** Integrate basic NLP techniques for improved query understanding, multi-language support, or filtering by metadata (e.g., document type, publication date).

Detailed Requirements

Requirement	Extra Info/Limitations
User Authentication	• Implement secure login/signup for admin users, optionally leveraging a 3rd party identity provider. • Enforce role-based access control to ensure only admins can access data upload and management features.
Admin Data Management	• Develop an admin interface that allows the uploading of files. • Enable the organization of files into hierarchical directory structures mirroring the “Access to Memory” archive. • Provide options to add/edit metadata for each file.
Search Interface & API	• Create a responsive and intuitive search UI inspired by Perplexity. • Build a RESTful API that accepts natural language queries and returns relevant results from the uploaded data. • **Ensure the API is general enough to allow integration with future interfaces (e.g., WhatsApp chatbots).**
Deployment on Azure	• Deploy the complete application (front end & back end) on Microsoft Azure. • Set up a CI/CD pipeline (e.g., using Azure DevOps) for automated testing and deployment.

Requirement	Extra Info/Limitations
Bonus	• Enhance the search functionality using NLP for query refinement or multi-language support. • Enable advanced filtering options based on metadata (e.g., document type, date, etc.).