ARRS project requirements

Project Name: Automated Resume Review System (ARRS)

Project Overview

The ARRS is designed to streamline the process of reviewing candidate resumes by automatically processing new submissions stored in a designated Google Drive folder. Each morning, the system identifies new resumes, evaluates them using a predefined set of questions through OpenAI's API, and records the responses in a CSV file for further analysis.

Functional Requirements

1. Google Drive Integration

- FR1.1: The system must connect to Google Drive using appropriate API credentials.
- FR1.2: The system must access a specified folder within Google Drive to check for new resume submissions.
- FR1.3: The system must differentiate between already processed and new resumes based on a timestamp or flagging mechanism.

2. Resume Processing

- FR2.1: The system must support common resume formats, including PDF and Word documents.
- FR2.2: The system must extract text from the resumes for analysis. This may require integration with a document parsing API or library capable of handling multiple file formats.

3. OpenAI API Integration

- FR3.1: The system must securely store and use OpenAI API credentials for querying.
- FR3.2: The system must format the extracted resume text and predefined questions into a suitable input for the OpenAI API.
- FR3.3: The system must handle API responses, extracting relevant answers or insights for each predefined question.

4. Data Recording and Reporting

- FR4.1: The system must generate a new CSV file each day for that day's resume reviews or append to an existing monthly file, based on configuration.
- FR4.2: For each resume, the system must record relevant details including the filename, submission date, and answers to the predefined questions in the CSV file.
- FR4.3: The system must ensure data integrity and prevent duplication in the CSV file.

Non-Functional Requirements

1. Performance

- NFR1.1: The system should process each resume and complete the review within a specified time frame, e.g., less than 5 minutes per resume.
- NFR1.2: The system must be capable of handling a surge in resume submissions, scalable to process up to 100 resumes per day.

2. Security

- NFR2.1: The system must securely handle API credentials and sensitive personal information from resumes, complying with relevant data protection regulations (e.g., GDPR, CCPA).
- NFR2.2: Access to the Google Drive folder, OpenAI API, and CSV files must be restricted to authorized personnel.

3. Reliability

- NFR3.1: The system should have an uptime of 99.9%, with error handling mechanisms in place to alert administrators of failures or issues in the process.
- NFR3.2: The system should perform automatic retries for transient errors in file processing or API requests.

System Architecture Requirements

- SAR1: The system should be developed using a modular architecture, allowing for easy updates or modifications to individual components, such as changing the set of questions without affecting the integration with Google Drive or OpenAI API.
- SAR2: The system should be deployable on cloud infrastructure to leverage scalability and performance benefits.

Compliance and Data Handling

- CDH1: The system must comply with all applicable data protection laws and best practices for handling personally identifiable information (PII).
- CDH2: The system must include documentation on data flow, storage, and access controls to support compliance audits.

Project Deliverables

- PD1: Source code for the Automated Resume Review System.
- PD2: Documentation covering system setup, configuration, and operation instructions.
- PD3: Compliance and data handling documentation.
- PD4: Test cases and results demonstrating system functionality and performance.

System Architecture Overview

Google Drive Integration Module: Responsible for authenticating with Google Drive, accessing the specified folder, and identifying new resume submissions based on timestamps or flags.
Document Processing Module: Handles the extraction of text from the resumes. This module is crucial for converting PDFs, Word documents, and other supported file formats into plain text for analysis.
OpenAI API Integration Module: Manages the communication with OpenAI's API, including sending requests and receiving responses. This module formats the extracted resume text along with predefined questions for analysis and interprets the API's responses.
Data Storage and Reporting Module: Organizes the information received from the OpenAI API, compiles it into a structured format, and records it in a CSV file. This module also handles data integrity and prevents duplications.
Scheduler and Automation Engine: Coordinates the execution of tasks, ensuring that the system runs at specified times (e.g., each morning) and efficiently processes batches of resumes.
Error Handling and Notifications Module: Detects, logs, and notifies administrators of any errors or issues in the workflow, ensuring timely intervention for any problems.

System Workflow

- Scheduled Trigger: Each morning, the Scheduler and Automation Engine triggers the Google Drive Integration Module to check for new resumes in the specified folder.
- Resume Identification: The Google Drive Integration Module authenticates with Google Drive, accesses the folder, and identifies new resume submissions based on the last processed timestamp or a flagging mechanism.
- Text Extraction: For each new resume, the Document Processing Module extracts the text from the file, supporting various formats like PDF and Word documents.
- Analysis Request: The extracted text, along with a set of predefined questions, is formatted into a request and sent to the OpenAI API Integration Module.
- OpenAI API Query: The OpenAI API Integration Module securely sends the request to OpenAI's API and waits for the analysis to complete.
- Response Processing: Once the analysis is received from OpenAI, the module processes and formats the responses.
- Data Recording: The Data Storage and Reporting Module takes the processed responses and records them in the CSV file along with the resume's metadata (e.g., filename, submission date). It ensures each entry is unique to avoid duplication.
- Error Handling: Throughout the workflow, the Error Handling and Notifications Module monitors for any issues. In case of errors (e.g., file format not supported, API downtime), it logs the issue and sends notifications to the administrators for intervention.
- Report Generation: After all new resumes have been processed for the day, the system may generate a summary report for administrators, highlighting the number of resumes processed, any issues encountered, and any action items required.

Function calls

1. Google Drive Integration Module

This module handles interactions with Google Drive to access and manage resume files stored in a specified folder.

- Authenticate(): Authenticate with Google Drive using OAuth 2.0 to access the API.
- ListFiles(folderId: string): List all files within the specified folder. Optionally, include parameters to filter by upload date or a custom "processed" flag.

- DownloadFile(fileId: string): Download a file based on its unique ID from Google Drive for processing.
- MarkFileAsProcessed(fileId: string): Update a custom property or move the file to a different folder to indicate it has been processed.

2. Document Processing Module

Responsible for converting resumes from various formats into a uniform text format for analysis.

- ExtractText(filePath: string): Extracts plain text from the resume file, supporting multiple formats (e.g., PDF, DOCX).
- NormalizeText(text: string): Apply normalization techniques (e.g., removing special characters, standardizing date formats) to prepare text for analysis.

3. OpenAI API Integration Module

Manages the sending of text to OpenAI's API and interprets the responses.

- InitializeAPI(apiKey: string): Initialize the OpenAI API client with the necessary credentials.
- SendAnalysisRequest(text: string, questions: string[]): Format and send a request to OpenAI's API with the resume text and predefined questions.
- ParseAPIResponse(response: object): Extract and format the relevant information from the API's response for storage.

4. Data Storage and Reporting Module

Handles the storage of analysis results and metadata about processed resumes.

- InitializeStorage(storagePath: string): Initialize the connection to the storage solution (e.g., local filesystem, cloud-based database) where the CSV files will be saved.
- WriteToCSV(data: object): Write the processed data into a CSV file, including resume metadata and answers to predefined questions. Ensure handling of data to avoid duplicates.
- GenerateReport(startDate: string, endDate: string): Generate a summary report for a given period, highlighting the volume of processed resumes, insights gained, and any anomalies detected.

5. Scheduler and Automation Engine

Coordinates the execution of tasks according to a predefined schedule.

- ScheduleJob(job: function, schedule: string): Schedule a job (e.g., checking for new resumes) to run at specific times, following cron syntax or similar scheduling formats.
- ExecuteJob(jobId: string): Manually trigger a job outside of its scheduled times, useful for testing or manual overrides.

6. Error Handling and Notifications Module

Provides a mechanism for logging errors and notifying administrators of system issues.

- LogError(error: object): Log an error message with details about the issue encountered.
- SendNotification(message: string, recipients: string[]): Send a notification (e.g., email, SMS) to a list of recipients about system errors or important information.