System Design and Evaluation Document (Current Scope)

This document provides a holistic view of the infrastructure supporting the application and clarifies the usage of different components within the project.

Purpose: The AlmaO1A FastAPI application is engineered to leverage OpenAI's GPT models for the analysis of CVs (Curriculum Vitae), transforming uploaded PDFs into insightful, actionable data. The application's design focuses on scalability, security, and ease of use, accommodating a broad range of deployment environments and user scenarios.

Design Choices:

- **FastAPI Framework:** Opted for its high performance and easy asynchronous support, ideal for IO-bound and high-concurrency operations.
- **OpenAI GPT Models:** Utilized for their leading-edge natural language processing, enabling sophisticated text analysis using GPT 4 and GPT-3.5 turbo.
- **PDFPlumber for PDF Processing:** Chosen for its reliable text extraction capabilities, critical for accurate data analysis.
- **Jinja2 Templating:** Employed for dynamic HTML response rendering, facilitating easy updates and management of the web interface.
- **Environment Variables:** Implemented for secure management of sensitive data such as API keys, enhancing security by avoiding hard-coded credentials.
- **Prompt Engineering:** Leveraged various prompting techniques like:
  - ✓ **Chain of Thought (CoT) Prompting**: mimic a step-by-step reasoning process
  - ✓ **Contextual Embedding:** setting the stage for the kind of analysis required
  - ✓ **Explicit Instruction**: instructs the model on the format and depth of the analysis required
  - ✓ **Simulated Role-Playing**: sets up a role for the language model to adopt—that of an immigration expert or consultant—which is a form of role-playing.
  - ✓ **Iterative Refinement:** An iterative approach to evaluating the CV, where each criterion is assessed independently on each criterion before a final judgment is made.
- **Deployment:**
  Heroku: Chosen for hosting live application, to leverage Heroku's dynamic scaling and operational flexibility, and operational calculations like **throughput, latency, and memory usage**.

Output: (For a sample resume attached on repo)

Structure of the Analysis Results: From samples Resume <Attached in the Git Repo>

- **Criterion-Based Evaluation:** Each of the eight criteria necessary for the O-1A visa has been individually assessed. The results are presented in a list format, with each criterion followed by a rating (Low, Medium, High) and a brief explanation justifying the rating:

- **Awards:** The candidate's CV did not mention any nationally or internationally recognized awards, resulting in a "Low" rating.
- **Membership:** No memberships requiring outstanding achievements were noted, thus a "Low" rating.
- **Press:** The absence of significant publications also leads to a "Low" rating.
- **Judging:** Lack of evidence regarding participation as a judge in the field results in a "Low" rating.
- **Original Contribution:** This category received a "High" rating, indicating significant contributions such as innovative algorithms or models noted in the CV.
- **Scholarly Articles:** Despite some publications, the lack of articles in professional journals resulted in a "Low" rating.
- **Critical Employment:** The candidate has held important roles in reputable organizations, meriting a "Medium" rating.
- **High Remuneration:** With insufficient information on salary or remuneration compared to peers, this criterion also received a "Low" rating.

Overall Evaluation:

- The summary compiles the ratings across all criteria, providing an overall evaluation of the candidate's qualifications for an O-1A visa:
  - **High:** The candidate meets high standards in original contributions.
  - **Medium:** Significant employment roles are acknowledged.
  - **Low:** Most criteria including awards, membership, press, judging, scholarly articles, and high remuneration did not meet the high standards required.

Architecture:

- The application is structured around a microservices architecture, delineating the web interface, API layer, and text processing services. This separation enhances maintainability and supports easier scalability.

Scalability:

- **Horizontal Scaling:** Achieved by deploying additional instances behind a load balancer to manage increased traffic and computational demand given large users access the website.
- **Asynchronous Processing:** FastAPI's asynchronous handling allows efficient management of numerous concurrent requests, crucial for performance under load.

Deployment:

- **Heroku**: Chosen for hosting the live applicationto leverage Heroku's dynamic scaling and operational flexibility and insightful operational calculations.
- **Procfile Integration**: Contains the command `web: uvicorn main:app --host 0.0.0.0 --port $PORT` to instruct Heroku on how to start the application.

Security and API Integration:

- **Heroku and OpenAI API Keys**: Securely stored as secrets within GitHub Actions and referenced as environment variables within the deployment process. This setup ensures that sensitive information such as API keys remains confidential and protected.

GitHub Integration:

- **GitHub Actions:** Automates workflows for continuous integration and deployment, ensuring that the application is always in a deployable state following any changes to the codebase.
- **GitHub Pages:** Used to host the project's static front-end, providing users with information about the application and access to the web interface.

Template Folder Usage:

- Contains **base.html** for the core HTML structure**,** **upload_form.html** for the CV upload interface, and **results.html** for displaying analysis results. These templates ensure a consistent look and feel across the application and facilitate easy updates to the UI.

Evaluation of Output:

- **Accuracy Measurement:** Text extraction and analysis accuracy are validated against a benchmark set of annotated CVs.
- **User Feedback:** Further, can be solicited to refine and enhance the model's accuracy and the overall user experience, also compare the accuracy with the currently approved O1-A applicant CVs to understand the analysis response the LLM provided.
- **Logging and Monitoring:** Integral for ongoing performance assessment, helping to identify and rectify issues swiftly.

Conclusion: The AlmaO1A FastAPI application stands out as a robust solution for CV analysis, designed with a focus on scalability, security, and user engagement. Continuous refinement based on user feedback and system monitoring ensures the application remains effective and relevant.

Live Application: Experience the features and functionalities of the AlmaO1A application on Heroku at this live link.

Further, the evaluation of the response can be compared with leveraging other Language Models like Claude, Mistral, and Gemini LLMs to understand the performance by careful evaluation against different metrics to get accurate and precise response.