**Technology Choices**

**I. Core Technologies**

* **Programming Language:** Python 3.10+ (or a specific version you choose for your environment)
* **Web Framework:** Flask (version 2.3.0 is the version we will be using, for repeatability)
* **AI/ML Library:** google-generativeai (version 0.3.2 is the version we will be using, for repeatability)
* **Data Analysis Library:** Pandas (version 2.1.4 is the version we will be using, for repeatability)
* **Data Generation Library:** Faker (version 20.2.2 is the version we will be using, for repeatability)
* **Frontend Technologies:** HTML, CSS (you may use a library like Tailwind CSS or similar), JavaScript
* **Version Control:** Git
* **Project Management:** ClickUp
* **Primary Backup:** OneDrive.
* **Secondary Backup:** Google Drive
* **Logging:** Python's logging module
* **API Secret Storage:** Google Cloud Secret Manager or OS Environment Variables

**II. AI Tools**

* **Google Gemini API:**
  + **Purpose:** Primary AI model for generating realistic client scenarios and potentially for future interactive client functionality.
    - We will be using the model gemini-1.5-pro-latest.
  + **Justification:** Chosen for its generative capabilities and Google's ecosystem integration.
  + **Version:** We will be using version 0.3.2 through the google-generativeai library.
  + **Authentication:** Requires an API key, which you can obtain from the Google Cloud Console. Workload Identity Federation is recommended for production, while OS environment variables can be used during development.
* **Cursor:**
  + **Purpose:** AI-powered code editor for development.
  + **Justification:** Used for its Python coding support, Git integration, and AI code suggestion and completion tools.
  + **Usage:** Code editor and IDE used to edit code in the Python project.
* **AI Studio (Vertex AI):**
  + **Purpose:** Development environment for prototyping and building AI applications.
  + **Justification:** Used for the Jupyter Notebook environment to prototype the DACS application, and perform the initial research and testing steps.
  + **Usage:** Using the provided Vertex AI environment with a custom library called gemini\_env.

**III. Other Technologies**

* **Python's logging module**
  + **Purpose:** Used to create logs that can be read to track errors and events that may be occurring in the code.
  + **Justification:** To be able to follow the process in your application, and quickly pinpoint issues and bugs.
  + **Usage:** To log events that can then be analyzed for debugging, and monitoring of your project.

**IV. Specific Choices and Reasoning:**

* **Python and Flask:** The choice for using Python was made due to its versatility, ease of use, and its vast collection of libraries for data science and machine learning. Flask was used for being light weight, and easy to setup.
* **Pandas:** Pandas was selected for being a robust and battle-tested library for dealing with data manipulation and data frames.
* **Faker:** Faker was chosen for being simple and robust for random generation of different data types.
* **Git:** We are using git for its reliability, and its capabilities for maintaining a clear history of development, code branching, and collaborative development.
* **ClickUp:** We are using ClickUp as our preferred project management tool, to coordinate, organize, and track our project.
* **Cloud Storage:** We are using Microsoft OneDrive as our primary source of file storage, and Google Drive for backup.
* **OS Environment Variables:** For simplicity we are using OS Environment Variables during the development phase, but it is encouraged that we use Google Cloud Secret manager for production environments.

**V. Future Considerations:**

* We should revisit these technology choices throughout the project lifecycle and evaluate alternative tools that we may be able to add to improve the DACS project.
* We will need to add a section about Javascript frameworks if we require them in future versions.
* We should keep the versions of all packages up to date in order to reduce the potential for conflict.