**Production Requirements Document: AI-Powered Data Analysis Practice Platform**

**1. Introduction/Overview**

This document outlines the production requirements for **Data Analyst Client Simulator (DACS)**, an AI-powered web application designed to simulate realistic client interactions for data analysts. The platform aims to provide a dynamic and challenging environment for users to practice their skills, build portfolios, and improve communication abilities. The core functionality involves an AI generating client scenarios (business tasks) and corresponding datasets, initially with the potential for future interactive client communication and feedback.

**2. Goals**

* Provide aspiring data analysts with a realistic environment to practice their skills on diverse, real-world-inspired problems.
* Generate varied and challenging data analysis scenarios beyond typical market/sales examples.
* Create corresponding datasets that are relevant to the scenarios and contain realistic data complexities.
* Offer a simple and intuitive platform for accessing these scenarios and datasets.
* (Future) Enable users to interact with an AI client to ask clarifying questions and adapt to new information.
* (Future) Provide feedback on users' data analysis approaches and results, simulating peer review.
* (Future) Serve as a source of project ideas for users to build their portfolios.

**3. Target Audience**

* Aspiring data analysts seeking to refine their skills and gain practical experience.
* Individuals looking to build a portfolio of data analysis projects.
* Students in data science or related fields.

**4. Functional Requirements**

**4.1 Scenario Generation:**

* **Requirement:** The system shall generate diverse and realistic client interaction scenarios (referred to as "business tasks").
  + **Details:** Scenarios should span various domains beyond typical business examples, including but not limited to botany, astronomy, sports, and historical events.
  + **Variety:** The system should maximize the variety of scenarios generated.
  + **Complexity (Future):** Future versions will allow users to select the complexity level of scenarios.
  + **Tagging (Future):** Future versions will allow scenarios to be tagged for thematic categorization and filtering.
  + Domain Selection (Future): The user interface will include elements to allow users to choose the area or domain of the data world for scenario generation (e.g., finance, botany, marketing).
* **Requirement:** Each generated scenario shall include:
  + **Client Description:** A brief background of the client or organization.
  + **Business Task:** A clear statement of the problem the data analyst needs to solve.
  + **Data Provided:** A description of the dataset being provided (types of data, potential sources, size).
* **Requirement:** The system shall ensure the generated scenario is coherent and logically sound.

**4.2 Dataset Generation:**

* **Requirement**: The system shall generate a CSV dataset that corresponds directly to the generated scenario.
  + Correspondence: The dataset's structure (columns and data types) must align with the "Data Provided" description in the scenario.
  + Completeness: The dataset may contain more information than explicitly mentioned in the scenario but should not lack any crucial information described.
* **Requirement**: The generated datasets shall be suitable for training effective machine learning models.
  + Data Volume: Datasets should contain a sufficient number of rows and columns. *(Open Issue: Need to define a specific range for row and column counts)*
  + Data Types: Datasets shall include a variety of relevant data types, potentially including numerical, categorical, date/time, and textual data.
* **Requirement**: The generated datasets shall include realistic data quality issues.
  + Issue Types: Include diverse but addressable data quality issues such as missing values, outliers, inconsistent formatting, and duplicates.
* **Requirement**: The system shall maintain the logical consistency between the scenario and the generated data.
* **Dataset Size Control (Future**): Future versions will allow users to influence the dataset size, potentially with a slider to generate more or fewer rows.

**4.3 User Interface (Initial Phase):**

* **Requirement:** The application shall have a minimalistic title page.
* **Requirement:** The title page shall contain a "Generate Scenario" button.
* **Requirement:** Upon clicking the "Generate Scenario" button, the system shall:
  + Generate a new scenario.
  + Generate the corresponding dataset.
  + Display the generated scenario in a clear and readable format on the page.
  + Provide a button or link to download the generated dataset as a CSV file.
* **Requirement:** The user interface shall be intuitive and self-explanatory.
* **Requirement:** The system shall display a non-obtrusive error notification (e.g., a pop-up) if an error occurs during scenario or dataset generation, prompting the user to try again.
* **Requirement:** The CSV dataset download shall be initiated by a user action (button click) and not occur automatically.

**4.4 (Future) Interactive Client:**

* **Requirement:** The system shall allow users to interact with an AI client in a text-based format.
* **Requirement:** The AI client shall be able to answer clarification questions related to the initial scenario.
* **Requirement:** If the answer to a question is not explicitly stated in the initial scenario, the AI client should be able to generate a plausible answer and adapt the scenario accordingly.

**4.5 (Future) Feedback Mechanism:**

* **Requirement:** The system shall provide feedback on the user's data analysis work, simulating a senior peer review.
  + *(Open Issue: Need to define the specific criteria and methods for feedback)*

**4.6 (Future) Portfolio Building Support:**

* **Requirement:** The platform shall provide a means for users to gain ideas for projects to include in their portfolios.

**5. Non-Functional Requirements**

* **Performance:**
  + Scenario and dataset generation should occur within a reasonable timeframe. *(Open Issue: Define acceptable generation time)*
* **Reliability:**
  + The system should consistently generate scenarios and datasets without errors in a majority of attempts. *(Open Issue: Define an acceptable error rate)*
* **Usability:**
  + The user interface should be easy to navigate and understand.
* **Maintainability:**
  + The codebase should be well-commented and structured for future modifications and scalability.
  + The system should be designed with a modular architecture to facilitate adding new features.
* **Scalability:**
  + The system architecture should be designed to accommodate future features and potential increases in usage.
* **Security:**
  + The system shall not permanently store the generated scenario text or datasets due to the potential for rapid accumulation. *(Open Issue: Define the scope and duration of temporary storage, if any)*

**6. Technical Requirements**

* **Programming Language:** Python.
* **Web Framework:** Flask (recommended, but open to alternatives).
* **AI/ML Libraries:** google-generativeai (for Gemini API), Pandas, potentially Faker.
* **Version Control:** Git.
* **IDE:** Cursor (user preference).
* **Deployment Environment:** *(Open Issue: Need to determine the initial deployment environment)*
* **Database (Potentially Future):** *(Open Issue: Determine if a database is needed for future features like user accounts or storing reports)*

**7. Data Requirements**

* **Scenario Data:** The AI model requires a foundation of information to generate realistic scenarios. This might involve training data or carefully crafted prompts.
* **Dataset Generation Logic:** The AI model for dataset generation requires logic to map the scenario elements to realistic data structures and entries, including the introduction of defined data quality issues.

**8. Success Metrics**

* **8.1 MVP Completion:** Successful development of a functional prototype with a basic web interface within the defined timeframe.
* **8.2 User Engagement (Future Testing):**
  + Number of users interacting with the AI client.
  + Frequency of use.
  + Completion of projects/scenarios.
* **8.3 User Feedback (Future Testing):**
  + Qualitative feedback on the realism of the AI client.
  + The usefulness of the tool for skill development.
  + Identification of areas for improvement in functionality and UI/UX.
* **8.4 Personal Satisfaction:** Your own assessment of the tool's value for skill development and understanding of AI scenario generation.

**9. Open Issues**

* Specific range for row and column counts in generated datasets.
* Acceptable scenario and dataset generation time.
* Acceptable error rate for scenario and dataset generation.
* Specific criteria and methods for AI feedback on user analysis.
* Definition of the initial deployment environment.
* Decision on whether a database is needed for future features.
* Scope and duration of temporary storage for generated data.
* **Specific LLM Choice:** Final decision on whether to use OpenAI API or another LLM provider.
* **Synthetic Data Tool Selection:** Evaluate and choose the most suitable synthetic data generation tool.
* **Detailed Design of "Unpredictable" Elements:** Create a more comprehensive list of potential unexpected events and changes for the AI client to use in future iterations.
* **Beta Testing Plan:** Define the specific criteria for selecting beta testers and the process for collecting and analyzing their feedback (for future testing phases).

**10. Future Considerations (Beyond Initial Phase)**

* Implementation of the interactive AI client.
* Development of the AI-powered feedback mechanism.
* Features to support portfolio building for users.
* User account management and progress tracking.
* Scenario customization options for users.
* Potential monetization strategies.