**ClickUp Task List Based on Roadmap:**

We'll organize the tasks using **Lists** in ClickUp, with each phase of the roadmap becoming a separate List. Within each List, the steps will become individual **Tasks**. For more complex steps, we can use **Subtasks**.

**ClickUp Space: Data Analyst Client Simulator (DACS)** (You can name your Space as you see fit)

**List 1: Phase 1 - Core Functionality - Scenario & Dataset Generation MVP**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Name** | **Description** | **Due Date (Estimate)** | **Priority** | **Assignee** |
| **Project Setup & Environment** | Set up Python environment, install libraries (Flask, google-generativeai, Pandas, Faker), initialize Git, set up basic Flask app. | End of Week 1 | High | You |
| **Gemini API Exploration & Integration** | Research Gemini API, get API keys, set up authentication, experiment with basic text generation. | End of Week 2 | High | You |
| **Scenario Generation Logic with Gemini** | Develop Python code to generate diverse, domain-based scenarios using Gemini API (focus on Business Task and Data Provided). | End of Week 3 | High | You |
| **Dataset Generation Logic** | Develop Python code using Pandas and Faker to generate realistic datasets based on scenario data, including data quality issues. | End of Week 4 | High | You |
| **Backend Integration - Scenario & Data** | Integrate Gemini scenario generation and Pandas dataset generation within the Flask application. Create routes and data flow logic. | End of Week 5 | High | You |
| **Basic Front-End Development** | Create basic HTML template, implement "Generate Scenario" button, display scenario details, provide CSV download link, implement basic error handling. | End of Week 6 | High | You |
| **Testing & Refinement - Scenario & Data** | Thoroughly test scenario diversity, data realism, dataset correspondence, UI functionality, error handling, and CSV download. | End of Week 8 | High | You |
| **Documentation & Initial Tracking Implementation** | Document core logic, create README, implement basic logging for scenario generation. | End of Week 9 | Medium | You |

**List 2: Phase 2 - Refinements and Preparation for Future Features**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Name** | **Description** | **Due Date (Estimate)** | **Priority** | **Assignee** |
| **Enhance Scenario Variety & Realism by Domain** | Refine Gemini prompts to create more nuanced scenarios specific to different data domains, including relevant data quality issues. | End of Week [Phase 2 - Week 2] | Medium | You |
| **Improve Dataset Logic & Size Control Foundation** | Refine dataset generation logic for domain consistency, design backend for dataset size control (explore Pandas functionalities). | End of Week [Phase 2 - Week 3] | Medium | You |
| **Plan for Domain Selection UI** | Design data model for domain categorization, plan UI element structure (e.g., dropdown). | End of Week [Phase 2 - Week 4] | Medium | You |
| **Front-End UI/UX Improvements (Basic Styling)** | Implement basic styling for a more visually appealing interface, improve clarity and guidance. | End of Week [Phase 2 - Week 5] | Low | You |
| **Code Refactoring and Modularization** | Refactor codebase to improve modularity and scalability, preparing for domain selection and size control. | End of Week [Phase 2 - Week 6] | Medium | You |
| **Advanced Testing & Error Handling (Phase 1 Features)** | Implement more robust error handling and logging. Write unit tests for scenario and dataset generation. | End of Week [Phase 2 - Week 7] | Medium | You |

**List 3: Phase 3 - (Future) Implementing Domain Selection and Dataset Size Control**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Name** | **Description** | **Due Date (Estimate)** | **Priority** | **Assignee** |
| **Develop Front-End for Domain Selection** | Implement the UI element for users to choose the data domain (e.g., dropdown). | TBD | Low | You |
| **Develop Front-End for Dataset Size Slider** | Implement the UI slider to control the number of rows in the generated dataset. | TBD | Low | You |
| **Backend Integration - Domain Selection** | Connect the front-end domain selection to the backend, filtering scenario generation by selected domain. | TBD | Low | You |
| **Backend Integration - Dataset Size Control** | Connect the front-end size slider to the backend, modifying dataset generation to produce datasets with the specified rows. | TBD | Low | You |
| **Test Domain Selection & Size Control** | Thoroughly test the new features, ensuring they function correctly and generate appropriate scenarios and datasets. | TBD | Low | You |

**List 4: Phase 4 - (Future) Implementing the Interactive AI Client**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Name** | **Description** | **Due Date (Estimate)** | **Priority** | **Assignee** |
| **Explore Gemini for Conversational AI** | Research Gemini's capabilities for understanding and responding to user queries in the context of data analysis scenarios. | TBD | Low | You |
| **Backend Integration - Gemini Interaction** | Develop backend logic to handle user input and communicate with the Gemini API for responses. | TBD | Low | You |
| **Front-End - Chat Interface Development** | Design and implement the user interface elements for interacting with the AI client (e.g., a chat window). | TBD | Low | You |
| **Implement Conversation State Management** | Develop logic to maintain conversation history and context. | TBD | Low | You |
| **Test Interactive AI Client** | Thoroughly test the interaction flow and the Gemini client's ability to understand and respond appropriately to data analysis questions. | TBD | Low | You |

**List 5: Phase 5 - (Future) Feedback Mechanism and Portfolio Building Support**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Name** | **Description** | **Due Date (Estimate)** | **Priority** | **Assignee** |
| **Define Feedback Criteria for Data Analysis** | Clearly define the criteria for evaluating user data analysis work. | TBD | Low | You |
| **Develop Feedback Logic (Potentially with Gemini)** | Explore using Gemini to analyze user submissions (code, reports) and provide feedback. | TBD | Low | You |
| **Backend Integration - Feedback Mechanism** | Integrate the feedback logic into the application. | TBD | Low | You |
| **Front-End - Feedback Display** | Design how the feedback will be presented to the user. | TBD | Low | You |
| **Implement Portfolio Idea Generation** | Develop features to suggest project ideas based on generated scenarios. | TBD | Low | You |
| **Implement User Account Management (If Needed)** | Develop user registration, login, and potentially progress tracking features. | TBD | Low | You |
| **Plan for Deployment and Scaling** | Research and plan for deploying the application to a more accessible platform. | TBD | Low | You |

**Other Ways to Use ClickUp for Project Tracking:**

* **Custom Fields:**
  + **Estimated Time:** Add a custom number field to estimate the time needed for each task.
  + **Actual Time Spent:** Use ClickUp's time tracking feature or a manual number field to track actual time spent.
  + **Technology/Tool:** Use a dropdown or text field to specify the primary technology or tool used for a task (e.g., "Gemini API," "Pandas," "Flask").
  + **Dependencies:** Use a relationship field to link tasks that are dependent on each other.
  + **Status Breakdown:** Create custom statuses within each List to further define progress (e.g., "Researching," "Developing," "Testing," "Review").
* **Views:**
  + **List View:** Your primary view for seeing all tasks.
  + **Board View (Kanban):** Visualize tasks by status to see the flow of work.
  + **Gantt Chart View:** Visualize the project timeline and dependencies. This will be particularly useful as you start assigning more concrete due dates.
  + **Calendar View:** See deadlines and scheduled tasks in a calendar format.
* **Goals:**
  + Create high-level Goals in ClickUp that correspond to the overall objectives of each roadmap phase.
  + Link individual tasks to these Goals to track progress towards larger objectives.
* **Time Tracking:**
  + Use ClickUp's built-in time tracking feature to record the time you spend on each task. This will help you understand how long tasks actually take and improve future estimations.
* **Docs:**
  + Use ClickUp Docs to store:
    - Your Production Requirements Document.
    - Meeting notes.
    - Research findings on Gemini API and data generation tools.
    - Design specifications for UI elements.
    - Any other relevant project documentation.
* **Templates:**
  + Once you've established a good workflow for a particular type of task (e.g., "Implement Backend Logic"), you can create a task template to quickly create similar tasks in the future.
* **Automations:**
  + Explore simple ClickUp Automations to streamline your workflow:
    - Automatically change the status of a task when all its subtasks are completed.
    - Assign yourself as the assignee when a new task is created in a specific list.
    - Set due dates based on triggers.
* **Dashboards:**
  + Create a ClickUp Dashboard to get a visual overview of your project progress. You can add widgets to track:
    - Tasks by status.
    - Tasks assigned to you.
    - Time tracked.
    - Progress towards Goals.

**Tips for Using ClickUp Effectively:**

* **Start Simple:** Don't try to implement all the features at once. Begin with the basic task list and gradually introduce other features as you become more comfortable.
* **Regularly Update:** Make it a habit to update task statuses, add comments, and track time regularly. This will ensure that ClickUp reflects the actual state of your project.
* **Experiment and Customize:** ClickUp is highly customizable. Experiment with different views, custom fields, and automations to find what works best for your workflow.
* **Focus on What's Useful:** Don't feel pressured to use every single feature. Focus on the features that provide the most value for your project tracking needs.