# AI Search Logic

**Main Requirement:**  
1)Show me X = search for X in the dataset If no result then = Sorry I couldn’t find X . I may have limited data

2)Sort by or filter by Y this means filter by Y but apply filter on earlier answer

3)What is Z? If Z is a company then extract its information from the context If Z is some category or vertical define it and ask they want to search Z

4)Go back to step Reverse previous action

5)How many companies are there in R category? Chat and suggest search for R make sure there are only two possible responses to any query.

Response should be short text response, and can suggest an action Action should be search, sort filter on data

**Possible Solution**

1)Perform a search based on user input.

2)Remember the previous results to apply additional filters.

3)Filter the results based on subsequent user queries.

**To achieve this, we'll need:**

1)A search mechanism (like embeddings or a search index).

2)A stateful system to remember the previous context and results.

3)Logic to filter results based on the user's query.

**For filtering part we might require,**  
Intent Detection: Use heuristics, pattern matching, or even a machine learning model to identify the intent of the user's query.

Entity Extraction: Extract the key entities or topics from the user's question. For example, in the question "Which of these companies can help with weather modeling?", "weather modeling" is the key entity.

**FLOW/STEPS**

Flow:

1.Initialization: Load the dataset and set up initial states.

2.User Query Handling: Wait for the user's input and handle the query.

3.Intent Detection: Determine the main intent of the user's query.

4.Entity Extraction: Identify the main topics or entities from the user's query.

5.Action Execution: Perform the desired action based on the detected intent and extracted entities.

6.Response & Suggestion Generation: Generate a response to the user's query and suggest the next possible actions.

7.Loop: Go back to the User Query Handling step and repeat.

Steps:

**1.Initialization:**

Load the dataset containing information about AI companies.

Initialize variables to keep track of previous results and user interactions.

2**.**UserQuery Handling:

Wait for the user to input a question or command.

Pass the user's input to the intent detection mechanism.

**3.Intent Detection**

Check if the user's query contains explicit commands like "sort by" or "go back".

If not, use heuristics, patterns, or an intent detection model to determine if the user is implying a filter, search, or other action.

**4.Entity Extraction**

If the user's intent is to filter or search, identify the main topics or entities in the user's query.

This can be done using keyword extraction, patterns, or more advanced NLP techniques like Named Entity Recognition.

**5.Action Execution**

Based on the detected intent and extracted entities, perform the appropriate action:

**Search:** Find relevant companies based on the user's query.

**Filter**: Refine the previously retrieved results based on the user's criteria.

**Sort**: Arrange the previously retrieved results based on the user's criteria.

**Go Back**: Revert to the results from the previous step.

**Count**: Return the number of companies based on a specific category.

**6.Response and suggestion generation**

Generate a response based on the results of the executed action.

Provide a suggestion for the next possible actions the user can take. This could be to further refine the results, view more details, or perform a new search.

**7.Loop**

Wait for the user's next input and repeat the process from the User Query Handling step.

**Implementation:**

To implement this flow, you'd structure your code in modular functions or methods (if using a class-based approach) that correspond to each of the outlined steps. The main loop would repeatedly execute the User Query Handling, Intent Detection, Entity Extraction, Action Execution, and Response & Suggestion Generation steps until the user decides to end the interaction.

**pseudocode**

**BEGIN**

**LOAD dataset into memory**

**INITIALIZE previous\_results as empty list**

**INITIALIZE history as empty list**

**WHILE true (or until user exits):**

**INPUT user\_query**

**IF user\_query is an explicit command:**

**EXECUTE explicit command**

**ELSE:**

**DETECT intent of user\_query**

**IF intent is "filter" OR "search":**

**EXTRACT entity from user\_query**

**IF intent is "filter":**

**FILTER previous\_results by entity**

**ELSE IF intent is "search":**

**SEARCH dataset for entity**

**END IF**

**ELSE IF intent is "sort":**

**SORT previous\_results based on user\_query**

**ELSE IF intent is "go back":**

**REVERT to previous state from history**

**ELSE IF intent is "count":**

**COUNT entities based on user\_query**

**END IF**

**END IF**

**GENERATE response based on the results of the executed action**

**DISPLAY response to user**

**SUGGEST next possible actions to user**

**END WHILE**

**END**