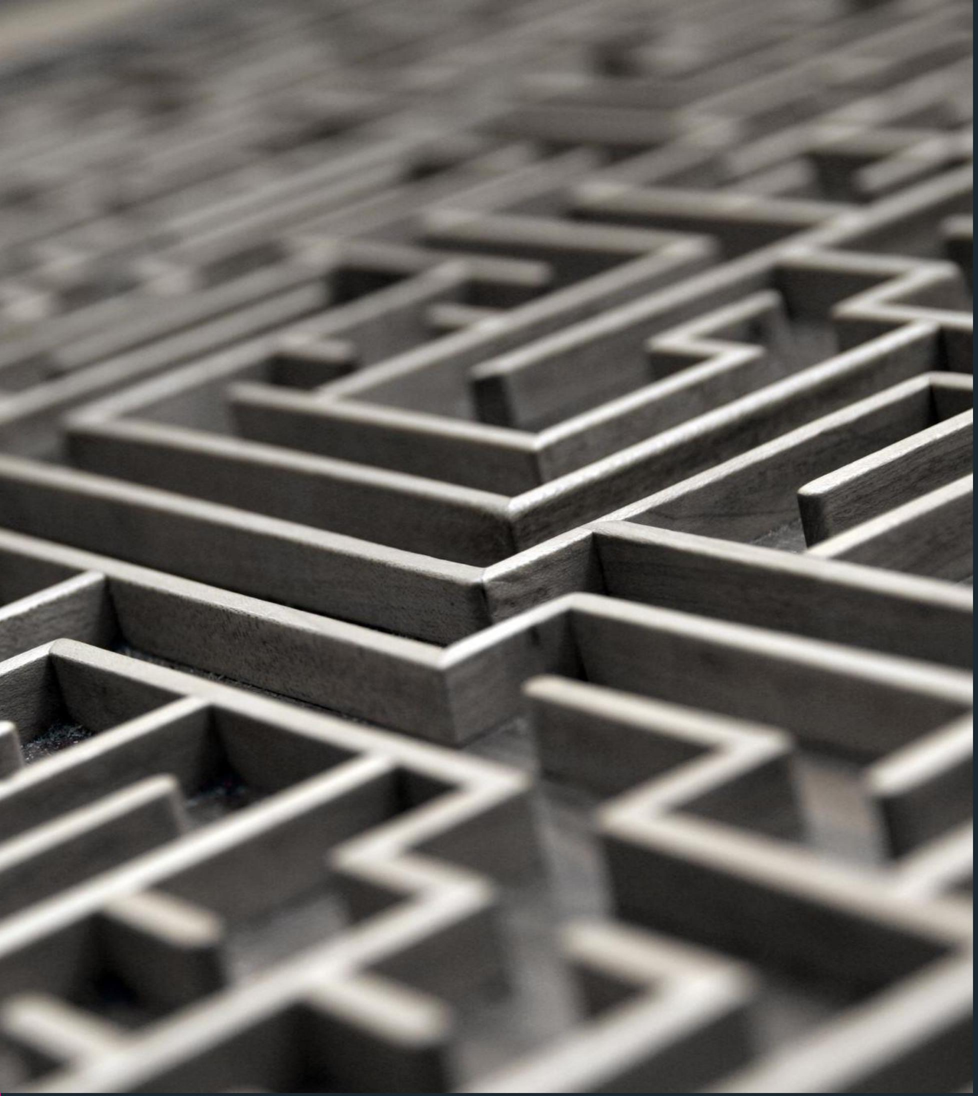
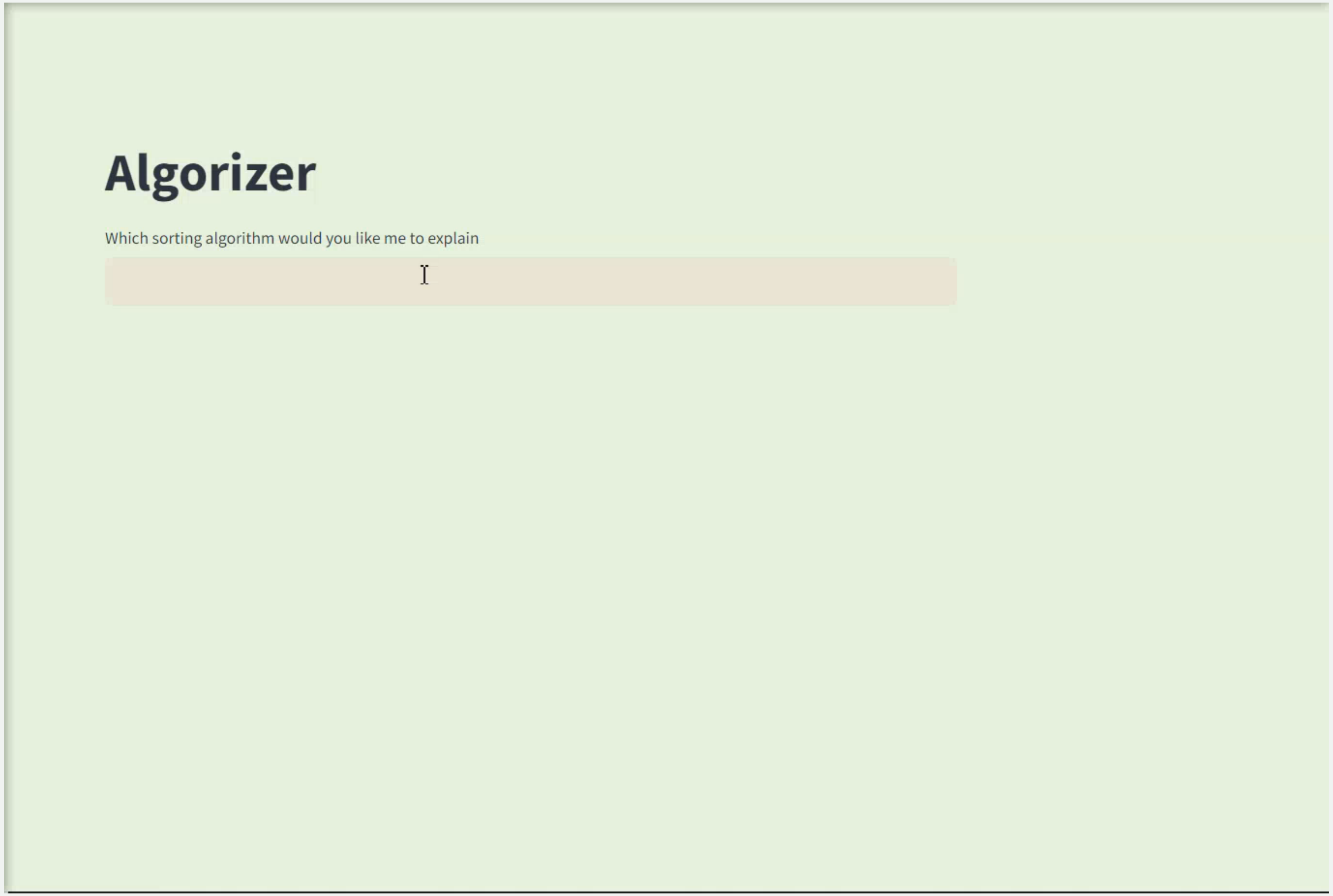
ALGORIZER



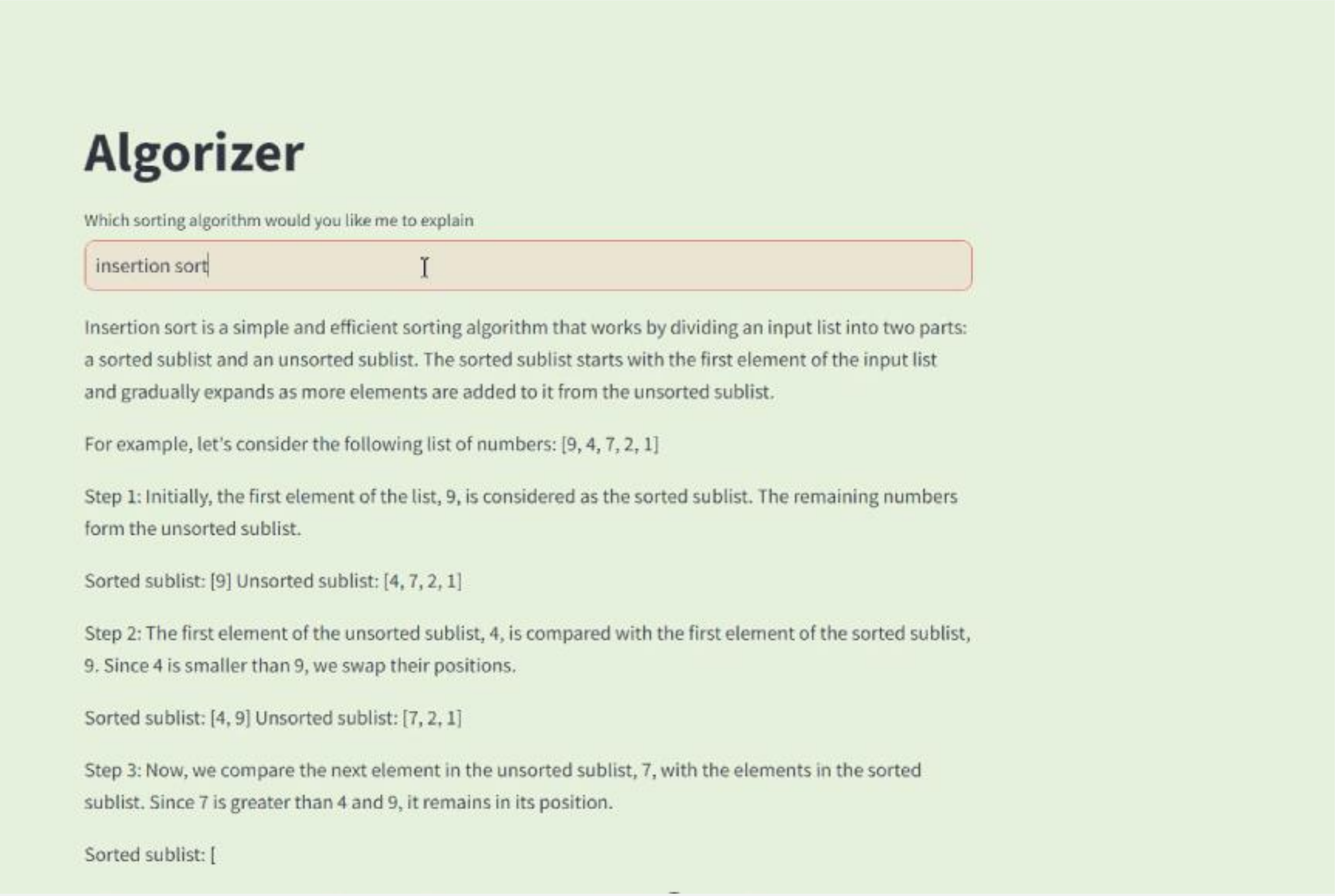
Why Algorizer?

Sorting algorithms are used extensively but understanding them can be difficult. My project explains popular sorting methods through interactive examples to help students and professionals better comprehend these fundamental computer science algorithms.



Algorizer

Which sorting algorithm would you like me to explain



Algorizer

Which sorting algorithm would you like me to explain

insertion sort

Insertion sort is a simple and efficient sorting algorithm that works by dividing an input list into two parts: a sorted sublist and an unsorted sublist. The sorted sublist starts with the first element of the input list and gradually expands as more elements are added to it from the unsorted sublist.

For example, let's consider the following list of numbers: [9, 4, 7, 2, 1]

Step 1: Initially, the first element of the list, 9, is considered as the sorted sublist. The remaining numbers form the unsorted sublist.

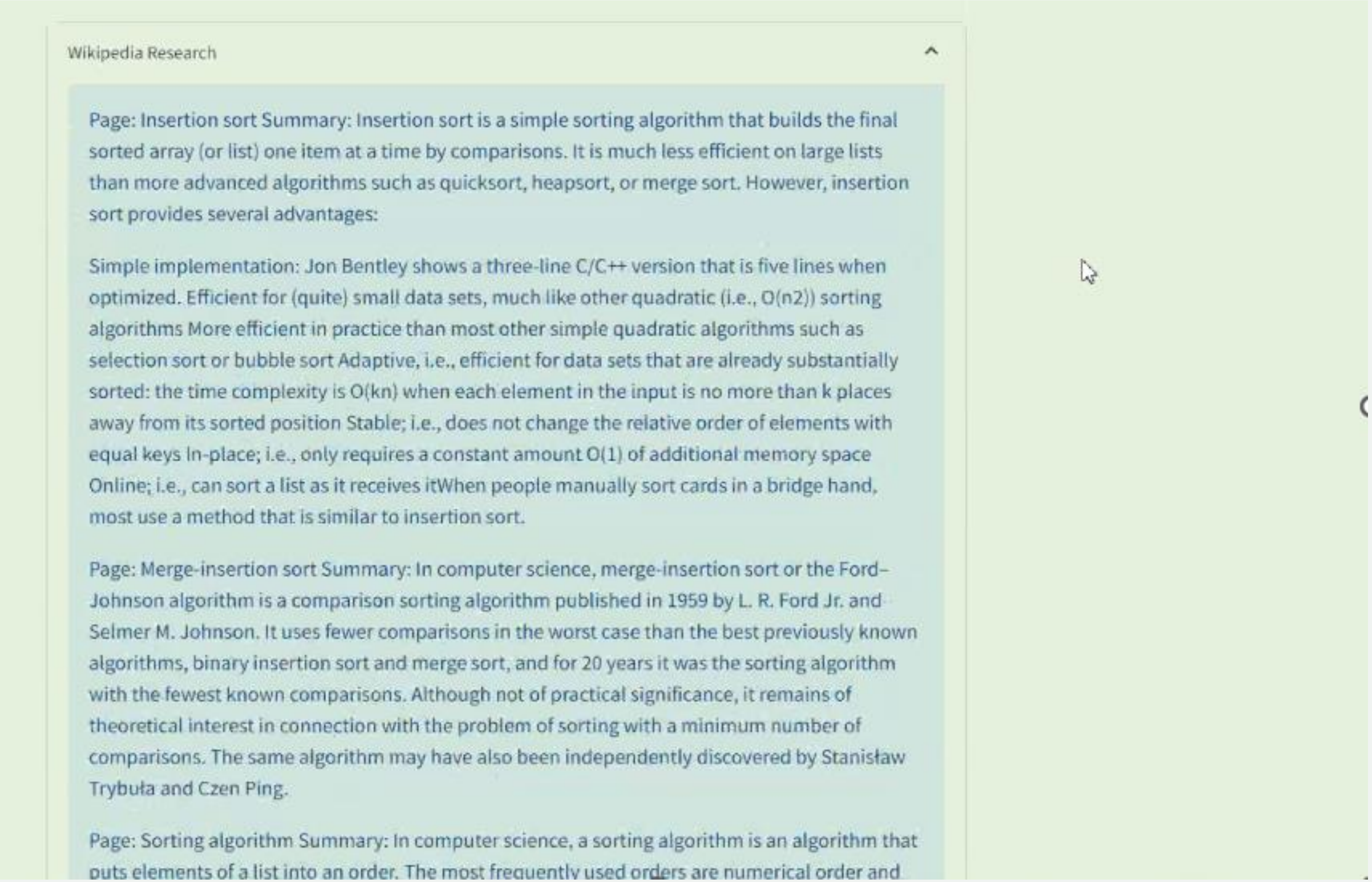
Sorted sublist: [9] Unsorted sublist: [4, 7, 2, 1]

Step 2: The first element of the unsorted sublist, 4, is compared with the first element of the sorted sublist, 9. Since 4 is smaller than 9, we swap their positions.

Sorted sublist: [4, 9] Unsorted sublist: [7, 2, 1]

Step 3: Now, we compare the next element in the unsorted sublist, 7, with the elements in the sorted sublist. Since 7 is greater than 4 and 9, it remains in its position.

Sorted sublist: [



Page: Insertion sort Summary: Insertion sort is a simple sorting algorithm that builds the final sorted array (or list) one item at a time by comparisons. It is much less efficient on large lists than more advanced algorithms such as quicksort, heapsort, or merge sort. However, insertion sort provides several advantages:

Simple implementation: Jon Bentley shows a three-line C/C++ version that is five lines when optimized. Efficient for (quite) small data sets, much like other quadratic (i.e., O(n2)) sorting algorithms More efficient in practice than most other simple quadratic algorithms such as selection sort or bubble sort Adaptive, i.e., efficient for data sets that are already substantially sorted: the time complexity is O(kn) when each element in the input is no more than k places away from its sorted position Stable; i.e., does not change the relative order of elements with equal keys In-place; i.e., only requires a constant amount O(1) of additional memory space Online; i.e., can sort a list as it receives itWhen people manually sort cards in a bridge hand, most use a method that is similar to insertion sort.

Page: Merge-insertion sort Summary: In computer science, merge-insertion sort or the Ford–Johnson algorithm is a comparison sorting algorithm published in 1959 by L. R. Ford Jr. and Selmer M. Johnson. It uses fewer comparisons in the worst case than the best previously known algorithms, binary insertion sort and merge sort, and for 20 years it was the sorting algorithm with the fewest known comparisons. Although not of practical significance, it remains of theoretical interest in connection with the problem of sorting with a minimum number of comparisons. The same algorithm may have also been independently discovered by Stanisław Trybuła and Czen Ping.

Page: Sorting algorithm Summary: In computer science, a sorting algorithm is an algorithm that puts elements of a list into an order. The most frequently used orders are numerical order and

What Technologies I used in this app.

- LangChain
- Chromadb
- Wikipedia
- Tiktoken
- Opengpt

Future Scope

- I plan on adding visualization and interactive arrays that can be sorted according to the algorithm selected by the user.
- The AI model can be fine tuned to explain every step of the example array being sorted according to a specific algorithm.