**Project Name:** Sorting Algorithms Visualizer

**Github Link:** https://github.com/projectsforstudents2022/Sorting\_Algorithms\_Visualizer.git

**Why was this project created?**

Sorting algorithms are a common component of modern computer software. On your PC, for instance, you might see files arranged in different ways if you launch file explorer. In sorted data, searching is more effective than in unsorted ones. Sorting algorithms are one of the many algorithms that computer science students begin learning in the beginning of study. Since I encountered sorting issues while designing algorithms in my starting year of study, it is understood that visual representation is an essential component of the learning process.

**What problem is it solving?**

The primary objective of this project is the development of a teaching aid programme that includes visualizations of the most popular sorting algorithms and their modifications. The programme offers the option of step-by-step execution, pseudocode, and the current state of variables, together with a graphic visualization of a few algorithms on a manually or arbitrarily formed array.

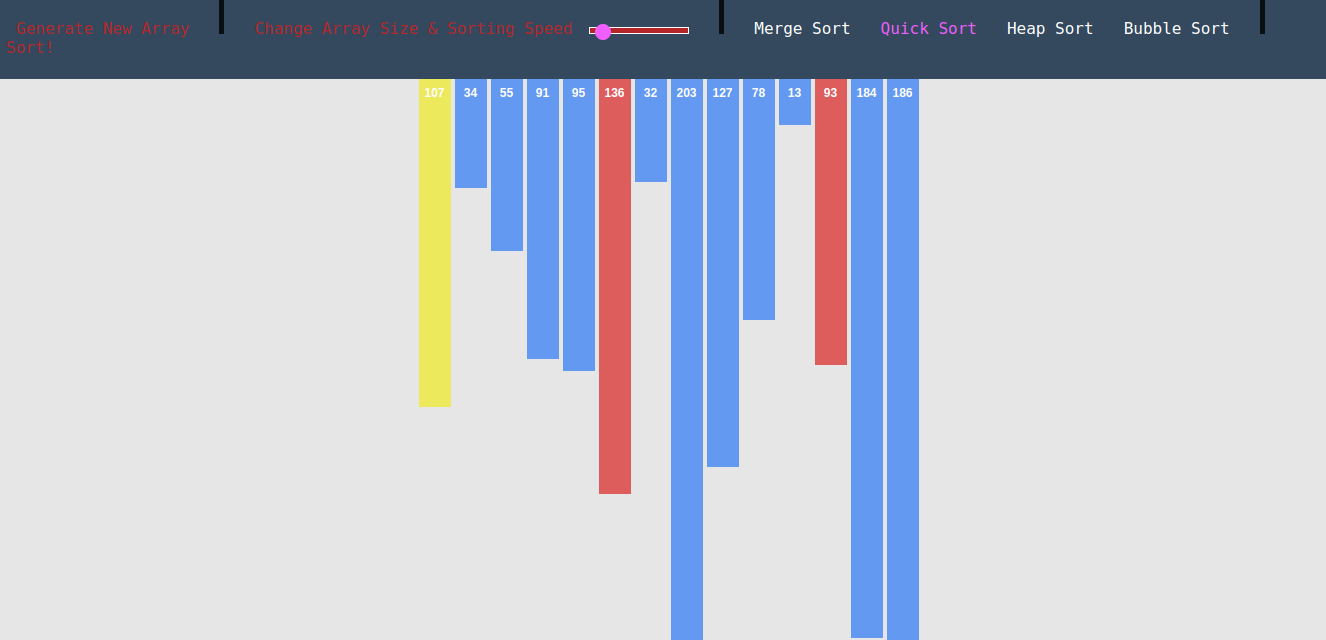
**Entire explanation of project**

* **PROPOSED APPROACH**

This software organizes the code using both object-oriented and functional programming paradigms. Only three objects, one to manage the canvas that displayed the animation, one to represent a piece of data, or "bar" object, and one to represent the places that each bar traveled to, or "pos'' objects, were employed in the design before it reached its final stage of development. The sortArea is responsible for creating the bar graph and uses a timer to keep the bars updated. As a result, everytime "Step" is called, the steps array (explained below) is used to update the bar values. The rectangles in the sortArea will be redrawn every second iteration of the timer with varying heights to reflect the updated values. When the "Step" button is selected, the change occurs instantly because the bars change sixty times per second. Each item of data in the sortArea is represented as a bar object. All facets of color, value, location, height, and sound are included in the statement.

The sorter, the only component of the model, is what makes it up. The algorithm's code, broken down into methods, is included in this object. The Start method uses a single integer constant to organize all potential algorithms. The four sorting algorithms described on the user interface as "Selection Sort, Bubble Sort, Insertion Sort, and Merge/Insertion Sort directly control this object. As soon as the user chooses a sorting method and clicks on one of the sort algorithm buttons, the sort algorithm method is activated.

* **RESULT**



* **CONCLUSION**

The application lets you navigate each represented algorithm by going ahead and backward. On a random or custom array, the user can perform sorting. The application displays pseudocode and up-to-date data on some variables during the demonstration run. I made an effort to produce top-notch software that could be utilized by professors, tutors, and students alike and had an intuitive and user-friendly interface.