Algorithm Visualization Project Report

Instructor: Prof. Shereen Taie

TA: Eng. Ismail Ahmed

Team Members

Karim Bassem Joseph (ID: 231000797) – Project Lead & Backend Developer

Nadeem Diaa Shokry (ID: 231000857) – Frontend Developer

Ahmed Khaled Shalaby (ID: 231000514) – Algorithm Specialist and UI/UX

Mahmoud Ahmed Farag (ID: 231002115) – Algorithm Specialist and Testing

Introduction

Algorithms are an essential part of computer science and software development. However, understanding how algorithms work internally can sometimes be difficult, especially for beginners.
Our project, Algorithm Visualizer, aims to solve this problem by providing an interactive web application where users can see how sorting algorithms behave step-by-step. By visualizing the process, users can better understand the logic behind each algorithm.

Project Objectives

The main goals of our project are:

1. Build a simple and intuitive web interface for visualizing sorting algorithms.
2. Develop animations that show each step of the algorithm.
3. Allow users to input their own datasets and watch how the algorithm handles them.
4. Provide performance comparisons (time and space complexity) between different algorithms.

Technologies Used

- Backend: Python (Flask Framework)
- Frontend: HTML, CSS, JavaScript
- Algorithms: Implemented in Python (Bubble Sort, Insertion Sort, Selection Sort, Merge Sort, Quick Sort)

Project Features

1. Algorithm Visualization

We created visual representations of multiple sorting algorithms like:

- Bubble Sort
- Insertion Sort
- Selection Sort
- Merge Sort
- Quick Sort

The user can see each individual operation (like swapping or comparing elements) clearly.

2. Interactive User Controls

Users can:

- Start, pause, and step through the algorithm.
- Choose how fast or slow the animation runs.
- Enter their own datasets to customize the visualization.

3. Educational Support

To make the tool more educational, we added:

- Information about time and space complexity for each algorithm.
- A section that allows users to compare different algorithms based on performance.

4. User Experience

The web application has:

- A clean and responsive design that works on laptops, tablets, and phones.
- Error handling (for example, if a user input invalid data).
- Smooth navigation between pages.

Conclusion

Through this project, we combined our knowledge of algorithms, backend development, and frontend design to create an educational tool that helps users visualize sorting processes easily.
We successfully achieved the objectives we planned, and the system is working as intended.
This project also helped us improve our teamwork, problem-solving, and software development skills.

We believe Algorithm Visualizer can be a useful tool for students who want to build a better understanding of sorting algorithms.