**PROJECT BASED LEARNING**

**Report on**

**“SORTING VISUALIZER”**

**SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF**

**SECOND YEAR COMPUTER ENGINEERING**

**SUBMITTED BY**

**SHIVAJI RAUT ROLL NO – 44**

**BHAVEN RATHOD ROLL NO – 43**

**MANASI GODSE ROLL NO – 22**

**SHIVAM KORADE ROLL NO – 50**

**UNDER THE GUIDANCE OF Prof.S.P.BAMANE**

# DEPARTMENT OF COMPUTER ENGINEERING

**TSSM’s**

# PADMABHOOSHAN VASANTDADA PATIL INSTITUTE OF TECHNOLOGY,BAVADHAN, PUNE-21.

**2022-23**

**AFFILIATED TO**



**CERTIFICATE**

**This is to certify that the seminar report entities**

“SORTING VISUALIZER TOOL”

**Submitted by**

**SHIVAJI RAUT**

**BHAVEN RATHOD**

**MANASI GODSE**

**SHIVAM KORDE**

**is a bonafide work carried out by above students under the guidance of Prof.Shobha Bamane and it is approved for the Computer Graphics Laboratory – Mini Project fulfilment of the requirement of Savitribai Phule Pune University.**

**Prof. S.P.BAMANE Prof. G.S.WAYAL**

**Subject Teacher HOD, Computer Engineering**

**Dr.R S.Pawar Principal**

# PADMABHOOSHAN VASANTDADA PATIL INSTITUTE OF TECHNOLOGY,BAVADHAN, PUNE-21.

**Place: Bavdhan,Pune**

**Date: / /**

# ACKNOWLEDGEMENT

With due respect and gratitude I take the opportunity to thank those who have helped me directly and indirectly. I convey my sincere thanks to Prof.G.S.Wayal HoD of computer Department. and Prof. Shobha Bamane their help in selecting The mini project topic and support. I thank to my seminar guide Prof. Prof.Shobha Bamane for her guidance, timely Help and valuable suggestions without which this seminar would not have been. Possible. Her direction has always been encouraging as well as inspiring for me. Attempts have been made to minimize the errors in the report. I would also like to express my appreciation and thanks to all my friends who knowingly have assisted and encourage me throughout my hard Work.

# Table of Contents

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Contents** | **Page No.** |
| 1 | Abstract | 5 |
| 2 | Introduction | 6 |
| 3 | Code Implementation | 7 |
| 4 | Screenshots (Outputs) | 11 |
| 5 | Applications | 13 |
| 6 | Future Prospects | 14 |
| 7 | Conclusion | 15 |
| 8 | References | 16 |

**ABSTRACT**

Although learning algorithms can be challenging, doing so is necessary for improving computational thinking and programming abilities. To solve this problem, our team created the Algorithm Visualizer interactive online tool, which gives students a fun and interactive way to learn how different algorithms are implemented. With three various learning speeds—slow, average, and fast—the application is made to support students in learning at their own pace. It also offers graphical representations of data and algorithms. The goal of the project is to make learning an amazing experience that inspires children to study more. This study examines the advantages of algorithm visualization as a tool for better comprehending data structures and algorithms and introduces an interactive web-based tool.

# INTRODUCTION

Welcome to our sorting visualizer tool! Sorting algorithms are a crucial concept in computer science that are applied in various applications, including data organization and optimizing search algorithms. Our sorting visualizer aims to provide users with a comprehensive tool to explore and understand various sorting algorithms in a visually engaging way.

The sorting visualizer tool currently supports several popular sorting algorithms, including bubble sort, insertion sort, selection sort, merge sort, and quicksort. Each of these algorithms works differently to sort elements in an input array and generate a sorted output. The tool offers an intuitive representation of each algorithm's sorting process, which helps users understand their differences and how they operate.

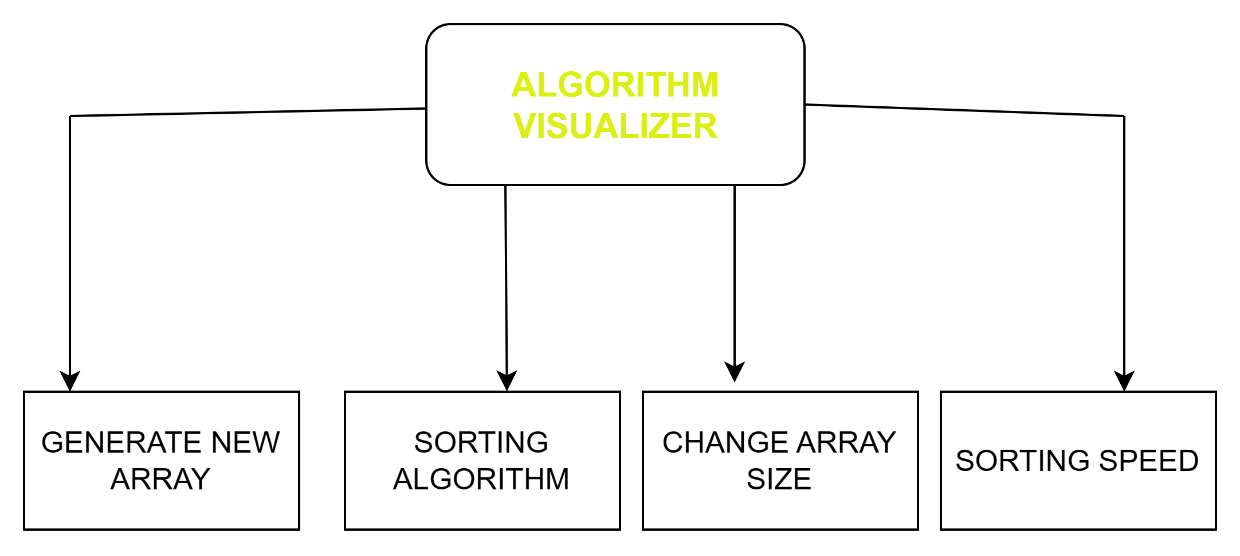
Our sorting visualizer tool allows you to customize various aspects of the visualization, such as the size of the input array, the speed of the animation, and the color scheme. You can experiment with different input sizes, adjust the animation speed, and select a color scheme that best fits your preferences.

One of the distinct features of our sorting visualizer tool is its ability to highlight the essential steps of each algorithm's sorting process. This feature enables users to understand the logic behind each algorithm's sorting process and how it operates on the input data. Additionally, we provide in-depth descriptions of each sorting algorithm, including their time and space complexity, to help users understand their theoretical aspects.

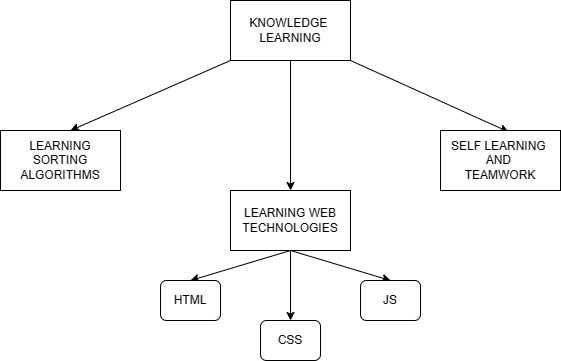
Our sorting visualizer tool caters to a wide range of users, from students learning sorting algorithms in a classroom setting to developers who want to optimize their code. Moreover, it is a fun and engaging way for anyone to explore and experiment with different sorting techniques.

So, what are you waiting for? Start exploring our sorting visualizer tool and compare different sorting algorithms. Whether you're interested in learning more about sorting algorithms or want to experiment with different settings and visualizations, our sorting visualizer tool has something to offer for everyone.

**SYSTEM ARCHITECTURE**



**MODELLING**



# CODE IMPLEMENTATION

<!DOCTYPE html>

<html>

<head>

<script

async

src="https://www.googletagmanager.com/gtag/js?id=G-4HHP888F7Y"

></script>

<script>

window.dataLayer = window.dataLayer || [];

function gtag() {

dataLayer.push(arguments);

}

gtag("js", new Date());

gtag("config", "G-4HHP888F7Y");

</script>

<meta charset="utf-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<title>Sorting Visualizer</title>

<meta name="description" content="" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<link

rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"

integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"

crossorigin="anonymous"

/>

<!-- CSS only -->

<link rel="stylesheet" href="style.css" />

</head>

<body>

<nav class="navbar dark-background">

<a class="navbar-brand text-color" href="#">Sorting Visualizer</a>

</nav>

<div id="sorting" class="flex-container"></div>

<div class="col range-input" id="input-panel">

<span id="size" class="label"

>Size

<input

id="size\_input"

type="range"

value="50"

min="1"

max="100"

style="margin-left: 60px"

/>

</span>

<span id="speed" class="label"

>Speed

<input

id="speed\_input"

type="range"

min="20"

max="300"

value="60"

style="margin-left: 60px"

/>

</span>

</div>

<div

class="btn-group-md btn-group-horizontal center"

style="margin-bottom: 10px"

>

<button

type="button"

class="btn btn-success new"

style="margin-right: 50px"

>

New Array

</button>

<button

type="button"

class="btn btn-danger stop"

style="margin-left: 5px"

disabled="true"

>

Stop Sorting

</button>

</div>

<div class="col-auto center" id="control-panel">

<div class="btn-group-md btn-group-vertical">

<button

type="button"

class="btn btn-outline-light bubbleSort"

style="

background-color: #83c5be;

border: 2px solid black;

margin-right: 200px;

"

>

Bubble Sort

</button>

<button

type="button"

class="btn btn-outline-light mergeSort"

style="

background-color: #83c5be;

border: 2px solid black;

margin-right: 200px;

"

>

Merge Sort

</button>

</div>

<div class="btn-group-md btn-group-vertical">

<button

type="button"

class="btn btn-outline-light selectionSort"

style="

background-color: #83c5be;

border: 2px solid black;

margin-right: 200px;

"

>

Selection Sort

</button>

<button

type="button"

class="btn btn-outline-light quickSort"

style="

background-color: #83c5be;

border: 2px solid black;

margin-right: 200px;

"

>

Quick Sort

</button>

</div>

<div class="btn-group-md btn-group-vertical">

<button

type="button"

class="btn btn-outline-light insertionSort"

style="

background-color: #83c5be;

border: 2px solid black;

margin-right: 200px;

"

>

Insertion Sort

</button>

<button

type="button"

class="btn btn-outline-light heapSort"

style="

background-color: #83c5be;

border: 2px solid black;

margin-right: 200px;

"

>

Heap Sort

</button>

</div>

</div>

<script

src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js"

integrity="sha384-b5kHyXgcpbZJO/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOnJ0"

crossorigin="anonymous"

></script>

<script src="sorting.js" async defer></script>

<script src="algorithms/bubbleSort.js" async defer></script>

<script src="algorithms/mergeSort.js" async defer></script>

<script src="algorithms/selectionSort.js" async defer></script>

<script src="algorithms/quickSort.js" async defer></script>

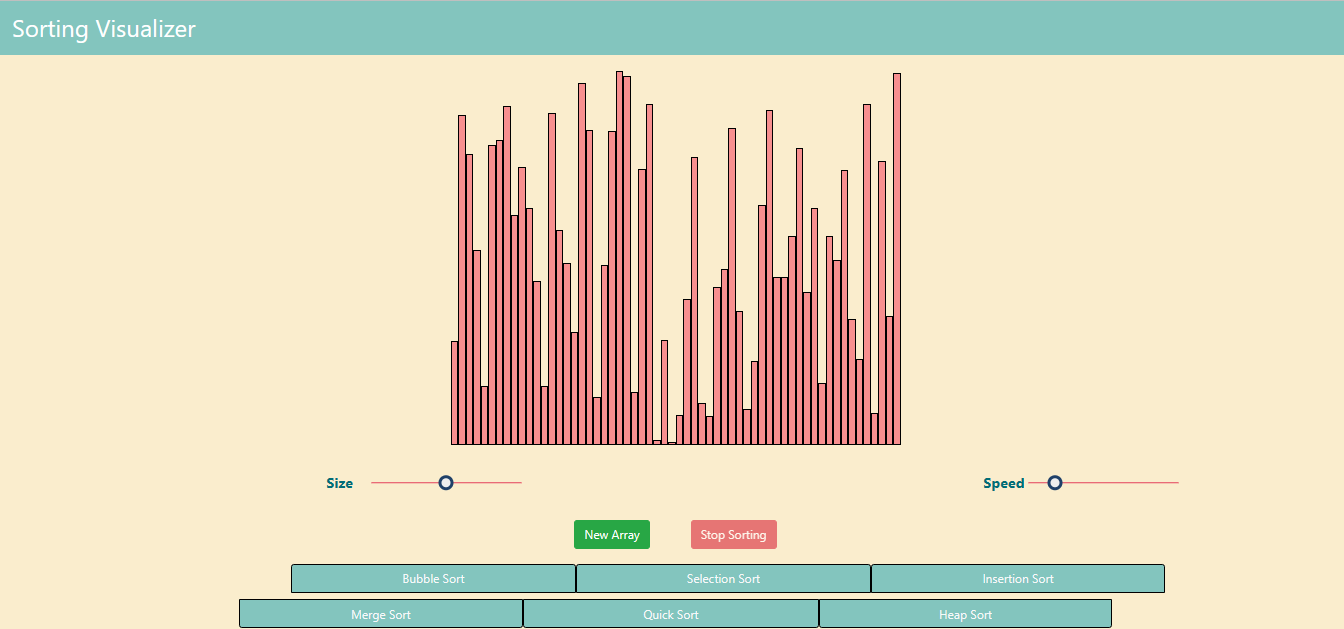
<script src="algorithms/insertionSort.js" async defer></script>

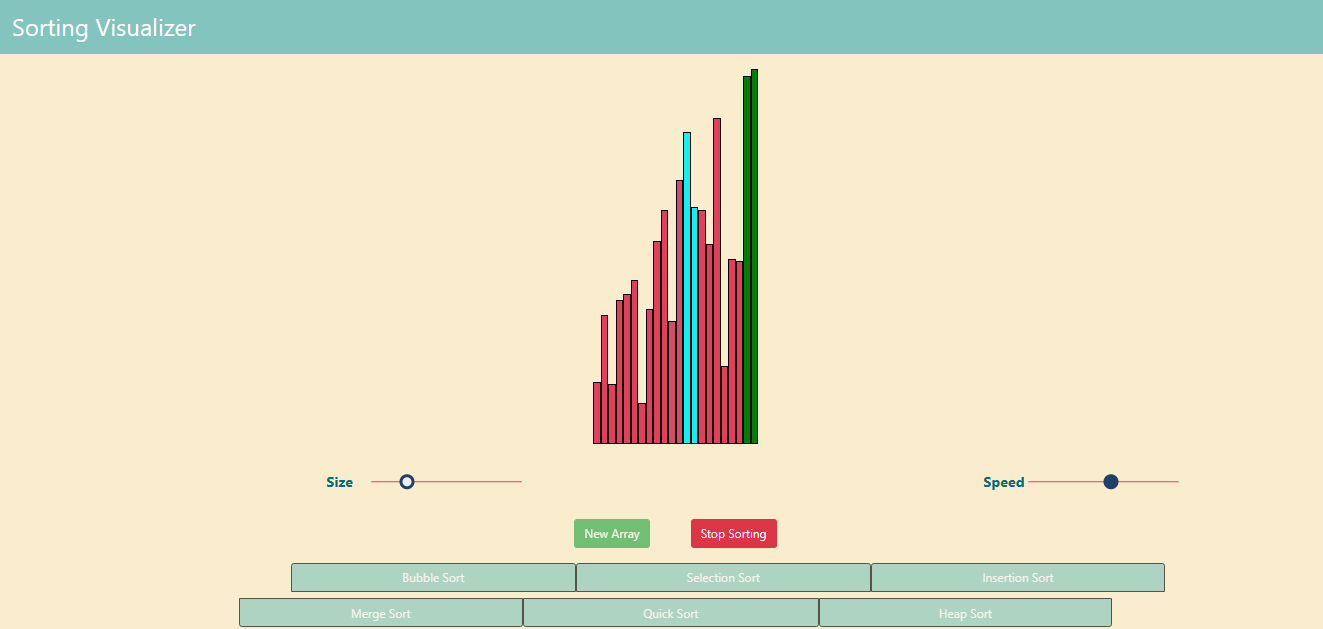
<script src="algorithms/heapSort.js" async defer></script>

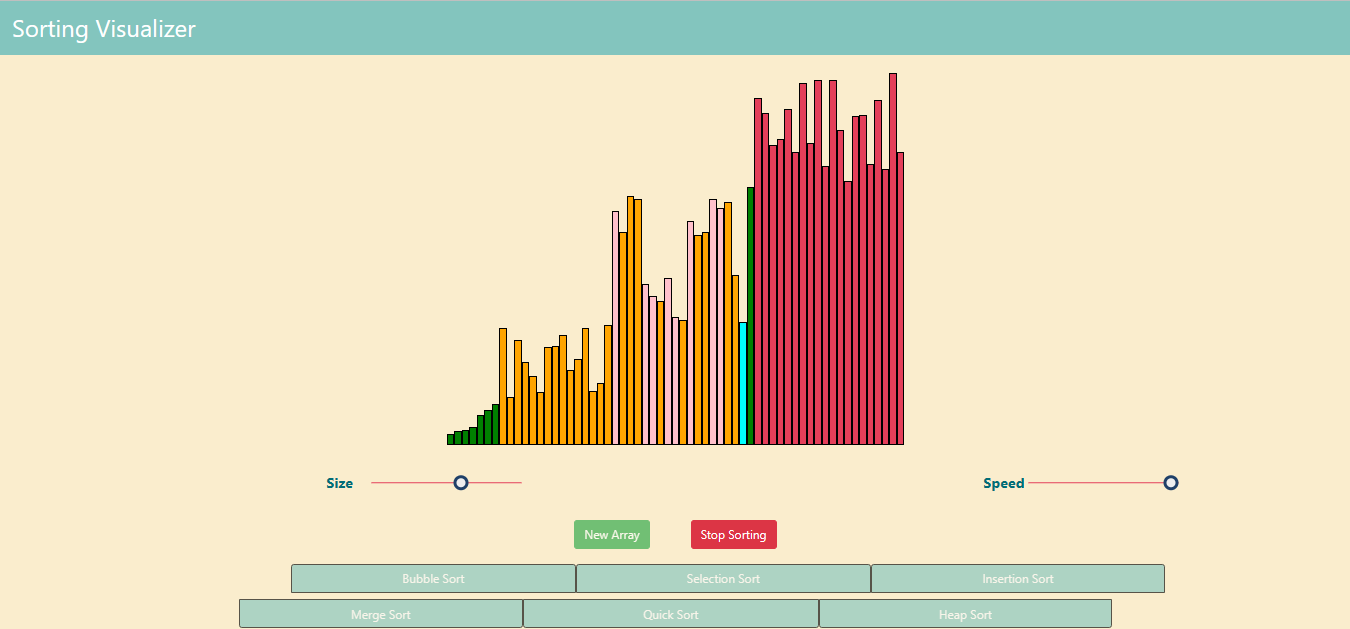
</body>

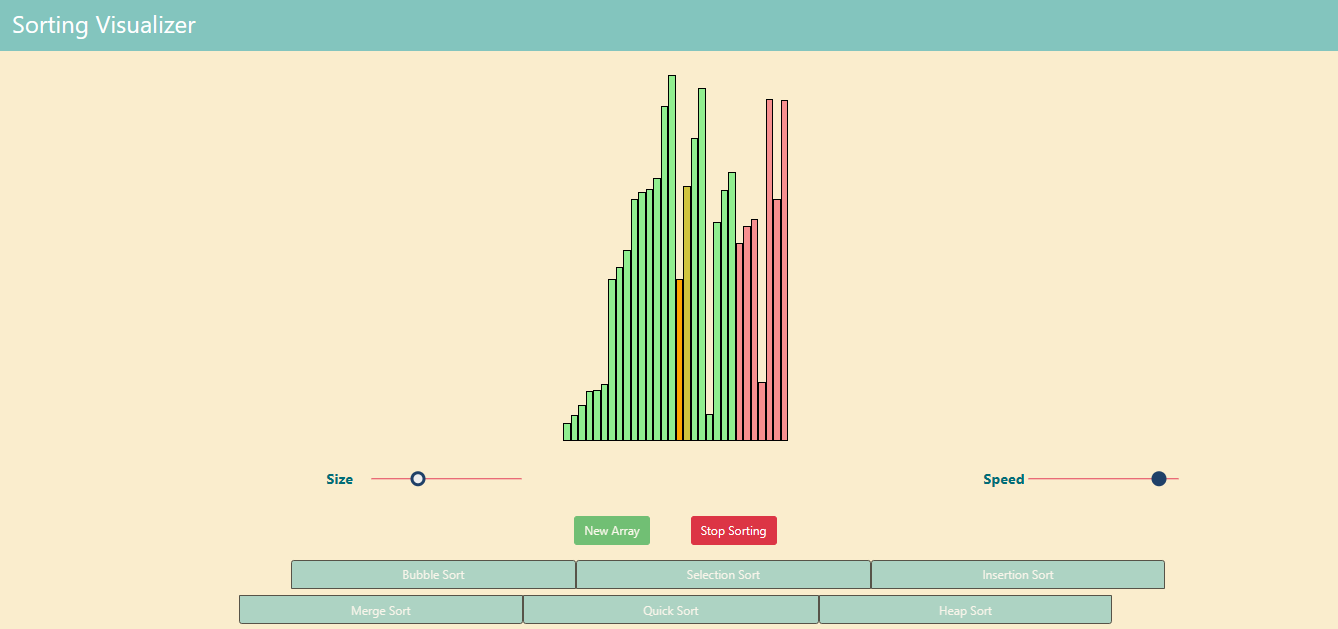
</html>

# ­­­­­­SCREENSHOTS









# APPLICATIONS

# Education: Sorting algorithms are a vital concept in computer science, and the sorting visualizer can be an effective tool for educating students about them. By displaying a visual representation of how different algorithms work, students can gain a deeper understanding of the concepts and apply them to other areas of computer science. The sorting visualizer is particularly beneficial for students who are just starting to learn about sorting algorithms and may have difficulty comprehending abstract concepts.

# Development: Developers frequently encounter situations where they need to sort data rapidly and efficiently. By using the sorting visualizer, developers can determine which algorithm would be best for their specific use case and optimize their code accordingly. The sorting visualizer can also assist in understanding the performance characteristics of different algorithms, such as their time and space complexity, and make informed decisions about which one to use.

# Research: Computer science researchers can use the sorting visualizer to investigate the performance and behavior of different sorting algorithms. This can assist them in creating new algorithms or enhancing existing ones. By visualizing the sorting process, researchers can gain insights into how algorithms behave under different circumstances and identify areas for further research.

# Gamification: Sorting algorithms may be tedious and uninteresting to learn about, but the sorting visualizer can make them enjoyable and engaging. By transforming the sorting process into a game, users can develop an interest in the subject and learn in a more enjoyable manner. This can be particularly helpful for younger students who may struggle to stay engaged with conventional teaching methods.

# Data Analysis: Data analysts and scientists often deal with large data sets that require rapid sorting. The sorting visualizer can assist them in determining which algorithm is best suited for their data and presenting the results in a meaningful manner. By providing a visual representation of the sorting process, data analysts can better understand how the algorithm works and make informed decisions about how to process and analyze their data.

# In summary, the sorting visualizer project has many potential applications and can be beneficial to a wide range of users, from students to professionals. By providing a visual representation of sorting algorithms, users can gain a better understanding of the concepts and make informed decisions about how to apply them in their work.

# FUTURE PROSPECTS

Expansion of Features: Your Sorting Visualizer project is already an excellent tool for visualizing the sorting process, but there are several potential features that could enhance the user experience. For instance, adding the ability to sort various data structures and allowing users to customize the visualization could make your project more useful and engaging to a broader audience.

Integration with Other Tools: Integrating your Sorting Visualizer with other tools or platforms could make it more valuable for users. For example, you could integrate it with popular development environments or online coding platforms, allowing developers to test and optimize their code efficiently. You could also integrate it with education platforms to make it easier for teachers to include your project in their lessons.

Community Building: Building a community around your Sorting Visualizer project could generate interest and engagement from users. You could create a forum or chatroom where users can share ideas and ask questions or organize coding challenges or competitions using your Sorting Visualizer. Building a community around your project can ensure its longevity and continued relevance in the future.

Expansion to Other Areas: Although your Sorting Visualizer project focuses on sorting algorithms, similar visualization tools could be useful in other areas of computer science. For instance, you could create a visualizer for searching algorithms, or for data structures like trees and graphs. By expanding your project to other areas, you could reach a wider audience and help users gain a deeper understanding of other essential computer science concepts.

Overall, the future prospects for your Sorting Visualizer project are bright. By improving and expanding your project, you can make it a valuable tool for students, developers, researchers, and anyone interested in computer science and data analysis.

# CONCLUSION

The sorting visualizer project is an indispensable tool for comprehending and visualizing sorting algorithms. By presenting a visual representation of how different algorithms work, users can gain a better understanding of the concepts and apply them to other areas of computer science. This project has wide-ranging potential applications, from education to development, research, and more.

As the project progresses and expands, there are many potential future prospects, such as incorporating new features, integrating with other tools, building a community, and extending into other areas of computer science. By continuously improving and expanding this project, it can become a valuable resource for anyone interested in computer science and data analysis.

In conclusion, the sorting visualizer project is a significant contribution to the field of computer science, and its potential applications and future prospects are exciting. By consistently improving and expanding the project, more users can gain a better understanding of sorting algorithms and other essential computer science concepts.

# REFERENCES

[1] “E-learning Tool for Visualization of Shortest Paths Algorithms” by Daniela Borissova and Ivan

Mustakerov, ResearchGate, July 2015.

[2] “Algorithm Visualization: The State” of the Field by Clifford A. Shaffer, Matthew L. Cooper, Alexander

Joel D. Alon, Monika Akbar, Michael Stewart, Sean Ponce and Stephen H. Edwardsacm Transactions on

Computing Education, Vol. 10, No. 3, Article 9, Pub. date: August 2010.

[3] “Visualizing sorting algorithms” by Brian Faria, Rhode Island College, 2017.

[4] P. Agrawal, H. Kaur, G. Singh, Indexed Tree Sort: An Approach to Sort Huge Data with

Improved Time Complexity, International Journal of Computer Applications 57(18)

(2012).

[5] A. D. Mishra and D. Garg, Selection of best sorting algorithm, International Journal of

Intelligent Information Processing 2(2) (2008), 363-368.

[6] R. Rahim, S. Nurarif, M. Ramadhan, S. Aisyah and W. Purba, December, Comparison

searching process of linear, binary and interpolation algorithm, Journal of Physics:

Conference Series, IOP Publishing (930)(1) (2017), 012007.