WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

AlgoViz – Where Data Structures Grow Visually

[1] Dr.Y.Baby Kalpana, [2] Sanmathi Priya K. S, [3] Manoj Kumar P, [4] Sachin Annadasan, [5] Om Prakash P

[12345]Sri Shakthi Institute Of Engineering And Technology,Coimbatore

A B S T R A C T:

Algoviz is a state-of-the-art learning tool that leverages live animations and interactive Visualizations to change the way developers and students learn about algorithms. Algoviz animates algorithm ideas in a way that immediately bridges the gap between the theoretical part of the algorithm and the abstraction of algorithmic ideas; being able to see the ideas come alive in a fast-paced animation makes them easily digestible. Algoviz allows students to do more than just write an initial code and then see nothing but a blank screen; they can now see how their innovations work with step-by-step animations and gain real coding experience and problem solving confidence. The Algoviz environment supports a whole variety of topics including graph traversal, sorting, searching, dynamic programming, and many more. There are visualizations that can be understood by students with widely-ranging levels of understanding; providing explanations and various programming language (and corresponding code samples) versions. Algo Vis allows for an authentic and engaged learning experience by allowing input of one's own data, a change of speed of the algorithms, and observing the various algorithmic flows in directions

INTRODUCTION

AlgoViz offers a distinct service to explore algorithms using interactive visuals and animation. The arrangement is structured in an manner that algorithmic processes can be "followed through" easily step by step and are less likely to cause someone to miss important components of the procedural nature of the algorithm. Exhaustive animation transitions will illustrate the highlighting steps in a way that makes the entire process less daunting and more engaging. There is device responsive control functionality that allows the user to modify the input values and see the effects of the differences in the process along with the results to inspire learning by discovery based processes Sorting methods, advanced data structures, searching methodology are all interactive visual. Decomposing for explanations in a stepwise matter encourages interaction while reducing static visuals and improving learning and understanding. Promotional insights of relative utility and complexity potentially encourage contrast in classroom juxtaposition reinforcing comparative assessments. Promoting interaction in real time encourages a real time learning making Algo Viz a fabulous resource to capitalize on algorithms it also allows for performance analysis beyond animation using detailed analysis of execution time, used memory, and computational trade offs. Cross platform functionality provides uninterrupted device experience which allows for ease of learning for students, teachers, and professionals. The interface promotes user engaged interaction.

REVIEW OF LITERATURE

Algorithm Visualization in Education

Algoviz employs real-time animations and interactive graphics to help students and developers understand algorithm functionality. In Algoviz, you can visualize processes such as sorting, searching, and graph traversal, as each component is visualized step-by-step to develop an understanding of how the process works. Algoviz provides an opportunity to visualize algorithm processes and remove the confusion from the abstract theoretical knowledge of algorithms by applying the theory to concepts.

User Interaction and Customization in Learning Tools

Learners can enter any data they wish, control the speed of algorithm execution, and watch how the flow occurs in real time! Engaging the learner in a hands-on experience fosters engagement and supports their effective problem solving and writing code with confidence.

Algorithm Visualization in Education

For every algorithm visualization, there are well-researched explanations with code samples in multiple programming languages with the potential of allowing students of various experience levels and programming languages to proceed with the learning outcomes.

EXISTING SYSTEMS

Currently, most existing platforms that teach algorithms do it via static materials (tutorials, texts, or videos (recorded)). Some platforms allow customized visualizations, but these customizations have restrictions. Most of the recorded content does little to foster real-time interactivity, sometimes even controlling the speed of the visualizations to see what an algorithm is doing or allowing a user to input their data. The tools we discussed only provide passive learning experiences, where learners have little user interactivity and no custom learning experiences. Many of the existing tools only indicate a small band of algorithms or only perform in one programming language, which may restrict their use to narrow curriculum or learning styles while concurrently not having explainers integrated in correlation to the visualization, meaning learners will have to sway back and forth between windows or sources of information. In addition, many existing tools do not account for learning diversity in students, especially for those that rely on sources or videos that are text-based.

FIELD OF INVENTION

For many developers and students, learning the subspecialty of computer science known as data structures and algorithms (DSA) can tend to be a largely painful process. It can be overwhelming for both students and developers on the outset, as even learning about the basic ideas about algorithms like pseudo code can be rather painful. Algoviz intends to provide help with this. Algoviz was developed with the intent to help ease the pain and even have fun learning algorithms.Algoviz approaches algorithms interactively to "play" algorithms to demonstrate how they artistically work. Users see the pseudo code for the algorithm and how it is executed step by step. Visualizing the algorithm to experiment with provides users a more constructive perspective of the algorithm. Additionally, users can add their own data to watch how the algorithm works, in real time, demonstrating what the algorithm does depending on their input data.Finally, Algoviz probably has its greatest strength in that it has taken DSA and made it fun instead of boring. It takes the abstract theory of DSA, makes it video game-like fun. Algoviz combines the visual learning space with the upstairs practice spaces to provide practically anyone - regardless of background - the ability to understand even the most complex algorithms, confidently.

SOFTWARE DESCRIPTION

- Frontend: HTML, CSS, JavaScript
- Backend:Flask
- Database:SqLite
- Quiz Interface:Tkinter

SCREENSHOTS

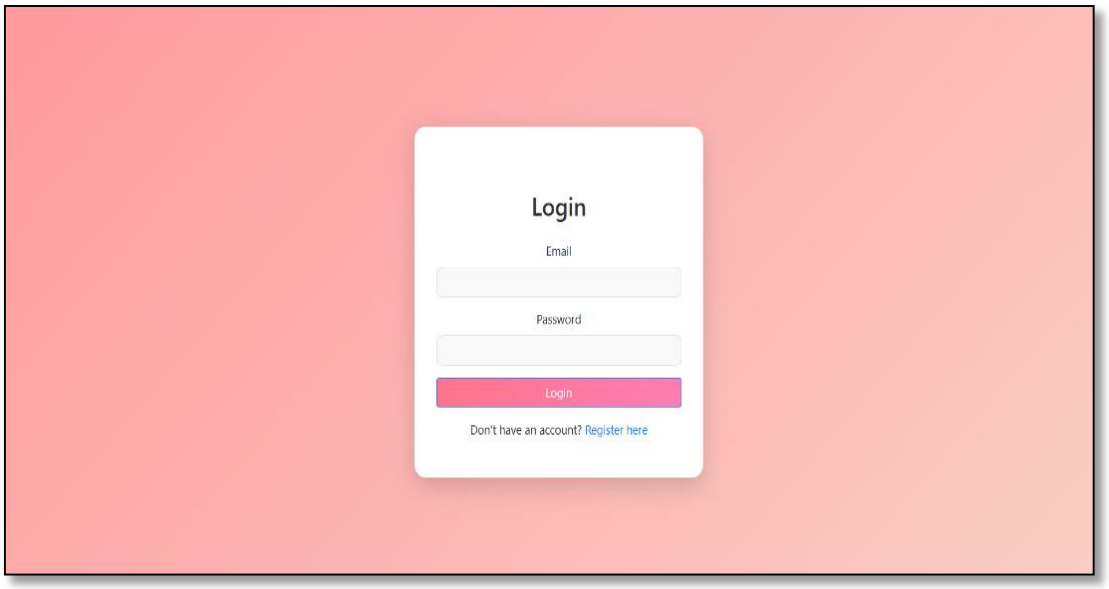

FIG 6..1.1 LOGIN PAGE OF ALGOVIZ

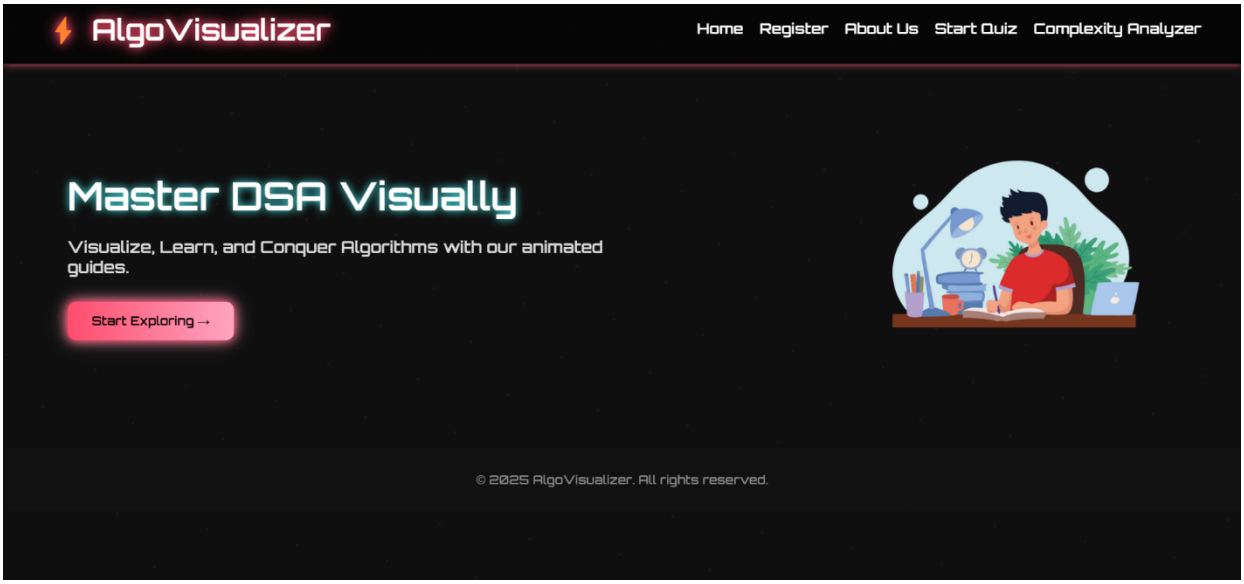

FIG 6.1.2 HOMEPAGE OF ALGOVIZ

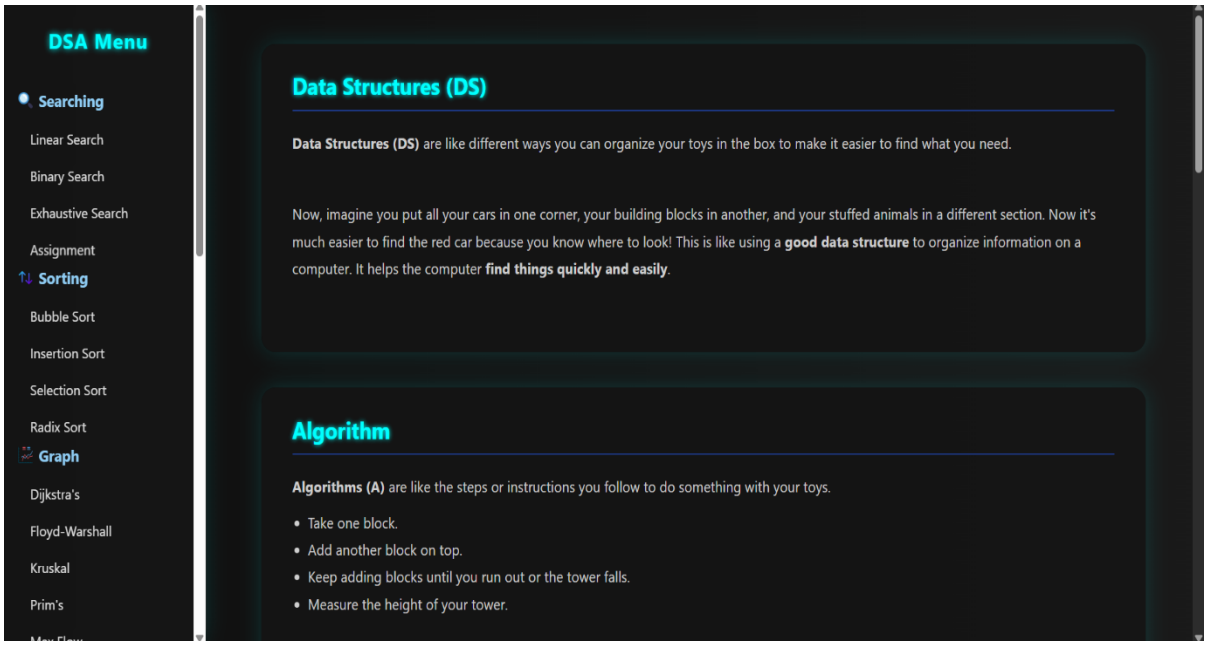

FIG 6.1.3 DATA STRUCTURES MENU

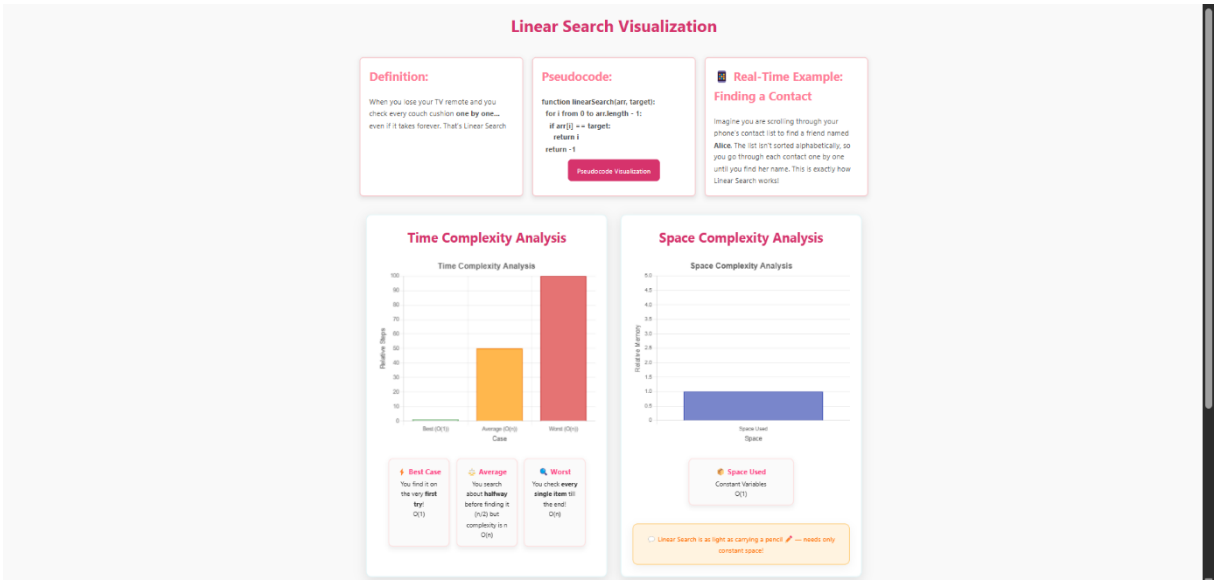

FIG 6.1.4 LINEAR SEARCH VISUALIZATION

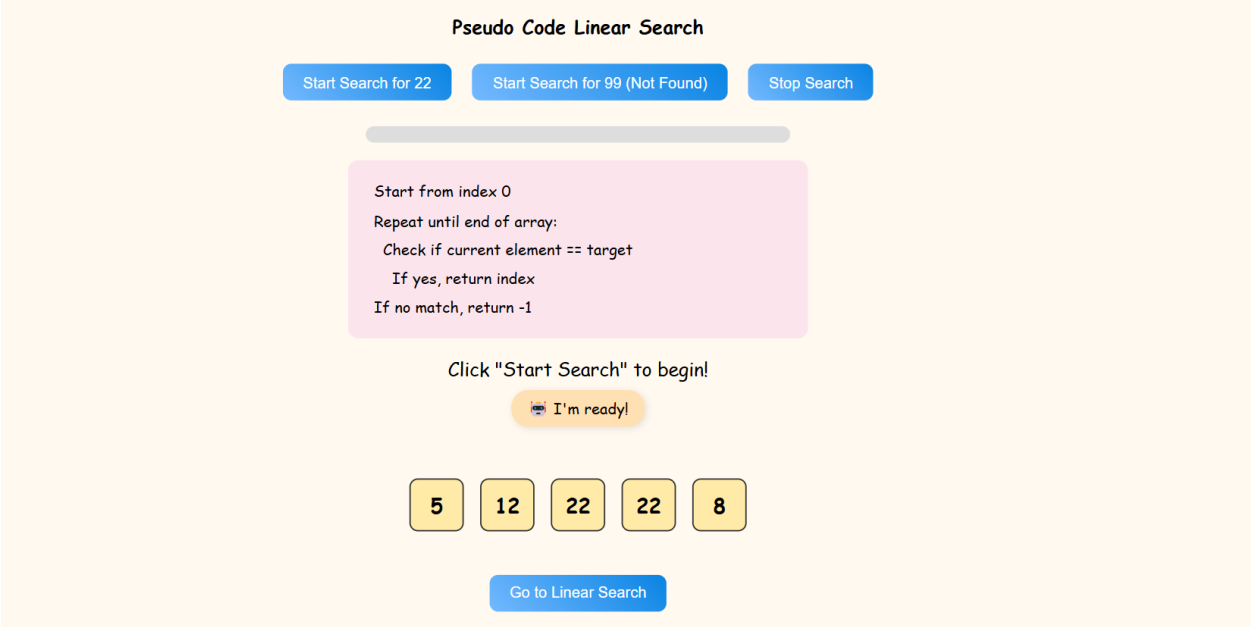

FIG 6.1.5 PSEUDOCODE VISUALIZATION

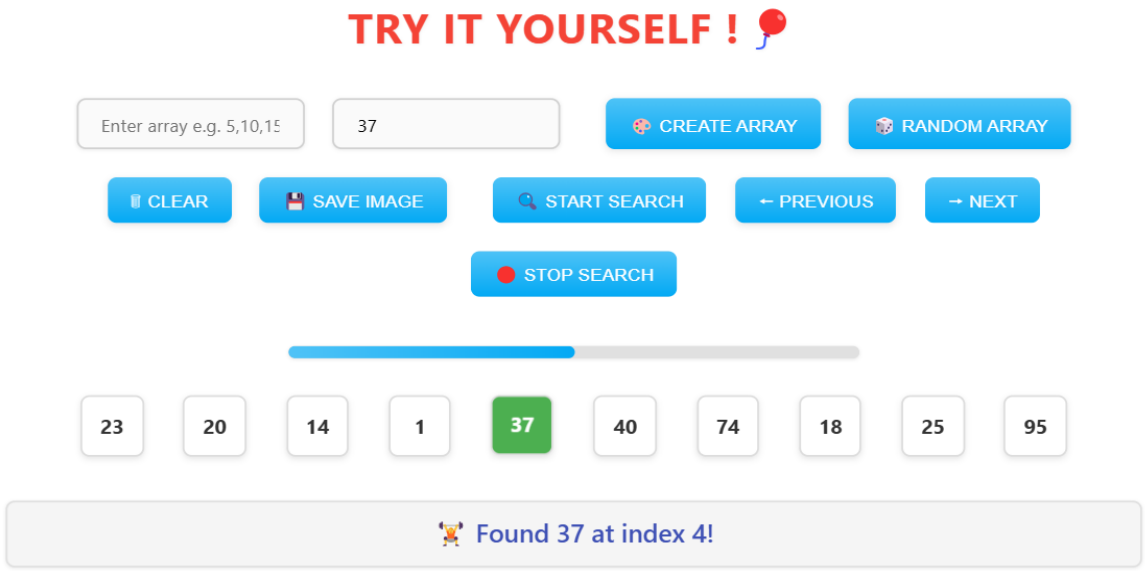

FIG .1.6 USER EXECUTION

CONCLUSION

This AlgoViz project has shown the effectiveness of the displaying complex algorithms as a means to improve understanding and learning. The project has done Algorithm visualization and algorithm interaction which are themselves dynamic and alive to help the users interact with different and complex algorithms in a more friendly and simpler manner.

- Here is what we were able to accomplish:
- Enhanced Algorithm Understanding: Users were able to reason better with some algorithms such as sorting, searching, and graph traversal when they were given the explained version implemented with visualization.
- Enhanced Learning Experience: Students greatly improved their understanding with the interactive features of the visualizations with respect to various parameters and were able to change the values with the results changing in real time.
- Practical Usefulness: The project has educational relevance in the real worlds for both expert and novice users for understanding the relevance of the complexity and provides them with the educational applicability.

Acknowledgements

I would like to express my sincere gratitude to my mentor, the department staff, and the Head of Department (HoD) for their invaluable guidance, support, and encouragement throughout the course of this project. Their expertise and constant assistance have been instrumental in the successful completion of this work.

REFERENCES:

1. Garg, K. (2021), "An Approach to Develop Web-Based Application for Simulation and Visualization of Operating System Algorithms," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 7, issue 2, ISSN 2456-3307, pp. 40–47.

2. Johnson, T., & Lee, H. (2022), "Comparative Studies on Sorting and Searching Algorithms with Visual Feedback," *International Journal of Educational Technology in Higher Education*, vol. 17, issue 2, ISSN 2365-9440, pp. 89–102.

3. Johnson, T., & Lee, H. (2022), "Comparative Studies on Sorting and Searching Algorithms with Visual Feedback," *International Journal of Educational Technology in Higher Education*, vol. 17, issue 2, ISSN 2365-9440, pp. 89–102.

4. Patil, R., Sharma, V., & Mehta, S. (2023), "Interactive Algorithm Visualization: Enhancing Learning through Visual Representations," *Journal of Computer Science Education*, vol. 19, issue 3, ISSN 1542-4048, pp. 211–225.

5. Patil, R., Sharma, V., & Mehta, S. (2023), "Interactive Algorithm Visualization: Enhancing Learning through Visual Representations," *Journal of Computer Science Education*, vol. 19, issue 3, ISSN 1542-4048, pp. 211–225.

6. Smith, L. (2024), "Optimizing Algorithm Performance: A Study on Visualization and Computational Efficiency," *Computer Applications in Engineering Education*, vol. 32, issue 1, ISSN 1099-0542, pp. 33–47.

7. Smith, L. (2024), "Optimizing Algorithm Performance: A Study on Visualization and Computational Efficiency," *Computer Applications in Engineering Education*, vol. 32, issue 1, ISSN 1099-0542, pp. 33–47.