Design and Developed Application for Algorithm Visualizer

Kshitija Shirke[1] Pratik Dhumal[2] Dharti Kshirsagar[3] Rutvik Wagh[4] Omkar Mhetras[5]
[1]Assistant Professor [2,3,4,5]UG Students
[1,2,3,4,5]Department of Information Technology
[1,2,3,4,5]SVPM's College of Engineering Malegaon(bk), Baramati, India

Abstract — In recent years, the popularity of algorithm visualization tools has grown significantly, enabling educators and students to comprehend complex algorithms through interactive visual representations. This abstract presents a novel approach to algorithm visualization through a Flutter-based platform, combining the power of Flutter's intuitive user interface with advanced algorithms to create an immersive and interactive learning experience. Our algorithm visualizer offers a seamless and engaging way to explore various algorithms, providing real-time visual feedback and step-by-step demonstrations. Leveraging Flutter's cross-platform capabilities, the visualizer ensures accessibility across multiple devices, fostering widespread adoption in educational settings. The user-friendly interface allows users to dynamically adjust input parameters, visualize algorithmic processes in real-time, and gain valuable insights into algorithm behaviours. By integrating cutting-edge algorithms with Flutter's flexibility, our visualizer empowers learners to grasp complex concepts with ease, making algorithm education more accessible and engaging for students and enthusiasts alike.

Keywords: Algorithm Visualizer, Sorting Algorithm Visualization, Pathfinding Algorithm Simulator, Interactive Algorithm Animation, Educational Algorithm Tool, Real-Time Algorithm Execution Etc

I. INTRODUCTION

In the rapidly evolving landscape of computer science and programming, understanding algorithms is paramount. Algorithms serve as the backbone of software development, solving complex problems and optimizing processes. Visualizing these algorithms not only enhances comprehension but also fosters a deeper appreciation for their intricacies. Introducing our innovative solution: a Flutter-based algorithm visualizer that revolutionizes the way algorithms are learned and understood. Leveraging the power of Flutter, a popular open-source UI toolkit, our visualizer offers an intuitive and interactive platform for programmers, students, and enthusiasts alike to explore algorithms in real-time. With its user-friendly interface and dynamic visualization capabilities, users can witness algorithms in action, grasp their underlying concepts, and experiment with different parameters, promoting a hands-on learning experience.

At the heart of our Flutter-based algorithm visualizer lies a rich collection of algorithms spanning various domains such as sorting, searching, graph traversal, and dynamic programming. Users can select specific algorithms, input their own data, and observe the step-by-step execution in a visually appealing and easy-to-understand manner. Real-time animations and colorful representations aid in demystifying complex algorithms, making the learning process engaging and enjoyable. Additionally, our visualizer supports customization, allowing users to modify algorithm parameters and input data on the fly, facilitating experimentation and deepening understanding.

The Flutter-based algorithm visualizer is a powerful tool designed to demonstrate the inner workings of various algorithms in a user-friendly and interactive manner. Utilizing Flutter's versatile framework, the visualizer provides a seamless and responsive user interface. Users can input different algorithms, such as sorting, searching, or graph traversal algorithms, and witness real-time visual representations of their execution. The application employs intuitive animations and graphics to illustrate the step-by-step process, aiding users in understanding complex algorithms with ease. This Flutter-based visualizer not only enhances algorithmic learning but also serves as a valuable resource for developers and students looking to grasp algorithm complexities visually and intuitively.

Furthermore, this Flutter-based visualizer is designed with educational institutions in mind, offering educators a powerful tool to enhance their teaching methodologies. By integrating this visualizer into the classroom, educators can illustrate abstract algorithms in a concrete and interactive way, catering to diverse learning styles. Students can collaborate, experiment, and gain valuable insights, thereby fostering a collaborative and dynamic learning environment. As we venture into this new era of algorithm education, our Flutter-based visualizer stands as a testament to the fusion of technology and education, empowering individuals to grasp the core concepts of algorithms effortlessly and inspiring the programmers and problem solvers of tomorrow.

II. LITERATURE SURVEY

A. Algorithm Visualizer: Its features and working

Barnini Goswami, Anushka Dhar, Akash Gupta and h Antriksh Gupta.

The main aims of these paper to engage the students by providing self-paced hands-on experience, fun filled games through mazes and patterns and interactive, perceivable visualizations for their better concept understanding of various algorithms. Our work presently focuses on path-finding, sorting and CPU scheduling algorithms.

B. Finding the shortest path in a graph and its visualization using C# and WPF

Radoslav, Metodi and Ivan

The presented work is an example of realizing and applying of Dijkstra's algorithm to find the shortest path between two vertices in a connected, undirected graph, which is often a solved problem at a time annual International Olympiad in Informatics. They only implemented a program that allows drawing an undirected graph, visualizing the shortest path between two vertices and finding its value.

C. E-learning Tool for Visualization of Shortest Paths Algorithms

Daniela Borissova and Ivan Mustakero

In this paper an e-learning tool for shortest paths algorithms visualization is described. The developed e-learning tool allows creating, editing and saving graph structure and visualizes the algorithm steps execution. It is intended to be used as a supplement to face-to-face instruction or as a stand-alone application. The conceptual applicability of the described e-learning tool is illustrated by implementation of Dijkstra algorithm. The preliminary test results provide evidence of the usability of the e-learning tool and its potential to support students' development of efficient mental models regarding shortest paths algorithms.

III. OBJECTIVE

Develop an interactive web-based platform to visualize various algorithms, enabling users to better understand their functionality and performance.

Provide a user-friendly interface that allows users to input algorithm parameters, data sets, and custom inputs for real-time visualization.

- Create a diverse library of algorithms spanning sorting, searching, graph traversal, dynamic programming, and more to cater to a wide range of educational and practical needs.
- Offer a step-by-step mode that animates each algorithm's execution, highlighting key steps and data transformations, aiding in comprehension.
- Implement a variety of visualization styles, including animations, charts, and graphs, to convey algorithmic concepts effectively.
- Allow users to adjust input sizes, algorithm speed, and visualization options to experiment and observe how algorithm behavior changes with different inputs.
- Enable the comparison of multiple algorithms side by side, fostering a deeper understanding of their relative efficiencies.
- Support educational purposes by providing explanations, pseudocode, and references for each algorithm.
- Incorporate debugging and error handling features to help users identify issues in their custom inputs or algorithm implementations.
- Ensure cross-browser compatibility and responsive design for accessibility on various devices.
- Continuously update and expand the algorithm library to include new algorithms and improvements based on user feedback and emerging trends.
- Foster a community of algorithm enthusiasts by allowing users to share visualizations, collaborate on algorithm development, and provide feedback on the platform's usability and features.

IV. ARCHITECTURE

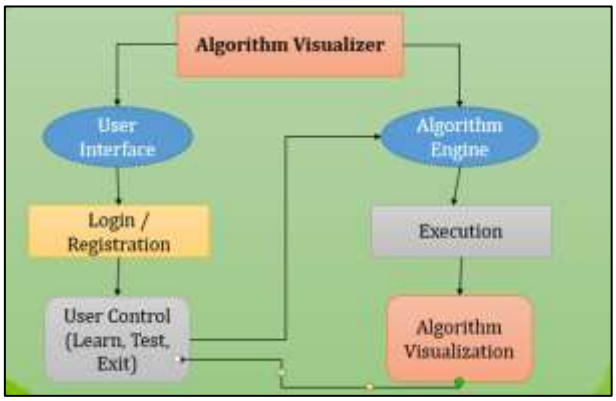

Fig. 1: System Architecture

V. PROPOSED SYSTEM

The proposed system represents a significant leap forward in algorithm visualization technology. It aims to provide a more immersive and interactive experience for users, allowing them to visualize algorithms in a dynamic and engaging manner. The proposed system intends to support a wide range of programming languages and algorithms, making it a valuable tool for learners, educators, and professionals alike. Additionally, the user interface is designed to be intuitive and user-friendly, ensuring that users of all skill levels can easily navigate and make the most of the visualizer. With advanced features such as real-time code execution, customizable visualization options, and collaborative coding environments, the proposed system promises to revolutionize the way algorithms are taught and learned, offering a comprehensive and user-centric platform for algorithm visualization.

VI. EXPECTED RESULT

As an experimental output, our system will gather typically aims to provide a graphical representation of how different algorithms work, allowing users to better understand their inner workings.

VII. CONCLUSION

In conclusion, the survey of a Flutter-based algorithm visualizer represents a significant advancement in the field of computer science education and algorithm understanding. By harnessing the power of Flutter, a versatile and user-friendly framework, we try to design a visualizer that provides an intuitive platform for students, developers, and enthusiasts to interactively explore complex algorithms and data structures. The seamless integration of interactive graphics and animations enhances the learning experience, making abstract concepts tangible and engaging. Additionally, the real-time visualization capabilities empower users to observe algorithms in action, fostering a deeper understanding of their behavior and performance. The application's cross-platform compatibility ensures accessibility across various devices, making it a valuable educational tool for a wider audience. As we move forward, this Flutter-based algorithm visualizer not only enriches the educational landscape but also serves as a testament to the innovative potential of mobile app

development, paving the way for future advancements in algorithm visualization and computational education.

ACKNOWLEDGEMENT

We take this opportunity to thank our project guide Prof. shirke k .k. and Head of the Department Prof. Dr.Gawade J.S. and Honorable Principal Prof. Dr.Mukane S.M. for their valuable guidance and for providing all the necessary facilities, which were indispensable in the completion of this project report. We are also thankful to all the staff members of the Department of Information Technology of SVPM's College of Engineering, Malegaon(Bk) for their valuable time, support, comments, suggestions and persuasion. We would also like to thank the institute for providing the required facilities, Internet access and important books.

REFERENCE

[1] Barnini Goswami, Anushka Dhar, Akash Gupta and h Antriksh Gupta (2023) "Algorithm Visualizer: Its features and working "978-1-6654-0962-9/21/$31.00 ©2021 IEEE
[2] Daniela Borissova and Ivan Mustakerov(2015) 'Elearning Tool for Visualization of Shortest Paths Algorithms', Research Gate.
[3] Neetu Goel and Dr. R.B. Garg(2012) 'A Comparative Study of CPU Scheduling Algorithms. ', International Journal of Graphics and Image Processing, Vol. 2, No. 4
[4] Radoslav, Metodi and Ivan(2020) 'Finding the shortest path in a graph and its visualization using C# and WPF', International Journal of Computers, Vol. 5
[5] Brian Faria(2017) 'Visualizing sorting algorithms', Rhode Island College
[6] Pedro Moraes and Leopoldo Teixeira(2019) 'Willow: A Tool for Interactive Programming Visualization to Help in the Data Structures and Algorithms Teaching-Learning Process', SBES 2019
[7] Q. Gao and X. Xu(2014) 'The analysis and research on computational complexity', The 26th Chinese Control and Decision Conference (2014 CCDC),pp.3467-3472. TihomirOrehovaˇcki (2012) 'ViSA: Visualization of Sorting Algorithms', Research Gate.
[8] Akoumianakis D. (2011). Learning as 'Knowing': Towards Retaining and Visualizing Use in Virtual Settings. Educational Technology & Society, 14 (3), 55-68.
[9] Ozyurt O., Ozyurt H., Baki A., Guven B. & Karal H. (2012). Evaluation of an adaptive and intelligent educational hypermedia for enhanced individual learning of mathematics: A qualitative study. Expert Systems with Applications, 39(15), 12092-12104.
[10] Nguyen V.A. & Yamamoto A. (2012). Learning from graph data by putting graphs on the lattice. Expert Systems with Applications, 39(12), 11172-11182.
[11] Karavirta V. (2007). Integrating Algorithm Visualization Systems. Electronic Notes in Theoretical Computer Science, 178(4), pp. 79-87.
[12] Seppala O. & Karavirta V. (2009). Work in Progress: Automatic Generation of Algorithm Animations for Lecture Slides. Electronic Notes in Theoretical Computer Science, 224, 97-103.
[13] Hundhausen C. D., Douglas S. A. & Stasko J. T. (2002). A meta-study of algorithm visualization effectiveness. Journal of Visual Languages and Computing, 13(3), 259-290.
[14] Fouh E., Akbar M. & Shaffer C. A. (2012). The Role of Visualization in Computer Science Education. Computers in the Schools, 29(1-2), 95-117.
[15] Roles J.A. & ElAarag H. (2013). A Smoothest Path algorithm and its visualization tool. Southeastcon, In Proc. of IEEE, DOI: 10.1109/SECON.2013.6567453.

