



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

# IJRSET

International Journal of Innovative Research in  
**SCIENCE | ENGINEERING | TECHNOLOGY**



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 4, April 2024

**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA

Impact Factor: 8.423

 9940 572 462

 6381 907 438

 [ijirset@gmail.com](mailto:ijirset@gmail.com)

 [www.ijirset.com](http://www.ijirset.com)

# Create & Develop the Algorithm Visualizer

Shirke K.K.<sup>1</sup>, Pratik Dhumal<sup>2</sup>, Dharti Kshirsagar<sup>3</sup>, Rutvik Wagh<sup>4</sup>, Omkar Mhetras<sup>5</sup>

Dept. of Information Technology, SVPM'S College Of Engineering Malegaon (Bk), Baramati, Pune, India<sup>1-5</sup>

**ABSTRACT:** Algorithm visualization tools have been increasingly popular in the last few years, making it possible for teachers and students to understand complicated algorithms using interactive visual aids. This abstract introduces a brand-new method for visualizing algorithms using a Flutter-based platform. By fusing sophisticated algorithms with Flutter's user-friendly interface, an immersive and dynamic learning environment is produced. Our algorithm visualizer provides step-by-step instructions and real-time visual feedback, making it an easy and entertaining method to experiment with different algorithms. By utilizing Flutter's cross-platform features, the visualizer guarantees accessibility on various devices, promoting extensive use in educational environments. Users can see algorithmic processes in real-time, dynamically modify input parameters, and obtain insightful knowledge about algorithm behaviors thanks to the user-friendly interface. Our visualizer helps learners understand difficult ideas more easily by combining state-of-the-art algorithms with Flutter's flexibility. This makes algorithm teaching more approachable and interesting for both hobbyists and students.

**KEYWORDS:** Algorithm Visualizer, Sorting Algorithm Visualization, Pathfinding Algorithm Simulator, Interactive Algorithm Animation, Educational Algorithm Tool, Real-time Algorithm Execution etc.

## I. INTRODUCTION

Comprehending algorithms is crucial in the quickly changing field of computer science and programming. Algorithms are the foundation of software development; they solve complicated issues and streamline procedures. Seeing these algorithms in action not only improves understanding but also cultivates a greater understanding of their complexity. We would like to present our creative solution, a Flutter-based algorithm visualizer that transforms the way people learn and comprehend algorithms. Our visualizer makes use of Flutter, a well-liked open-source UI framework, to provide programmers, students, and enthusiasts with an easy-to-use and engaging platform for real-time algorithm exploration. The dynamic visualization capabilities and user-friendly interface of the platform enable users to observe algorithms in operation, understand their fundamental ideas, and experiment with various settings, hence facilitating a practical learning experience.

Our Flutter-based algorithm visualizer is built on an extensive library of algorithms covering a wide range of applications, including graph traversal, sorting, searching, and dynamic programming. Users can input their own data, choose particular algorithms, and see the step-by-step execution in an aesthetically pleasing and easily comprehensible way. Colorful illustrations and real-time animations help to demystify complicated algorithms, which makes learning interesting and fun. Furthermore, our visualizer is customizable, enabling users to instantly add data and change algorithm settings, which promotes experimentation and enhances comprehension.

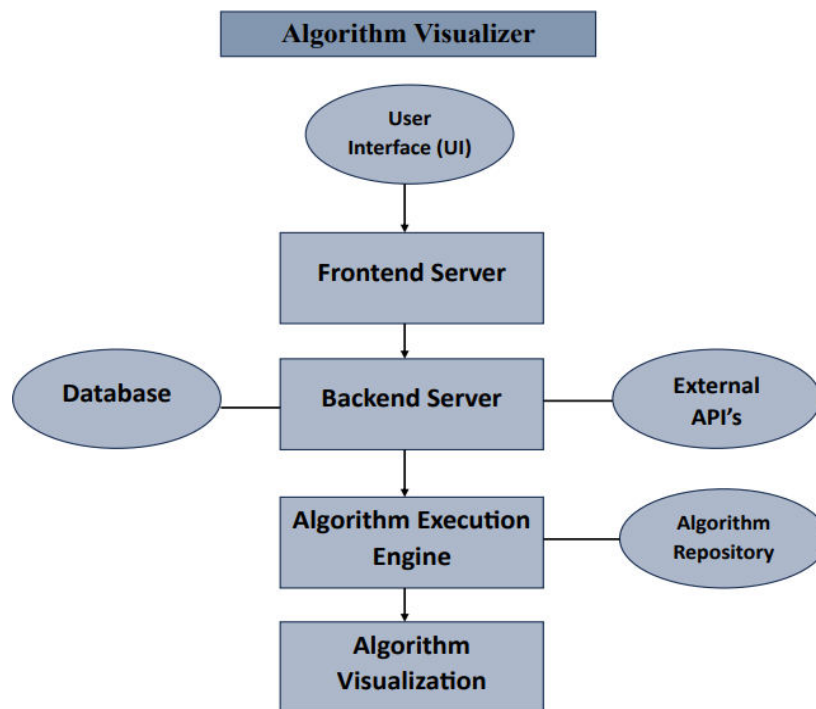
An effective tool for interactively and easily explaining the inner workings of different algorithms is the Flutter-based algorithm visualizer. By leveraging the flexible foundation of Flutter, the visualizer offers a smooth and adaptable user experience. Users can input various algorithms, including searching, sorting, and graph traversal methods, and see graphic representations of those algorithms being executed in real time. The application makes use of user-friendly animations and graphics to walk users through each step of the process, making it easier for users to understand complex algorithms. In addition to improving algorithmic learning, this Flutter-based visualizer is an invaluable tool for developers and students who want to understand algorithm difficulties intuitively and visually.

Additionally, this visualizer, which is built on Flutter, was created with educational institutions in mind, providing teachers with an effective tool to improve their teaching approaches. Educators may accommodate a variety of learning styles by using this visualizer to show abstract algorithms in a tangible and participatory manner in the classroom. Students can work together, try new things, and acquire important lessons, which promotes a dynamic and cooperative learning environment. Our Flutter-based visualizer is a testament to the marriage of technology and education as we enter this new era of algorithm education, enabling people to easily understand the fundamentals of algorithms and motivating the programmers and problem solvers of the future.

## II. RELATED WORKDONE

Significant progress has been made in the field of algorithm visualization in the last several years. The initial research in this field concentrated on using flowcharts and diagrams to represent algorithms in a static way. Later attempts, such as VisuAlgo and Algomation, have made use of interactive web-based platforms to produce dynamic visualizations. Furthermore, in order to improve users' comprehension and engagement, researchers have investigated the use of augmented reality (AR) and virtual reality (VR) to submerge users in algorithmic settings. Additionally, machine learning techniques have been used to automatically produce code-based representations. To further serve a wider range of practitioners and learners, recent work has tackled the difficulty of visualizing intricate, real-world data structures and algorithms. All things considered, the associated literature presents a varied and developing terrain of algorithmic visualization methods that accommodate different learning styles.

## III. SYSTEM ARCHITECTURE



## IV. METHODOLOGY

There are various processes involved in implementing an algorithm visualizer system, and the approach chosen will vary depending on the scale of the project, team preferences, and organizational requirements. An algorithm visualizer system's implementation process is broadly outlined below:

### 1. Requirements Analysis:

Define and gather requirements for the algorithm visualizer system.

Identify key features, user interactions, and performance expectations.

### 2. System Design:

Create a detailed system design that outlines the architecture, components, and interactions.

Design the user interface, specifying how algorithms will be visualized and how users will interact with the system.

**3. Choose Technologies:**

Select the programming languages, frameworks, and tools that are suitable for the project.

Consider technologies for frontend development (HTML, CSS, JavaScript) and backend development (Node.js, Django, Flask, etc.).

**4. Database Design (if applicable):**

If the system involves data storage (e.g., saved visualizations, user preferences), design the database schema.

Choose a database management system (e.g., MySQL, MongoDB) and plan data retrieval and storage.

**5. Prototyping:**

Develop a prototype or a minimum viable product (MVP) to validate the design and gather early feedback.

Focus on core functionalities and the visualization aspect.

**6. Frontend Development:**

Implement the user interface, considering user experience (UX) and responsiveness.

Integrate visualization components and ensure a smooth interaction with algorithms.

**7. Backend Development:**

Implement the backend logic to handle user requests, process algorithms, and manage data if applicable.

Develop APIs (Application Programming Interfaces) for communication between frontend and backend components.

**8. Algorithm Implementation:**

Implement the algorithms that users will visualize.

Ensure the algorithms can be executed step-by-step, and their state can be visualized at each step.

**9. Integration:**

Integrate the frontend and backend components to create a cohesive system.

Ensure data flow and communication between different parts of the system.

**10. Testing:**

Perform unit testing to validate the functionality of individual components.

Conduct integration testing to ensure smooth interactions between different modules.

Implement user testing to gather feedback on the usability and effectiveness of the visualization.

**11. Refinement and Iteration:**

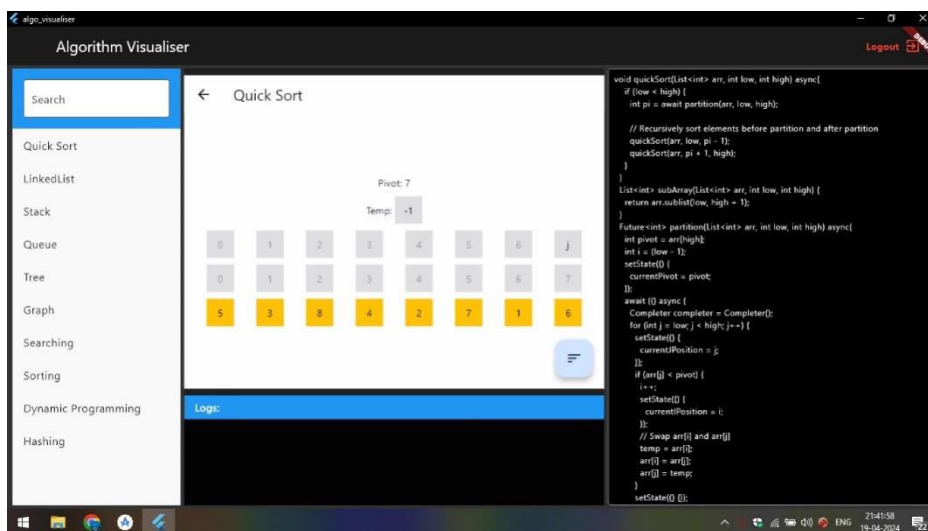
Refine the system based on user feedback and testing results.

Iteratively improve the implementation, addressing any identified issues.

**12. Documentation:**

Create comprehensive documentation for the system, including user guides, developer documentation, and any relevant technical documentation.

**V. RESULTS**



## VI. 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.

## REFERENCES

- [1] Barnini Goswami, Anushka Dhar, Akash Gupta and 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, Metodí 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. Tihomir Orehovački (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.



INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  [ijirset@gmail.com](mailto:ijirset@gmail.com)



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details