

# Graph Traversal and Path Finding Visualizer

A PROJECT REPORT

*Submitted by*

**Sameer Bid - 719**

*in partial fulfilment for the award of the degree*

*of*

**M.Sc. Data Science and Artificial Intelligence – Part 1**



**RAMNIRANJAN JHUNJHUNWALA COLLEGE OF ART'S,  
SCIENCE & COMMERCE (AUTONOMOUS),  
GHATKOPAR (W)**

April – 2023



**RAMNIRANJAN JHUNJHUNWALA COLLEGE OF ART'S,  
SCIENCE & COMMERCE (AUTONOMOUS), GHATKOPAR  
(W)**



*(Affiliated to University of Mumbai)*

**Certificate**

*This is to certify that the Project entitled **Graph Traversal and Path Finding Visualizer** is bonafide work of **Sameer Bidi** bearing Seat No **719** submitted in partial fulfilment of the requirements for the award of Degree **Master of Science** in **Data Science & Artificial Intelligence**.*

**Signature of Internal Guide**

**Signature of Co-ordinator**

**College Seal and Date**

**Signature Examiner**

---

## Abstract

This web-based tool provides users with an interactive platform to visualize the working of pathfinding algorithms such as BFS, DFS, etc. The tool uses SVG for drawing and animating the graph, while the algorithms are implemented in Python.

Users can add nodes and edges to the graph, select a pathfinding algorithm and run it, and customize the graph as per their requirements. The visualization provides users with a better understanding of how the algorithms work and how they can be used to solve real-world problems.

This project aims to help students, researchers, and developers understand the implementation and working of pathfinding algorithms in a more visual and interactive manner.

## **Acknowledgement**

Before we get into thick of things, I would like to add a few heartfelt words for the people who were part of Graph Traversal and Path Finding Visualizer project in numerous ways, people who gave unending support right from the stage the project idea was conceived.

A project report is such a comprehensive coverage; it would not have been materialized without the help of many. The four things that go on to make a successful endeavour are dedication, hard work, patience and correct guidance. Able and timely guidance not only helps in making an effort fruitful but also transforms the whole process of learning and implementing into an enjoyable experience.

In particular, I would like to thank our Mentor/Director Dr. (Mrs.) Usha Mukundan, R.J. College. I would like to give a very special honour and respect to our teacher, Prof. Mujtaba Shaikh who took keen interest in checking the minute details of the project work and guided us throughout the same. A sincere quote of thanks to the non-teaching staff for providing us software their time. I appreciate outstanding co-operation by them, especially for the long Lab timings that we could receive.

## Declaration

I hereby declare that the Project entitled, “*Graph Traversal and Path Finding Visualizer*” done at R. J. COLLEGE, Ghatkopar(W), Mumbai, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my Knowledge other than me, No one has submitted to any other University. The Project is done in partial fulfilment of the requirements for the award of degree of *Master of Science Data Science and Artificial Intelligence* to be submitted as mini project as part of our curriculum.

# Index

• Abstract	3
• Acknowledgement	4
• Declaration	5
• Index	6
1. Introduction	7
2. Problem Statement	8
3. Methodology	9
4. Challenges	10
5. Output Image	11
6. Future Work	12
7. Conclusion	13
8. References	14

# Introduction

Pathfinding algorithms are widely used in various applications such as maps, robotics, and video games to find the shortest or fastest path between two points. However, understanding the implementation and working of these algorithms can be challenging, especially for beginners.

This web-based tool aims to provide users with an interactive and visual way to understand the implementation and working of pathfinding algorithms such as BFS, DFS, etc. The tool uses SVG for drawing and animating the graph, while the algorithms are implemented in Python.

Users can add nodes and edges to the graph, select a pathfinding algorithm and run it, and customize the graph as per their requirements.

The visualization provides users with a better understanding of how the algorithms work and how they can be used to solve real-world problems. This tool is designed to help students, researchers, and developers understand the implementation and working of pathfinding algorithms in a more visual and interactive manner.

By providing an interactive platform, users can experiment with different algorithms and learn from their mistakes in a safe and controlled environment.

The rest of this mini documentation will describe the features, user guide, technical details, and future developments of this web project.

## Problem Statement

Pathfinding algorithms are used to find the shortest or fastest path between two points, and they have numerous applications in various domains such as maps, robotics, and video games. However, understanding the implementation and working of these algorithms can be challenging, especially for beginners.

Traditional methods of learning through lectures or textbooks may not be sufficient to provide a clear understanding of these algorithms. Therefore, there is a need for a more interactive and visual approach to learning pathfinding algorithms.

To address this problem, this web-based tool provides an interactive platform to visualize the implementation and working of pathfinding algorithms such as BFS, DFS, etc. The tool uses SVG for drawing and animating the graph, while the algorithms are implemented in Python. Users can add nodes and edges to the graph, select a pathfinding algorithm and run it, and customize the graph as per their requirements. The visualization provides users with a better understanding of how the algorithms work and how they can be used to solve real-world problems.

This tool aims to provide a more interactive and visual approach to learning pathfinding algorithms, thereby making it easier for beginners to understand these algorithms. By providing an interactive platform, users can experiment with different algorithms and learn from their mistakes in a safe and controlled environment.



## Methodology

The web-based tool uses a combination of SVG for drawing and animating the graph, and Python for implementing the pathfinding algorithms.

The following steps outline the methodology used in the project:

1. **Graph creation:** Users can create a graph by adding nodes and edges to the canvas. The graph is represented using an adjacency matrix.
2. **Pathfinding algorithm:** Users can select a pathfinding algorithm such as BFS or DFS. When the algorithm is run, it returns the path between the start and end nodes.
3. **Animation:** The output of the pathfinding algorithm is used to animate the graph nodes to visualize the path. The path is highlighted by changing the color of the edges or nodes, depending on the algorithm used.
4. **Customization:** Users can customize the graph by changing the color, shape, or size of the nodes and edges. They can also change the starting and ending nodes.
5. **User interface:** The user interface is designed to be intuitive and easy to use. Users can interact with the canvas using mouse or touch inputs.
6. **Implementation:** The tool is implemented using HTML, CSS, and JavaScript for the frontend, and Python for the backend. The frontend communicates with the backend through a RESTful API.
7. **Deployment:** The tool is deployed on a web server using a cloud hosting service such as AWS or Heroku.

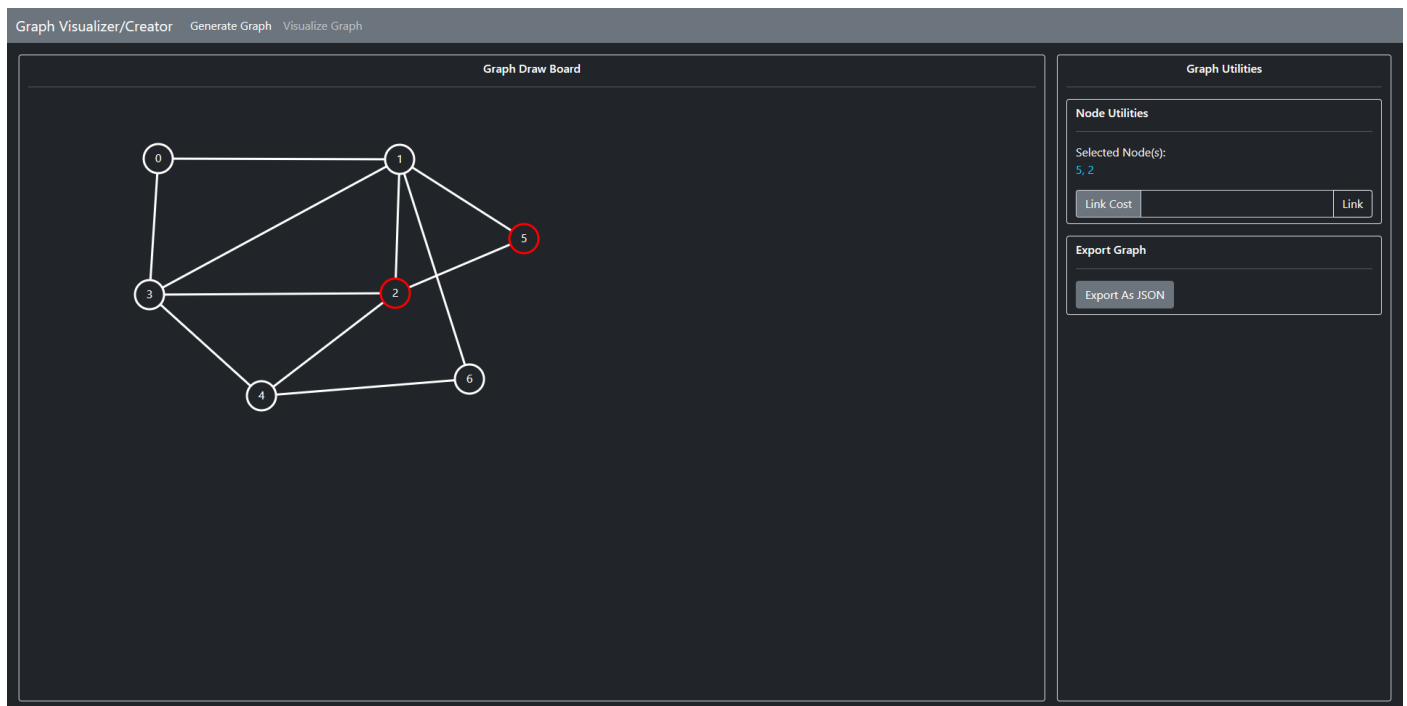
By using Python for the pathfinding algorithms and SVG for animating the graph, the tool provides a visual and interactive approach to learning pathfinding algorithms.

The use of a RESTful API allows for easy communication between the frontend and backend, making the tool responsive and efficient. The customization options and intuitive user interface make it easy for users to experiment with different algorithms and learn from their mistakes in a safe and controlled environment.

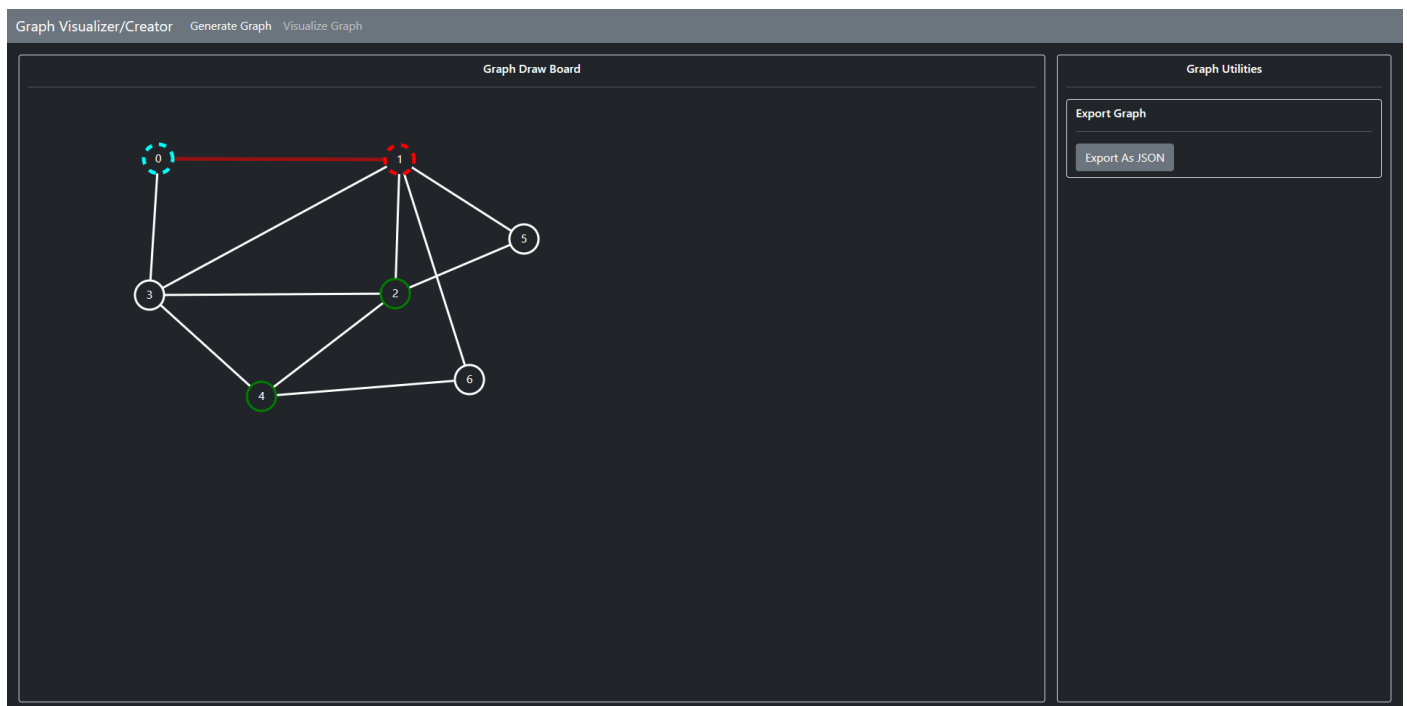
# Challenges

1. Algorithm efficiency:
  1. Pathfinding algorithms can be computationally intensive, especially when working with large graphs.
  2. Ensuring that the algorithms run efficiently and quickly is important to provide a smooth user experience.
2. User input validation:
  1. Users can create custom graphs by adding nodes and edges to the canvas.
  2. Ensuring that the user input is valid and error-free can be challenging, especially when dealing with complex graphs.
3. Cross-browser compatibility:
  1. The use of SVG for drawing and animating the graph may cause compatibility issues across different browsers.
  2. Ensuring that the tool works seamlessly across all major browsers can be challenging.
4. User interface design:
  1. Designing an intuitive and easy-to-use interface can be challenging, especially when dealing with complex functionality such as pathfinding algorithms.
5. Deployment and hosting:
  1. Deploying the tool on a web server and ensuring that it is hosted securely and efficiently can be challenging.
6. Handling errors and exceptions:
  1. The tool may encounter errors or exceptions, especially when dealing with complex user input or large graphs.
  2. Ensuring that the tool can handle these errors gracefully and provide useful feedback to the user is important for a good user experience.
7. Future developments:
  1. As new pathfinding algorithms are developed or as new features are requested by users, ensuring that the tool can be easily updated and expanded can be challenging.

# Output Image



The main interface consisting of a Draw Board where you can draw graphs and Graph Utilities which you can use to edit the Graph



Graph being animated

## Future Work

1. Additional algorithms:
  1. While the tool currently supports BFS, DFS, and other common pathfinding algorithms, there are many other algorithms that could be added in the future.
2. Algorithm comparison:
  1. Users could benefit from being able to compare the results and performance of different algorithms side-by-side.
  2. Adding a feature to compare the results of different algorithms could be a valuable addition to the tool.
3. Real-time graph editing:
  1. Currently, users must create the graph before running the pathfinding algorithm.
  2. Adding a feature to allow real-time editing of the graph could make the tool more dynamic and user-friendly.
4. Mobile optimization:
  1. While the tool is designed to work on desktop, optimizing the user interface for smaller screens could make the tool more accessible to a wider audience.
5. Localization:
  1. The tool is currently only available in one language. Adding support for multiple languages could make the tool more accessible to a wider audience.

## Conclusion

In conclusion, the web project on visualizing pathfinding algorithms provides a powerful tool for learning and experimenting with different pathfinding algorithms. The tool uses Python to run the algorithms and SVG for drawing and animating the graphs.

By providing a visual representation of the algorithms in action, users can gain a deeper understanding of how the algorithms work and how they can be applied to real-world problems.

While the project currently supports several popular pathfinding algorithms, there is still room for expansion and improvement. Future work could include adding additional algorithms, improving the user interface, and adding new features to make the tool even more useful and accessible to a wider audience.

This project demonstrates the power of combining programming and visualization to create an interactive and engaging learning experience. By providing a hands-on approach to learning about pathfinding algorithms, this tool has the potential to inspire the next generation of computer scientists and programmers.

## References

1. [SVG Tutorial – How to Code Images with 7 Examples \(freecodecamp.org\)](https://www.freecodecamp.org/svg/)
2. [A Guide to SVG Animations \(SMIL\) | CSS-Tricks - CSS-Tricks](https://css-tricks.com/a-guide-to-svg-animations-smil/)
3. [An Introduction to SVG Animation | Toptal®](https://www.toptal.com/designers/svg/introduction/)