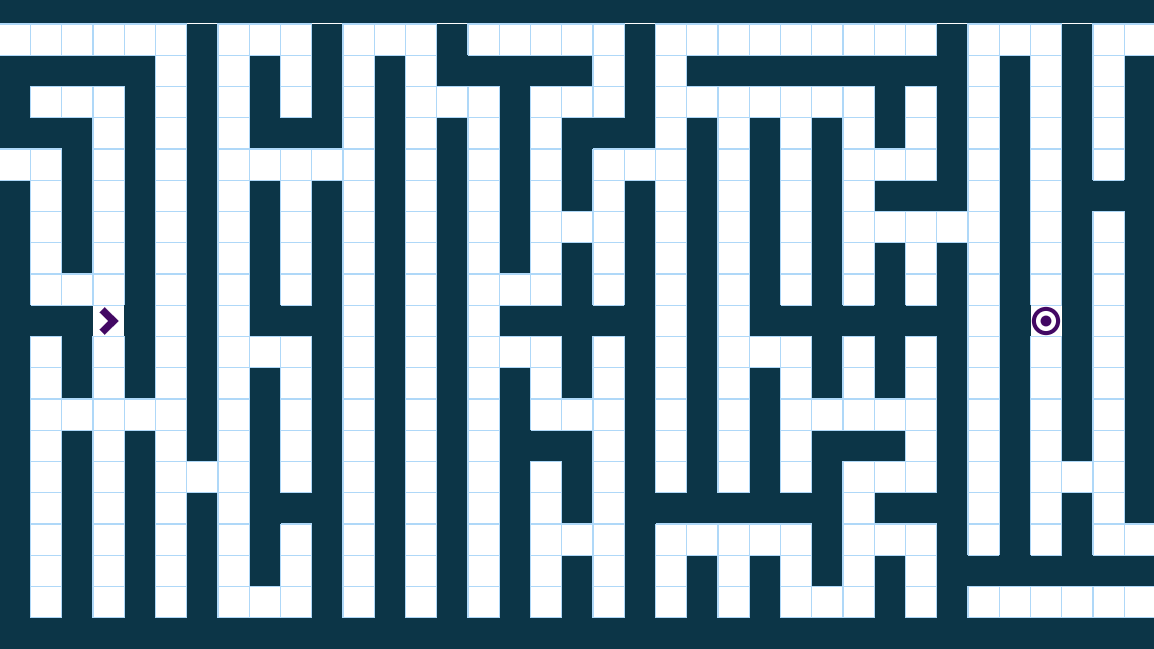
***Path and Sorting Visualizer:-***

**1). Path finding Visualizer:** - An easy to use Interactive Graph Path visualizer to implement searching algorithms in a 2-D grid between source and destination.

****

Source and destination in a 2-D Maze.

Features:-

# All the Basic and Advance Path Finding algorithms.

# Fast, Medium, Slow and Step Wise execution for better understanding of Algorithm.

# Customizable Grid system (Colors, Execution Speed, Start, Stop)

# Some of Maze Generation Algorithms included

# Random Maze Generator

# Implemented Path Finding Algorithms:-

# Dijkstra's Algorithm

# A\* Algorithm

# Greedy Best First Search

# Bidirectional Greedy Search

# Breadth First Search

# Depth First Search

# Swarm Algorithm

**2) Sorting Visualizer: -** A sorting algorithm is an algorithm that puts the elements of a list in a certain order. While there are a large number of sorting algorithms, in practical implementations a few algorithms predominate. In this implementation of sorting visualizer, we'll be looking at some of these sorting algorithms and visually comprehend their working. The sorting algorithms covered are-:

\* Selection Sort,

\* Insertion Sort,

\* Bubble Sort,

\* Merge Sort,

\* Quick Sort

\* Heap Sort.

You can randomize the list of numbers and select any type of sorting algorithm to call on the list from the given options. Here, all sorting algorithms will sort the elements in ascending order. The sorting time being visualized for an algorithm is not exactly same as their actual time complexities. The relatively faster algorithms like Merge Sort, etc. have been delayed so that they could be properly visualized.

Technologies which we will use:-

HTML, CSS, JavaScript, Python.

Framework which we will use:-

1. Node.js(Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser) or

# **Django** (Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design.)