# Algorithm Visualizer

submitted in fulfilment of the requirements for

Major-Project Sem-VII

by

**Mr. Darshan Chanchad (08)**

**Mr. Jigar Darji (12)**

**Mr. Harsh Moradiya (70)**

**Mr. Nikunj Rupavatiya (112)**

Supervisor:

**Dr. Jitendra Saturwar**

**Department of Computer Engineering**

**VIDYA VIKAS EDUCATION TRUST'S**

**UNIVERSAL COLLEGE OF ENGINEERING**

**KAMAN, VASAI - 401208**

**UNIVERSITY OF MUMBAI**

**2022-2023**

# Vidya Vikas Education Trust's

# Universal College of Engineering, Vasai (E)

Department of Computer Engineering

## CERTIFICATE

This is to certify that, the Major Project entitled **"Algorithm Visualizer"** is the bonafide work of **Mr. Darshan Chanchad (08), Mr. Jigar Darji (12), Mr. Harsh Moradiya (70) and Mr. Nikunj Rupavatiya (112)** submitted to the University of Mumbai in fulfilment of the requirement for the Major Project-I SemesterVII project work of B.E. Computer Engineering at Universal College of Engineering, Vasai, Mumbai at the Department of Computer Engineering, in the academic year 2020-2021,Semester – VII

Dr. Jitendra Saturwar

**Supervisor**

Dr. Jitendra Saturwar

**Head of Department**

Dr. J.B. Patil

**Principal**

# Major Project-I Report Approval for B. E.

This project report entitled “Algorithm Visualizer” by Mr. Darshan Chanchad (08), Mr. Jigar Darji (12), Mr. Harsh Moradiya (70) and Mr. Nikunj Rupavatiya (112) approved for the Major Project-I Semester VII project work of B.E Computer Engineering at Universal College of Engineering, Vasai, in the academic year 2020-2021.

**Internal Examiner** **External Examiner**

Date:

# Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

**Mr. Darshan Chanchad (08)**

**Mr. Jigar Darji (12)**

**Mr. Harsh Moradiya (70)**

**Mr. Nikunj Rupavatiya (112)**

Date:

# Abstract

Even though they are a hard subject, algorithms serve as a student's basis for computational thinking and programming abilities, as we have seen throughout the years. So, the project idea was developed to lessen the difficulties faced by pupils. Algorithm Visualizer is an engaging and interactive programme for kids. It provides the students with practical experience using the algorithms. It encourages their imagination to help them comprehend while also assisting teachers in helping their pupils comprehend. With our three learning speeds—slow, average, and fast—every learner may study at their own rate with this project. This interface is intended to encourage concentration and complete engagement. For our project, we used JavaScript as the main language and React.js as the framework. The goal of this project is to transform studying into an amazing experience that inspires students to want to learn even more.

***Keywords- Visualization, Algorithms, e-learning, path finding, sorting***

# Contents

# List of Figures

# List of Tables

# List of abbreviations

1. AVs – Algorithm Visualizer
2. HTML – Hyper text markup language
3. UI – User Interface
4. IEEE - Institute of Electrical and Electronics Engineers
5. RAM – Random Access Memory
6. IDE – Integrated Development Environment
7. DFD – Data Flow Diagram
8. UML – Unified Modelling Language
9. JS- Javascript

# Chapter 1

# Introduction

Nowadays sorting algorithms are widely used in computer software. For example, if you open file explorer on your PC, you may see files sorted in different ways. Searching in sorted data is more efficient than in not sorted ones. Students of computer science start learning different algorithms in the first year of studies and sorting algorithms are among them. The main goal of the thesis was to create a program which would serve as a tool for understanding how most known sorting algorithms work. There was an attempt to make the best possible user experience

## 1.1 Project Idea

We developed a method of learning through visualization and hand-on experience over different searching and sorting algorithms which is bound to help the students and teachers. Good visualizations bring algorithms to life by graphically representing their various states and animating the transitions between those states, especially dynamic algorithm visualization which shows a continuous, movielike presentation of an algorithm’s operations. Visualization allows the human visual system to extend human intellect; we can use it to better understand these important conceptual processes, other things, too. Also, we are well aware of the fact that the more we do things ourselves and engage the more we tend to learn about a particular topic. Thus, engaging in various game like activities can surely help the users get a hold on the topics. This technique is quite simple to explain to someone in conversation, but more advanced sorting algorithms, such as Quick Sort, which requires the data to be moved around a pivot point, are not easy to grasp using text alone. I wanted the animation to appeal to a wide spectrum of individuals utilizing various technology media, and so I had it made in a web-based format. Instead of requiring the user to install extra software or attempt to organize setups to use the tool, this helps to remove this source of anxiety. It uses HTML5 (Hypertext Markup Text Language) JavaScript, and CSS for the website's layout (Cascading Style Sheets).

# Chapter 2

# Review of Literature

A literature survey was carried out to find various papers published in international journals such as IEEE etc. related to algorithm visualizer to identify the implementation of algorithms and their visualization.

## 2.1 Existing System

Java was the most common implementation technology for AVs until recently, but with the adoption of HTML5 and JavaScript, the importance of Java on the web is diminishing. In the past few years, HTML5-based AV systems have been developed. Let see the some existing AVs, JAWAA [1] is a language for writing animations as well as a system for visualizing those animations. Animations can include graphical primitives, as well as some data structures. TRAKLA2 [2] is a learning environment that includes algorithm simulation exercises. The exercises require students to construct an AV by simulating the steps for an algorithm. The system supports multiple data structures, has several exercises included, automatically assesses student solutions, and gives visual feedback. VisuAlgo [3] is an online collection of AVs. Users can learn algorithms by seeing their execution (in case of, for example, sorting algorithms) as a slideshow, or by exploring operations on a data structure (for example, a binary heap). An interesting feature of the site is the online quiz system, which generates questions about the content for the learner to answer. In many of the questions, the answer is provided by direct interaction with a data structure. The data structures supported include arrays, trees, graphs, and lists.

After studying all this existing system we noticed that the systems were made in java and now the the poplar is javascript and html5 and two system were not having the speed control that changes the speed of animations in it is not there and also the animation that is no that much smooth so this all this things we will be added in our Algorithm visualizer. It will contain many basic and advance level algorithms.

## 2.2 Literature Survey

We have examined various research papers in the domain of AVs for our project to delve deeper into the details of the various researches conducted in the field of AV. Table 2.1 shows survey of the research paper done for the project.

Table 2.1 – Literature Survey table

| Sr. No. | Paper Name | Year of Publication | Author | Publication | Proposed Work | Research Gap |
|---|---|---|---|---|---|---|
| 1. | Algorithm Visualization System for Teaching Spatial Data Algorithms | 2010 | Jussi Nikander, Juha Helminen, and Ari Korhonen | JITE | In this they have only done the work on spatial algorithms. They have made the software that provides algorithm simulation exercises that can be automatically graded. | This is the research we are going to further enhance and implement in our project. Here, the gap is only the software is not online available it has to be downloaded and visualization of transition is not done. |

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | ALGORITM VISUALIZR | 2021 | Barnini Goswami, Anushka Dhar, Akash Gupta, Antriksh Gupta | IRJMETS | The proposed work was very user friendly and easy to understand by the student. The project has main two algorithm path finding and sorting algorithm. | This project has very short range of algorithms only two have been used i.e path finder and sorting not wide variety of algorithms is used in it, the project is very small. |
| 3 | Using algorithm visualizations in computer science education | 2015 | Slavomír Šimoˇnák | CEJCS | the paper is discussing the possibility of enriching the standard methods of teaching algorithms, with the algorithm visualizations. | A model that could be implemented in modern day facial recognition systems but needs a lot of improvisation in tracing and face recognition models. |

| | | | | | | |
|---|---|---|---|---|---|---|
| 4 | Design and evaluation of a web-based dynamic algorithm visualization environment for novices | 2013 | Euripides Vrachnos, Athanassios Jimoyiannis | ScienceDirect | The research shows the web based dynamic algorithm visualizer designed to for understanding to secondary students. | The research is not quite specific in the field where it is about what algorithms ŧuses. Although,there are many algorithms can be used. the research is quite restricted to the usage of only two algorithms. |
| 5 | Web-Based Dynamic Algorithm Visualizer Along with Chat Engine SDK for Neophyts | 2021 | Arijit Goswami, Biswarup Bhattacharjee, Rahul Debnath, Ankita Sikder | IJRESM | This research showcases that the web-based algorithm visualizer is made and for support the chat function is also added in it for asking the question related to the time complexity and etc. | This research fails to establish a point of visualization, the animation in this is not done smoothly and application is more over towards the chatting. The sorting algorithm is added no other algorithm. |

| | | | | | | |
|---|---|---|---|---|---|---|
| 6 | Algorithm Visualization - Modern Web-based Visualization of Sorting and Searching Algorithms | 2017 | Ashish Kumari, Manav Mittal, Vipul Jha, Abhishek Sahu, Manish Kumar, Neeti Sangwan and Navdeep Bohra | AAMS | This paper states the web based online algorithm visualizer in which it has searching and sorting algorithm and compares the different algorithm based on time and space complexity. | This paper has failed to added the other algorithms it have only added the sort algorithms and search algorithms. The vizualization is good but only used for limited algorithms. |
| 7 | Visualizing Algorithms Over the Web with the Publication-Driven Approach | 2001 | Camil Demetrescu, Irene Finocchi , Giuseppe Liotta | Crossmark | According to the publication-driven approach, algorithms run on a developer ’s remote server and their data structures are published on blackboards held by the clients. | The whole system is controlled by the teacher in which there is no interaction of students. There is no animation it, running like given input to program and it generates the input. |

| | | | | | | |
|---|---|---|---|---|---|---|
| 8 | Algorithm Visualization | 2019 | Armine Khachatryan | IEEE | The interface was created with the java in which he added the algorithms like Counting Inversions, Kruskal's Minimum/Maximum Spanning Tree, and Steiner Tree | The research proves to be very effective but the algorithm used for visualization are of high level not the basic ones like sorting, path finder nqueen , etc. |

In this paper presented by Jussi Nikander, Juha Helminen, and Ari Korhonen, they introduce a novel learning environment for spatial data algorithms, SDA-TRAKLA2, which has been implemented on top of the TRAKLA2 system. Spatial data items are identified by a set of coordinates, such as x and y for two-dimensional spatial data. The spatial environment contains new visualizations for representing spatial data, and a number of new exercises that cover a variety of spatial data algorithms. [1].

In the paper presented by Barnini Goswami, Anushka Dhar, Akash Gupta & Antriksh Gupta, our application Algorithm Visualizer is both interactive and alluring to students. It gives the students hands on experience of the algorithms' implementation. It feeds into their imagination to help them get a better understanding while also helping teachers to help make their students understand better. Through this project every student can learn at their own pace with our three speeds of learning: slow, average and fast. This interface is designed to make one feel fully engaged and concentrated. We have made use of React.js as framework and JavaScript as primary language for our project. [2].

In the paper presented by Slavomír Šimoˇnák, they have discussed the possibility of enriching the standard methods of teaching algorithms, with the algorithm visualizations. As a step in this direction, we introduce the VizAlgo algorithm visualization platform, present our practical experiences and describe possible future directions, based on our experiences and exploration

performed by means of a simple questionnaire. [3].

In This paper Euripides Vrachnos, Athanassios Jimoyiannis presents DAVE, a web-based dynamic algorithm visualization environment designed to support secondary education students' learning about basic algorithms. DAVE facilitates students' experimentation with array algorithms by allowing the modification of both code and data. The presentation of preliminary results, obtained from an evaluation study, provided evidence of the usability of the system and its potential to support students' development of efficient mental models regarding basic array algorithms [4].

In the paper presented by Arijit Goswami, Biswarup Bhattacharjee, Rahul Debnath, Ankita Sikder, it presents main purpose of this paper is to ensure effective and reliable methods of detecting various algorithms. Using this web application anyone can learn algorithms quickly and easily. Such applications are already available, but the efficiency of the available algorithm visualizers is not achieved thoroughly. This newly developed application proposes to take a step further and visualize all types of algorithms along with a customized chat engine SDK that enhances visual accuracy on a much larger scale. [5].

In the paper presented by Faizan Ahmad, Aaima Najam and Zeeshan Ahmed, it evaluates various face detection and recognition methods, provide complete solution for image-based face detection and recognition with higher accuracy, better response rate as an initial step for video surveillance. Solution is proposed based on performed tests on various face richdatabases in terms of subjects, pose, emotions, race and light. The actual advantages of face- based identification over other biometrics are uniqueness and acceptance. As human face is a dynamic object having high degree of variability in its appearance, that makes face detectiona difficult problem in computer vision [6].

In the paper presented by Ashu Kumar, Munish Kumar and Amandeep Kaur, states face detection is a computer technology that determines the location and size of a human face in a digital image. Face detection has been a standout amongst topics in the computer vision literature. This paper presents a comprehensive survey of various techniques explored for face detection in digital images. Different challenges and applications of face detection are also presented in this paper. At the end, different standard databases for face detection arealso given with their features. Furthermore, special discussions on the practical aspectstowards the development of a robust face detection system are done and this paper concludes with several promising directions for future research [7].

In the paper presented by Armine Khachatryan, created an algorithm visualization (AV) tool for CSUN faculty members, students, and anyone else who is studying algorithms to use. This AV tool helps instructors focus less on drawing complicated diagrams or graphs in class and focus more on explaining the actual logic behind the algorithm. AVs are more helpful when they are user-friendly and interactive. I came across several AV tools that were not very effective in my opinion because they used a more technical approach, so I decided to recreate some of them. The algorithms I created AVs of are Counting Inversions, Kruskal's Minimum/Maximum Spanning Tree, and Steiner Tree. [8].

## 2.3 Problem Statement and Objective

The most proven method to learn any algorithm is visualizing them. Especially for subjects like Data Structures, where algorithms form its basis, it is not practical to memorize them theoretically without comprehending them practically. It becomes quite confusing and tedious to visualize them on our own since abstract thinking plays a crucial role in forming its conceptual model in our mind. Thus, to develop a website so that students can learn various data structures and algorithms with the help of visualization

The main objective of the thesis project is to create a web application as a visualization tool. A single-page web application built using modern JavaScript technology that will visualize the flow and logic of various sorting algorithms. The UI will contain options to select one of the sorting algorithms which were implemented and several items or elements in the data array, control buttons to start, pause, navigate to previous or next steps along with an option for sorting speed and color mode. The data array of the selected size will be filled in with randomly generated unique values. The data set is represented as a vertical bar with the height of their respective values. After the sorting is started, the stepwise arrangement of data in ascending order based on their value/height will be visualized in the UI.

## 2.4 Project Scope

A visualization tool for visualizing some basic geometric algorithms along with data structure algorithms and operations associated with them has been presented. This tool provides an easy way to play and learn data structure concepts with its user-friendly and self-explanatory interface. In this system, only some commonly used and basic algorithms are implemented like arrays, queues, stacks, linked lists, linear and binary search tree, various sorting methods etc. Its scope can be extended by implementing more complex algorithms in the software. It can also be categorized for a more systematic interface. Developing and implementing a mechanism for the software package to recognize the user- defined observable data structures, and leave the implementation to the user is yet another way to extend its current scope, allowing users to use their own observable data structures, thus adding more flexibility to the software.

# Chapter 3

# Proposed System

The proposed system involves the simulation of the different type of sorting algorithms codes. The scope has its limitations. Only 6 types of sorting algorithms codes are created which are bubble sort, insertion sort, selection sort, heap sort, merge sort and quick sort. Once the synthesize and simulation of the codes for the Website have been run, the following animations will show how successfully data sets from distinct unsorted data can be sorted using distinct algorithms.

## 3.1 Analysis/Framework/ Algorithm

### 3.1.1 Sorting Algorithms

Sorting algorithm is an effective algorithm in computer science. It carries out a vital undertaking that puts data of a list in a specific request or organizes a compilation of items into a specific request. Sorting data has been produced to orchestrate the array values in different routes for a database. For example, sorting will dictate an array of numbers in ascending or descending order. Usually, it sorts an array into ascending or descending order. Most basic sorting algorithms include two stages which are compare two items and swap two items or copy one item. It will keep on executing until the data has been fully arranged

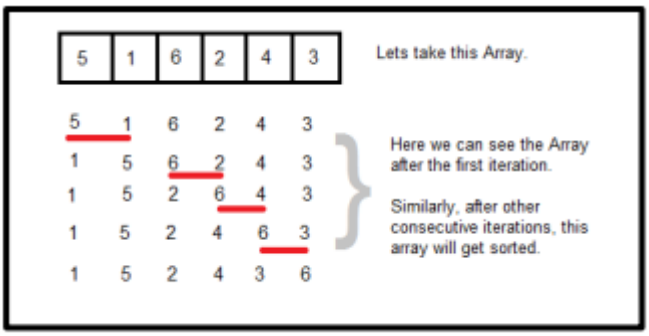



Figure.3.1 – Example for sorting algorithm.

### 3.1.2 Bubble Sort

Bubble sort is a casual and common sorting algorithm. It frequently compares the pairs of adjoining elements and swap the element when they are in reversed order. The algorithm passes through the list until the entire list has been sorted. Bubble sort is quite inefficient for sorting large data but it is stable and adaptive

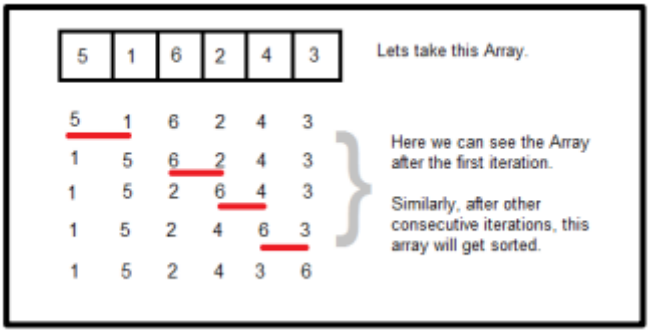



Figure.3.2 – Example for bubble sort.

### 3.1.3 Quick Sort

Quick sort as the name suggests, it is a fast sorting algorithm. It is very fast and requires less additional space but it is not a stable search. Quick sort takes a divide and conquer approach to sorting lists. It divides the list of elements to be sorted into 2 segments and after that call the quick sort strategy recursively to sort the 2 segments. For example, divide the problem into 2 smaller segments and overcome by understanding the smaller segments.

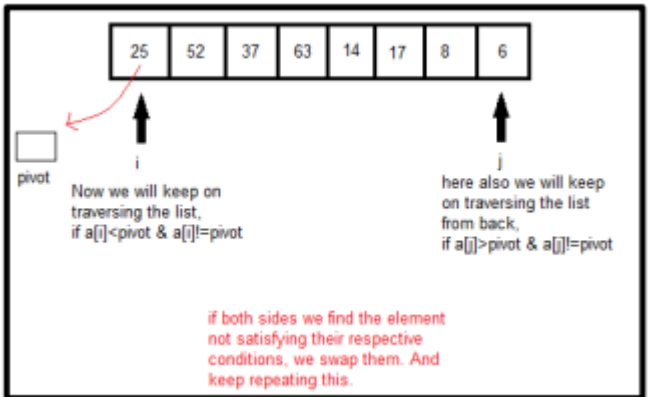



Figure.3.3 – Example for bubble sort.

### 3.1.4 N-Queens Problem

The N-queens is a problem to place N queens on an N N chess board such that no queen can attack another. For example, the 6-queens problem has four solutions, as shown in Figure 1. This problem is commonly used as a benchmark program [1, 2] for computer systems and as an example in computer

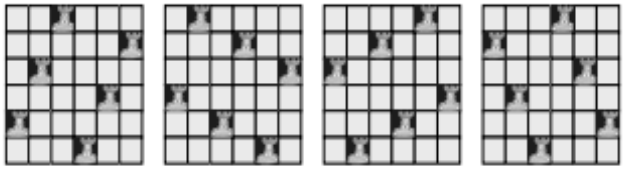
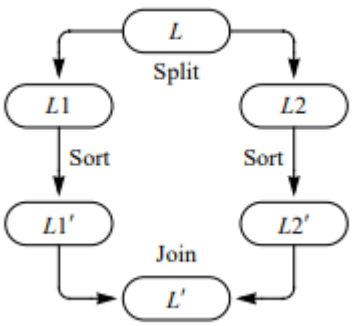


Figure.3.4 – Example for bubble sort.

### 3.1.5 Recursive Sorting

The idea of a recursive sort is to sort a large list assuming you can recursively sort a smaller part of the list. Figure 20.3 shows the general approach. Suppose you have a list of elements, L. To sort the list, you split it into two sublists, L1 and L2. The sublists are each smaller than the original list, L. The recursive idea lets you assume that you have the solution to the problem of sorting the smaller lists. So you recursively sort L1, producing the sorted sublist L1′. Then you recursively sort L2, producing the sorted sublist L2′. The last step is to join the two sorted sublists, L1′ and L2′, into the final sorted list, L′.



## 3.2 System Requirements

This section will provide the user the required specification of the hardware and software components on which the proposed system is to be implemented.

### 3.2.1 Hardware Requirements

This subsection will provide the minimum requirements that must be fulfilled by the hardware components. The hardware requirements are as follows: -

- A smart phone with
    1) RAM – minimum 2 gigabytes
    2) Processor – minimum dual core
- A desktop with
    1) RAM – minimum 4 gigabytes
    2) Processor – minimum quadcore or hexacore

### 3.2.2 Software Requirements

This subsection will provide the versions of software applications that must be installed.

The software requirements are as follows: -

- Google chrome
- Or Any Browser
- IDE
- Hosting

Mobile should be connected to internet to make use of the app efficiently.

## 3.3 Design Details

In design details, we analyse the System Architecture and System Modules in detail. We study the flow and process of the entire project in order to develop the project in an orderly and systematic manner. There are 3 modules in Algorithm Visualizer, encoding the Sorting and Path Finding through various algorithms

### 3.3.1 System Architecture

HTML5, CSS, and JavaScript make up the back-end code. There are three varieties of code in one .html file and all three can be executed from this file alone. Including different types of web languages in a single page is one of the shortcomings of HTML 5. Since, therefore, there were three different types, each had been segregated, producing three different files (plus the miscellaneous sound and image files). Readability and keeping relevant code together are benefits of excellent programming practices. However, in the end, I opted not to break the code into two separate sections because of these two reasons. By just having to worry about one project file instead of three, the project may be more easily transported and sent. And because the changes to the coding languages are identified unambiguously in the project file, they do not reduce readability

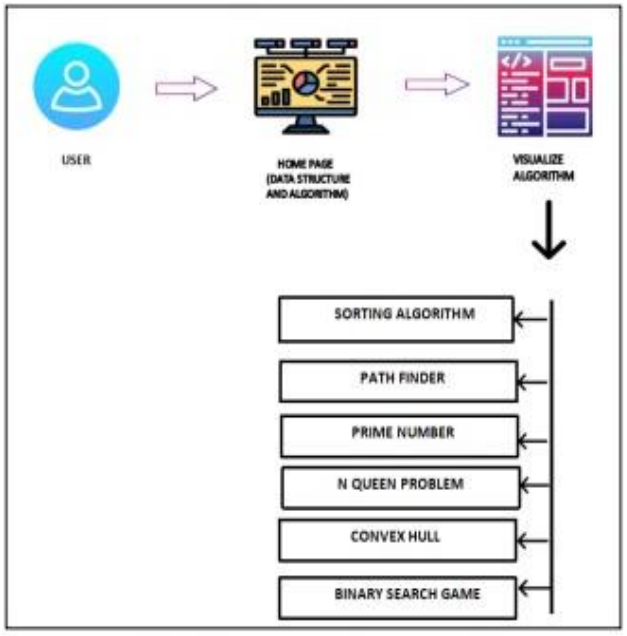



Figure. 3.1 – System Architecture

### 3.3.2 Entity Relationship Model

Figure 3.4 shows the Entity Relationship Diagram of the proposed system. Entity Relationship diagram is a data modelling technique that graphically illustrates an information system's entities and the relationships between those entities. As we are able to see from the above model that the centre of attraction of our application is the user thus, we need to ensure great user experience (UX) which would enhance the overall impact of our application. Since we did not have much complex relationships to manage in our application we decided to implement our app using some lightweight frameworks and scripting languages. Thus, JavaScript as the base language was an obvious choice owing to its lightweight nature and wide variety of framework options. We then went through most of the popular JavaScript frameworks. We tested each of them by trying to implement a sample page and came to a unanimous decision that React.js was the best choice due to its features like reusability, easy testing and debugging, and component based approach.Now, the only thing left was to decide how to structure our application to maximize its effectiveness. For this we analyzed a few existing designs over the internet and we finally decided on an architecture which has already been explained in the methodology section

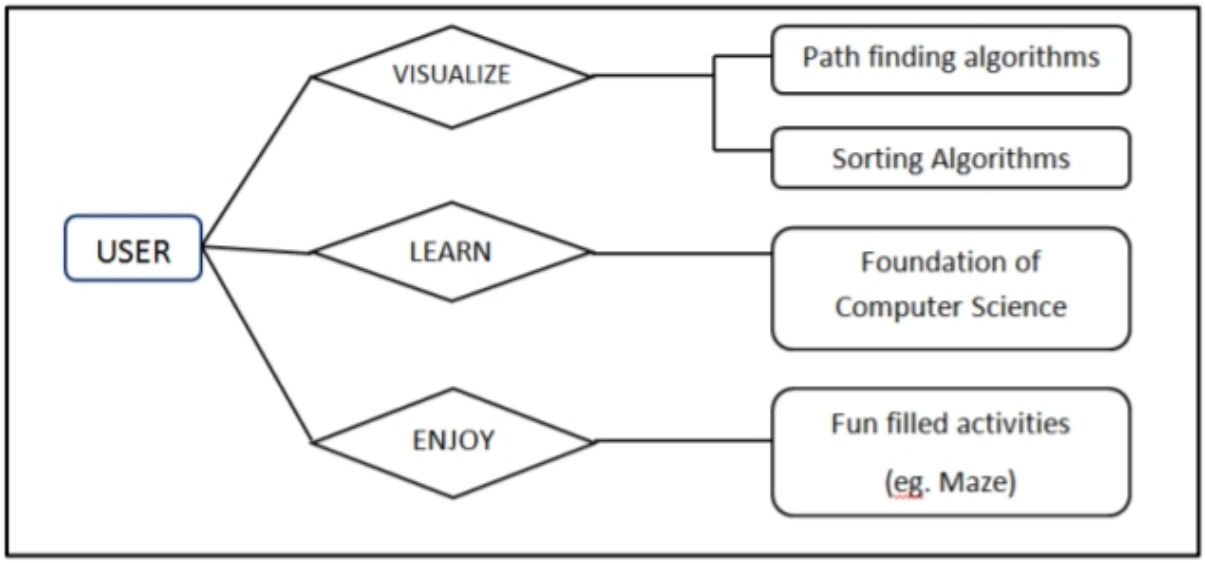



Figure 3.4 - Entity Relationship Diagram

## 3.4 Fundamental Model

Fundamental model of the project gives overall idea about the project. How the entities are related to each other, what are the attributes of the entities, how the data flows between the entities is shown by the fundamental model.

### 3.4.1 Data Flow Model

Data Flow Diagram (DFD) shows graphical representation of the" flow" of data through an information system, modelling its process aspects. It includes data inputs and outputs, data stores, and the various subprocesses the data moves through. DFDs are built using standardized symbols and notation to describe various entities and their relationships.

**DFD LEVEL 0**

In Data Flow Diagram we Show the flow of data in our system, In DFD0 we show the base DFD in which rectangle present External entity (an outside system that sends or receives data)and circle show a Process(process that changes the data, producing an output).The arrows towards the process shows input while the arrows away from the process shows output.

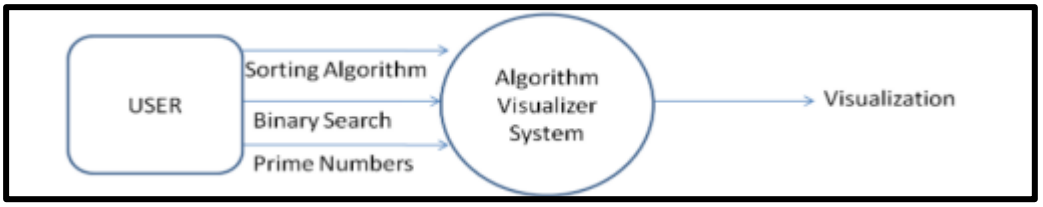



Figure 3.5.1 – DFD Level 0

**DFD Level 1**

Figure 3.6 shows the Level 1 Data Flow Diagram of the proposed system. It is exactly the same as the Level 0 DFD, but much simplified. The Level 1 DFD shows how the system is divided into sub-systems i.e. subprocesses, each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It breaks down the main processes into subprocesses that can then be analysed and improved on a more intimate level

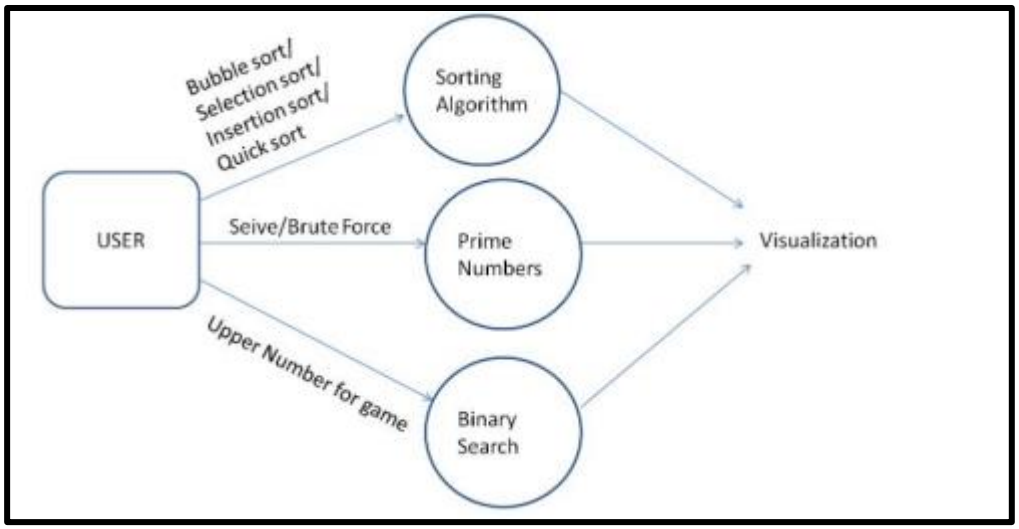
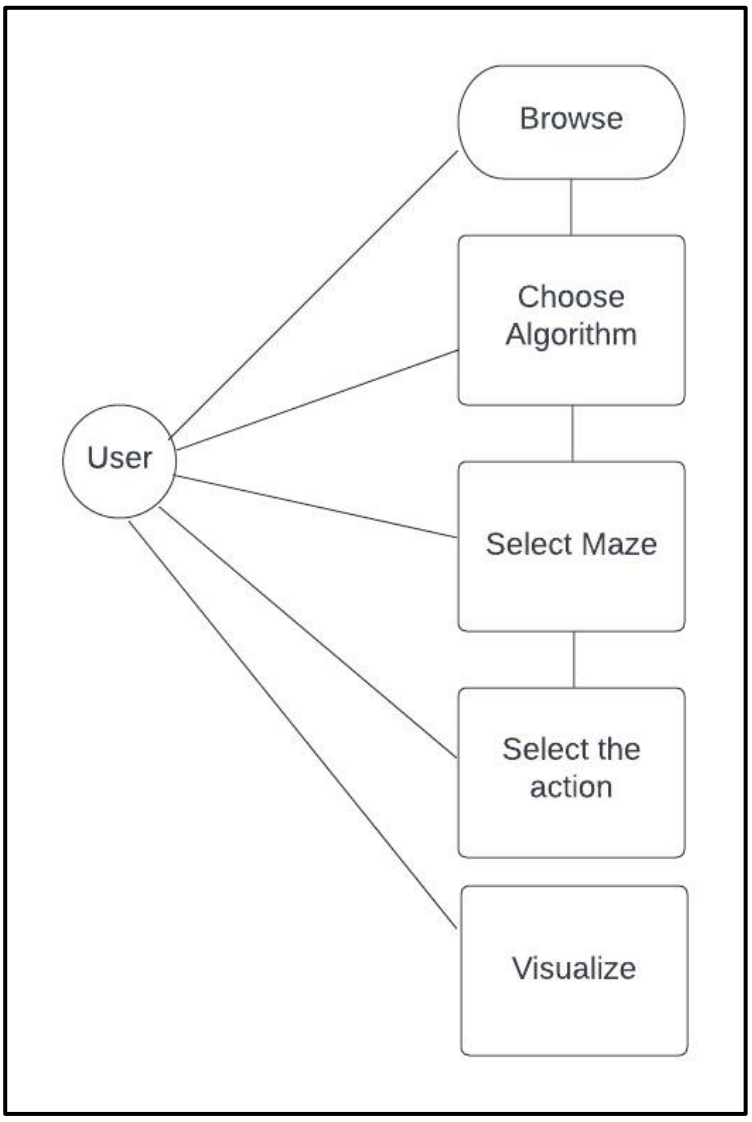



Figure 3.5.2 – DFD Level 1

## 3.5 UML (Unified Modelling Language) Diagram

The Unified Modelling Language is a general-purpose, developmental, modelling language in the field of software engineering that is intended to provide a standard way to visualizethe design of a system. We have prepared and designed the UML diagrams of – Use Case, Activity, Component, Deployment and Sequence Diagrams.



## 3.6 Methodology

1. Path-finding Algorithm

   The navbar of path-finding algorithm consists of the following options:-

a) Algorithms

   We have included:

- Path finder Algorithm
- Sorting Algorithm
- Recursive Algorithm
- N- queen Algorithm

The algorithms present in the navbar are chosen on the basis of their popularity and difficulty level. Students find it difficult to understand these algorithms theoretically. When they will see the visualization of these algorithms then they will be able to understand it better. User will be able to differentiate between the functionalities of different algorithms on the basis of time complexity after the visualization is over.

The algorithms present in the navbar are chosen on the basis of their popularity and difficulty level. Students find it difficult to understand these algorithms theoretically. When they will see the visualization of these algorithms then they will be able to understand it better. User will be able to differentiate between the functionalities of different algorithms on the basis of time complexity after the visualization is over.

b) Mazes and Patterns Maze and patterns are included to ensure better and clear understanding of algorithms. As there will be walls or obstruction between the starting node and the goal node, user can relate the visualization with real world like situation. Also, user will be able to figure out which algorithm is better based on algorithm time complexity. Especially for users looking for a playful option for understanding these complex topics these fun filled options can turn out to be the appropriate way.

c) Speed The project contains speed bar for maintaining the speed of visualization, this feature is included because everyone has a different learning rate so the user can vary the speed of visualization according to his/her choice. Designing Grid structure will be used to represent each node. Computer generated starting and ending node will be displayed initially. User can change the positions of start and end node according to his/her will.

# Chapter 4

# Result and Discussion

This chapter includes the snapshots of the actual outputs that were seen by the user and this chapter also contains the results of the proposed system.

## 4.1 Proposed System Result

The existing system that is implemented will help the Students to participate in learning the algorithms. It will facilitate the learning of different algorithms like path finder, sorting, recursive sorting, N-queen. Figure 4.1 shows theGUI of Home Page of Algorithm Visualizer which has the functionalities of different algorithms in top where students can choose the algorithm which they want to learn.

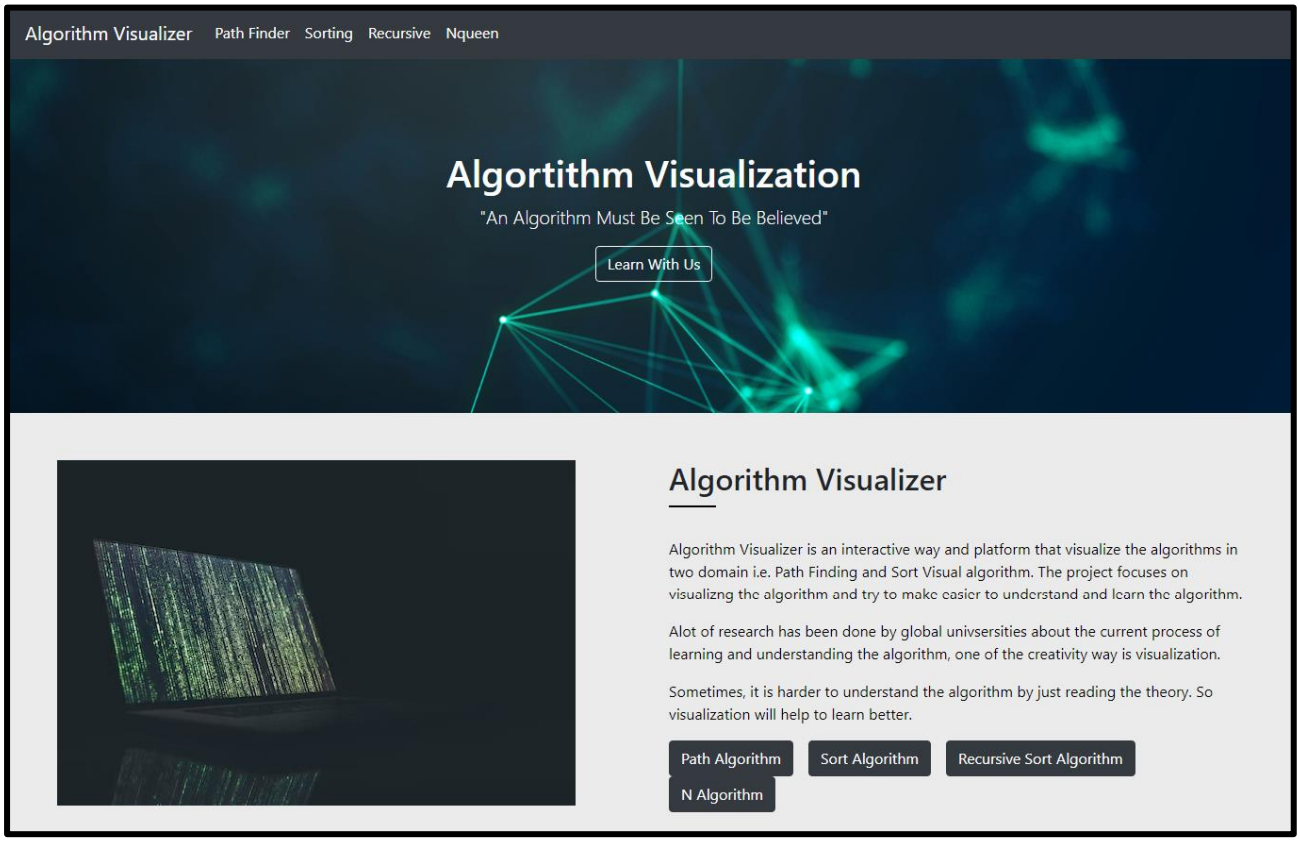



Figure 4.1 – GUI of Home Page

Figure 4.2 contains the screenshot of the first algorithm i.e. Path finding algorithm it has the navbar where user can select the type algorithm in it hase 3 types Dijkstra, bfs, dfs. It has the maze creation tool also in which it will create the random maze and 4 buttons create maze, visualize, reset path and rest board.

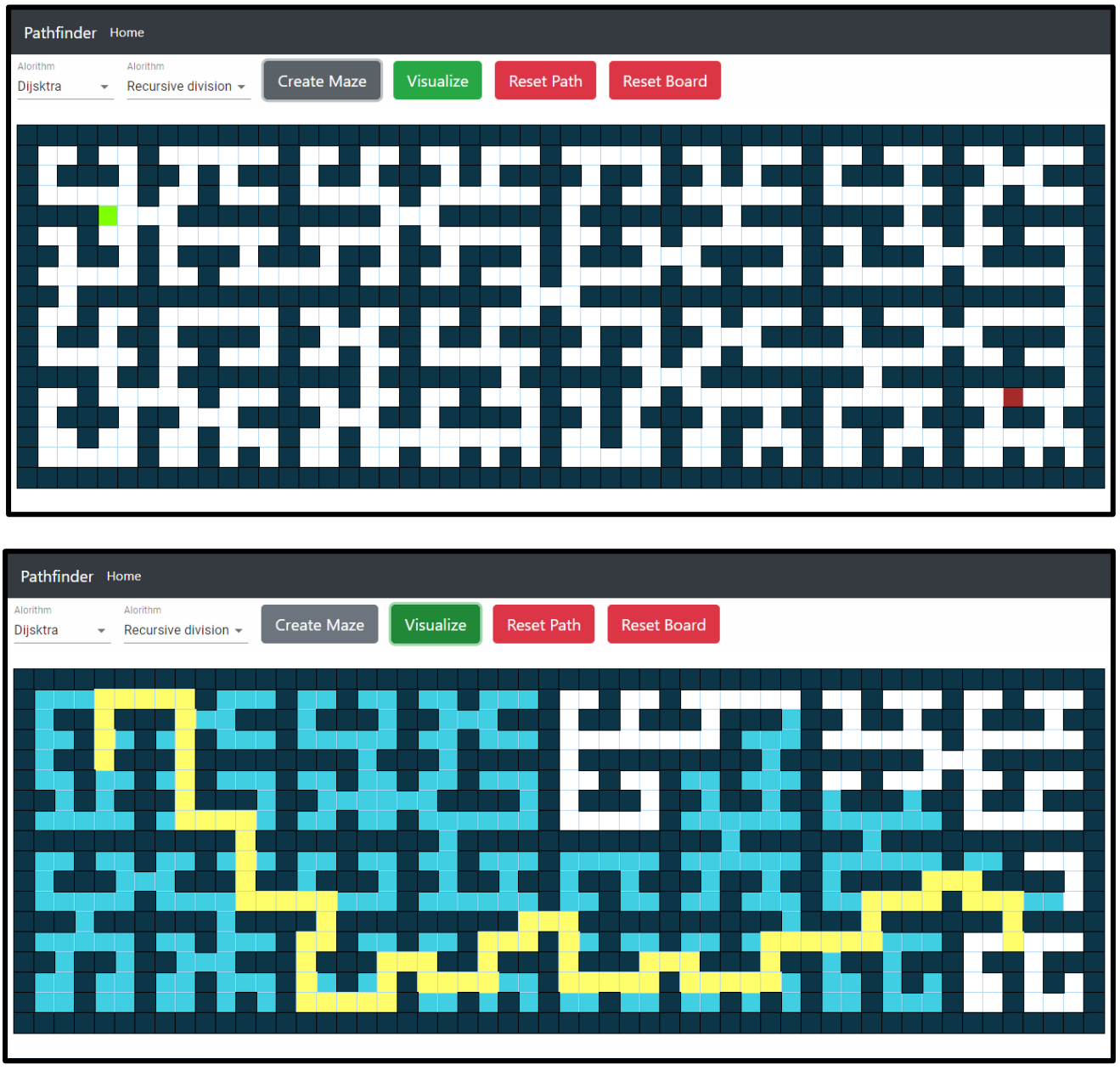



Figure 4.2 – GUI of Path Finding algorithm visualizer


Figure 4.3 GUI of the sorting algorithm.

Figure 4.3 contains the screenshot of the second algorithm i.e. sorting algorithm it has the navbar where user can adjust the number of bars, speed of animation, also there are two visuals though which we can see the difference between two algorithms in nav bar the options are provided to select the algorithm to visualize.

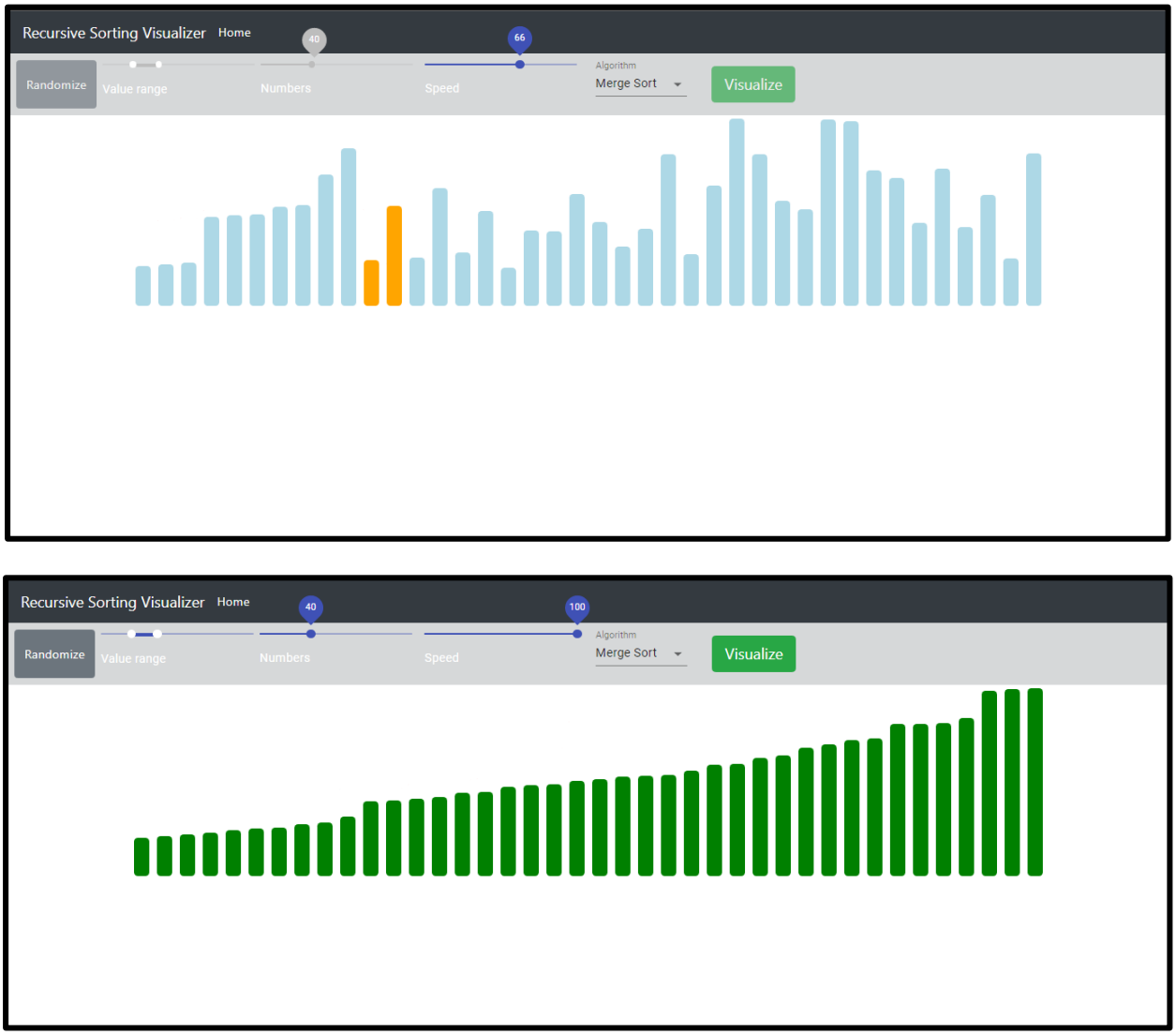



Figure 4.4 – GUI of Recursive Sorting Algorithm

Figure 4.4 shows the screenshot of the third algorithm i.e. recursive sorting algorithm it has the navbar where user can adjust the number of bars, speed of animation, also there are two visuals though which we can see the difference between two algorithms in nav bar the options are provided to slect the algorithm to visualize.

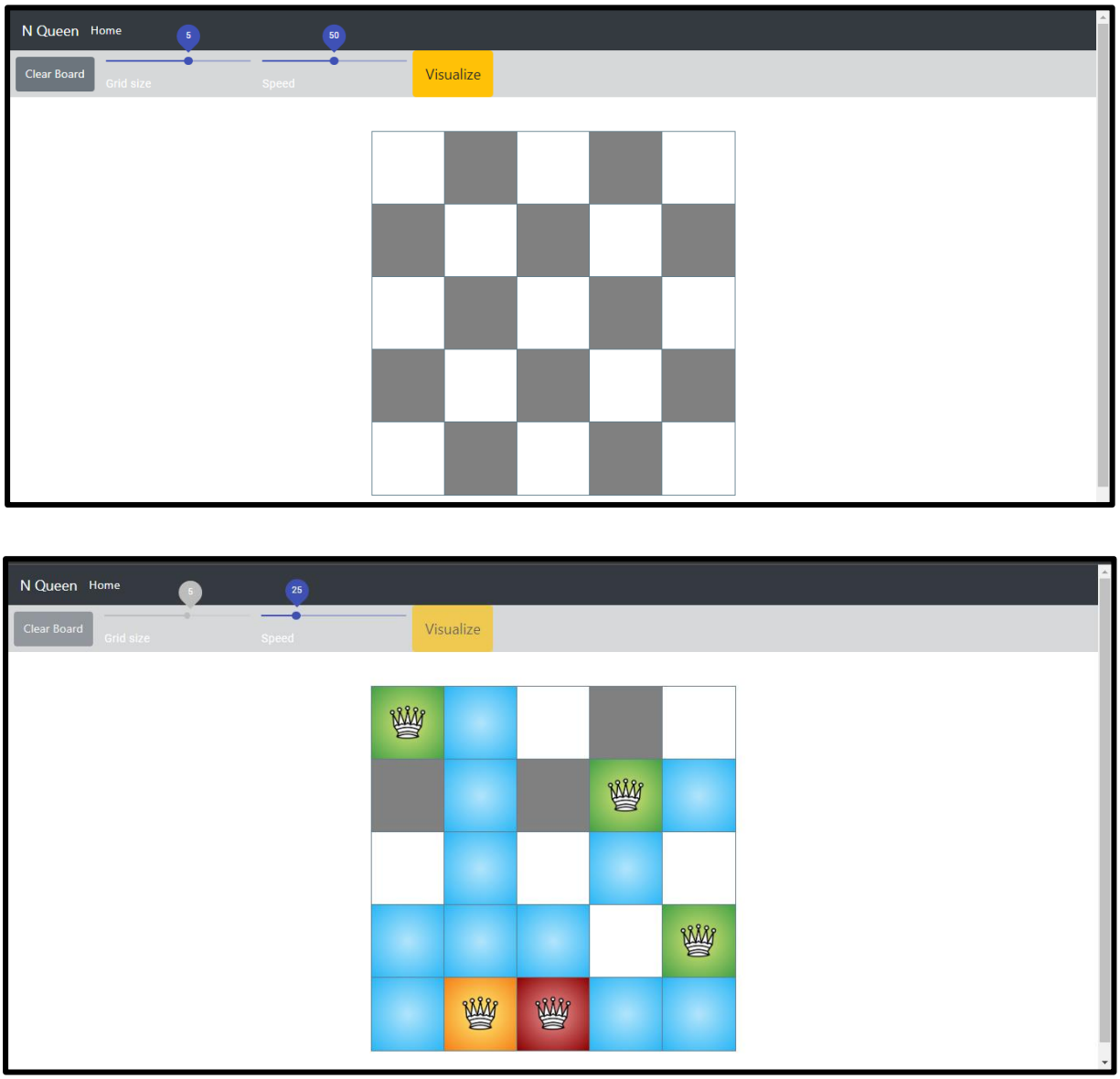



Figure 4.5 – GUI of N-queen algorithm

Figure 4.5 shows the screenshot of the third algorithm i.e. N-queen algorithm it has the navbar where user can adjust the grid size and speed of animation. Then rest the visualization will happen based on the custom speed.

# Conclusion

We began our endeavour by researching some well-known algorithm representations that have been created over a period of time. Our research indicates that algorithmic visualisation is frequently seen as a helpful auxiliary tool, used in conjunction with conventional methods of computer science instruction. We were able to successfully complete this project and achieve our goal of integrating Graph Path Finding with Visualization and differentiating their performances. There has been a significant disconnect between theoretical comprehension and actual understanding of algorithms realised, as is true for the majority of instructional disciplines. In particular for the Dijkstra algorithm, this is frequently true for shortest routes algorithms. The project's primary objective is to be used by researchers, educators, and students to teach and study the already-known graph algorithms. The system's major goal is to offer a collaborative learning environment for teachers and students in order to enhance learning in an efficient manner.

# Appendix

1) https://app.creately.com/

Creately tremendously helped is making the UML diagrams in the project. The various UML diagrams made in the project are – Data Flow Diagrams, Use Case Diagrams and the Entity Relationship Diagrams.

2) Visual Studio IDE

Visual Studio was used for setting up the entire project.

3) React JS

React is a free and open-source front-end JavaScript library for building user interfaces based on UI components.

4) Firebase

Firebase is a platform developed by Google for deploying web applications.

# References


[1] Jussi Nikander, Juha Helminen, and Ari Korhonen, “Algorithm Visualization System for Teaching Spatial Data Algorithms”, JITE, 2010.

[2] Barnini Goswami, Anushka Dhar, Akash Gupta, Antriksh Gupta, “ALGORITM VISUALIZR”, *International Research Journal of Engineering and Technology (IRJET),* 2021.

*[3]* Slavomír Šimoˇnák “Using algorithm visualizations in computer science education”, CEJCS*),* 2015.

[4] Euripides Vrachnos, Athanassios Jimoyiannis, “Design and evaluation of a web-based dynamic algorithm visualization environment for novices”, ScienceDirect, 2013.

[5] Arijit Goswami, Biswarup Bhattacharjee, Rahul Debnath, Ankita Sikder, “Web-Based Dynamic Algorithm Visualizer Along with Chat Engine SDK for Neophyts”, *International Journal of Engineering Science and Computing (IJESC),* 2021.

[6] Ashish Kumari, Manav Mittal, Vipul Jha, Abhishek Sahu, Manish Kumar, Neeti Sangwan and Navdeep Bohra, “Algorithm Visualization - Modern Web-based Visualization of Sorting and Searching Algorithms”, AAMS*),* 2017.

[7] Camil Demetrescu, Irene Finocchi , Giuseppe Liotta, “Visualizing Algorithms Over the Web with the PublicationDriven Approach”, *Crossmark,* 2018.

[8] Armine Khachatryan, “Algorithm Visualization”, *Institute of Electrical and Electronics Engineers* (*IEEE),* January 2019.


`

# Acknowledgement

We take this opportunity to express our deep sense of gratitude to our project guide and project co-ordinator**, Dr. Jitendra Saturwar**, for her continuous guidance and encouragement throughout the duration of our miniproject work. It is because of her experience and wonderful knowledge, we can fulfil the requirement of completing the miniproject within the stipulated time. We would also like to thank **Dr. Jitendra Saturwar**, Head of computer engineering department for his encouragement, whole-hearted cooperation and support.

We would also like to thank our Principal, **Dr. J. B. Patil** and the management of Universal College of Engineering, Vasai, Mumbai for providing us all the facilities and the work friendly environment. We acknowledge with thanks, the assistance provided by departmental staff, library and lab attendants.

**Mr. Darshan Chanchad (08)**

**Mr. Jigar Darji (12)**

**Mr. Harsh Moradiya (70)**

**Mr. Nikunj Rupavatiya (112)**