

Exploring All the Features And User Concerns Of Home Away Your Ultimate Housing Solution

Project Guide: - Prof. Ashish Bhise (H.O.D. of AI&DS Dept.)

Anupam Pandey
(Final year B.E)
anupam.pandey@mmit.edu.in

Yash Bambal
(Final year B.E)
Yash.bambal@mit.edu.in

Sakshi Bhor
(Final year B.E)
sakshi.bhor@mmit.edu.in

Karn Datta Godse
(Final year B.E)
karn.godse@mmit.edu.in

Abstract :- In today's era, there's a rapid change in lifestyle with this modern lifestyle approach, the definition of housing and renting is changing globally. With the aim to enhance the experience of renting and providing people comfortable houses, which can be their “ Home Away from home”, our Project “ Home Away” is a web based application, developed through synergy of HTML, CSS And JAVA for webpage creation which is then integrated with a database where user data and insights are stored through which suitable output(suggestions) are produced for the customers. This is a dynamic solution that facilitates direct communication between property owners and potential owners. It is a completely customised app, where consumers can pick and choose their stay as per their needs.

Keywords – Enhance experience, Insights, Database, Dynamic Solution, Rental,

I. INTRODUCTION

Home Away is an innovative web application designed to revolutionize the way individuals seek rental accommodations and streamline the rental process. This platform is a one-stop solution for all your housing needs, offering an array of services. At its core, Home Away enables users to search and rent houses, be it for long-term leases or short-term stays.

One of its standout features is the ability to create and manage rental agreements online, simplifying the often complex and paper-intensive legal aspects of renting. Users can draft, review, and sign digital contracts, enhancing transparency and security. Additionally, Home Away fosters seamless communication between landlords and tenants through

its built-in messaging system, making it easy to discuss property details, negotiate terms, and address concerns in real-time.

With an intuitive interface and a comprehensive database of rental listings, Home Away is poised to redefine the way people find, rent, and manage their homes, offering convenience, efficiency, and peace of mind in the housing market

II. ALGORITHMS

We must ensure that the fundamentals of algorithms are understood before continuing on to the further sections, which describes algorithms and the software. The definition of the algorithm will be our first step. We can simply define an algorithm as a series of steps or instructions that addresses a certain problem. Problems are some tasks that need to be solved. We face many different problems everyday like:- quickest way to work or home etc. However, all these problems may not fit our algorithm definition. Problems need to be specified with its inputs and all the inputs should have an output. Solving a problem means producing an output for all the inputs

Currently a large number of algorithms exist. However, many of them can be classified into the following different types of algorithms:

- **Recursive Algorithm:** A recursive algorithm calls itself repeatedly until a problem is solved. In these algorithms, the input becomes smaller and smaller as the algorithm calls itself. Some common problems which are solved using this approach are: Tower of Hanoi and Searching in a Binary Search Tree.

- **Divide and Conquer Algorithm:** In Divide and conquer algorithm, a large problem is divided into smaller sub problems and then each sub problem is solved individually. After solving each sub problem, the solutions of all the sub problems are combined to form the solution of the whole problem. Merge Sort and Quicksort are major Divide and conquer algorithms.

- **Dynamic Programming Algorithm:** Some algorithms may take a large amount of time to execute completely. To avoid this and reduce the time taken by the algorithm to execute completely, more space is used by the algorithm in order to store the already calculated results. In Dynamic Programming Algorithm, additional space is used to store the results of already calculated sub problems for future use. This reduces the time required to calculate the same results again. Fibonacci Series is a good example of Dynamic Programming Algorithms. MMIT, Department of Computer Engineering 2022-23 8

- **Greedy Algorithm:** Optimization Problems are solved using these problems. In this algorithm, to find the optimal solution for the whole problem, locally optimal solution is found at various steps of the algorithm. Popular Greedy algorithms include Huffman Coding and Dijkstra's algorithm.

- **Brute Force Algorithm:** This is the simplest of all the algorithms. It generates all the possible solutions and to search for one or more possible solutions of the given problem. Linear Search is an example of Brute Force Algorithm.

- **Backtracking Algorithm:** It is used to solve a problem in an incremental way. We keep moving forward with the solution until we find the correct solution. If the solution fails we backtrack, i.e. we move backwards, and continue finding the solution from the previous step. NQueens Problem is solved using the Backtracking Approach.

III. METHODOLOGY AND FRAMEWORK

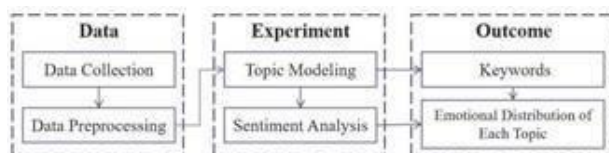


Figure 1: Framework

Fig:1 shows the framework of the study. We use the LDA method to extract the underlying themes of the user reviews and use SnowNLP from sentiment analysis and polarity analysis to study user rating and sentiment distribution of HOME AWAY listings.

Software Requirements Specifications (SRS) are created during the requirements stage of the software development process. It is also called a

Requirements document. When all requirements are gathered and analyzed, this report is constructed, laying the groundwork for software engineering efforts. SRS is a formal report that serves as a software representation, allowing clients to assess if the report (SRS) satisfies their needs. It also includes specific descriptions of system requirements as well as user needs for the system.

User Classes and Characteristics : In our proposed systems, we have finalized the following components to make it a success, as shown in the following illustration fig2:

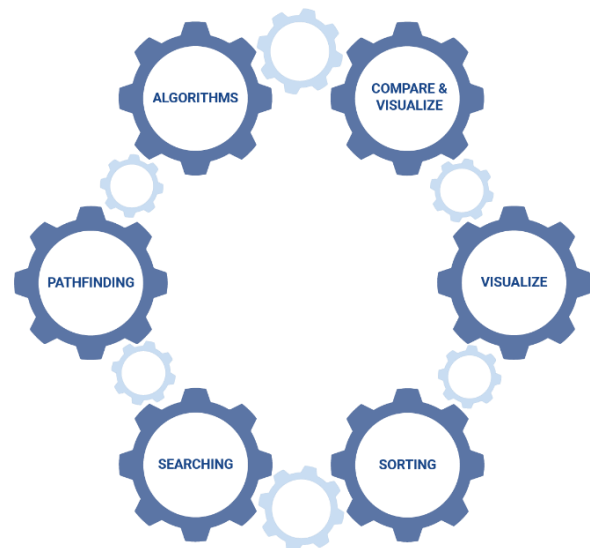


Figure 2: Proposed System Components

Assumptions and Dependencies: It is assumed that the user is familiar with an Internet Browser and also knows how to operate a keyboard and a mouse.

Since our application is a Web Based Application. An Internet connection is required. It is assumed that the user has a decent internet connection.

IV. FUNCTIONAL REQUIREMENTS

Functional requirements are attributes or capabilities of a product that developers must implement to allow users to carry out their duties. For the development team as well as the stakeholders, it is crucial to make them apparent. Functional requirements often explain how a system will behave under particular circumstances.

Visualization: Users will be able to learn and understand the working of various algorithms through visualization. Learning algorithms through visualizations can help students grasp algorithms better.



Figure 1: Visualization

Hardware Interface: As this is a software project (not a hardware project), this system does not require any Hardware Interface to facilitate communication.

Software Interface: This dynamic web application makes use of Browser, and localhost ports to run while development and testing. The system will have an intuitive GUI.

Communication Interface: The communication medium between the user and system is through user input objects or default generated values provided to the system.

V. NON - FUNCTIONAL REQUIREMENTS

Performance Requirements: Performance refers to how quickly a software system, or a specific component of it, reacts to specific user actions while handling a specific workload.

Given the current user base as a whole, this statistic often shows how long it takes a user must wait before the goal operation occurs (the page renders, a transaction is executed, etc.). Since we are using ReactJS, which is considered one of the best in terms of rendering performance, the visualization of algorithms will be fast and smooth.

Safety Requirements: Firebase will be used as the backend of our Web Application. Firebase services logically segregate client data and encrypt it in transit via HTTPS.

Security Requirements: Security is a non-functional criterion that guarantees that every piece of data inside the system will be safe from malware assaults and unauthorized access.

The Web Application will have an SSL certificate. An SSL certificate is a digital certificate that permits an encrypted connection and verifies the legitimacy of a website.

Software Quality Attributes:

→ **Scalability:** Scalability measures the highest

workloads that the system can handle while still delivering the required levels of performance. When workloads increase, your system can grow vertically or horizontally using one of two methods.

The Web Application will be able to handle multiple clients simultaneously as Firebase is designed with scalability and performance in mind.

→ **Usability:** Usability standards come in a variety of forms.

→ **Learnability:** Users will be able to learn to operate our system very easily as our user interface is very user-friendly. We will also provide a tutorial about how to use the application when the user opens the application for the first time.

→ **Efficiency:** We have only focused on essential topics and relevant components to ensure the user does not fiddle around.

→ **Memorability:** Users will remember how to use the application as we provide tutorials when the user opens the application for the first time and also our system is very easy to use without any complex steps.

→ **Errors:** To avoid user errors, we guide them about how to use the system with the help of tutorials.

→ **Satisfaction:** Our design is pleasing to use with appropriate colors and animations that look good and don't strain the eyes of the user. Users will be able to easily use the Web Application without any help from others.

→ **Reliability :** The dependability of a system or component indicates the likelihood that it will function well for a predetermined amount of time under specific circumstances.

The application will always provide the correct information and intended visualization to the user when the input is provided to it.

→ **Responsiveness:** The Web Application can adapt to various screen sizes and will also work properly on mobile devices.

Integrability:

All the components of our system are integrated with each other. The components can communicate, respond and interact with each other

VI. SYSTEM REQUIREMENTS

Database Requirements: Firebase will be used for storing information about the algorithms. Firebase offers a collection of hosting services. It offers NoSQL and real-time database, content, social authentication, and notification hosting as well as other services, like a

real-time communication server.

Software Requirements (Platform Choice):

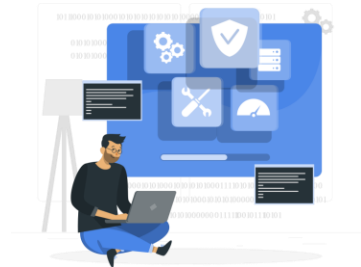


Figure 3: system Requirements Analysis

Minimum Software Requirements:

For User:

→ Browsers: Microsoft Internet Explorer, version 9 or later. Mozilla Firefox, version 29 or later. Chrome, version 34 or later.

Used For Development:

→ Visual Studio Code
 → ReactJs libraries
 → NodeJS
 → HTML
 → CSS
 → Javascript

Hardware Requirements:

Figure 1: System Hardware Requirement Analysis

Minimum Hardware Requirements:

→ Laptop or PC
 → Pentium 4 or newer processor that supports SSE2
 Mac - Mac computer with an Intel x86 or Apple silicon processor
 → 512MB of RAM / 2GB of RAM for the 64-bit version
 → 200MB of hard drive space

VII. SYSTEM ARCHITECHTURE

The system has 3 main components: - Pathfinding Algorithms, Searching Algorithms and Sorting Algorithms.

Pathfinding Algorithms: The pathfinding algorithms will be visualized through a grid where users can create

a maze and set their starting point and end point. Users can choose an algorithm and click on visualize to visualize the chosen algorithm. Users can clear the board, clear the path, adjust speed and also add a bomb to the maze as a barrier.

Searching Algorithms: Searching Algorithms like Linear Search and Binary Search will be visualized with the help of blocks where the number of blocks indicates the array size and each block will have a value inside it.

Searching Algorithms like DFS and BFS will have circles which represent nodes and lines which represent the edges of a graph or tree. Users can drag circles to add new nodes and connect circles with the help of lines to form edges.

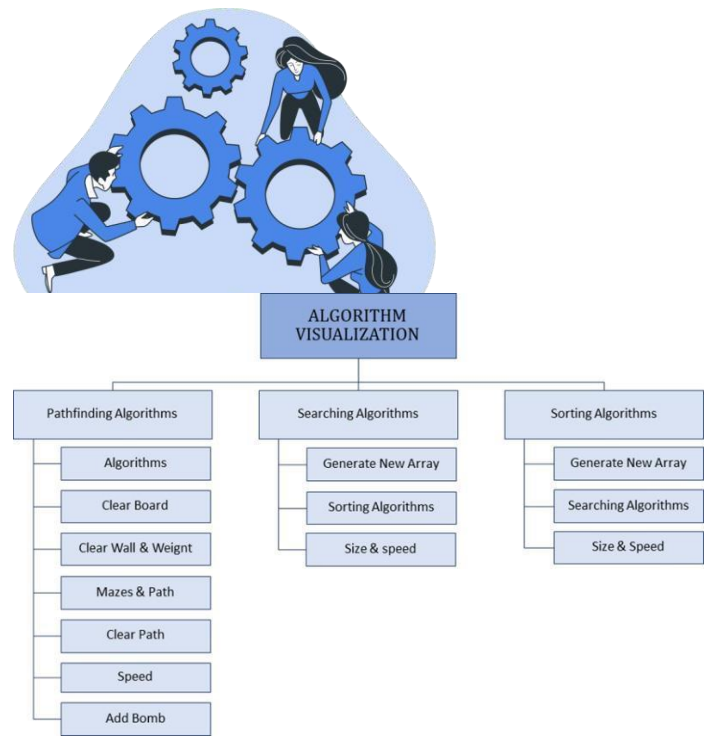
Sorting Algorithms: Sorting algorithms will be visualized with the help of bars where the number of bars indicates the array size and the height of the bar indicate the value. Users can generate random arrays or insert their own arrays

VIII. DATA FLOW DIAGRAMS

The classic visual representation of how information moves through a system is a data flow diagram (DFD). A tidy and understandable DFD can graphically represent the appropriate quantity of the system demand. It can be done manually, automatically, or both.

It demonstrates how information enters and exits the system, what modifies the data, and where information is kept.

A DFD's main goal is to outline a system's overall



scope and bounds. It can be used as a tool for communication between a system analyst and anyone involved in the process of establishing the order that will serve as the basis for a system redesign. A bubble chart or data flow diagram are other names for the DFD. Below are the Level 0, Level 1 and Level 2 Data Flow Diagrams of our System:

In Data Flow Diagram Level 0, the complete system is shown as a single bubble. It is sometimes referred to as the core system model, with input and output data marked by incoming and exiting arrows. The system components that each of these bubbles represents are then broken down and documented as ever-more-detailed DFDs.

In the above Data Flow Diagram Level - 0, we have represented our system as a single process and the external entities for our system are User and Admin.

A context diagram is divided into various bubbles and processes in the Level 1 Data Flow Diagram. At this stage, we draw attention to the system's primary goals and deconstruct the high-level DFD process into its component parts.

In the above diagram, we have broken down our system into various processes which are:

- ➔Searching Visualizer
- ➔Sorting Visualizer
- ➔Path Finding Visualizer
- ➔Comparing Algorithms

We Also have an Algorithms Database which will store all the information of the algorithms.

Users will provide data to the algorithms and then get the visualization and analysis in return. Admin can add new algorithms to the Algorithms Database.

Parts of Level 1 DFD are further processed in Level 2 DFD. It can be utilized to project or document precise/important information about how the system operates. In Level 2 DFD, each of the processes which are shown in the DFD Level 1 Diagram is broken down into sub-processes.

In the above Level 2 DFD we have shown all the sub-processes of the Sorting Visualizer Process from the Level 1 DFD. The Sorting Visualizer Process consists of the following sub-processes:

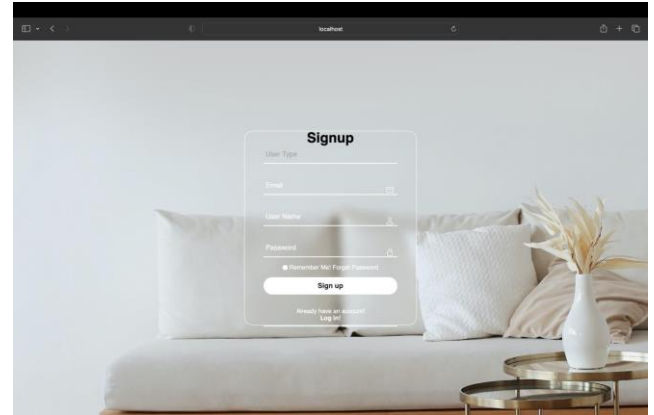
- ➔ Initially, the user needs to choose an algorithm from various available sorting algorithms. The Sorting algorithms are fetched from the Algorithms Database.
- ➔Next user gives the input array which needs to be sorted.

➔The backend will then generate all the graphical components required for visualization and then the visualization will take place.

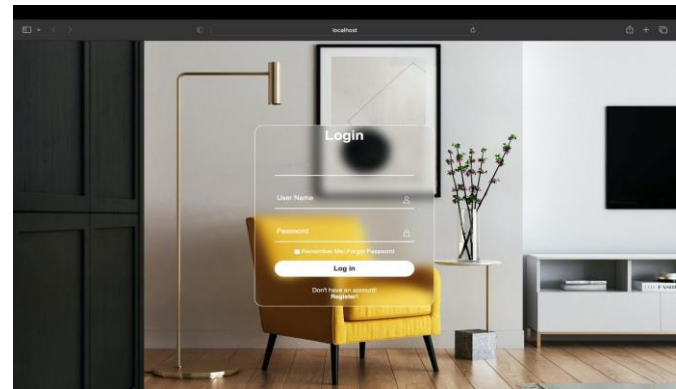
IX. Results

Snapshot of Outcome

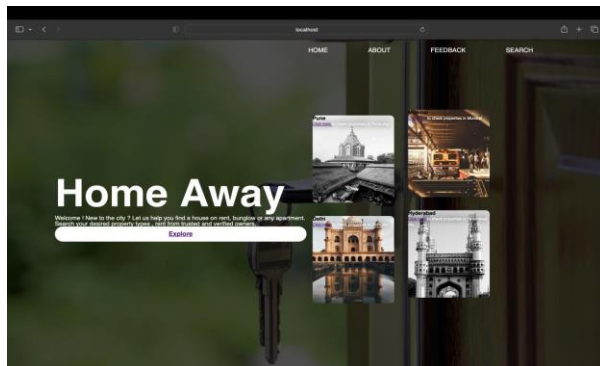
Registration



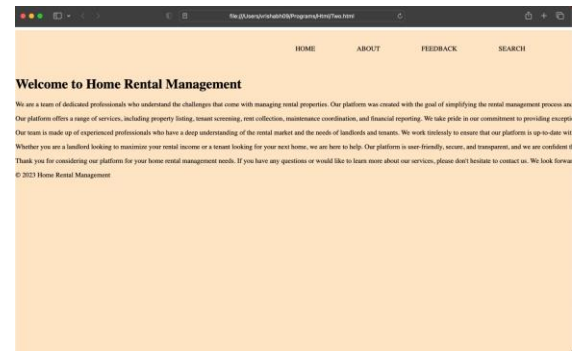
Login



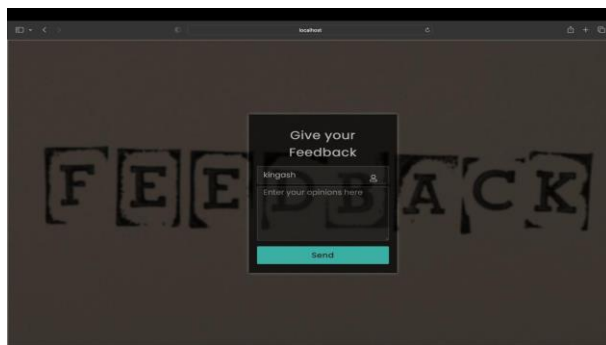
Homepage



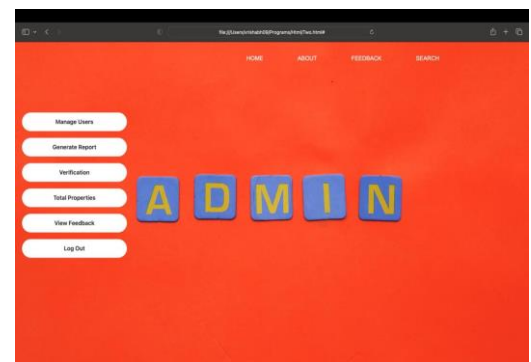
About page



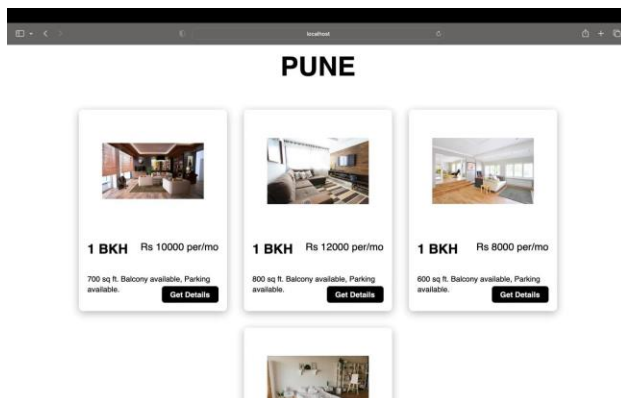
Feedback



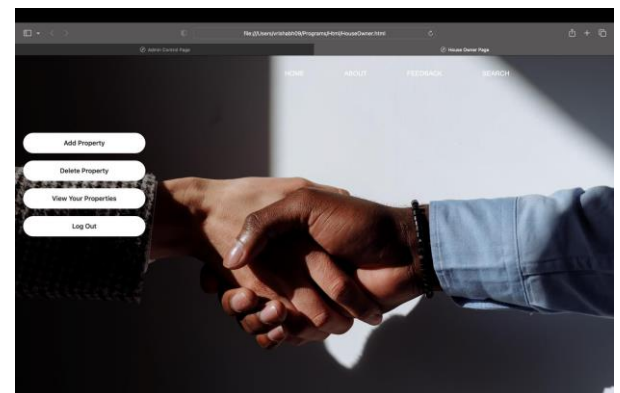
Admin Page



Property page



House owner Interface



Search Page

Select location	
Select Type	
Select Price	
Book	Apply

Name	Type	Location	Price	Jump
Posti	3BHK	Suburban	2000000	View

Add Property Page

Search page results

Services

Total users page

House

Name	Type	Location	Username	Price
Patil_Wada	3BHK	Sahakar Nagar	bob123	200000
Uttam_Townscape	3BHK	Yerawada	bhakti0205	150000
Sonigara_Township	3BHK	PUNE	bob123	100000
Skytree	2BHK	Koregaon Park	ketan123	100000
Lunkad	1BHK	Viman_Nagar	ketan123	300000
Peter_Wada	Bungalow	Vadgaon	ketan123	1500000

Customer

Name	Username	Password	Email	Phone
Omkar Nagnure	omkar1234	Osn@12345	omkarnagnure98@gmail.com	8390829546
Shruti	shruti123	Shruti@123	shruti@gmail.com	4354627897
Bhakti	bhakti123	Bhakti@1234	bhakti@gmail.com	4356789043

House Owners

Name	Username	Password	Email	Phone
Omkar Nagnure	osn123	123	omkarnagnure98@gmail.com	08390829546
Bob	bob123	bob12	bob@gmail.com	123456
Bhakti	bhakti0205	0987	bhakti@gmail.com	3245665472
Omkar Nagnure	hbhbh	ybhub	omkarnagnure98@gmail.com	8390829546
Ketan	ketan123	Ketan@123	ketan@gmail.com	8390829546

[Back](#) [Print](#)

X. CONCLUSION

We have comprehended through our research what the drawbacks of earlier-developed algorithm visualizers are and how to work around them to make them better. In order to successfully make the desired targets and particular, we will be creating a sophisticated system for algorithms visualization. This will be very helpful for students, teachers and corporate trainers for a better understanding of algorithms. Easing the way of e-learning and demonstrating through the dynamic aspect of visuals is our sole impulse to undertake this project.

Future Scope

- ➔ Internet is required to run the website so in future, a Mobile Application can be developed which will have all the functionalities of the website and will run offline.
- ➔ Many more algorithms can be added.
- ➔ A descriptive idea about the algorithm that is being visualized.
- ➔ Pause-and-play feature.

XI. APPENDIX

[1] G. Prabhakar, S. Gaur, L. Deshwal and P. Jain, "Analysis of Algorithm Visualizer to Enhance Academic Learning," 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), 2022, pp. 279-282, doi: 10.1109/ICIPTM54933.2022.9753906.

They created an algorithm visualizer using the Tkinter Module in Python. It was a desktop application which could be downloaded from an easy-to-navigate website made by them.

[2] N. Yadav, K. Dhameja and P. Chaubey, "Path Finding Visualizer Application for Shortest Path Algorithm" 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), 2021, pp. 1669- 1672, DOI: 10.1109/ICAC3N53548.2021.9725716.

They created a path-finding visualizer which uses A*, Dijkstra and DFS algorithms to find the shortest path between the starting node and the destination node.

[3] B. Goswami, A. Dhar, A. Gupta and A. Gupta, "Algorithm Visualizer: Its features and working," 2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), 2021, pp. 1-5, DOI: 10.1109/UPCON52273.2021.9667586.

They visualized various Path Finding Algorithms, Sorting Algorithms and CPU Scheduling Algorithms. In a survey conducted by them, they found that 60 per cent of students would like to learn algorithms through such visualizations rather than books.

[4] Sengar, Nishant. (2021). Pathfinder Visualizer of Shortest Paths Algorithms. International Journal for Modern Trends in Science and Technology. 6. 479-483. 10.46501/IJMTST061293.

They created a visualizer to visualize BFS and Dijkstra's Algorithm. They also provided a comparison and analysis between the two algorithms.

[5] B. Nagaria, B. C. Evans, A. Mann and M. Arzoky, "Using an Instant Visual and Text Based Feedback Tool to Teach Path Finding Algorithms: A Concept," 2021 Third International Workshop on Software Engineering Education for the Next Generation (SEENG), 2021, pp. 11-15, doi: 10.1109/SEENG53126.2021.00009.

They emphasized the necessity for an interactive teaching tool that may assist students who learn best visually at any moment, regardless of the current academic teaching assistance. They provide a mechanism that allows students to submit their path-finding formulas and receive visual comments.

[6] A. B. Ghandge, B. P. Udhane, H. R. Yadav, P. S. Thakare, V. G. Kottawar and P.

B. Deshmukh, "AlgoAssist: Algorithm Visualizer and Coding Platform for Remote Classroom Learning," 2021 5th International Conference on Computer, Communication and Signal Processing (ICCCSP), 2021, pp. 1-6, doi: 10.1109/ICCCSP52374.2021.9465503. They created AlgoAssist, a platform with integrated features where students may improve their coding abilities, teachers can assess students' work, and the emphasis is on "Algorithm Visualization" to help students comprehend the flow and operations of algorithms. It integrates lab functionality into a single platform, making it easier for professors and students to utilize.

[7] A. Jain, "Realizing Algorithms Using GUI," 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART), 2021, pp. 451- 455, doi: 10.1109/SMART52563.2021.9676269.

They created a Sorting Algorithms Visualizer which can be used to visualize various Sorting Algorithms like: Insertion Sort, Bubble Sort, Selection Sort etc. This application allows users to generate a random array for visualization. Further, one can also control the speed of sorting and the array size.

[8] Abedalrahim, Jamil & Alsayaydeh, Jamil & Zainon, Maslan & Oliynyk, A. & Aziz, Azwan & Rahman, A & Baharudin, Z.A. & Teknologi, Fakulti & Elektrik, Kejuruteraan & Elektronik, Dan & Teknikal, Universiti & Melaka, Malaysia & Jaya, Hang & Tunggal, Durian. (2020). THE DEVELOPMENT OF SYSTEM FOR

ALGORITHMS VISUALIZATION USING SIMJAVA. Journal of Engineering and Applied Sciences. 15. 3024-3034.

XII. REFERENCES

- [1] G. Prabhakar, S. Gaur, L. Deshwal and P. Jain, "Analysis of Algorithm Visualizer to Enhance Academic Learning," 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), 2022, pp. 279-282, doi: 10.1109/ICIPTM54933.2022.9753906.
- [2] N. Yadav, K. Dhameja and P. Chaubey, "Path Finding Visualizer Application for Shortest Path Algorithm" 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), 2021, pp. 1669- 1672, DOI: 10.1109/ICAC3N53548.2021.9725716. B. Goswami, A. Dhar, A. Gupta and A. Gupta, "Algorithm Visualizer: Its features and working," 2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), 2021, pp. 1-5, DOI: 10.1109/UPCON52273.2021.9667586.
- [3] Sengar, Nishant. (2021). Pathfinder Visualizer of Shortest Paths Algorithms. International Journal for Modern Trends in Science and Technology. 6. 479-483. 10.46501/IJMTST061293.
- [4] B. Nagaria, B. C. Evans, A. Mann and M. Arzoky, "Using an Instant Visual and Text Based Feedback Tool to Teach Path Finding Algorithms: A Concept," 2021 Third International Workshop on Software Engineering Education for the Next Generation (SEENG), 2021, pp. 11-15, doi: 10.1109/SEENG53126.2021.00009.
- [5] A. B. Ghandge, B. P. Udhane, H. R. Yadav, P. S. Thakare, V. G. Kottawar and P. B. Deshmukh, "AlgoAssist: Algorithm Visualizer and Coding Platform for Remote Classroom Learning," 2021 5th International Conference on Computer, Communication and Signal Processing (ICCCSP), 2021, pp.1-6,doi: 10.1109/ICCCSP52374.2021.9465503.
- [6] A. Jain, "Realizing Algorithms Using GUI," 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART), 2021, pp. 451-455, doi: 10.1109/SMART52563.2021.9676269.
- [7] Abedalrahim, Jamil & Alsayaydeh, Jamil & Zainon, Maslan & Oliinyk, A. & Aziz,

& Tunggal, Durian. (2020). THE DEVELOPMENT OF SYSTEM FOR

ALGORITHMS VISUALIZATION USING SIMJAVA. Journal of Engineering and Applied Sciences. 15. 3024-3034.