D.Y.PATIL COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institute)

KASABA BAWADA, KOLHAPUR

A PROJECT REPORT

ON

"Algorithm Visualizer"

Submitted By

Name	Roll No
Mr. Shreyas Chandrashekhar Sawant	36
Mr. Vikas Vinod Patil	37
Mr. Harshvardhan Sunil Solankure	39
Mr. Ruapm Ravindra Jadhav	38

Under the guidance of

Prof. S. N. Patil

D.Y.PATIL COLLEGE OF ENGINEERING AND TECHNOLOGY

KASABA BAWADA, KOLHAPUR

DEPARETMENT OF COMPUTER SCIENCE AND ENGINEERING

D Y PATIL
COLLEGE of
ENGINEERING & TECHNOLOGY
(AN AUTONOMOUS INSTITUTE)
KASABA BAWADA, KOLHAPUR

CERTIFICATE

It is to certify that a project report titled "**Algorithm Visualizer**" carried out by **Mr. Shreyas Chandrashekhar Sawant DSE21110990**, **Rupam Ravindra Jadhav, Vikas Vinod Patil DSE21143798, Harshvardhan Sunil Solankure DSE21139653**, are a bona fide student of D Y Patil College of Engineering & Technology, Kasaba Bawada Kolhapur submitted in partial fulfillment of the requirements for the award of BACHOLER OF TECHNOLOGY in the DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING during the year 2022-2023. It is certified that all correction/suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies requirements for the said degree.

Prof. R. J. Dhanal
H.O.D CSE

Prof. A. P. Budaragade
Project-II Coordinator

Prof. Dr. S. D. Chede
Principal

Prof. S. N. Patil
Project Guide

Prof.
External Examiner

Date:

Place:

ACKNOWLEDGEMENT

Firstly, we would like to express our profound gratitude to our internal guide of the project, Mrs. Sharayu Nilesh Patil for the continuous support of our Project work, for her patience, motivation, encouragement, and immense knowledge.

We would like to express our gratitude and deep regards to department project coordinator Prof. Avinash Pratap Budaragade for his guidance and support throughout the completion of project work. We would like to express our heart full gratitude to department H.O.D Prof. Raddheka Dhannal for her continuous encouragement & motivation

It is our pleasure to acknowledge the help we have been received from institute and the Individual. We would like to thank our Principal Dr. Santosh Chede Principal, D. Y. Patil College of Engineering and Technology, Kolhapur in particular for always giving encouragement, support and the excellent facilities provided.

Abstract

This report outlines a study that tested the benefits of animated sorting algorithms for teaching. To visualize four sorting algorithms, a web based animation application was constructed. A visualization of data is implemented as a bar graph, after which a data sorting and algorithm may be applied. The resulting animation is then performed either automatically or by the user, who then sets their own pace. This is research on the computer science curriculum's approach to learning algorithms. The experiment featured a presentation and a survey, both of which asked students questions which may illustrate improvements in algorithm comprehension. These findings and reactions are catalogued in this document and compared to earlier investigations. We present our findings of algorithm visualization, based on extensive search and analysis of links to hundreds of visualizations. We seek to answer questions such as how content is distributed among topics, who created algorithm visualizations and when, the overall quality of available visualizations, and how visualizations are disseminated. We found a wiki that currently catalogs over 350 algorithm visualizations, contains the beginnings of an annotated bibliography on algorithm visualization literature, and provides information about researchers and projects. Unfortunately, we found that most existing algorithm visualizations are of low quality, and the content coverage is skewed heavily toward easier topics exception to some . There are no effective repositories or organized collections of algorithm visualizations currently available. Thus, the field appears in need of improvement in dissemination of materials, informing potential developers about what is needed, and propagating known best practices for creating new visualizations. The aim is to automate its existing manual system by the help of computerized equipment's and computer software, fulfilling user requirements. Basically the thesis describes how to manage for good performance and better services for the users.

Algorithm analysis and design is a great challenge for both computer and information science students. Fear of programming, lack of interest and the abstract nature of programming concepts are main causes of the high dropout and failure rates in introductory programming courses. With an aim to motivate and help students, a number of researchers have proposed various tools. Although it has been reported that some of these tools have a positive impact on acquiring programming skills, the problem still remains essentially unresolved. This paper describes LightAlgo, a tool for visualization of sorting algorithms. LightAlgo is an easy-to-set-up and fully automatic visualization system with step-by-step explanations and comparison of sorting algorithms. Design principles and technical structure of the visualization system as well as its practical implications and educational benefits are presented and discussed.

INDEX

1. Introduction

Now-a-days 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. Since I faced the problems of sorting during the course of algorithm design in the first year of our studies, there is an understanding that the visual representation is a vital part of the studying process. During working on the project it was very exciting to learn different techniques of sorting algorithms into the depth. The main goal of the project 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. The demonstration software is made in a user-friendly and easy-to-use style. To gain maximal benefit from learning you can try each sorting algorithm on your data. The text of the thesis describes principles of the most known sorting algorithms which are demonstrated in the computer program. It might be used as a source for learning algorithms by students. Also, the program might be easily used as a demonstration by lecturers and tutors during classes. Besides, there is programmer documentation and user guide to the provided software. Readers of this text are expected to have some programming experience to know basic data structures such as arrays, lists, trees and understand recursive procedures. Also, knowledge of some simple algorithms and their implementations could be helpful. In order to understand the topic better, knowledge of linear algebra and calculus is involved.

Sorting Visualizer a GUI based python program to visualize common Sorting Algorithms. This project is built using HTML, CSS, and JS. This project sorting visualizer is a very simple UI and it allows the users to select the sort algorithm, select the array size, and speed of the visualization. The main reason we chose this project to become more familiar with the javascript concepts, and CSS styling. So, we didn't use any frameworks other than HTML, CSS, and JS. We have learnt sorting algorithms like bubble sort, selection sort, insertion sort, quick sort. But often we fail to understand the core idea of a particular algorithm maybe because we are unable to visualize how they work. So the most important thing to understand about these algorithms is visualization. That’s why we are making this project to let everyone understand how these algorithms work and through this thesis you also will get a deep understanding of such sorting algorithms. This project is a good start for beginners and a refresher for professionals who have dabbled in data structures and algorithms using Javascript before and also web developers. The methodology can be applied to showcase any algorithm of one's choosing. We wanted the animation to be web-based to appeal to a wide spectrum of people using different technology media. This way, the user would not need to worry about installing special software or trying to organize configurations to use the tool. The webpage is coded with HTML5 (Hypertext Markup Language, Version 5), JavaScript, and CSS (Cascading Style Sheets). The physical elements of the webpage (buttons and layout) are coded with very minimal HTML5 code. The next biggest contributor would be the CSS code, which is responsible for the appearance and behavior of the buttons and text. Finally, the rest is devoted to JavaScript, responsible for the histogram generation, movement, algorithm design, and sound. All the buttons refer to designated parts of the JavaScript code to perform the tasks.

2. Literature Survey

On previous studies, the UI was pretty decent but not gave proper context of the algorithm i.e. bar representation. The bar doesn't give the proper information about a problem. So we will try to implement such feature where user can give his input and we will figure out how to work with it and give him the desired output. We will Provide more features like custom elements, values, array size control or different UI for visuals i.e. shapes or we will give the static value to the elements so the user can get perfect idea. The only thing here is how many percentage we can simplify the UI i.e. shapes or other UI components so it will be easy to understand. Some previous projects on this topic had very dull design for the website and there was no attraction towards the working of actual function of visual. It minimizes the interest in the visuals. We will figure it out and provide an interactive website with good design, animations and the main visual function. The data structure complexities and definition can be also provided on the below of the visual. Which will make the website fully functional with proper information. In this milestone the basic structure of this website will be made. challenges for us are through out this are- 1. Site responsiveness - To bring responsiveness to the site, we have gone back to learn CSS styling. It is like going back to the time when we was learning the CS fundamentals and web technologies as a beginner. 2. Writing sort algorithms in JS - And coming to the second challenge we are going to face are writing sort algorithms in JS. When we want to write some code or learn algorithms for solving coding problems, we usually prefer Python/Java. Initially, we will take more time to implement the first sort algorithm i.e., Bubble sort to if it perfectly works. After that, for the remaining algorithms, we can start implementing them so quickly one after the other. Some few studies as below

Java And Web-based Algorithm Animation (JAWAA) JAWAA [Pierson and Rodger, 1998] uses simple command language to create animations of data structures and display them through a web browser. The system is built on JAVA so it can be used on any machine. Animation to be created is written in simple script language. A text file can be easily transcribed into any text editor or can be generated as an output. Texts have one command or image function per line. The JAWAA applet returns the command file and executes it. The system interprets commands by line and performs specific functions for each command. JAWAA provides commands that allow users to create and move both archeological objects such as circles, rows, text, rectangles, etc., and data structure objects such as columns, stacks, rows, columns, trees, graphs etc. The interface of the system contains animation. canvas and panel that gives users controls such as start, stop, pause and step animation. To control the animation speed the scroll bar is also provided.

In 2008, paper “AlCoLab: Architecture of Algorithm Visualization System” concerns the style of script supported algorithm visualization systems for educational purposes, focusing on the support and the improvement that those systems provide in the process of teaching of an conceptual subject such as algorithms. In 2019, paper “Towards Developing an Effective Algorithm Visualization Tool for Online Learning” reports a work-in-progress research project at Athabasca University on developing an effective algorithm visualization tool for online learning. In 2019, paper “Open Interactive Algorithm Visualization ” presents a work-in-progress project form developing an open interactive algorithm visualization website. In 2021, paper “AlgoAssist: Algorithm Visualizer and Coding Platform for Remote Classroom Learning” focuses on "algorithm visualization", which allows a better understanding of its flow and operation. It supports the combination of the lab into a single application dedicated to pre-assessment, algorithm explanation, visualization, coding, and post-assessment. In 2021, paper “Algorithm Visualizer ” aims to simplify and deepen the understanding of algorithms operation. Within the paper we talk about the possibility of improving the standard methods of teaching algorithms, with the algorithm visualizations.

After the implementation and studying previous researches and works, We figured out that our work was easy to function with rather than some other works i.e. Visualgo etc. They have given so many data structures to interact with but a normal user cant figure it out at first place. Although There are different functions like create random array or actual working of algorithm along with visuals which we tried to implement but that was quite hard and we did not had that much time to work along. So we kept our interface for search and sort simple and clean so a user can easily understand it by providing less but effective features. In the final work, We have given user to select of various sort or search techniques in the same page with a beautiful UI we should say. For us the main factor is the UI is the main thing we will compare with other works on this project. We have seen some dull UI in some of the works. Why our focus was on the dark or RGB/Neon theme was it looks very attractive to us or we will say to everyone. It has its own benefits of its not very hazardous to our eyes instead of the white bright colors and many more. It feels very smooth and if we are comfortable with something that means we can understand and work with it well.

3. Objectives and Proposed Work

3.1 Objectives

This project is for educational purpose. The main objective of this project is to help beginners and students to be able to visualize the basic algorithms and get a better understanding of the underlying operations. With the shift of remote and digital literacy, a combined platform serving the effective literacy requirements of students is required. Algorithm Visualizer is a combined platform that's a comprehensive result for educators and students to educate and learn online effectively. It substantially focuses on “algorithm visualization", which allows a better understanding of its inflow and operation. In future we are going to add other algorithms to get more understanding about algorithms to students and teachers. Also, we are going to add more related stuff about algorithms like tutorial videos, assignments and examples to get more knowledge about algorithms. Creating the website's User Interface (UI) using HTML, CSS and enhancing it further using Bootstrap; without actually implementing any of the app's core features. Implementation of animations, effects and core functionalities (sorting algorithms) using JavaScript. Publish to GitHub and host your project live using Netlify.

3.2 Proposed work

3.2.1 Problem with existing system

There are some websites to learn algorithms working but We developed this project to get easy understanding about algorithms using GUI. There are some drawbacks of those websites are as follows:

1) Complex way to understand
2) Worse Graphical User Interface
3) Not user friendly
4) Default range of elements
5) Requires more bandwidth

3.2.2 Need of proposed system

Over the years we've observed that algorithms even tough being a complex subject are the foundation of computational thinking and programming skills of a student. So to ease up the hardships of students this idea of the project was formed.

Our application Algorithm Visualizer is both interactive and alluring to students.

As we’ve observed a lot of students face difficulties while solving the real time Data Structures & algorithmic questions. As DSA is one of the important part for the Realtime Project Development & It gives the students hands on experience of the algorithms' implementation 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.

3.2.3 Features

1) Attractive UI for better understanding
2) Responsive code IDE
3) Platform Independent
4) Understanding Concepts through videos and documents
5) User friendly
6) Take less bandwidth
7) Free to use

4. Modules and Design

4.1 Modules

1. Home -

About - On this page user can see the basic introduction of our project and if he wants to contact or know about us more he can click our socials. This page is basically for giving a user a basic idea of our work and about us.

Documentation - The resources about the project are available here, Anyone can download it and further do study on it. The main idea was behind this was make this project open sourced.

More - We tried to implement some other data structures here which can make our web app more usable to the user. i.e. Stack, Binary Tree etc.

1) Stack – In this, We have given user to actually enter his random values and execute push and pop functions on the stack and can see visually how it gonna work. Their is also a prompt at bottom which triggers and prompts some message at particular scenarios. i.e. empty stack, nothing to pop etc.

2) Binary Tree – As in the name user can create his own tree by entering the nodes values and search for specific node by various traversal techniques.

2. Choice -

Search – In the choice page, We have given 2 options for search and sort functions which was our main objective to cover. In this module there are 2 basic types of search which are

Linear Search - In computer science, a linear search or sequential search is a method for finding an element within a list. It sequentially checks each element of the list until a match is found or the whole list has been searched.

Binary Search - In computer science, binary search, also known as half-interval search, logarithmic search, or binary chop, is a search algorithm that finds the position of a target value within a sorted array. Binary search compares the target value to the middle element of the array.

Sort – In the sort module, There are 4 options which user can choose from between

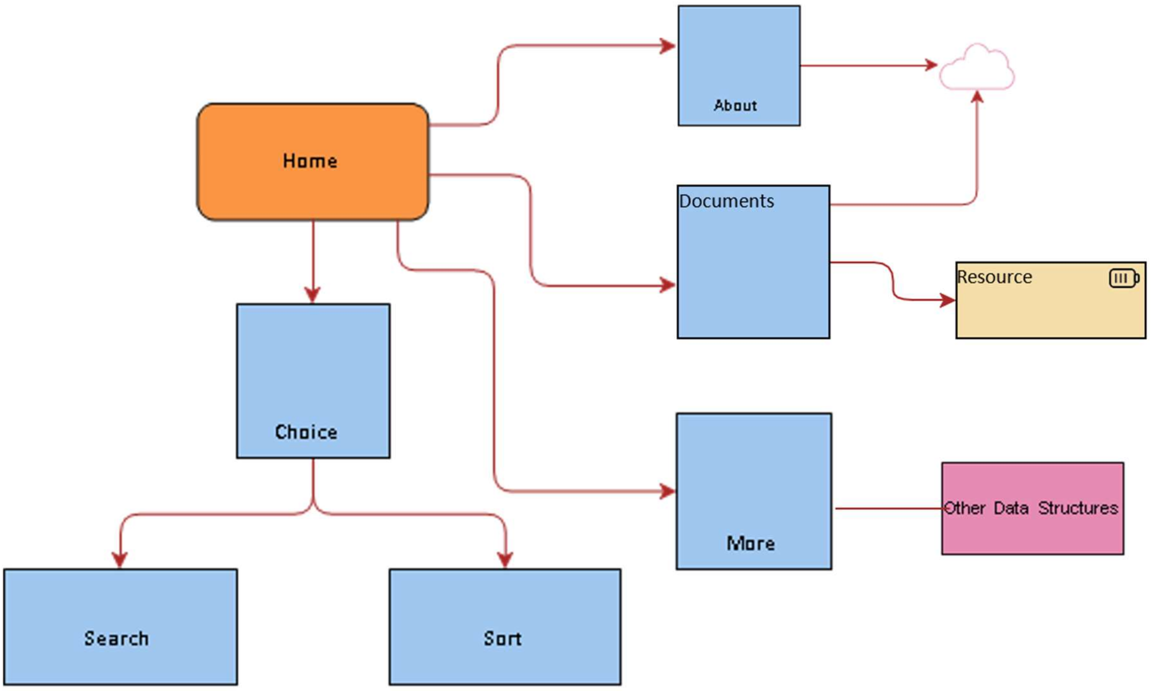
Bubble - Bubble sort, sometimes referred to as sinking sort, is a simple sorting algorithm that repeatedly steps through the input list element by element, comparing the current element with the one after it, swapping their values if needed.

Selection - In computer science, selection sort is an in-place comparison sorting algorithm. It has an $O(n^2)$ time complexity, which makes it inefficient on large lists, and generally performs worse than the similar insertion sort.

Quick - Quicksort is a divide-and-conquer algorithm. It works by selecting a 'pivot' element from the array and partitioning the other elements into two sub-arrays, according to whether they are less than or greater than the pivot. For this reason, it is sometimes called partition-exchange sort.

Merge - The Merge Sort algorithm is a sorting algorithm that is based on the Divide and Conquer paradigm. In this algorithm, the array is initially divided into two equal halves and then they are combined in a sorted manner.
User can pick any algorithm from the above list of available algorithms. Algorithms are chosen on the basis of their popularity and importance according to the existing curriculum. Designing For proper visualization we have used different colors to differentiate between the sorted bars and unsorted bars and even for bars which are currently in the process of comparison and sorting. After sorting the colors of bars will change to same color which will be different from initial colors of array and array elements will be arrange in ascending order.

4.2 Design



4.3 Architecture diagram for the website

As you can see in the diagram, The main modules are given and the relationships between them are showed up by lines.

As we start from Home we can go to about, documents and more section and redirect back to home. And also the about and documents are provided with some external links for resources in the internet. In the More page, We tried to implement some extra data structures like stack and trees for just future purpose. If we worked on this project to make it 100% perfect so how many data structures and what can we implement to make this better? Such points we will try to fit in more section.

Choice section gives us 2 options i.e. search and sort

Search – Binary and Linear

Sort – Bubble, Quick, Merge and Selection

All the pages have options for redirection to the Home page and choice page in case user wants to do something else.

5. Implementation

Home page -

The first component that we implemented was the home page. For this 1st we thought about the theme. A theme which can be perfect for our whole project and it will give us a advantage over other works. Then after some study, We found out dark theme was perfect with its own benefits. So what would be the color scheme with the dark UI? Neon colors! This combination was very good to watch as user view. We even asked our fellow classmates about how its looking they said it was very good and smooth.

So starting from the home page, We gave 3 options to the nav bar which were About, Documentation and More. The Home page itself represents our project. The effect on the "ALGORITHM" logo gives techno vibes and the bg image shows that we are visualizing the whole universe.

Documentation -

There are 3 images of what we are providing to user for future work of his. They are our ppt for the project which we presented to our mentor. The pdf of synopsis and report where he can see the actual steps what we followed and last the github, Where our repository is available from which user can download and work on the project itself. Which, This all makes the project open source.

Bellow the page, There are some our references link from which we gathered basic data for our project.

About -

On this page there is a small paragraph about our project which gives us basic info of what actually it is and its working. Then bellow that there is a mail where user can contact us if have some query or anything to discuss about the website. Then our socials are provided at end where we are generally active.

More -

So there was no plan of making this but the idea behind this was what if we want to expand this project in future. So on what can we or what could be implemented if we make this mini project for bigger platform. What things can make it perfect. Such types of statements we discussed and the answers where some more data structures like Tree, Graph or basic Stack

1) Stack -

For stack, After clicking create stack button, There is a input bar where user can enter the values and if he pushed the push button the element will be pushed in the array and visually in the stack which he can see there. Works on FIFO manner, he can push as many elements he wants. The main thing is the bellow prompt which prompts the current action of the operations which are going like “x element was pushed in stack” or if we tried to pop empty stack then “stack is empty nothing to pop” and some more. The algorithm behind this same as real stack the push inserts the user's element to the array and the array is mapped with the visuals and pop removes current pushed element from array which leads removing it also from the visual. The prompt just works on actions and conditions i.e. if array is empty and user clicks pop button then the “empty stack nothing to pop” statement will appear on the prompt and much more.

2) Binary Tree -

We kept this simple, The add button will add the elements then the elements will be placed in circular <div> and the div's will be connected by lines. The main solution of all this visualization was JS. Js did all of our job of the project. We can directly map the java algorithm with the visuals and perform the operations. For tree, we can search particular element and find its path by highlighted lines.

Search -

As I said in previous paragraph, The Js did all our job easy for mapping the algorithm with the visuals. There are 2 searches which we implemented was linear and binary.

In linear, The working is you will be given an array and you have to search for particular element by comparing it with every other element in that array. So what we did is created individual <div> for every element and gave static values on the top of the divs. In the input box, User can enter the value he wants to search and if founds on the array the element <div> will be highlighted and a message will pop “Element found” and vice versa. In the visual, We can see how the comparing works. For that we gave different color to the compared element and different to the searched element.

On the other hand, The binary search the concept is you will be given a sorted array from which you have to search for particular element. So we statically provided an array which is already sorted with assigned values. The thing is binary search works like, Low + High = Mid then compare the value with Mid, If its small Then Mid will be the new High else the Mid + 1 will be new low and repeatation of this. The visualization show how it select high, mid then new high mid and operates further same like previous search in other aspects.

The reset buttons randomizes the array everytime for better understandings and also stops the current working.

Sort -

Sort was our first component after the home page to be implement. The working is same as explained in same each sorts algorithm will be mapped with the visuals i.e. <div>. Here we implemented 2 more features that were, Speed and Size. The Speed will control the visualization speed and the Size will manage the array size i.e. 10 default and 40 max. The concept for each sort is

Bubble - compares the linear elements with all further elements and swaps according to their size.

Selection - Brings smallest element to first.

Quick – Selects an element as pivot and compares other elements with it and further more.

Merge – Breaks down array into parts then reassembles it by comparing the arrays.

So the backend for these are same as their algorithm, In the code you can see the algorithm for each sort the only difference is the array[i] means array of elements are just replaced with array of divs and performed the actions on that. The res9ets the page or stops the working and generate new array, Creates new array.

From each module i.e. Search and Sort redirection to Home and Choice is done for user direction.

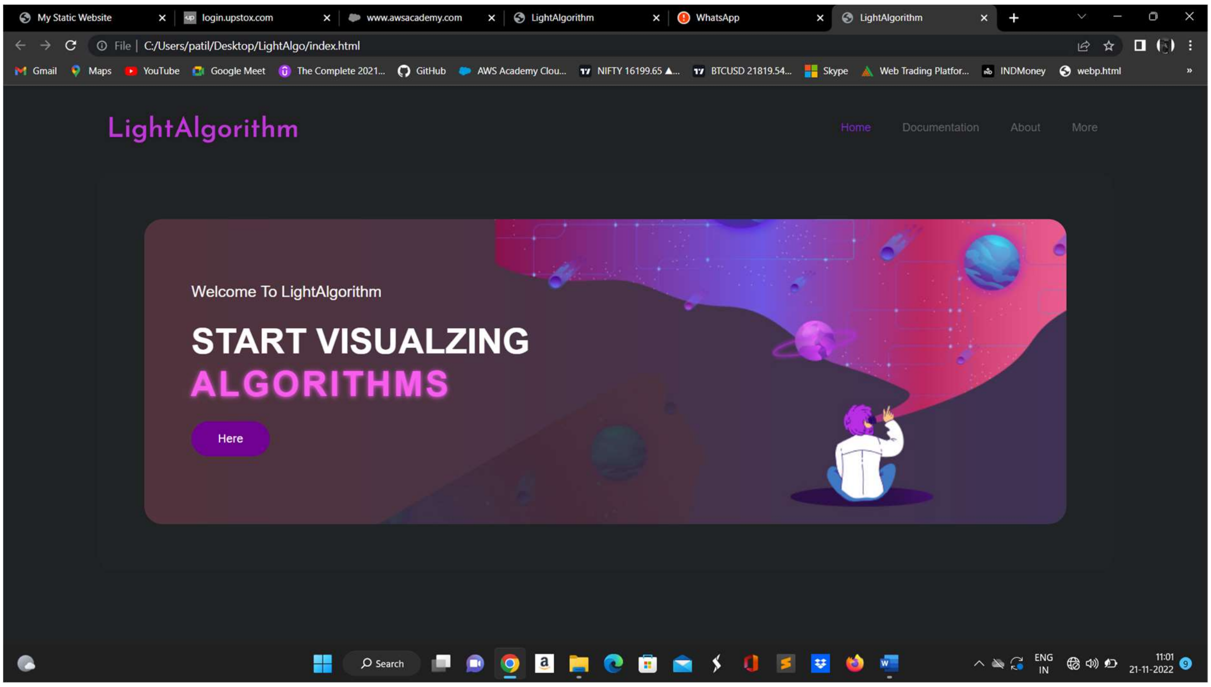
The final thing is, We've developed this project locally and its web app then how can we deploy it so now others can see it from anywhere using there browser. For this Github came handy, It did our 2 jobs 1st storing all over code which we can provide to user and 2nd deploying our code as a Git pages i.e. an url. Which can we send to anyone and he can see the output on the browser screen.

Also, Responsiveness was an important part of this because we are making a web app and user can view it on any device any browser etc. That time bootstrap was the solution to this. The size of the components will auto place according to their screen ratio and alignments will be done. So basically there no need of any requirements for the user just a browser is enough for it.

Github repo - https://github.com/Vikypatil1/LightAlgo

Result

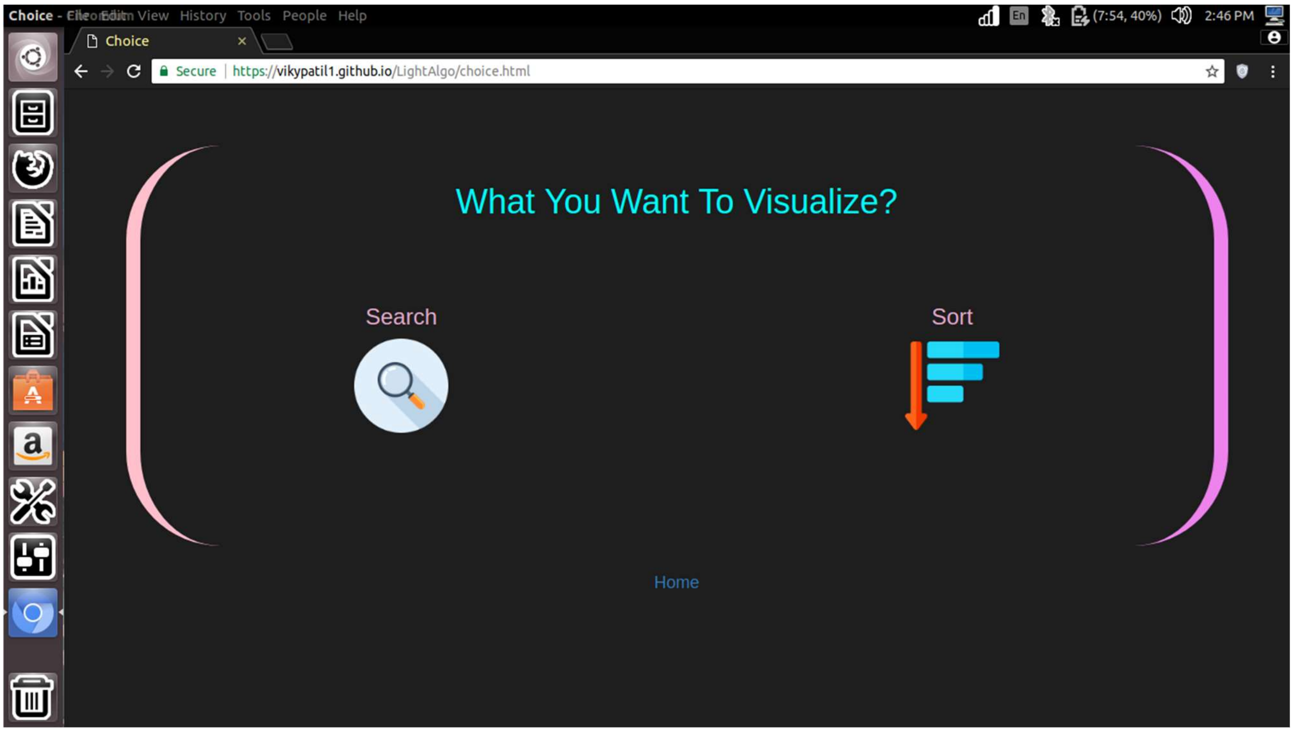
Website - https://vikypatil1.github.io/LightAlgo/index.html



Welcome To LightAlgorithm

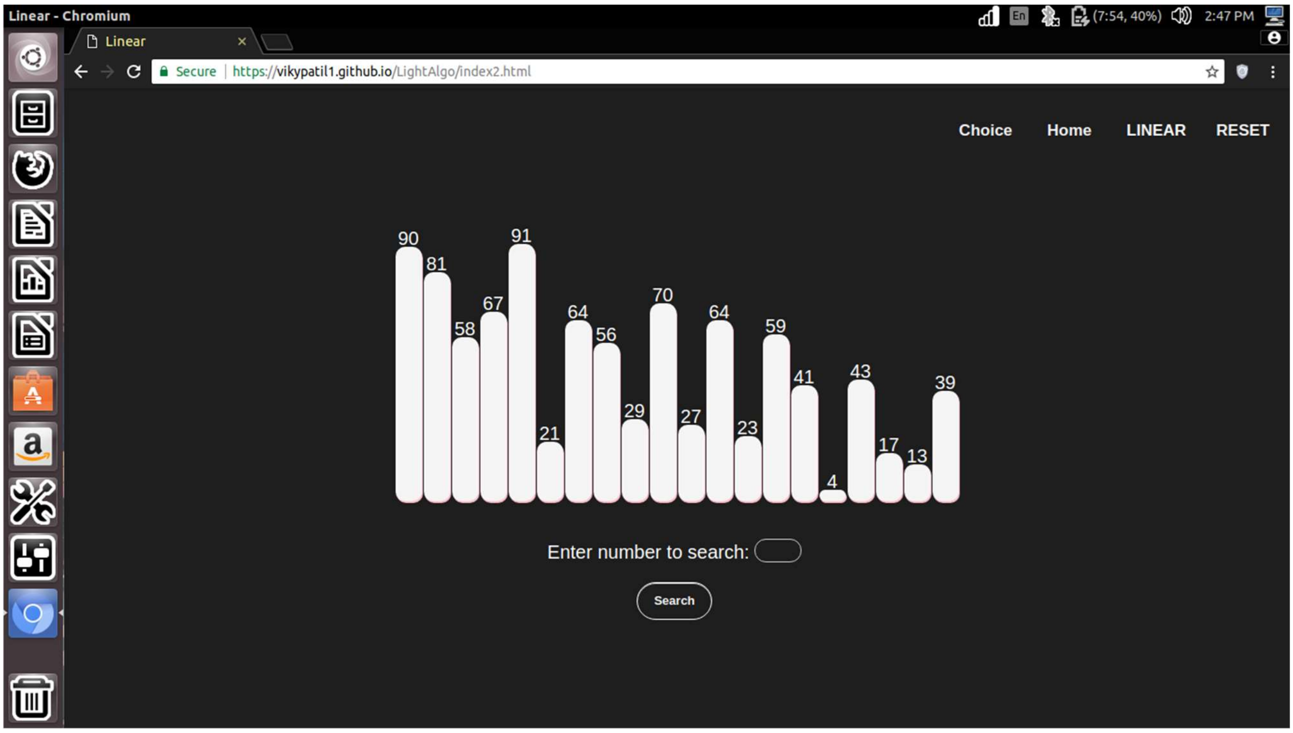
START VISUALZING ALGORITHMS

Home Screen

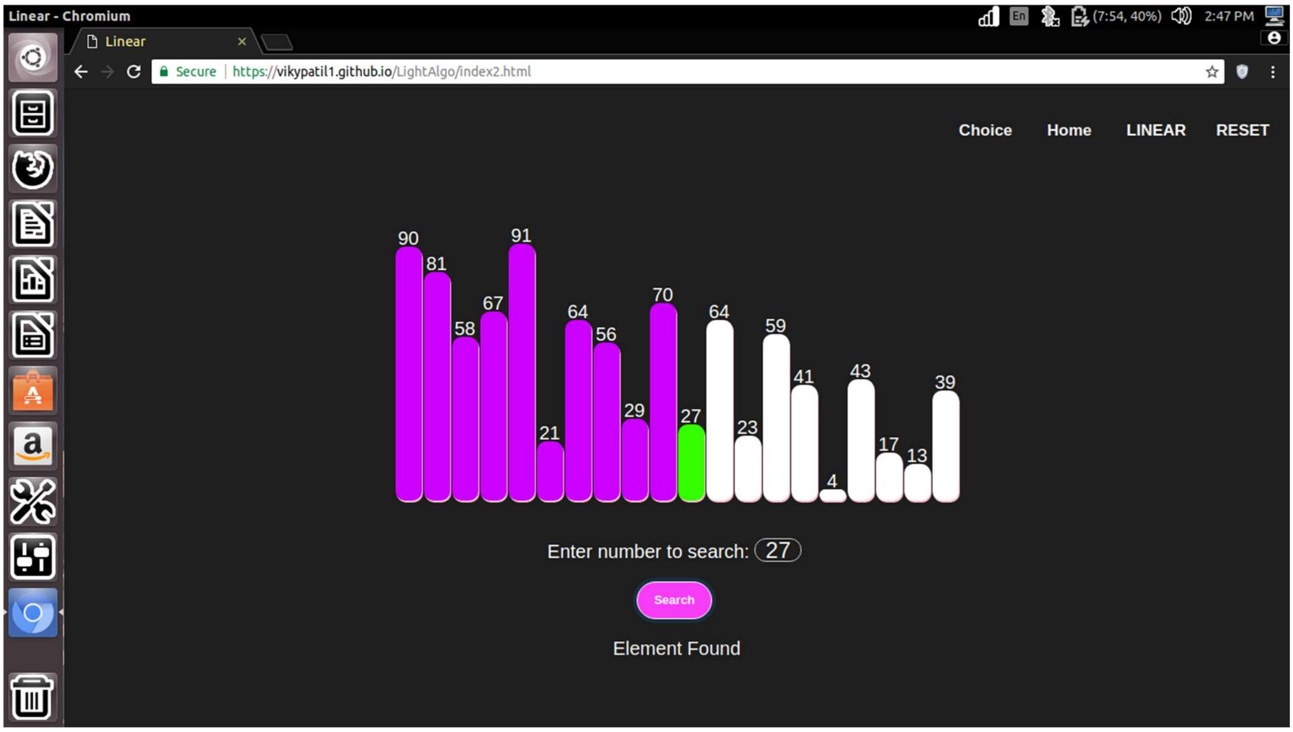


What You Want To Visualize?

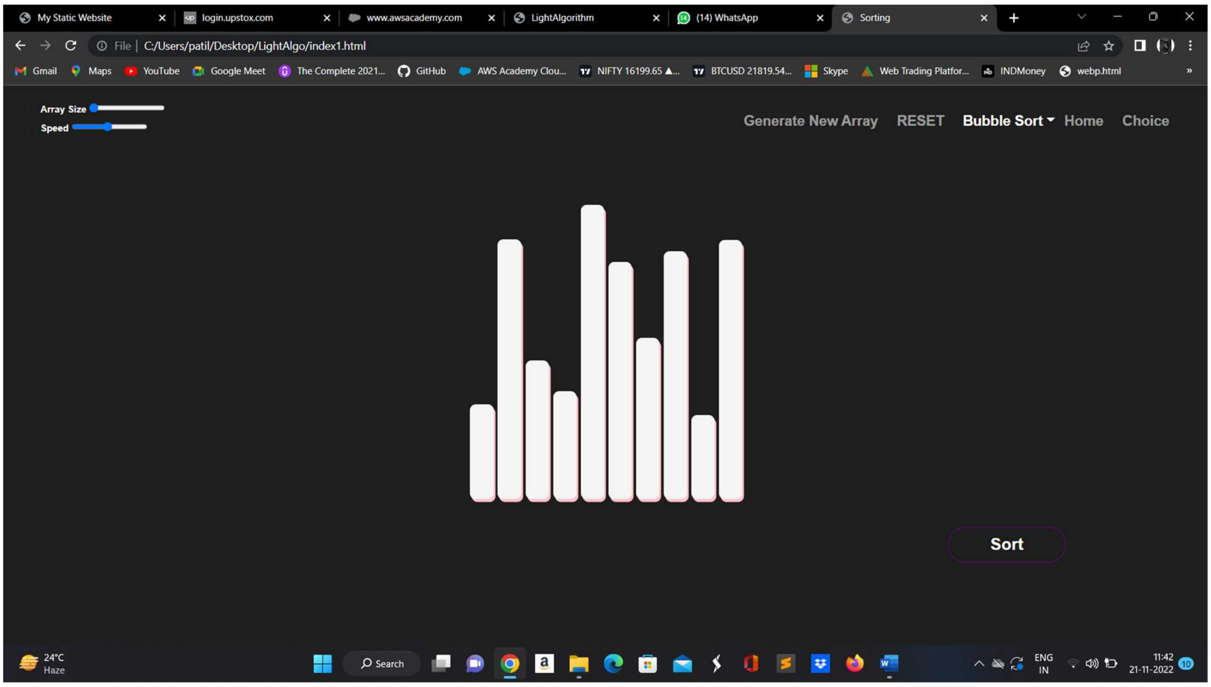
Choice



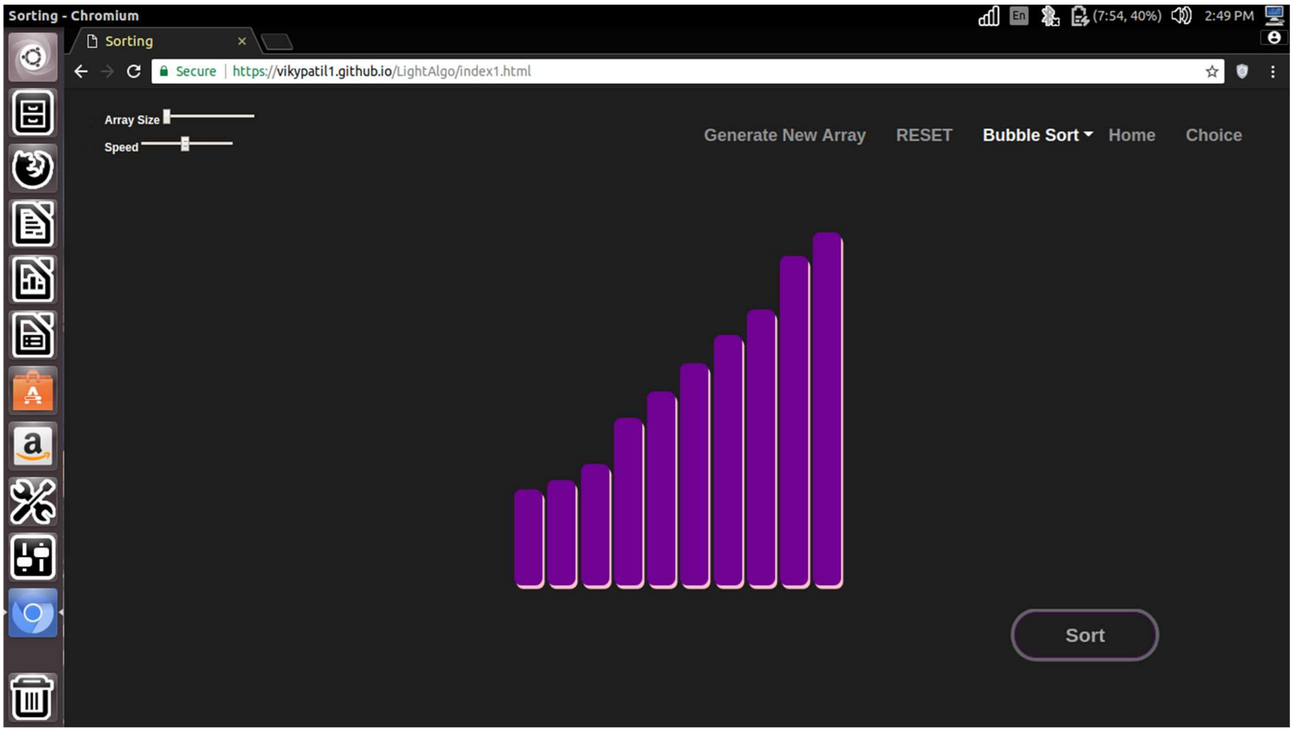
Search



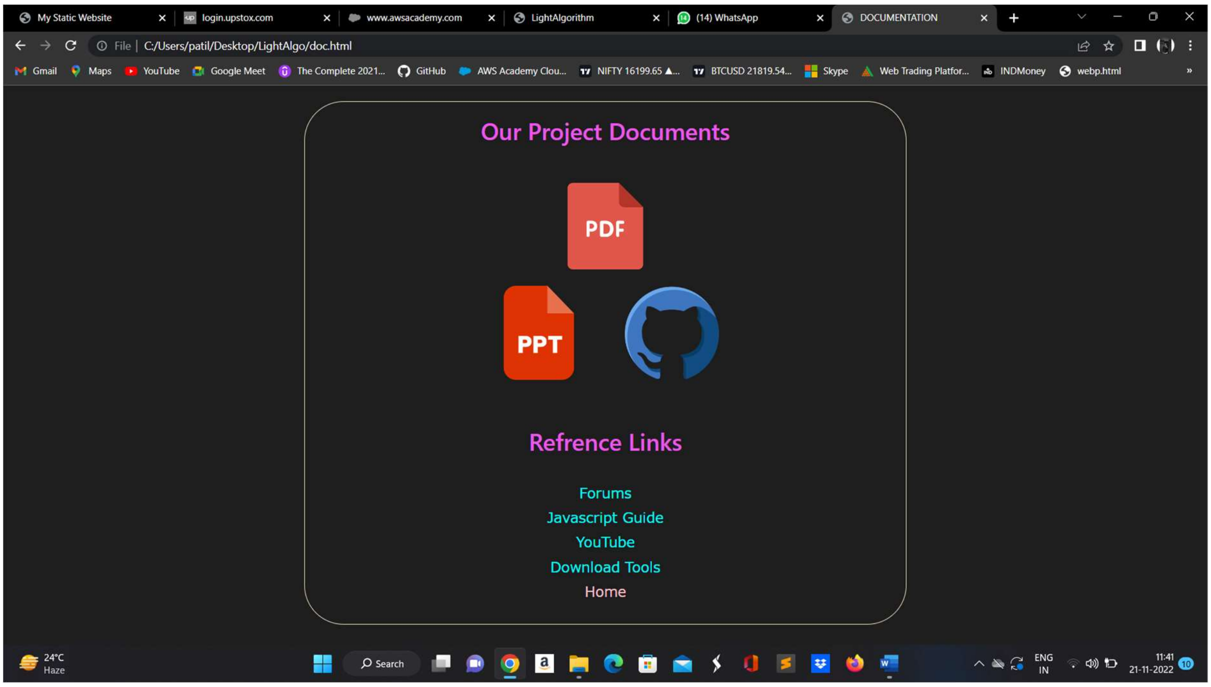
Successful search



Sort



After sort

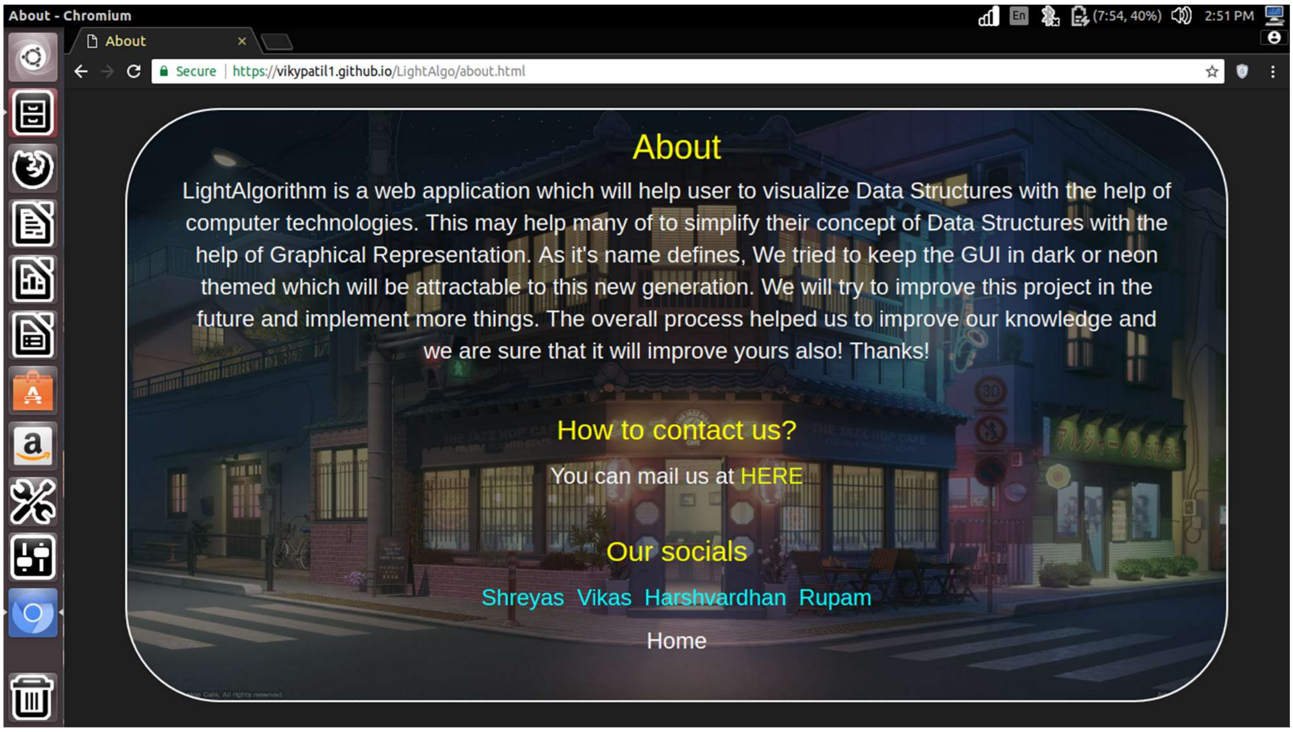


Our Project Documents

Refrence Links

Forums
Javascript Guide
YouTube
Download Tools
Home

Documentation



About

LightAlgorithm is a web application which will help user to visualize Data Structures with the help of computer technologies. This may help many of to simplify their concept of Data Structures with the help of Graphical Representation. As it's name defines, We tried to keep the GUI in dark or neon themed which will be attractable to this new generation. We will try to improve this project in the future and implement more things. The overall process helped us to improve our knowledge and we are sure that it will improve yours also! Thanks!

How to contact us?

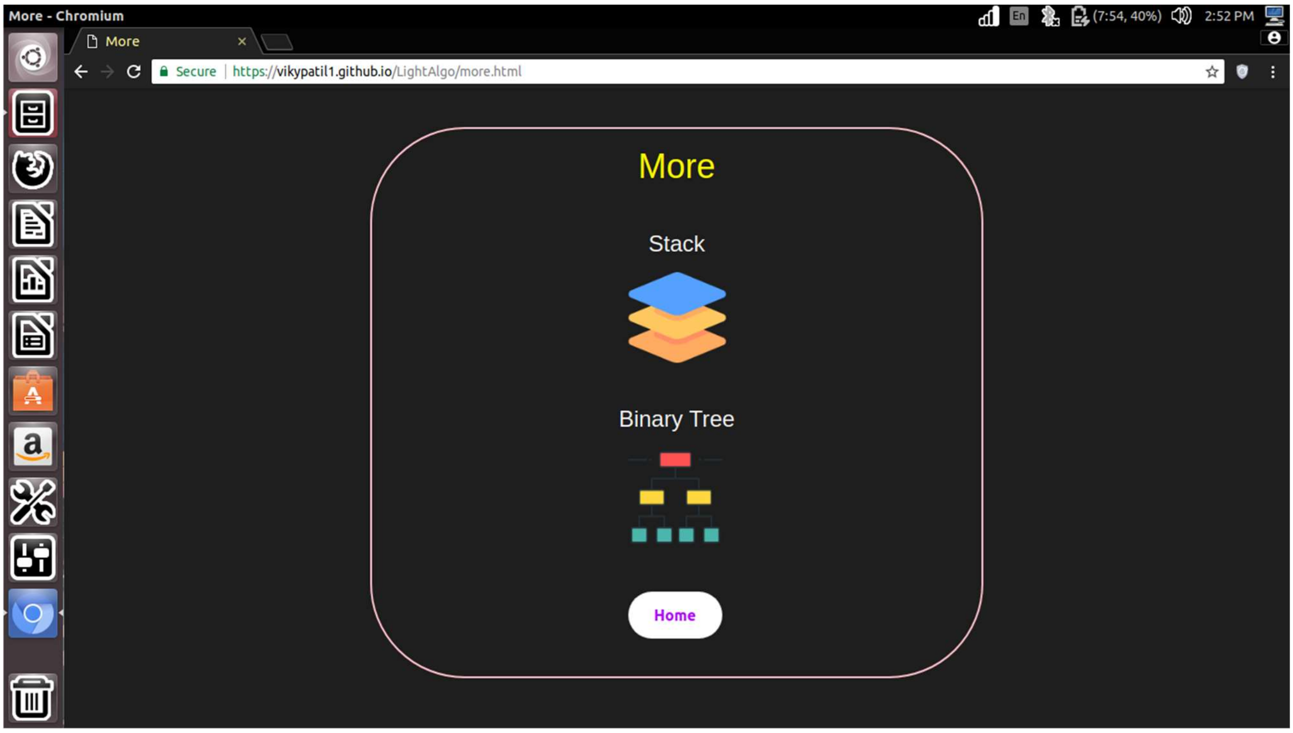
You can mail us at HERE

Our socials

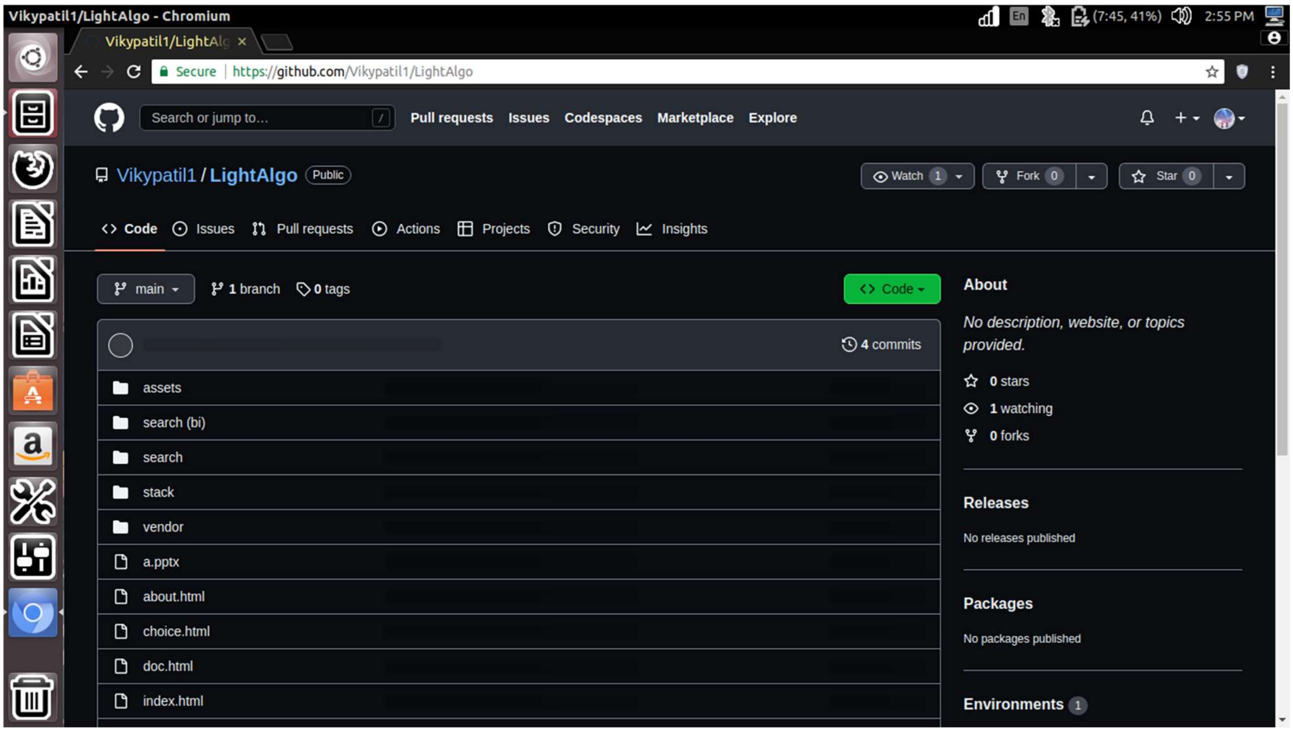
Shreyas Vikas Harshvardhan Rupam

Home

About



More



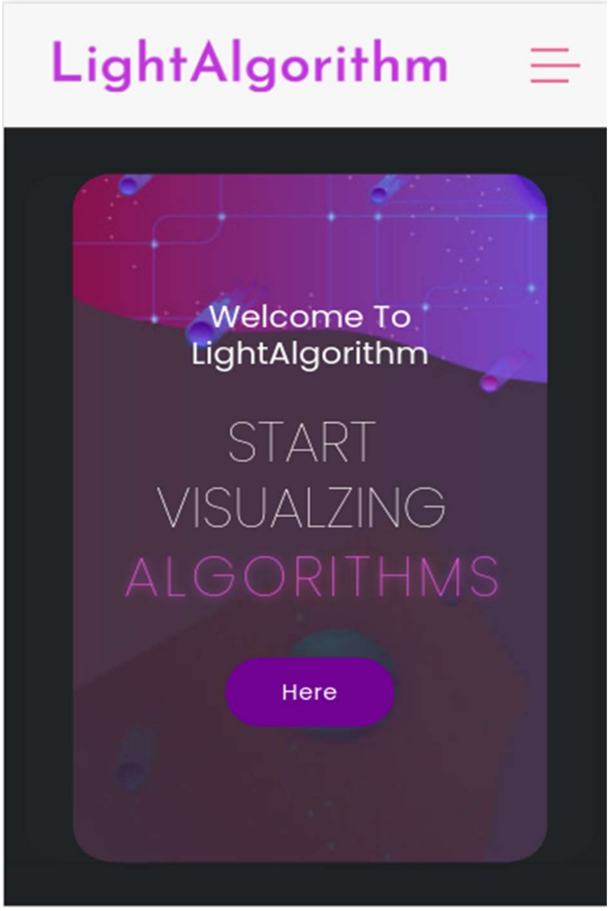
No description, website, or topics provided.

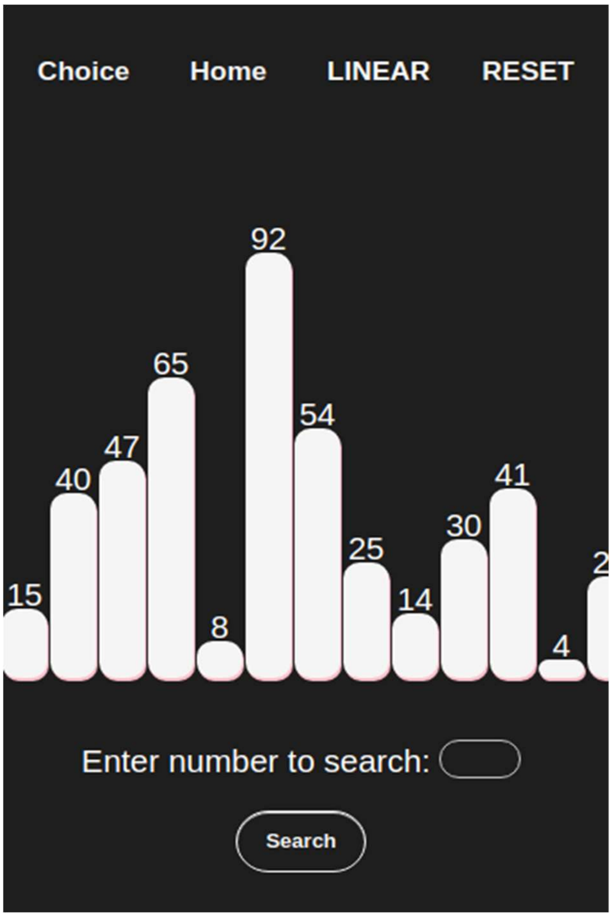
No releases published

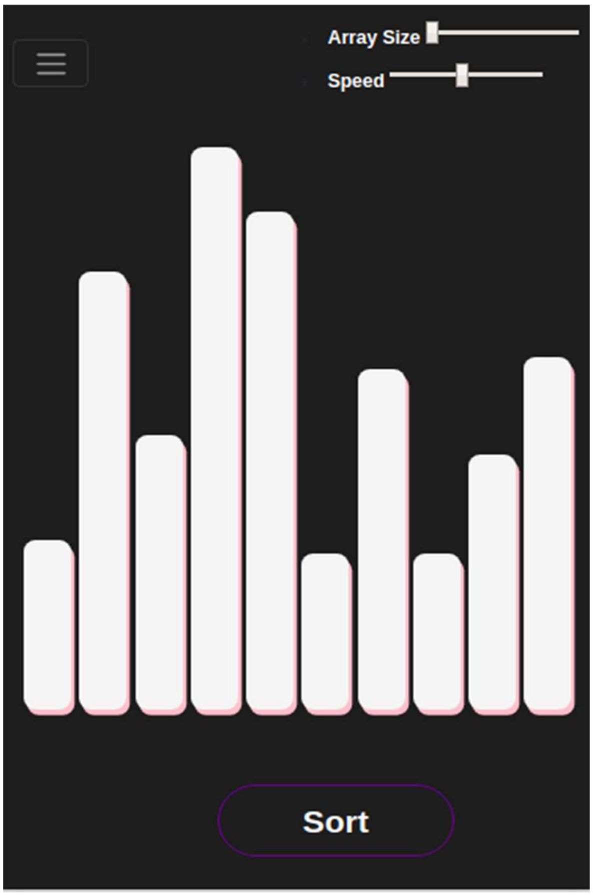
No packages published

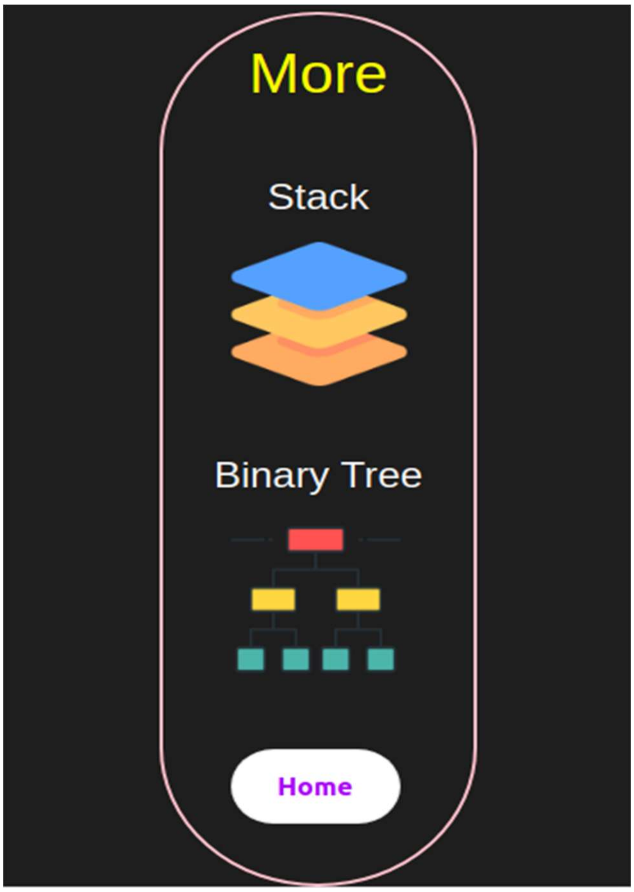
Github Repo

Mobile Views









6. Conclusion

We started our project by studying a number of the well-known algorithm visualizations that are developed over a few years. According to our findings, algorithmic visualization is often seen as a valuable supporting tool, utilized in addition to straightforward ways of education within the field of computer science. With the execution of this project, we have got with success attain our objective of our project is to sorting and searching techniques with Visualization and differentiate their performance. As is the case with most other teaching areas, there has been a major gap between the idea and practical understanding of algorithms realization. The main goal of the project is to use it from research educators and students for teaching and studying the existing known searching, sorting, stack and tree algorithms. The main plan of the system is to provide an associate educational environment for both instructors and students to facilitate the learning process in economical way.

7. References

Some papers-

1)https://theses.cz/id/3w4s3a/Mykhailo-Klunko-bc-thesis.pdf

2)https://www.irjmets.com/uploadedfiles/paper/volume3/issue_7_july_2021/14236/1628083553.pdf

3)https://www.theseus.fi/bitstream/handle/10024/507342/Bikram_Karki.pdf;jsessionid=E1EE7681B67796961237C7B8258787FF?sequence=2

4)https://people.cs.vt.edu/~shaffer/Papers/SIGCSEWiki.pdf

Guides-

1) https://www.crio.do/projects/javascript-sorting-visualiser/

2) https://github.com/topics/algorithm-visualization

Technologies-

1) https://www.geeksforgeeks.org/html-basics/

2) https://www.w3schools.com/js/

3) https://getbootstrap.com/docs/5.0/getting-started/introduction/

Components-

1) https://nicepage.com/html-templates/

2) https://freefrontend.com/css-forms/

3) https://codepen.io/collection/GZnZWD