

ALGORITHM VISUALIZER

Project report in partial fulfilment of the requirement for the award of the degree of

Bachelor of Technology

In

COMPUTER SCIENCE

Submitted By

NAMES	ENROLLMENTS NUMBERS
BISWARUP BHATTACHARJEE	University Enrolment No. 12019009023003
SUBHAJIT PATI	University Enrolment No. 12019009023006
RAHUL DEBNATH	University Enrolment No. 12019009023001
ARIJIT GOSWAMI	University Enrolment No. 12019009001175
SUBHADIP MAJI	University Enrolment No. 12019009001345
ANKITA SIKDER	University Enrolment No. 12019009023005

GROUP NO. 42

Under the guidance of

PROF. SUDIPTO KUMAR MONDOL

Department of Computer Science



*UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA University
Area, Plot No. III – B/5, New Town, Action Area – III, Kolkata – 700160.*

CERTIFICATE

This is to certify that the project titled **ALGORITHM VISUALIZER** submitted by **BISWARUP BHATTACHARJEE** (University Roll No. **12019009023003**), **SUBHAJIT PATI** (University Roll No. **12019009023006**), **RAHUL DEBNATH** (University Roll No. **12019009023001**), **ARIJIT GOSWAMI** (University Roll No. **12019009001175**), **SUBHADIP MAJI**(University Roll No. **12019009001345**) and **ANKITA SIKDER**(University Roll No.**12019009023005**) students of UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA, in partial fulfilment of requirement for the degree of Bachelor of Computer Science, is a bonafide work carried out by them under the supervision and guidance of Prof. **SUDIPTO KUMAR MONDOL** during 4th Semester of academic session of 2020 - 2021. The content of this report has not been submitted to any other university or institute. I am glad to inform that the work is entirely original and its performance is found to be quite satisfactory.

← →
Prof. Sudipto Kumar Mondal
Assistant Professor
Department of Computer Science
Science

↓
Prof. Sukalyan Goswami
Head of the Department
Department of Computer

ACKNOWLEDGEMENT

We would like to take this opportunity to thank everyone whose cooperation and encouragement throughout the ongoing course of this project remains invaluable to us.

We are sincerely grateful to our guide Prof. **Sudipto Kumar Mondol** of the Department of Computer Science, UEM, Kolkata, for his wisdom, guidance and inspiration that helped us to go through with this project and take it to where it stands now.

We would also like to express our sincere gratitude to Prof. **Sukalyan Goswami**, HOD, Computer Science, UEM, Kolkata and all other departmental faculties for their ever-present assistance and encouragement.

Last but not the least, we would like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

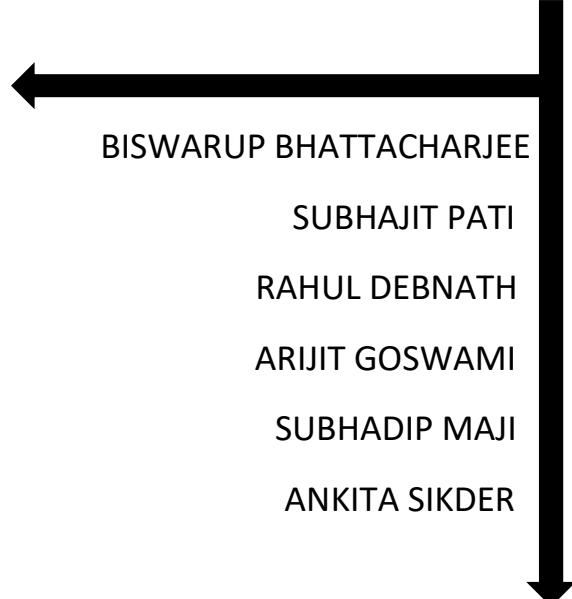


TABLE OF CONTENTS

ABSTRACT	→	<i>pgNo. 5</i>
<u>CHAPTER - 1 :</u>		
INTRODUCTION	→	<i>pgNo. 5-6</i>
<u>CHAPTER - 2 :</u>		
LITERATURE SURVEY	→	<i>pgNo. 6-8</i>
<u>CHAPTER - 3 :</u>		
PROBLEM STATEMENT	→	<i>pgNo. 9</i>
<u>CHAPTER - 4 :</u>		
PROPOSED SOLUTION	→	<i>pgNo. 9-12</i>
<u>CHAPTER - 5 :</u>		
EXPERIMENTAL SETUP AND RESULT ANALYSIS	→	<i>pgNo. 13-20</i>
<u>CHAPTER - 6 :</u>		
CONCLUSION & FUTURE SCOPE	→	<i>pgNo. 21-22</i>
BIBLIOGRAPHY :		
	→	<i>pgNo. 22</i>

ABSTRACT

Algorithm Visualizer is an interactive online platform that visualizes **algorithms** from code. Learning an **algorithm** gets much easier with visualizing it. **Algorithm-visualizer** is a web app written in React. It contains UI components and interprets commands into visualizations. Algorithms are a fascinating use case for visualization. To visualize an algorithm, we don't merely fit data to a chart; there is no primary dataset. Instead there are logical rules that describe behaviour. This may be why algorithm visualizations are so unusual, as designers experiment with novel forms to better communicate. This is reason enough to study them. The main objective of this is to ensure effective and reliable approaches for visualization of various algorithms. Using this web application anyone can learn algorithms fast and easily.

INTRODUCTION

This is a web application which is used to visualise algorithms. It has been created using JavaScript, CSS, SCSS, ReactJS, HTML5, Vercel, Chat Engine XDK, nodejs and Formik. User can approach this website through a provided link. First of all it has a sign up or login page. If user is opening it for first time he has to select sign up option. Here he has to give username, email id, password, verify password. If he has already signed up, then he has to go to login option and then he has to give email, password, confirm password. Then you have entered in the main page or home page. Here user can see your email id and a firebase-provided password. Here four buttons are given. VISUALIZE ALGORITHMS, ALGORITHMS, DISCUSSION, FEEDBACK. By clicking on VISUALIZE ALGORITHMS user can watch and understand working of some specific algorithms. If user goes to ALGORITHMS he can see total 72 algorithms. Sorting, Searching and Others algorithms. If he clicks any of them in learn more button then their details will be shown in a page. The DISCUSSION part will take the user to a login page where he has to give username and password (firebase-provided password in home page). Then he will be added in a discussion group for doubt clearing where the developers will answer his questions regarding this website. In the feedback portion user can write about how much this website is useful or any idea for improvement in this website in the comment section. In mathematics and computer science, an algorithm is a finite sequence of well-defined, computer-implementable instructions, typically to solve a class of problems or to perform a computation. Algorithms are always unambiguous and are used as specifications for performing calculations, data processing, automated reasoning, and other tasks. As an effective method, an algorithm can be expressed within a finite amount of space and time, and in a well-defined formal language for calculating a function. Starting from an initial state and initial input (perhaps empty), the instructions describe a computation that, when executed,

proceeds through a finite number of well-defined successive states, eventually producing "output" and terminating at a final ending state. Here we can see different type of algorithms and their details. The DISCUSSION part will take the user to a login page where he has to give username and password (firebase-provided password in home page). Then he will be added in a discussion group for doubt clearing where the developers will answer his questions regarding this website. Discussion is important for any queries of user. When developers guide the user properly it will very helpful for the user to explore the web site. Website feedback is information obtained directly from website users—through on-page surveys, feedback widgets, and other techniques—to help organizations understand what people think (and how they feel) about their websites and landing pages. This entire web application makes a good practice of visualization-based algorithms study.

LITERATURE SURVEY

Web development is the work involved in developing a Web site for the Internet (World Wide Web) or an intranet (a private network). Web development can range from developing a simple single static page of plain text to complex web applications, electronic businesses, and social network services. A more comprehensive list of tasks to which Web development commonly refers, may include Web engineering, Web design, Web content development, client liaison, client-side/server-side scripting, Web server and network security configuration, and e-commerce development. A web application (or web app) is application software that runs on a web server, unlike computer-based software programs that are run locally on the operating system (OS) of the device. Web applications are accessed by the user through a web browser with an active network connection. These applications are programmed using a client-server modelled structure—the user ("client") is provided services through an off-site server that is hosted by a third-party. Visualization or visualisation is any technique for creating images, diagrams, or animations to communicate a message. Visualization through visual imagery has been an effective way to communicate both abstract and concrete ideas since the dawn of humanity. Examples from history include cave paintings, Egyptian hieroglyphs, Greek geometry, and Leonardo da Vinci's revolutionary methods of technical drawing for engineering and scientific purposes. Visualization today has ever-expanding applications in science, education, engineering (e.g., product visualization), interactive multimedia, medicine, etc. Typical of a visualization application is the field of computer graphics. The invention of computer graphics (and 3D computer graphics) may be the most important development in visualization since the invention of central perspective in the Renaissance period. The development of animation also helped advance visualization. The present world appreciates visualization-based study. Algorithm is a

step-by-step procedure, which defines a set of instructions to be executed in a certain order to get the desired output. Algorithms are generally created independent of underlying languages, i.e. an algorithm can be implemented in more than one programming language. Searching Algorithms are designed to check for an element or retrieve an element from any data structure where it is stored. Based on the type of search operation, these algorithms are generally classified into two categories: Sequential Search: In this, the list or array is traversed sequentially and every element is checked. For example: Linear Search. Interval Search: These algorithms are specifically designed for searching in sorted data-structures. These type of searching algorithms are much more efficient than Linear Search as they repeatedly target the center of the search structure and divide the search space in half. For Example: Binary Search. A Sorting Algorithm is used to rearrange a given array or list elements according to a comparison operator on the elements. The comparison operator is used to decide the new order of element in the respective data structure. It has been created using JavaScript, CSS, SCSS, ReactJS, HTML5, Vercel, Firebase, Chat Engine SDK, nodejs and Formik. JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities. CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files. CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files. React (also known as React.js or ReactJS) is an open-source, front end, JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality. HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and last major HTML version that is a World Wide Web Consortium (W3C) recommendation. The current specification is known as the HTML Living Standard. It is maintained by a consortium of the major browser vendors (Apple, Google, Mozilla, and Microsoft), the Web Hypertext Application Technology Working Group (WHATWG). HTML5 was first released in a public-facing form on 22 January 2008, with a major update and "W3C Recommendation" status in October 2014. Its goals were to improve the language

with support for the latest multimedia and other new features; to keep the language both easily readable by humans and consistently understood by computers and devices such as web browsers, parsers, etc., without XHTML's rigidity; and to remain backward-compatible with older software. HTML5 is intended to subsume not only HTML 4 but also XHTML 1 and DOM Level 2 HTML. Vercel is a deployment and collaboration platform for frontend developers. Vercel puts the frontend developer first, giving them comprehensive tools to build high-performance websites and applications. Vercel enables developers to host websites and web services that deploy instantly and scale automatically – all without any configuration. Firebase is a Backend-as-a-Service (BaaS). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google's infrastructure. Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents. Chat Engine is an API which makes it easy to build chat services. Building a chat from scratch takes a lot of time, code, and is expensive. It's better to use a product instead of writing it from scratch. We make it easy to build your chat idea in minutes. Chat Engine makes building chat very easy for people on a budget. It's perfect for people like freelancers and students, projects like start-ups and side-hustles. Other SDKs such as Stream and SendBird start at \$300 per month after a 14 day trial. That's too expensive! We believe everybody should have fair access to a great Chat API while building their next big thing! Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine. Let's face it, forms are really verbose in React. To make matters worse, most form helpers do way too much magic and often have a significant performance cost associated with them. Formik is a small library that helps you with the 3 most annoying parts: Getting values in and out of form state, Validation and error messages, Handling form submission. An **API** is basically a set of data, often in JSON format with specified endpoints. ... When we access data from an **API**, we want to access specific endpoints within that **API** framework. For example, if in the scenario we are using today, we are going to be using the random user **API**. **Axios** is a promise based HTTP client for the browser and Node.js. **Axios** makes it easy to send asynchronous HTTP requests to REST endpoints and perform CRUD operations. It can be used in plain JavaScript or with a library such as Vue or React. **Ant Design** is heavily based on psychological principles to anticipate—and be customized for—user behavior. **Ant Design** is built for **React**. In this tutorial, you will build a small **React** app that displays transactions to the user based on the **Ant Design** principles. You will use Okta to secure your web application.

PROBLEM STATEMENT

The first goal of this project visualises various sorting and searching algorithms, and create a web application to visualize this algorithms. This visualization portion's goal is create two types of visualization, first is path finding approach and second is sorting visualization approach, In this path finding approach's goal is create simple java script logic to visualize searching algorithms like Dijkstra algorithm, a* search algorithm etc and sorting visualization's goal is generate random arrays with random size and then create some simple java scripts logic to sort these arrays as a result we can simply visualize sorting algorithms. In this project contains three single page applications these are nothing but three functionalities of our main project. Another goal of this is create a backend database in firebase to store user's email and password ,after store it we will fetch the user id which will provide by firebase. After that pass this provided user id as password into react-chat-engine sdk, then we will create three react single page applications and host these applications through netlify. The second goal of our project is that store and show the details of many algorithms to learn easy, and the details are time complexity of algorithms, and their recurrence relations, logical approach etc. Third important goal of our project is create a chat application, by this portion users will discuss their doubts together. Last portion's goal is basically create collection contacts, feedbacks in firestore database which is use as backend and another goal of this portion is create also a frontend to send feedbacks and contact us portions using bootstrap and sass(syntactically awesome style sheets).After create total three single page applications and the visualization portion then the next goal is merge all functionalities and create a multi-page responsive application using vercel development, create a home page of our project after connect it to firebase login-signup page. Then the project will be done.

PROPOSED SOLUTION

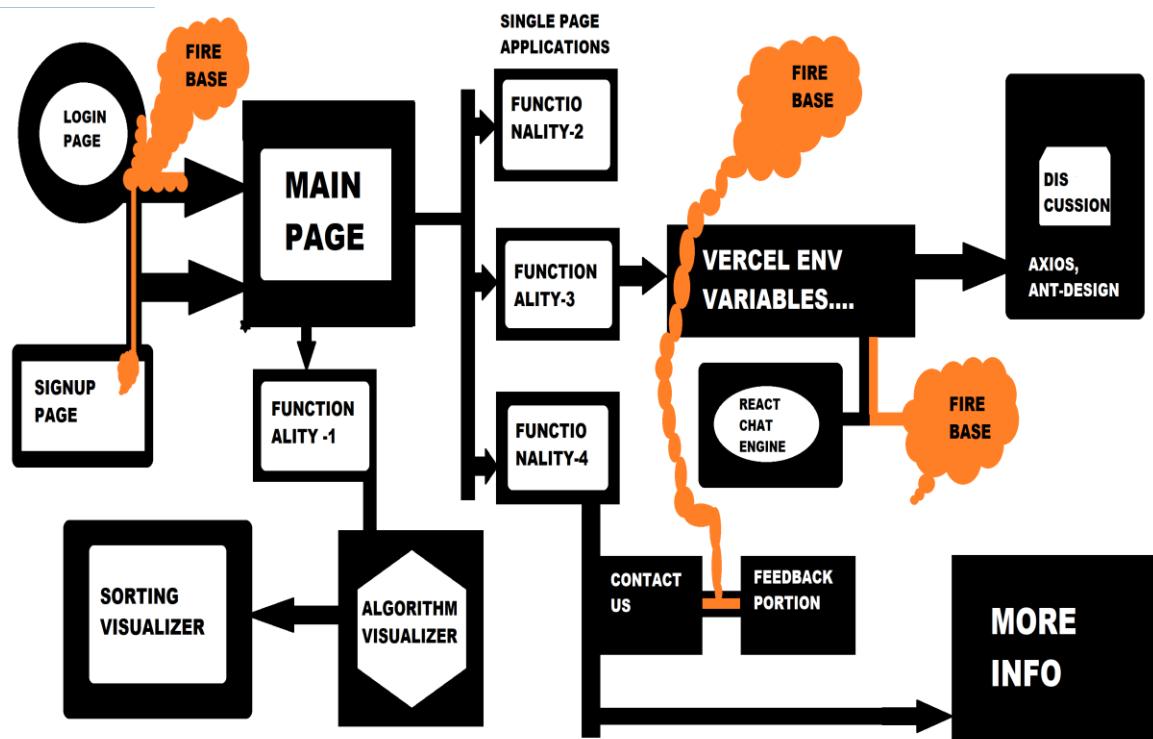
At first we create a project in firebase then this project set to web application based. Basically **firebase** is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google's infrastructure. **Firebase** is categorized as a NoSQL database program, which stores data in JSON-like documents. In our project we create a custom email password authentication system in firebase. After that we use react library in node js to create login and signup page, to create these forms pages we use Formik library in react-js. **React** js is an open-source JavaScript library that is **used** for building user interfaces specifically for single-page applications. **React** allows developers to create large web applications that can change data, without reloading the page. The main purpose of **React** is to be fast, scalable, and simple. After that we create the main page of our project In this page we use css blob effect using svg. **SVG** is an image format for vector graphics. It literally means Scalable Vector Graphics. Basically, what you work with in Adobe

Illustrator. After creating the mainpage we added four functionalites,1st is visualization portion here we create two sub parts also 1st is path finder approach and 2nd is sorting visualizer approach. In the path finding approach we implement some java scripts logic to visualize searching algorithms. Here we visualize seven popular searching algorithms 1st **Dijkstra's Algorithm** (weighted): the father of pathfinding algorithms ,guarantees the shortest path,2nd **A Search*** (weighted): arguably the best pathfinding algorithm, uses heuristics to guarantee the shortest path much faster than Dijkstra's Algorithm,3rd **Greedy Best-first Search** (weighted): a faster, more heuristic-heavy version of A*,4th is does not guarantee the shortest path, 5th is **Swarm Algorithm** (weighted) : a mixture of Dijkstra's Algorithm and A*, does not guarantee the shortest-path,6th is **Convergent Swarm Algorithm** (weighted): the faster, more heuristic-heavy version of Swarm; does not guarantee the shortest path,7th is **Bidirectional Swarm Algorithm** (weighted): Swarm from both sides; does not guarantee the shortest path,8th is **Breadth-first Search** (unweighted): a great algorithm; guarantees the shortest path,9th is **Depth-first Search** (unweighted): a very bad algorithm for pathfinding; does not guarantee the shortest path,On top of the pathfinding algorithms listed above, I implemented a **Recursive Division** Maze Generation algorithm. The Swarm algorithm is essentially a mixture of Dijkstra's Algorithm and A* Search; more precisely, while it converges to the target node like A* , it still explores quite a few neighbouring nodes surrounding the start node like Dijkstra's. The algorithm differentiates itself from A* through its use of heuristics: it continually updates nodes' distance from the start node while taking into account their estimated distance from the target node. This effectively "balances" the difference in total distance between nodes closer to the start node and nodes closer to the target node, which results in the triangle-like shape of the Swarm Algorithm. We named the algorithm "Swarm" because one of its potential applications could be seen in a video-game where a character must keep track of a boss with high priority (the target node), all the while keeping tracking of neighbouring enemies that might be swarming nearby. 2nd subpart of the first functionality is sorting visualization part here we also implement java scripts logic to visualize sorting algorithms. A Sorting Algorithm is used to rearrange a given array or list elements according to a comparison operator on the elements. The comparison operator is used to decide the new order of element in the respective data structure. In this project contains three single page applications these are nothing but three functionalities of our main project. We create a backend database in firebase to store user's email and password ,after store it we will fetch the user id which will provide by firebase. After that pass this provided user id as password into react-chat-engine sdk. Chat Engine is an API which makes it easy to build chat services. Building a chat from scratch takes a lot of time, code, and is expensive. It's better to use a product instead of writing it from scratch. We make it easy to

build your chat idea in minutes. We will create three react single page applications and host these applications through netlify. **Netlify** is a web developer platform that multiplies productivity. By unifying the elements of the modern decoupled web, from local development to advanced edge logic, **Netlify** enables a 10x faster path to much more performant, secure, and scalable websites and apps. After that store and show the details of many algorithms to learn easy, and the details are time complexity of algorithms, and there recurrence relations, logical approach etc. To create these single page applications we use react router dom. Routing is a process in which a user is directed to different pages based on their action or request. ReactJS Router is mainly used for developing Single Page Web Applications. React Router is used to define multiple routes in the application. When a user types a specific URL into the browser, and if this URL path matches any 'route' inside the router file, the user will be redirected to that particular route. React Router is a standard library system built on top of the React and used to create routing in the React application using React Router Package. It provides the synchronous URL on the browser with data that will be displayed on the web page. It maintains the standard structure and behaviour of the application and mainly used for developing single page web applications. After that we create a chat application using react chat engine which is already discussed, by this portion users will discuss their doubts together. To create this discussion portion we use axios to produce requests in react application. **Axios** is a promise based HTTP client for the browser and Node. js. **Axios** makes it easy to send asynchronous HTTP requests to REST endpoints and perform CRUD operations. It can be **used** in plain JavaScript or with a library such as Vue or React. Consuming **REST APIs** in a React Application can be done in various ways, but in this tutorial, we will be discussing how we can consume **REST APIs** using two of the most popular methods known as **Axios** (a promise-based HTTP client) and Fetch **API** (a browser in-built web **API**). **Axios** is a library that helps us make http requests to external resources. In our **React** applications we often need to retrieve data from external APIs so it can be displayed in our web pages. After we retrieve the data, we typically add it to the state, so it's ready to be **used** by our application. Last we create collection contacts, feedbacks in firestore database which is use as backend and another goal of this portion is create also a frontend to send feedbacks and contact us portions using bootstrap and sass(syntactically awesome style sheets). **Firestore** is **Firebase's** newest database for mobile app development. **Firestore** allows you to run sophisticated ACID transactions against your document data. This gives you more flexibility in the way you structure your data. Focus on your **application** development using **Firestore** client-side

development libraries for Web, iOS, **Android**, Flutter, C++, and Unity. **Bootstrap** is a framework to help you design websites faster and easier. It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels, etc. ... Here are some additional reasons to **use Bootstrap**: **Bootstrap's** responsive CSS adjusts to phones, tablets, and desktops. Here we use sass to create frontend, **Sass** (which stands for 'Syntactically awesome style sheets) is an extension of CSS that enables you to use things like variables, nested rules, inline imports and more. It also helps to keep things organised and allows you to create style sheets faster. **Sass** is compatible with all versions of CSS. After create total three single page applications and the visualization portion then merge all functionalities and create a multi-page responsive application using vercel development. **Vercel** is the best place to deploy any frontend **app**. Start by deploying with zero configuration to our global edge network. Here the main conceptual solution is firebase auth is connected to react chat engine sdk through vercel. Here we set firebase projects settings credentials and private key of the project which is create in react-chat-engine in to vercel environment's variables. After deploy this project using vercel deploy. Below we can see a table for analysis time-complexities of algorithms which are visualize in the 1st functionalities of our project.

WORKING FLOW:-



TIME COMPLEXITY ANALYSIS:-

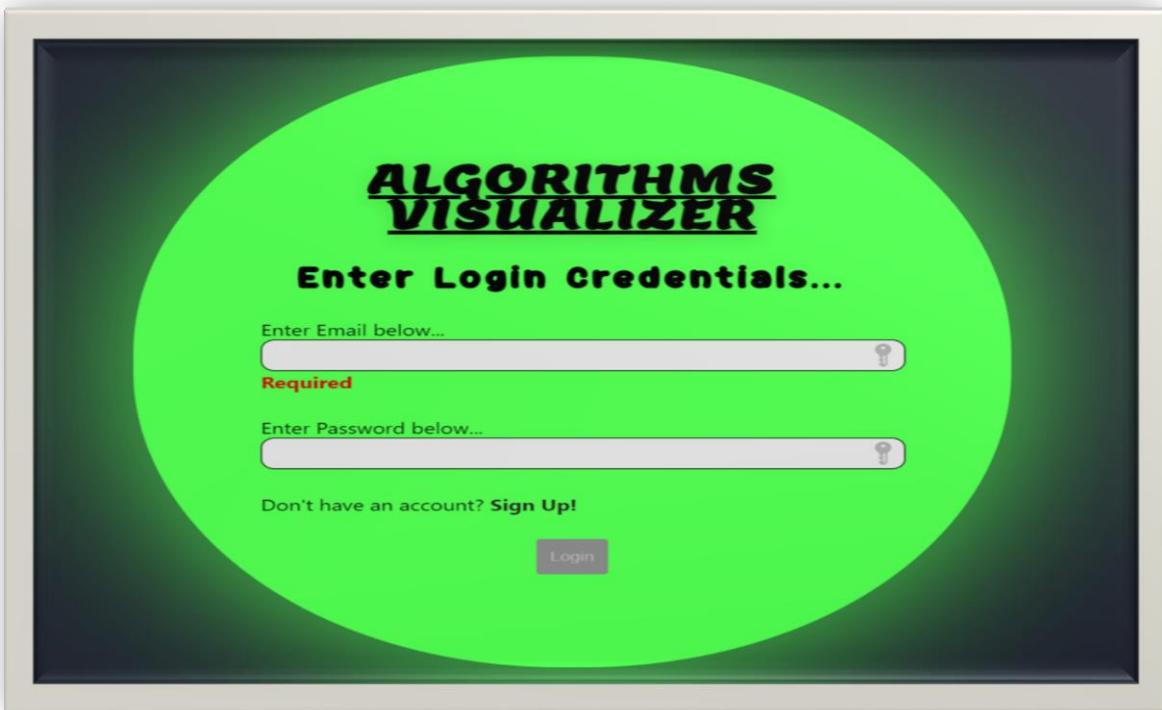
ALGORITHMS	TIME COMPLEXITIES
DIJKSTRA'S ALGORITHM	$O(V^2)$
BFS ALGORITHM	$O(V+E)$
DFS ALGORITHM	$O(V+E)$
A * SEARCH ALGORITHM	$O(E)$
BUBBLE SORT ALGORITHM	$O(n^2)$
SELECTION SORT ALGORITHM	$O(n^2)$
INSERTION SORT ALGORITHM	$O(n)$
MERGE SORT	$O(n \log n)$
QUICK SORT	$O(n \log n)$
HEAP SORT	$O(n \log n)$

EXPERIMENTAL SCREENSHOTS:

SINGUP-PAGE:



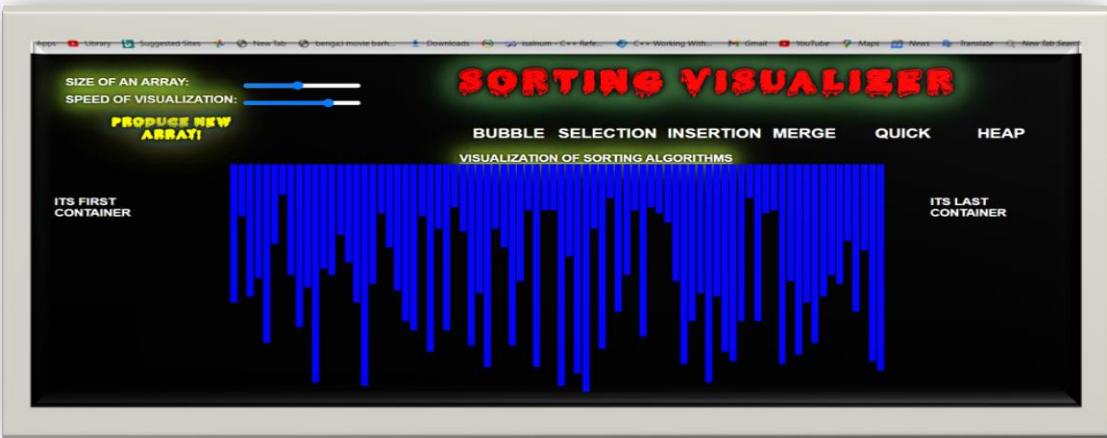
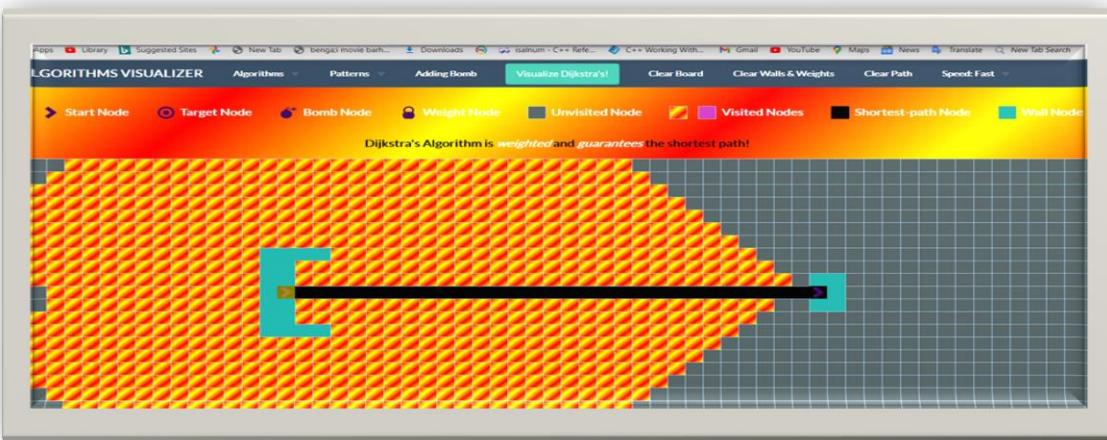
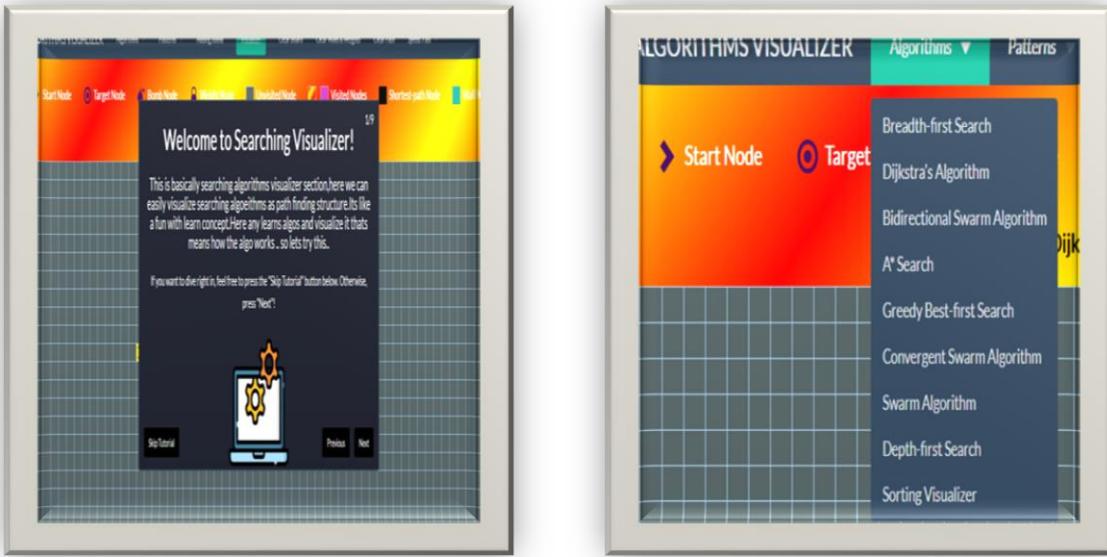
LOGIN-PAGE:

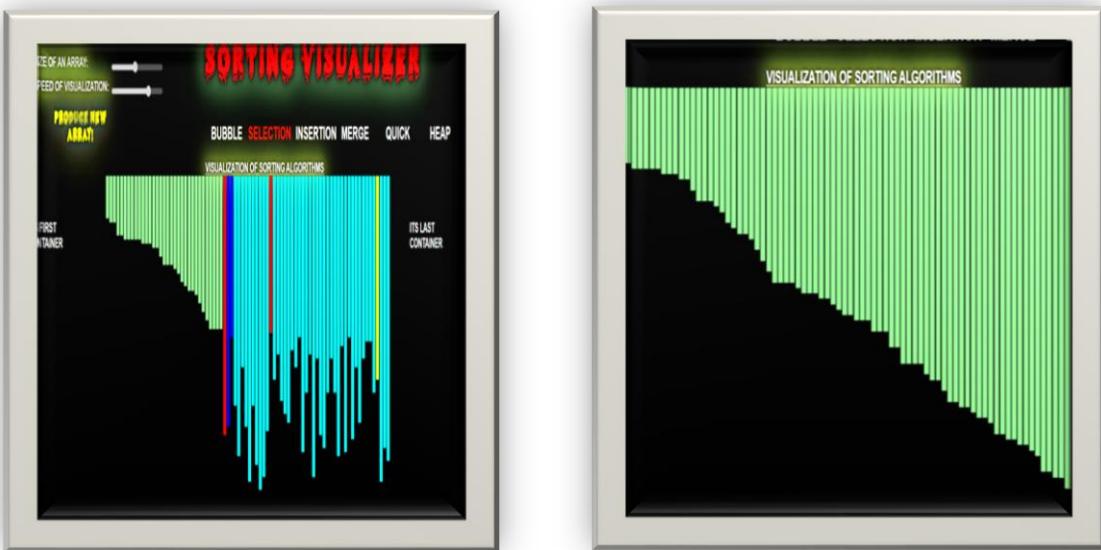


MAINPAGE:



1ST FUNCTIONALITIES:





2ND FUNCTIONALITES:





DIJKSTRA's ALGORITHM

An algorithm for finding the shortest paths between nodes in a graph

TIME COMPLEXITY

BEST: $\Theta(E+V \log V)$ where V is the number of vertexes and E is the number of edges

WORST: $\Theta(E+V \log V)$ where V is the number of vertexes and E is the number of edges

AVERAGE: $\Theta(E+V \log V)$ where V is the number of vertexes and E is the number of edges

SPACE COMPLEXITY

Space Complexity of an algorithm is total space taken by the algorithm with respect to the input size. Space complexity includes both Auxiliary space and space used by input.

Space complexity is a parallel concept to time complexity. If we need to create an array of size n, this will require $O(n)$ space. If we create a two dimensional array of size n^2 , this will require $O(n^2)$ space.

AVERAGE: $\Theta(V)$ where V is the number of vertexes

DEFINITION

Dijkstra's algorithm (or Dijkstra's Shortest Path First algorithm, SPF algorithm) is an algorithm for finding the shortest paths between nodes in a graph, which may represent, for example, road networks. It was conceived by computer scientist Edsger W. Dijkstra in 1956 and published three years later.

3RD FUNCTIONALITES:

JOIN DISCUSSION

USERNAME

PASSWORD

SUBMIT

DEVELOPERS DISSCUSSION

ankita arijit bisa rahul subhadip subhajit

how does this web app work?

don't know

how many algorithms are here?

72 algorithms

Type your questions...

DEVELOPERS DISSCUSSION

Image Apr 18

new
it is me

friends Zone
my hello!

ok
my hello!

how does this web app work?

don't know

how many algorithms are here?

72 algorithms

ankita arijit bisa rahul subhadip subhajit

DEVELOPERS DISSCUSSION

People

- AN ankita
- AR arijit
- BI bisa
- RA rahul
- SU subhadip
- SU subhajit

Type a username

Photos

Options

New Chat

Type your questions...

4TH FUNCTIONALITES:



Algorithm Visualizer About UserGuide ContactUs Submit Feedback

ALGORITHM VISUALIZER

OVERVIEW

This is a web application which is used to visualise algorithms. It has been created using JavaScript, CSS, SCSS, ReactJS, HTMLs, Vercel, Firebase, Chat Engine SDK, nodejs and Formik. User can approach this website through a provided link. First of all it has a sign up or login page. If user is opening it for first time he has to select sign up option. Here he has to give username, email id, password, verify password. If he has already signed up, then he has to go to login option and then he has to give email, password, confirm password. Then you have entered in the main page or home page. Here user can see your email id and a

Algorithm Visualizer About UserGuide ContactUs Submit Feedback

ALGORITHM VISUALIZER ABOUT PAGE

This is a web application which is used to visualise algorithms.
It has been created using JavaScript, bootstrap4,CSS, SCSS, ReactJS, HTML5, Vercel, Firebase, Chat Engine SDK, nodejs and Formik.

VISUALIZER SECTION	ALGORITHMS SECTION	DISCUSSION SECTION	THIS PORTION
Algorithm Visualizer is an interactive online platform that visualizes algorithms from code. Learning algorithms is often much easier with visualizations. It Algorithm visualizer is a web app written in React. It contains UI components and interprets commands into visualizations. Algorithms are a fascinating use case for visualization. To visualize an	In mathematics and computer science, an algorithm is a finite sequence of well-defined, computer-implementable instructions, typically to solve a class of problems or to perform a computation. Algorithms are always unambiguous and are used as specifications for performing calculations, data processing and automated reasoning and	The DISCUSSION part will take the user to a login page where he has to give username and password. If user has not given password in home page. Then he will be added in a discussion group for doubt clearing where the developers will answer his questions regarding this website. Discussion is important for any queries of user. When developers guide the user	Feedback occurs when outputs of a system are routed back as inputs because a chain of cause-and-effect, that forms a circuit or loop. The system can then be said to feed back into itself. The notion of cause-and-effect has to be handled carefully when applied to feedback systems. Simple causal reasoning about a feedback system is difficult because the first system influences the

A screenshot of a contact form titled 'CONTACT US'. The form is divided into sections for 'NAME', 'EMAIL', 'PHONE', and 'MESSAGE'. Each section contains an input field and a descriptive label. A 'Submit' button is located at the bottom of the form.A screenshot of a feedback form titled 'GIVE FEEDBACK'. It includes fields for 'USERNAME', 'EMAIL', 'RATING' (with a note indicating it should be a number between 0 and 100), and 'COMMENTS'. A 'Submit' button is positioned at the bottom of the form.

CONCLUSION

Algorithm Visualizer is an interactive online platform that visualizes algorithms as per user's choice. It has been created using JavaScript, CSS, SCSS, ReactJS, HTML5, Vercel, Firebase, Chat Engine XDK, nodejs and Formik. User can approach this website through a provided link. First of all it has a sign up or login page. If user is opening it for first time he has to select sign up option. Here he has to give username, email id, password, verify password. If he has already signed up, then he has to go to login option and then he has to give email, password, confirm password. Then you have entered in the main page or home page. Here user can see your email id and a firebase-provided password. Here four buttons are given. VISUALIZE ALGORITHMS, ALGORITHMS, DISCUSSION, FEEDBACK. By clicking on VISUALIZE ALGORITHMS user can watch and understand working of some specific algorithms. If user goes to ALGORITHMS he can see total 72 algorithms. Sorting, Searching and Others algorithms. If he clicks any of them in learn more button then their details will be shown in a page. The DISCUSSION part will take the user to a login page where he has to give username and password (firebase-provided password in home page). Then he will be added in a discussion group for doubt clearing where the developers will answer his questions regarding this website. In the feedback portion user can write about how much this website is useful or any idea for improvement in this website in the comment section.

APPLICATION:-

1. Learning an **algorithm** gets much easier with visualizing it. Algorithm Visualizer lets user visualize various algorithms and data structures. User can understand the algorithms and their working flow easily. It contributes three types of algorithms. They are sorting algorithms, searching algorithms and others algorithms.
2. This web application has a descriptive portion of 72 algorithms which is very useful for students of Computer Science and Mathematics subjects. Here time complexity, space complexity, definition and other information are gathered for students so that they don't have to search all information in google spending lot of time.
3. Students or any one who want to learn algorithm they can interact with the developers or any other user for any query, related to this web app. This idea of interaction and safe chatting area provides user nice way of learning.

4. This web app can be useful for students, interns, office workers or anyone who want to learn algorithms and their working flow. This online interactive platform base study is trending these days.

FUTURE SCOPE:-

1. Students while working on a course have to solve assignments. For solving exercises and finishing assignments students can use this algorithm visualizer.
2. Students like discussion-based study through an interactive platform. So that they will use it.
3. Without searching about algorithms students will get enough information about algorithms. It will save their time.
4. In the age of online study it is a great opportunity for students of learning algorithm by visualizing the working flow. Visualization based study is a trend of present days and the way is very efficient for understanding.

BIBLIOGRAPHY:-

1. <https://firebase.google.com/>
2. <https://reactjs.org/>
3. <https://chatengine.io/>
4. <https://vercel.com/>
5. <https://www.netlify.com/>
6. <https://nodejs.org/en/>
7. <https://www.npmjs.com/package/react-axios>
8. <https://formik.org/>