

**MINI PROJECT**  
**(2021-22)**  
**“ALGORITHM VISUALIZER”**  
**Project Report**



**Institute of Engineering & Technology**

**Submitted By -**

Deepanshi Mittal(191500247)

Komal Bindal(191500402)

Kritika Sharma (191500411)

**Under the Supervision Of**

**Ms. Ruchi Gupta**

**Technical Trainer**

**Department of Computer Engineering & Applications**



Department of Computer Engineering and Applications

GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,

Chaumuha, Mathura – 281406 U.P (India)

## **Declaration**

I/we hereby declare that the work which is being presented in the Bachelor of technology. Project “**Algorithm Visualizer**”, in partial fulfillment of the requirements for the award of the **Bachelor of Technology** in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of my/our own work carried under the supervision of **Ms.Ruchi Gupta, Technical Trainer, Dept. of CEA, GLA University.**

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

**Sign:** Kritika Sharma **Sign:** Komal Bindal

**Name of Candidate:** Kritika Sharma **Name of Candidate:** Komal Bindal **University**

**Roll No.:** 191500411 **University Roll No.:** 191500402

**Sign:** Deepanshi Mittal

**Name of Candidate:** Deepanshi Mittal **University Roll No.:** 191500247



**Department of Computer Engineering and Applications**  
**GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,**  
**Chaumuha, Mathura – 281406 U.P (India)**

## **Certificate**

This is to certify that the project entitled “Algorithm Visualizer”, carried out in Mini Project – I Lab, is a bonafide work by Kritika Sharma, Komal Bindal, Deepanshi Mittal and is submitted in partial fulfillment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

**Signature of Supervisor:**

**Name of Supervisor:** Ms. Ruchi Gupta

**Date:** 26-11-2



**Department of Computer Engineering and Applications**  
**GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,**  
**Chaumuha, Mathura – 281406 U.P (India)**

## **ACKNOWLEDGEMENT**

Presenting the ascribed project paper report in this very simple and official form, we would like to place my deep gratitude to GLA University for providing us the instructor Ms Ruchi Gupta, our technical trainer and supervisor.

He has been helping us since Day 1 in this project. He provided us with the roadmap, the basic guidelines explaining on how to work on the project. He has been conducting regular meeting to check the progress of the project and providing us with the resources related to the project. Without his help, we wouldn't have been able to complete this project.

And at last but not the least we would like to thank our dear parents for helping us to grab this opportunity to get trained and also my colleagues who helped me find resources during the training.

Thanking You

**Sign:** *Kritika Sharma***Sign:** *Komal Bindal*

**Name of Candidate:**Kritika Sharma **Name of Candidate:**Komal Bindal **University**

**Roll No.:**191500411 **University Roll No.:**191500402

**Sign:** *Deepanshi Mittal*

**Name of Candidate:**Deepanshi Mittal **University Roll No.:**191500247

# ABSTRACT

# ALGORITHM VISUALIZER

"THERE IS NO ALGORITHM FOR CREATIVITY"

The ALGORITHM VISUALIZER is intended for the students as well as for all the people who would like to know the working of various algorithms through their live demonstration. This tool will prove to be of great help for the students who are struggling to learn data structures and it will prove to be of great help as it will help them to brush up their concepts in a more right way.

## Advantages

- **Beginner friendly**
- **It will help the students to make their concepts even more clear by having a visual representation.**

## Tech Stack

- IDE: Visual Studio Code
- Language: HTML , CSS , Javascript

# CONTENTS

Cover Page .....	i
Declaration .....	ii
Acknowledgement .....	iii
Abstract .....	iv
Content .....	v
List Of figures.....	vi
Chapter 1 Introduction.....	1
1.1 Context.....	1 · 1.2
Motivation.....	1 · 1.3
Objective.....	2 · 1.4
1.2 Existing System.....	2 · 1.4
Sources.....	3
Chapter 2 Software Requirement Analysis.....	4
· 2.1 Impact Of Algorithms On Daily Life.....	4 ·
2.2 Problem Statement.....	5 · 2.3
Hardware and Software Requirements.....	6 · 2.4
Modules and Functionalities.....	6 · 2.5
Algorithm Visualizer.....	7
Chapter 3 Software Design.....	8
· 3.1 Use Case Diagram.....	8 · 3.2
Data Flow Diagram.....	11 · 3.3
Sequence Diagram.....	12
Chapter 4 Technology Used.....	13

4.1 FullstackDevelopment.....	13	· 4.2 Tools and Languages.....	15	· 4.4 Basic Terminology.....	16
Chapter 5 Implementation and User Interface.....	19	· 5.1 Implementation .....	19	· 5.2 User Interface.....	23
Chapter 6 Testing.....	33	· 6.1 Installation Testing.....	33	· 6.2 Unit Testing.....	34
		· 6.3 User Testing.....	37	· 6.4 Performance Testing.....	38
		· 6.5 Compatibility Testing.....	39		
Chapter 7 Conclusion.....	40				
<b>References</b> .....	41				

# CHAPTER-1

## INTRODUCTION

### 1.1 CONTEXT

This Algorithm Visualizer has been submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering at GLA University, Mathura supervised by Ms.Ruchi Gupta. This project has been completed approximately three months and has been executed in modules, meetings have been organised to check the progress of the work and for instructions and guidelines.

### 1.2 MOTIVATION

In the recent years, we have realized the importance of virtual learning and how important it is for us to have our resources online. Books have been the greatest source of learning all the while and having them at the reach of our fingertips would be an opportunity hardly any student would afford to miss.

In the century we are living the world is progressing at a really great pace, a lot number of technologies come up every single day. To keep up with the technology is also important to survive in this world of digitalization and learning. Along with this we need to have a place to keep the resources for areas of our interest so we thought of developing a website which could provide us to learn Data structure and algorithms virtually.

Moreover this kind of application can be used in areas/schools where guardians /parents cannot afford to buy books. This would be an excellent effort to provide education without any boundaries to all.

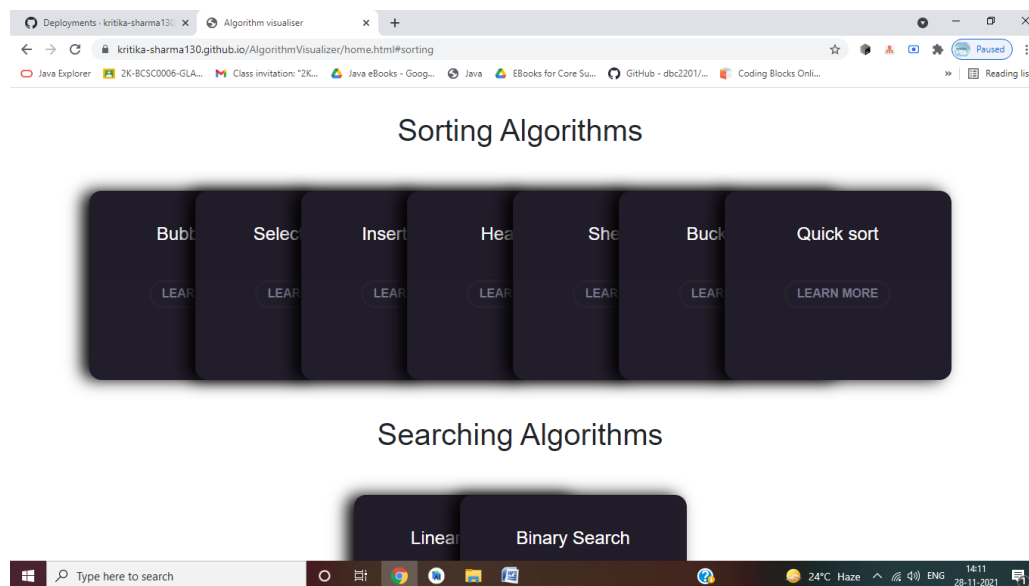
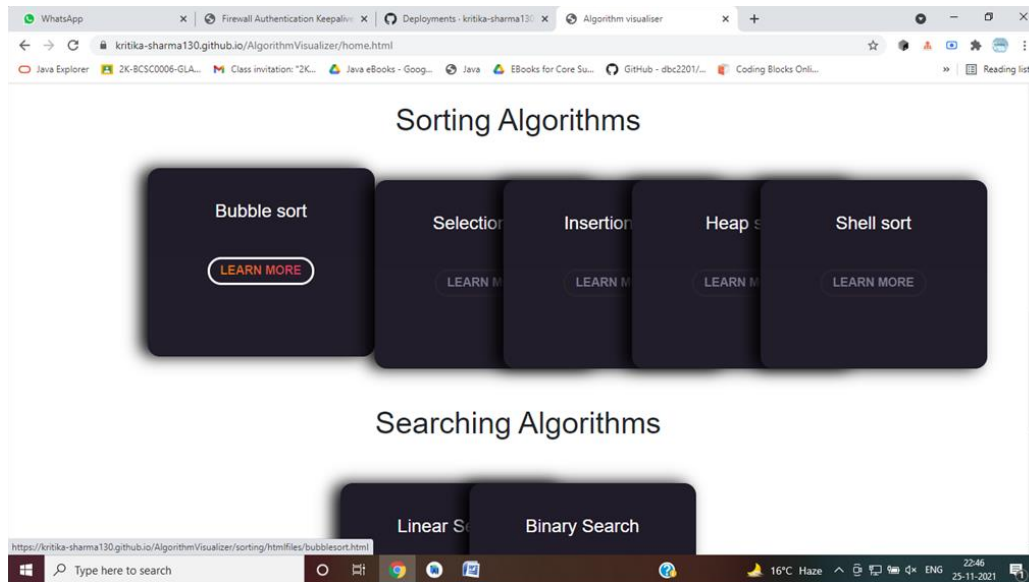
### 1.3 OBJECTIVE

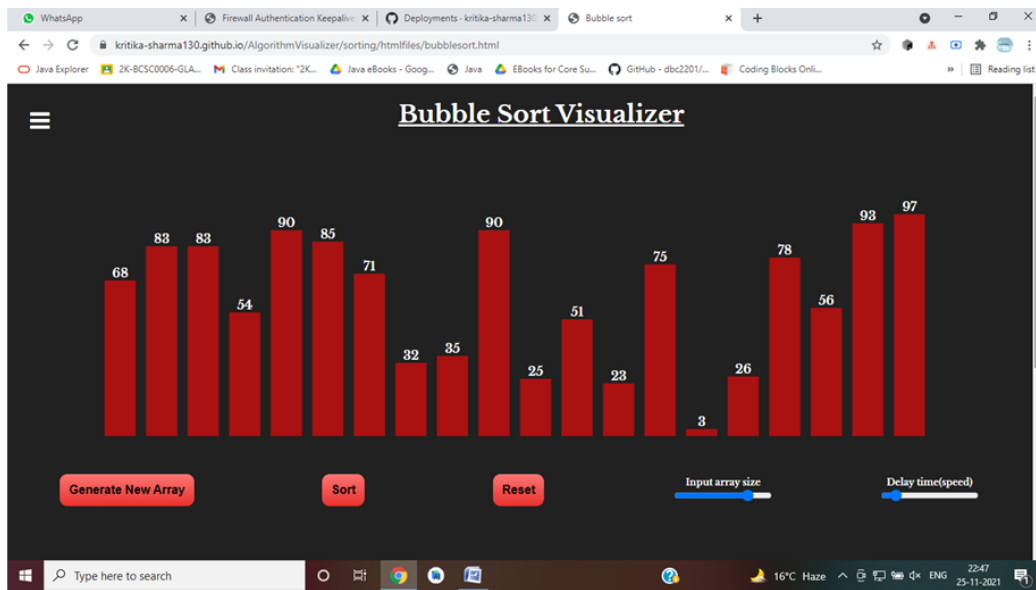
The main objective of this website is to make the learning process of learning data structure and algorithms easy by virtually learning through this website. Along with the implementation the website also contains visual features which are beginner friendly and makes understanding and learning easy. This application developed can be used at a variety of places, at education hubs and have its significance. The goal of the app was to provide a way to the learners and users to get all the learning visually they desire to read at a particular location rather than randomly surfing the Internet.



## 1.4 EXISTING SYSTEM

In the present scenario, we are in such a situation where every individual in a direct or indirect way is associated with data structures and algorithms. The website focuses upon the visual representation of different algorithms along with the implementation of their corresponding codes.





**Figure-1: Existing System**

## 1.5 SOURCES

The source of our project (including all the project work, documentations and presentations) will be available at the following link <https://github.com/kritika-sharma130/AlgorithmVisualiz>

# CHAPTER -2

## SOFTWARE REQUIREMENT ANALYSIS

### 2.1 IMPACT OF ALGORITHMS ON OUR DAILY LIFE

As a computer scientist, it is important to understand all of these types of algorithms so that one can use them properly. If you are working on an important piece of software, you will likely need to be able to estimate how fast it is going to run. Such an estimate will be less accurate without an understanding of runtime analysis. Furthermore, you need to understand the details of the algorithms involved so that you'll be able to predict if there are special cases in which the software won't work quickly, or if it will produce unacceptable results.

Of course, there are often times when you'll run across a problem that has not been previously studied. In these cases, you have to come up with a new algorithm, or apply an old algorithm in a new way. The more you know about algorithms in this case, the better your chances are of finding a good way to solve the problem. In many cases, a new problem can be reduced to an old problem without too much effort, but you will need to have a fundamental understanding of the old problem in order to do this.

As an example of this, let's consider what a switch does on the Internet. A switch has  $N$  cables plugged into it, and receives packets of data coming in from the cables. The switch has to first analyze the packets, and then send them back out on the correct cables. A switch, like a computer, is run by a clock with discrete steps – the packets are sent out at discrete intervals, rather than continuously. In a fast switch, we want to send out as many packets as possible during each interval so they don't stack up and get dropped. The goal of the algorithm we want to develop is to send out as many packets as possible during each interval, and also to send them out so that the ones that arrived earlier get sent out earlier. In this case it turns out that an algorithm for a problem that is known as “stable matching” is directly applicable to our problem, though at first glance this relationship seems unlikely. Only through pre-existing algorithmic knowledge and understanding can such a relationship be discovered.

### 2.2 PROBLEM STATEMENT

The ALGORITHM VISUALIZER is intended for the students as well as for all the people who would like to know the working of various algorithms through their live demonstration. This tool will prove to be of great help for the students who are struggling to learn data structures and it will prove to be of great help as it will help them to brush up their concepts in a more right way.

Through this the students can visually learn the data structures and algorithms.

## 2.3 HARDWARE AND SOFTWARE REQUIREMENTS

### Hardware Requirement

- Processor :intel i5
- Operating System :Any Operating System
- RAM : 4 GB (or higher)
- Hard disk : 256GB

### Software Requirement

- Software used: Visual Studio Code
- Language used : HTML,CSS,Javascript
- Database: Firebase
- User Interface Design : Website

## 2.4 MODULES AND FUNCTIONALITIES

- **Home Screen:** The first screen with which the user interacts will be this screen containing the website's name .
- **Login Page:** This page is for those users who have already registered themselves on the app and have a username and a password. There is also a way on this page for the new users to register themselves which will take them to the registration page.
- **Registration Page:** This is page is solely designed for the new users of the app who are willing to register themselves. This page takes input of the various details of the user and stores it in the database, later helping the user to login into the account with credentials they have provided.
- **Footer Drawer:** This is the most important part of the application that provides interactivity within the website as it connects the various activities together like it is a side bar on which the profile, the dashboard, the favourites section, the feedback section

,the About page of the page are linked and on clicking on each you can visit the pages.

- **Algorithm Description Page:** After clicking on the respective algorithm the user can have a complete view of the corresponding algorithm.
- **Feedback Page:** Initially the page is empty, but the user wants to give feedback, the user can use it for rating the website.

## 2.5 ALGORITHM VISUALIZER

Algorithm Visualizer is basically a website where the user can see the implementation of code as well as the visual effects related to the code. Through this platform the user can discover, learn and innovate new concepts in an interactive manner.

# CHAPTER- 3

## SOFTWARE DESIGN

### 3.1 USE-CASE DIAGRAM:

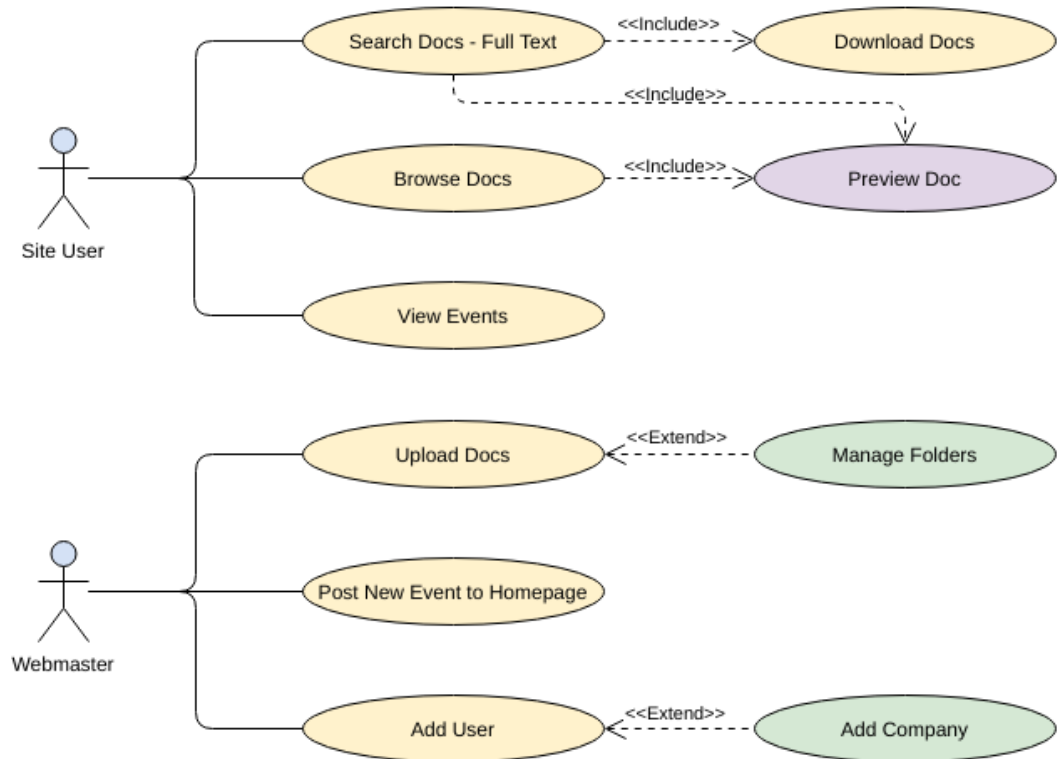
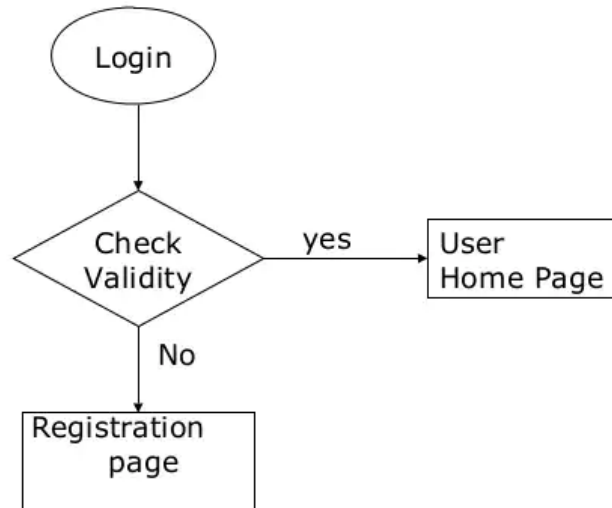


Figure-2: Use-Case Diagram

## 3.2 DATA FLOW DIAGRAM

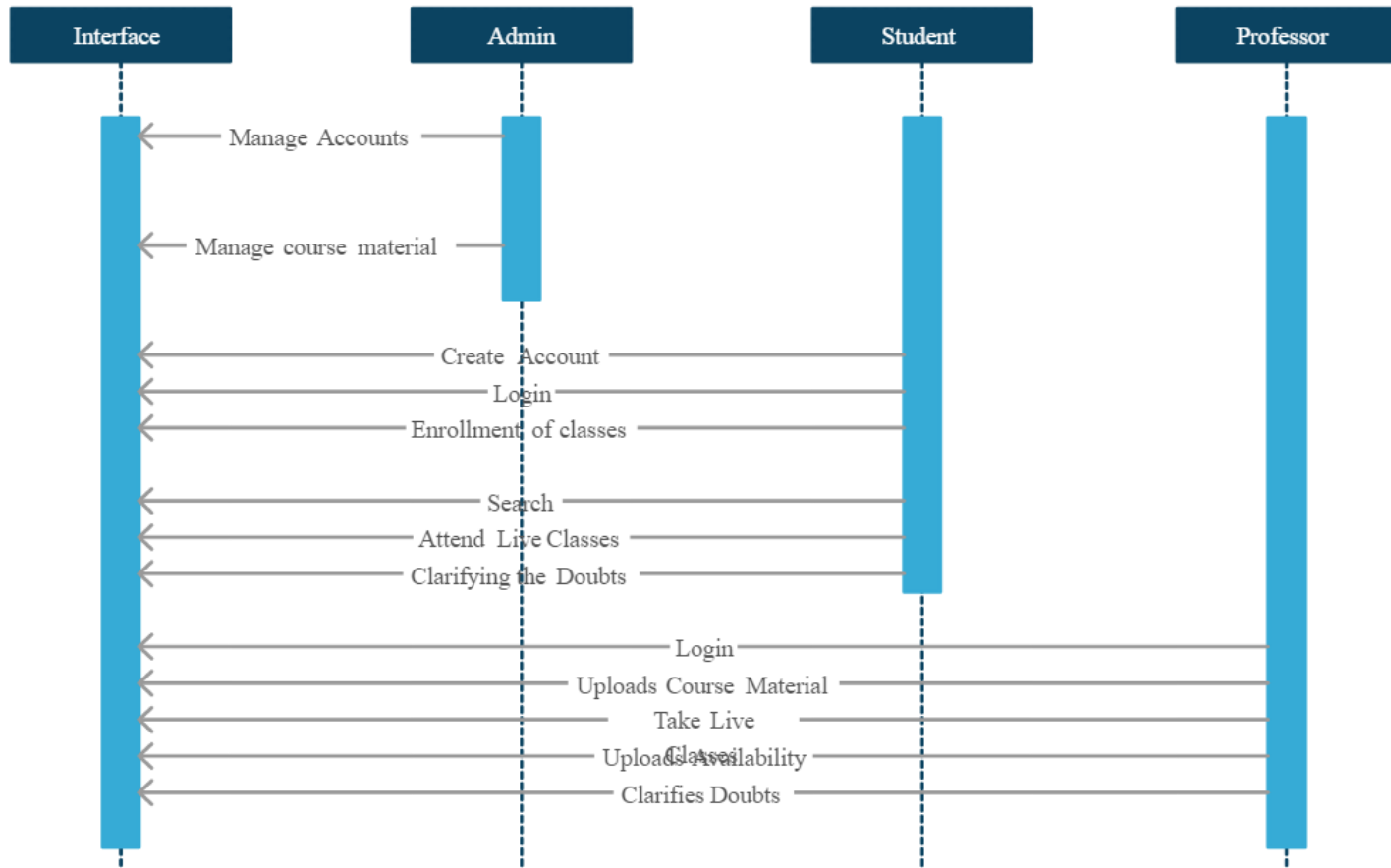
### DFD For E Learning System

**Data Flow Diagram**



**Figure-3: Data Flow Diagram**

### 3.3 SEQUENCE DIAGRAM



**Figure-4: Sequence Diagram**



# CHAPTER-4

## TECHNOLOGY USED

### 4.1 FRONTEND DEVELOPMENT

Full stack web development is the practice of working on both the front-end and back-end of a program. It is a term mostly used for those working in web development. The developers have background on creating user interface and user experience for front-end, and also have strong knowledge in a programming language that is used for handling the logic of the application, hence back-end.[1]

Full Stack is a layer of software or web development consisting of the front-end and the back-end portions of an application. Front-end is what the users will see or interact with on your application. Back-end part is what users do not see, such as application's logic, database, server, etc. A full-stack web developer is comfortable working with both back-end and front-end technologies which make a website or application function properly

14

### 4.3 TOOLS AND LANGUAGES

Tools used to build the Android App are:-

- **Visual Studio Code:** Visual Studio Code is a [source-code editor](#) made by [Microsoft](#) for [Windows](#), [Linux](#) and [macOS](#).<sup>[9]</sup> Features include support for [debugging](#), [syntax highlighting](#), [intelligent code completion](#), [snippets](#), [code refactoring](#), and embedded [Git](#). Users can change the [theme](#), [keyboard shortcuts](#), preferences, and install [extensions](#) that add additional functionality. Visual Studio Code was first announced on April 29, 2015, by Microsoft at the 2015 [Build](#) conference. A [preview](#) build was released shortly thereafter.
- **HTML:** The **HyperText Markup Language**, or **HTML** is the standard [markup language](#) for documents designed to be displayed in a [web browser](#). It can be assisted by technologies such as [Cascading Style Sheets](#) (CSS) and [scripting languages](#) such as [JavaScript](#).

[Web browsers](#) receive HTML documents from a [web server](#) or from local storage and [render](#) the documents into multimedia web pages. HTML describes the structure of a [web page semantically](#) and originally included cues for the appearance of the document.

- **CSS:** Cascading Style Sheets (CSS) is a [style sheet language](#) used for describing the [presentation](#) of a document written in a [markup language](#) such as [HTML](#).<sup>[1]</sup> CSS is a cornerstone technology of the [World Wide Web](#), alongside HTML and [JavaScript](#).<sup>[2]</sup>

CSS is designed to enable the separation of presentation and content, including [layout](#), [colors](#), and [fonts](#).<sup>[3]</sup> This separation can improve content [accessibility](#); provide more flexibility and control in the specification of presentation characteristics; enable multiple [web pages](#) to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be [cached](#) to improve the page load speed between the pages that share the file and its formatting.

- **JAVASCRIPT:** JavaScript ([/'dʒɑ:və\\_skrɪpt/](#)),<sup>[10]</sup> often abbreviated as JS, is a [programming language](#) that conforms to the [ECMAScript](#) specification.<sup>[11]</sup> JavaScript is [high-level](#), often [just-in-time compiled](#) and [multi-paradigm](#). It has [dynamic typing](#), [prototype-based object-orientation](#) and [first-class functions](#).

Alongside [HTML](#) and [CSS](#), JavaScript is one of the core technologies of the [World Wide Web](#).<sup>[12]</sup> Over 97% of [websites](#) use it [client-side](#) for [web page](#) behavior,<sup>[13]</sup> often incorporating third-party [libraries](#).<sup>[14]</sup> All major [web browsers](#) have a dedicated [JavaScript engine](#) to execute the code on the [user's](#) device.

As a multi-paradigm language, JavaScript supports [event-driven](#), [functional](#), and [imperative programming styles](#). It has [application programming interfaces](#) (APIs) for working with text, dates, [regular expressions](#), standard [data structures](#), and the [Document Object Model](#) (DOM).

## 4.4 BASIC TERMINOLOGY

- **Layout:** Layout is the parent of view. It arranges all the views in a proper manner on the screen.
- **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. Firebase has three core services: a real-time database, user authentication and hosting. With the Firebase iOS SDK, you can use these services to create apps without writing any server code.

**JSON** stands for JavaScript Object Notation. It is an independent data exchange format and is the best alternative for XML. JSON is used for data interchange (posting and

retrieving) from the server. Hence knowing the syntax and it's usability is important. JSON is the best alternative for XML and its more readable by human

## **CHAPTER -5**

### **IMPLEMENTATION AND USER INTERFACE**

Creating an app concept design with screen sketches and functional flow diagrams is the best way to communicate your vision to the mobile app developer. Making the concept clear to the developer is probably the most important factor in successful mobile app development. Yet it is one of the most common problems or obstacles in a mobile app development outsourcing project.

No matter what the marketing and profit goals are or if you are outsourcing an app for your personal use, you need to fully design and document the app concept if you expect a programmer to make your vision a reality. Developers are not mind readers and even descriptions given during conversations can be very fleeting or interpreted differently. Fully documenting your concept, therefore, leaves little to chance. The two most important things to do are: A) make a comprehensive description of how the app works and what it does (functionality) and B) create a comprehensive description of what the user sees and does (look and feel).

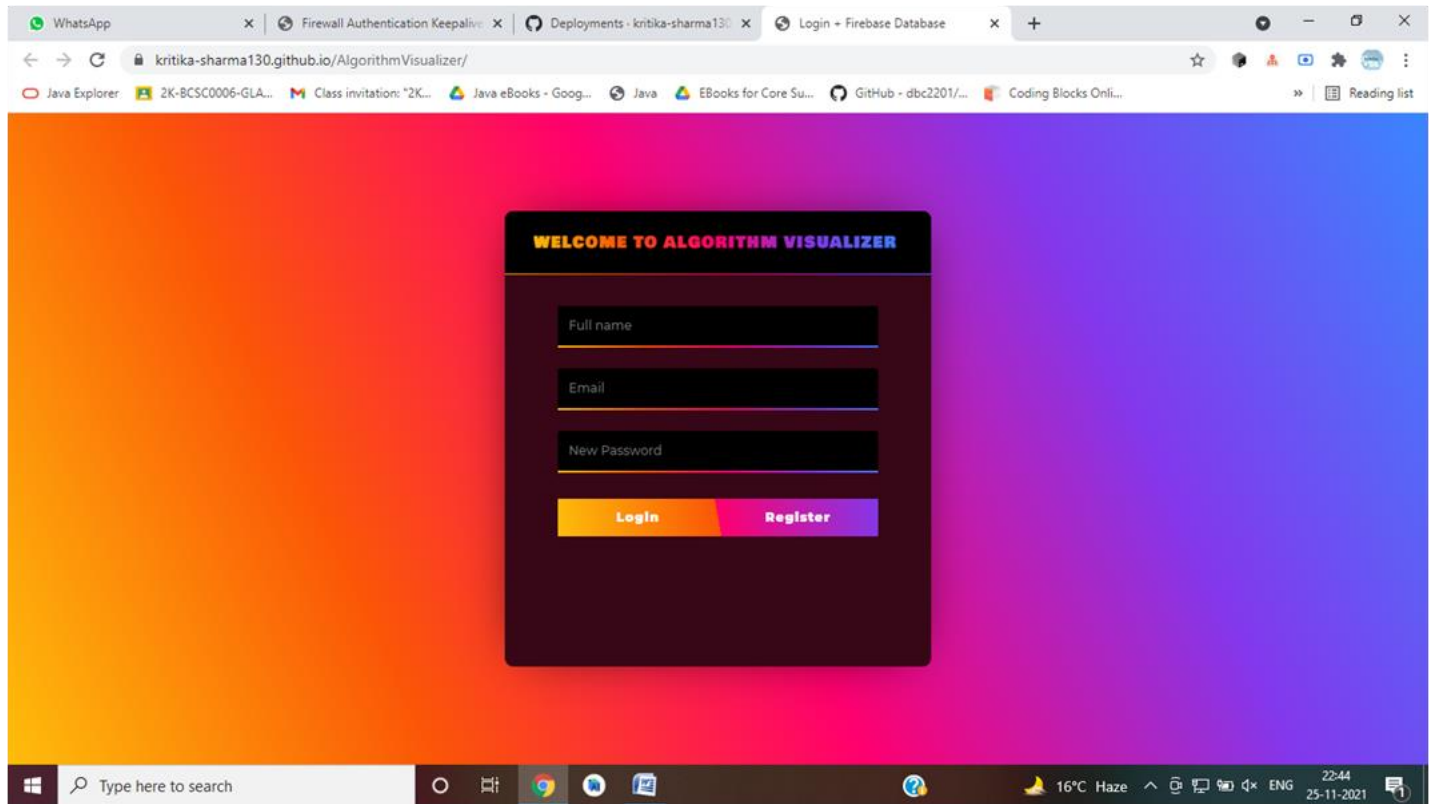
#### **5.1 Implementation of the Algorithm Visualizer:**

Implementation of AlgorithmVisualizer is taken place in various phases. Firstly we build the login interface then select the respective algorithm to be visited and finally we parse the Jason object to get the data in the required format and then display the result.

#### **4.5 Step to be followed by the user**

1. Firstly, we have build a home screen.
  2. Then, we have the Login activity which consists of following steps
    - Register : for new User
    - Login: For existing as well as new user
    - 3. We authenticate and store the user information from the Firebase authentication. 4. After that, we made a Drawer layout of our Bookopedia app which includes various functionality
    - Home Page: To see the layout of the whole screen.
    - Responsive Cards:Show the responsive cards. ·
- Footer: It comprises of all the necessary questions.

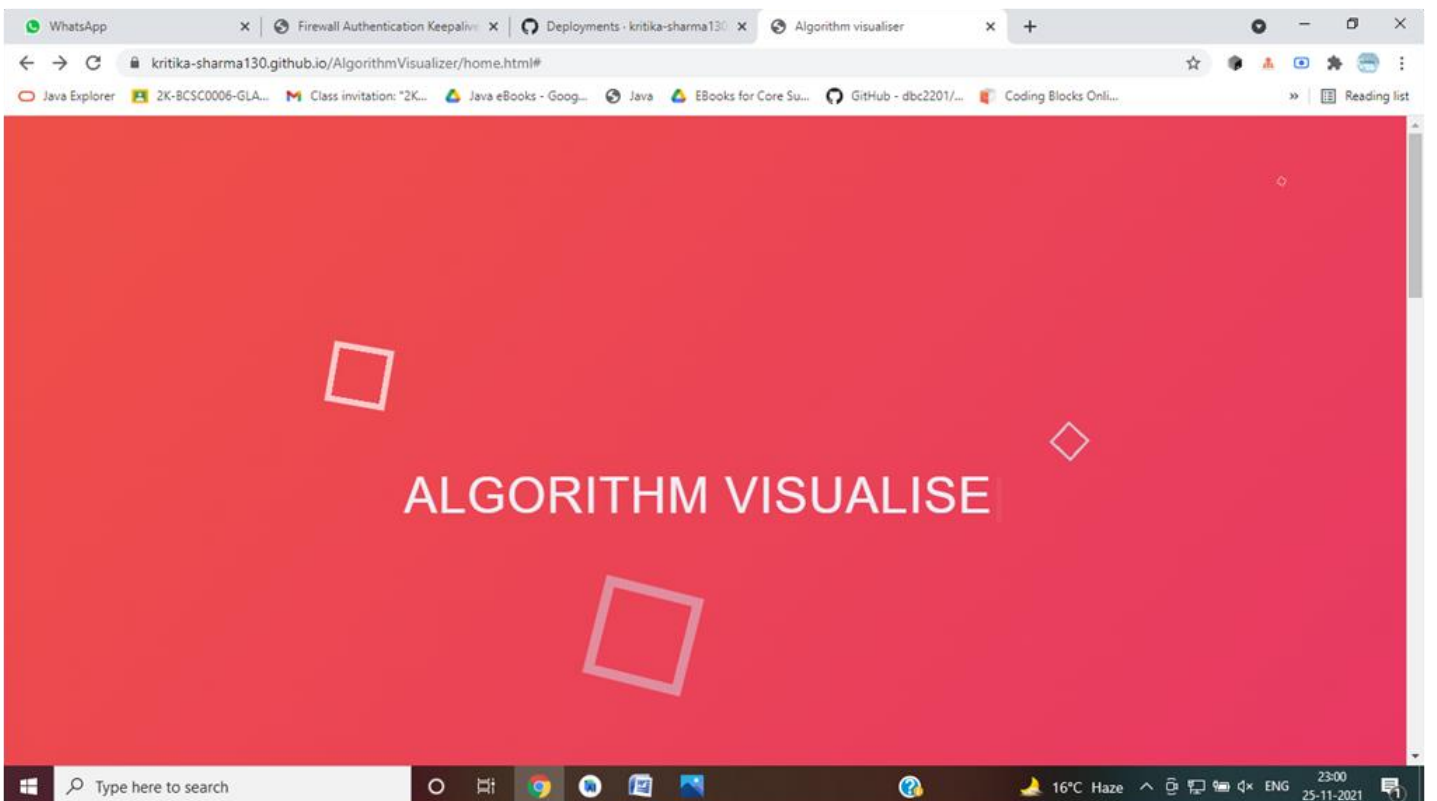
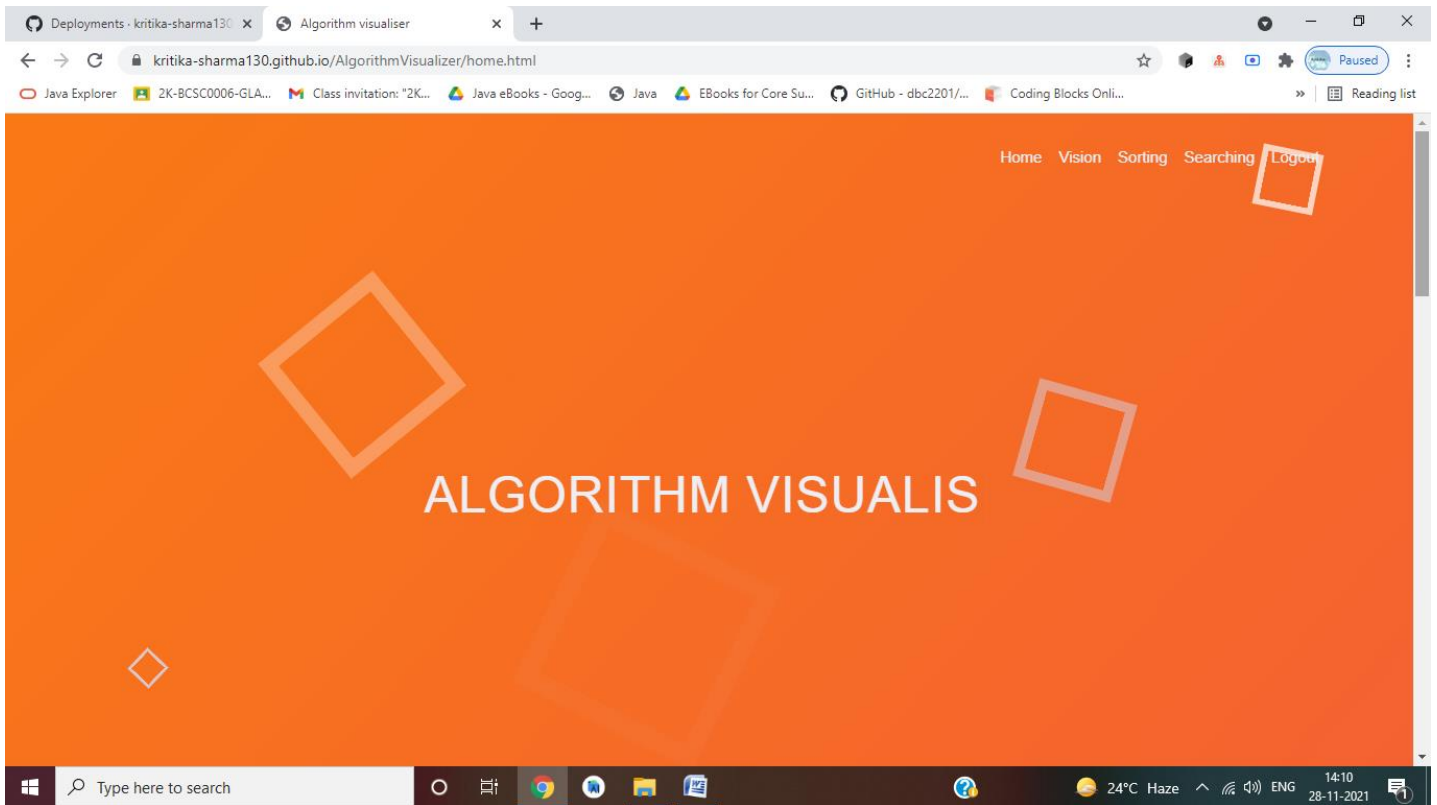
# SYSTEM ANALYSIS

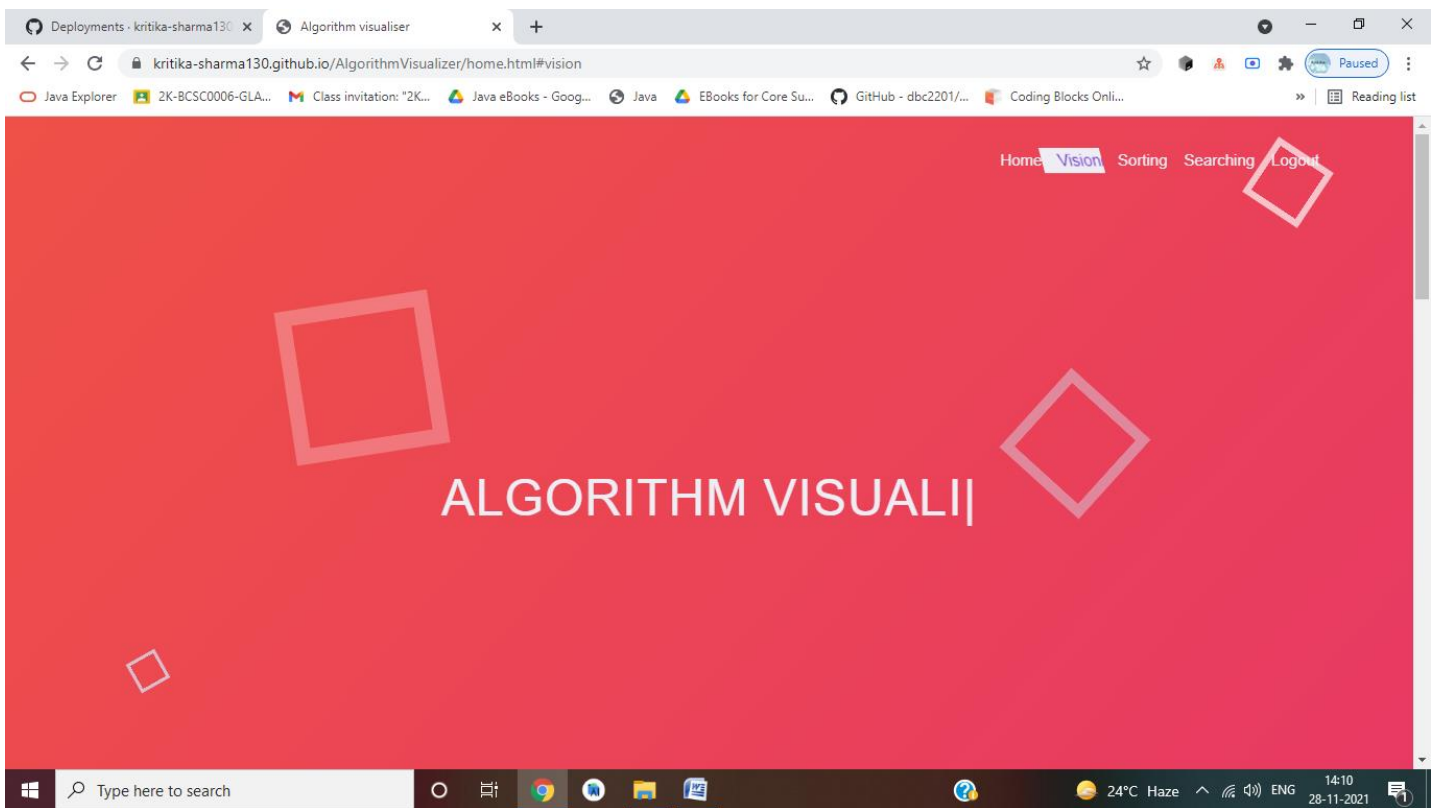
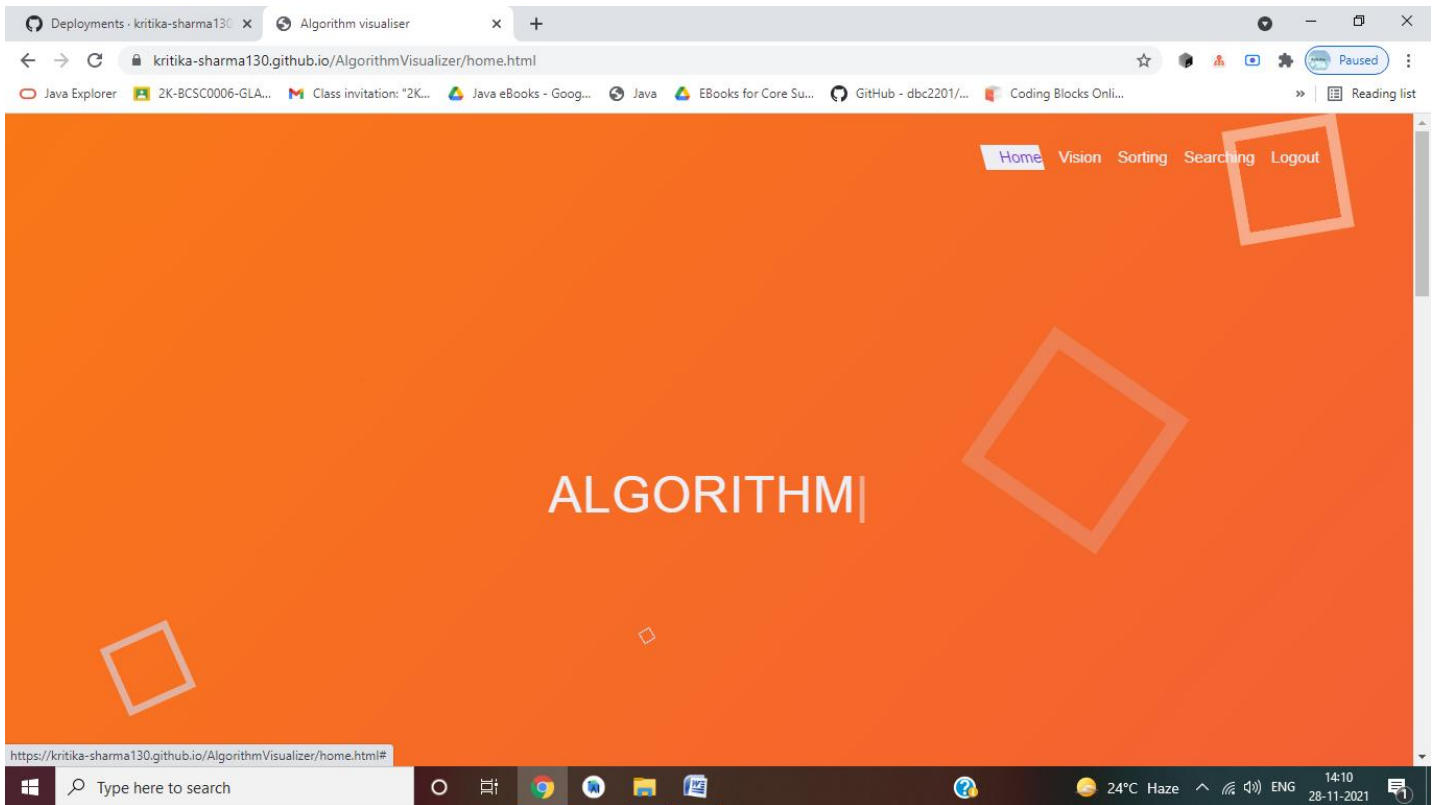


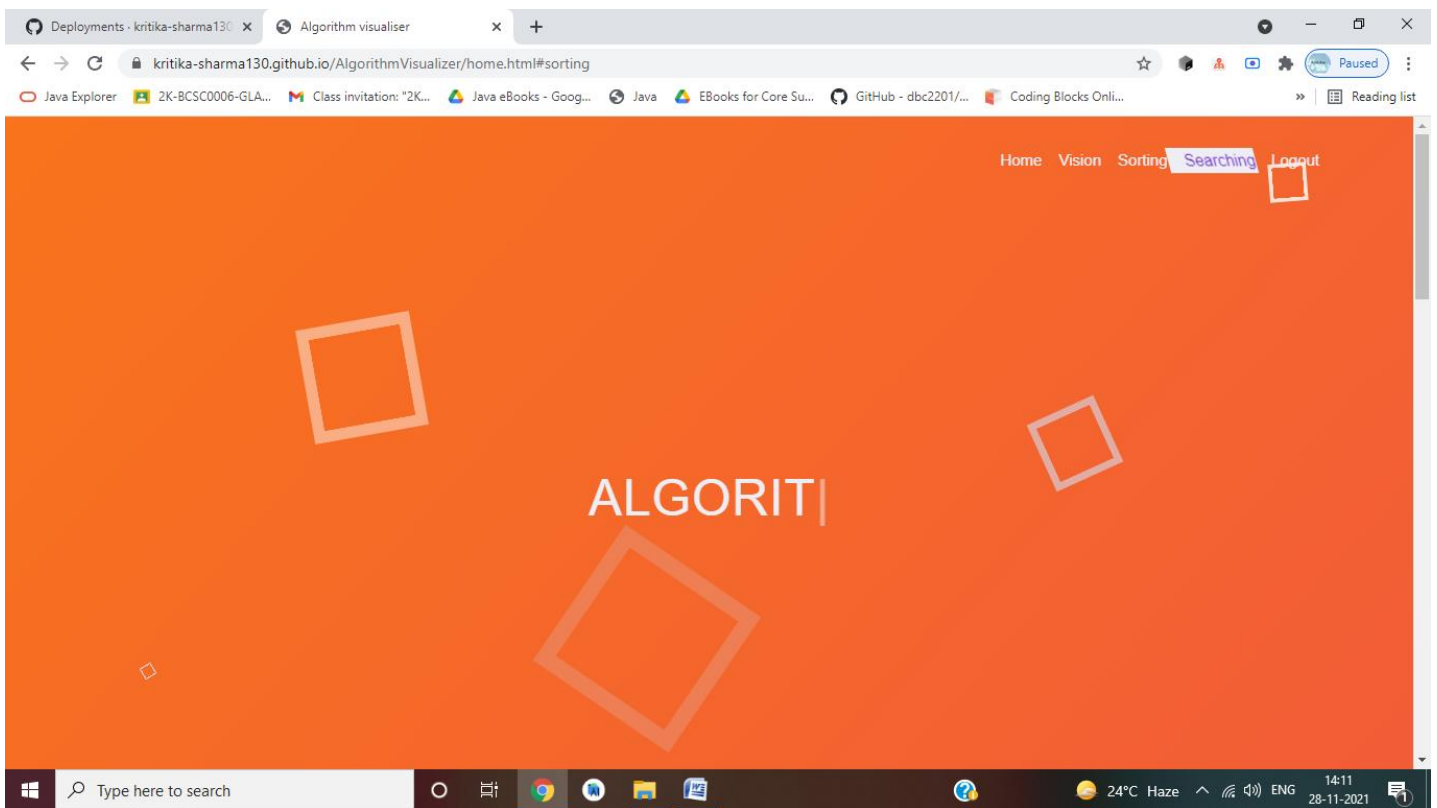
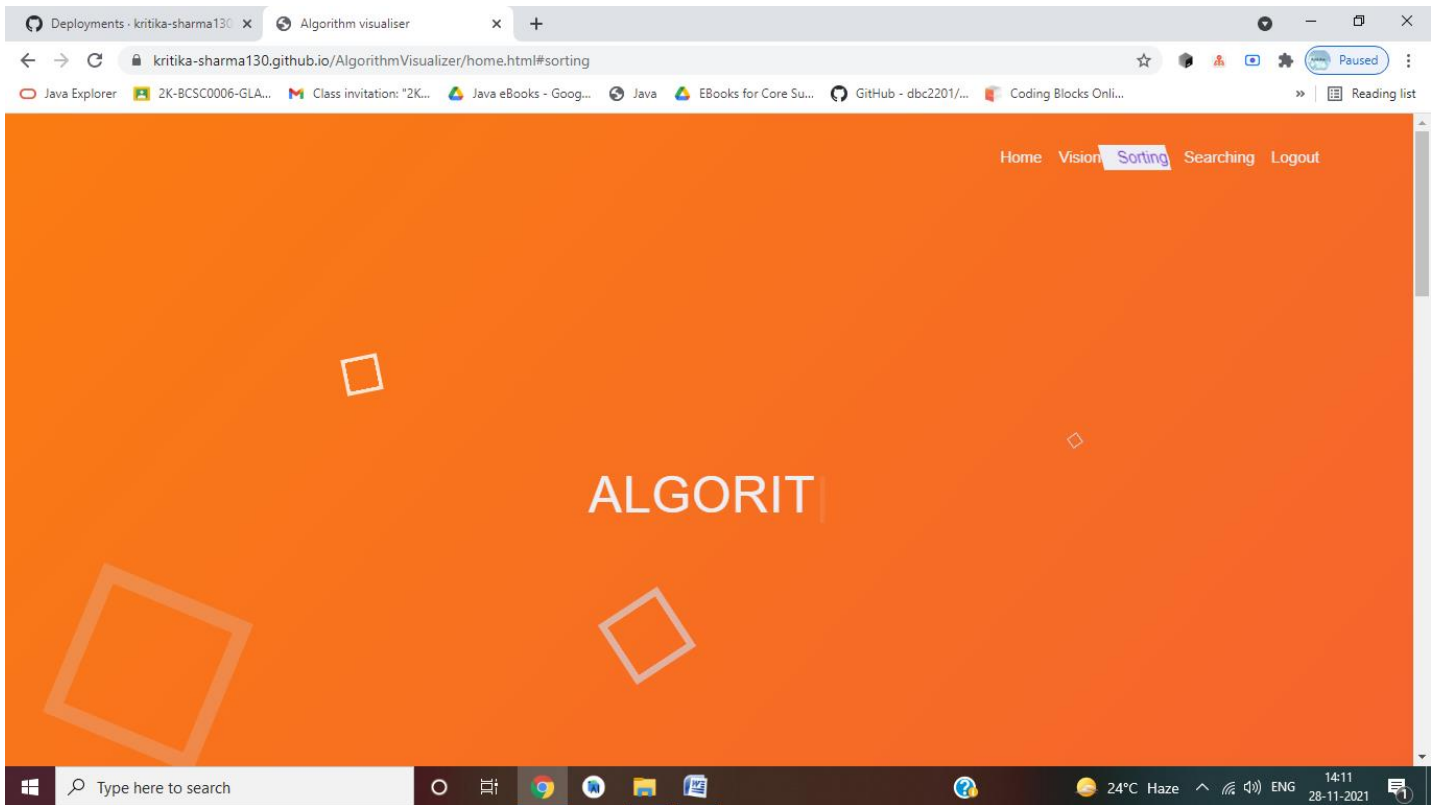
The picture given above shows the login and register page for the website. It allows the user to register if they are first visiting the website or login into the website if the user is already registered by entering the name, email and password.

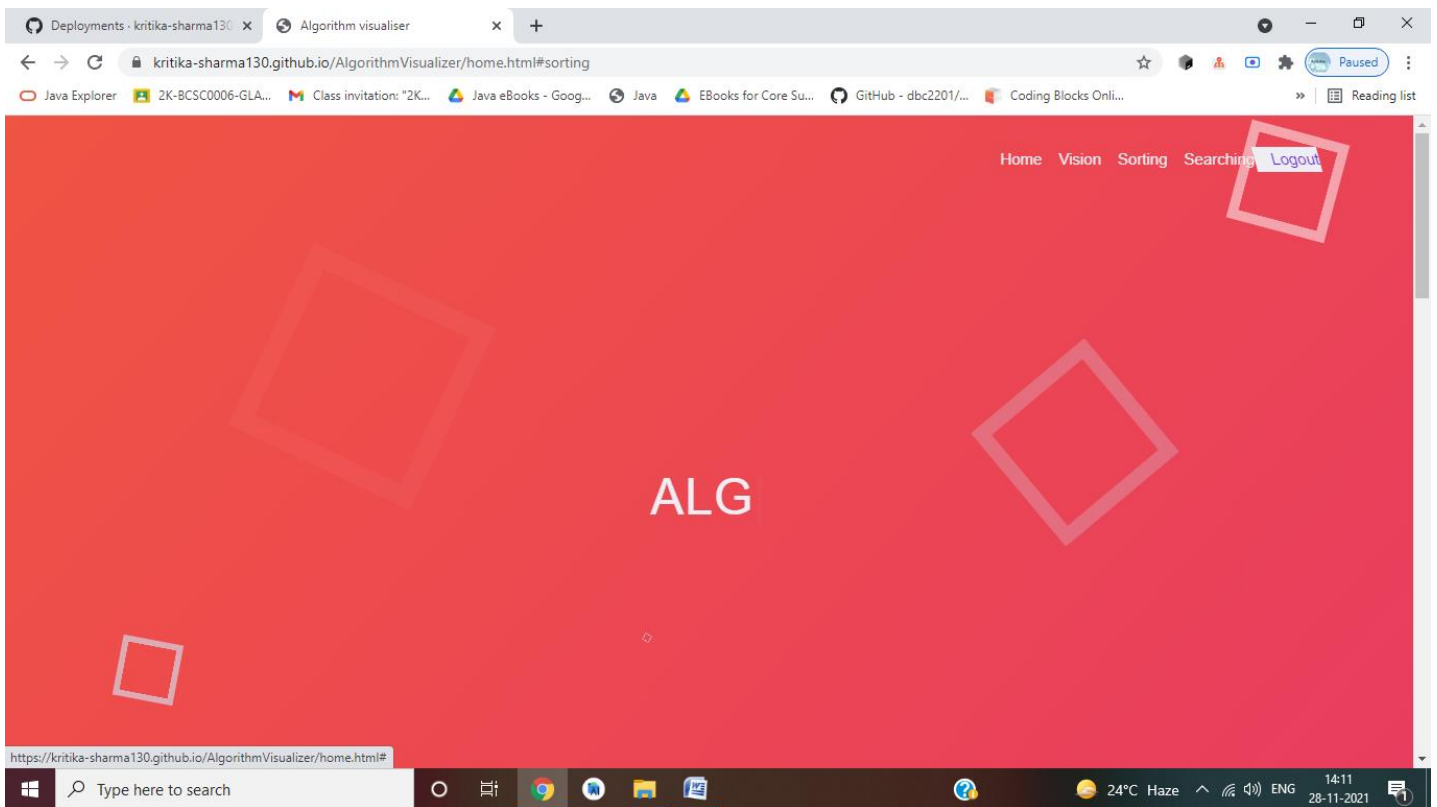
The page has two buttons:

1. **Login:** To login as an existing user.
2. **Register:** To register as a new user .









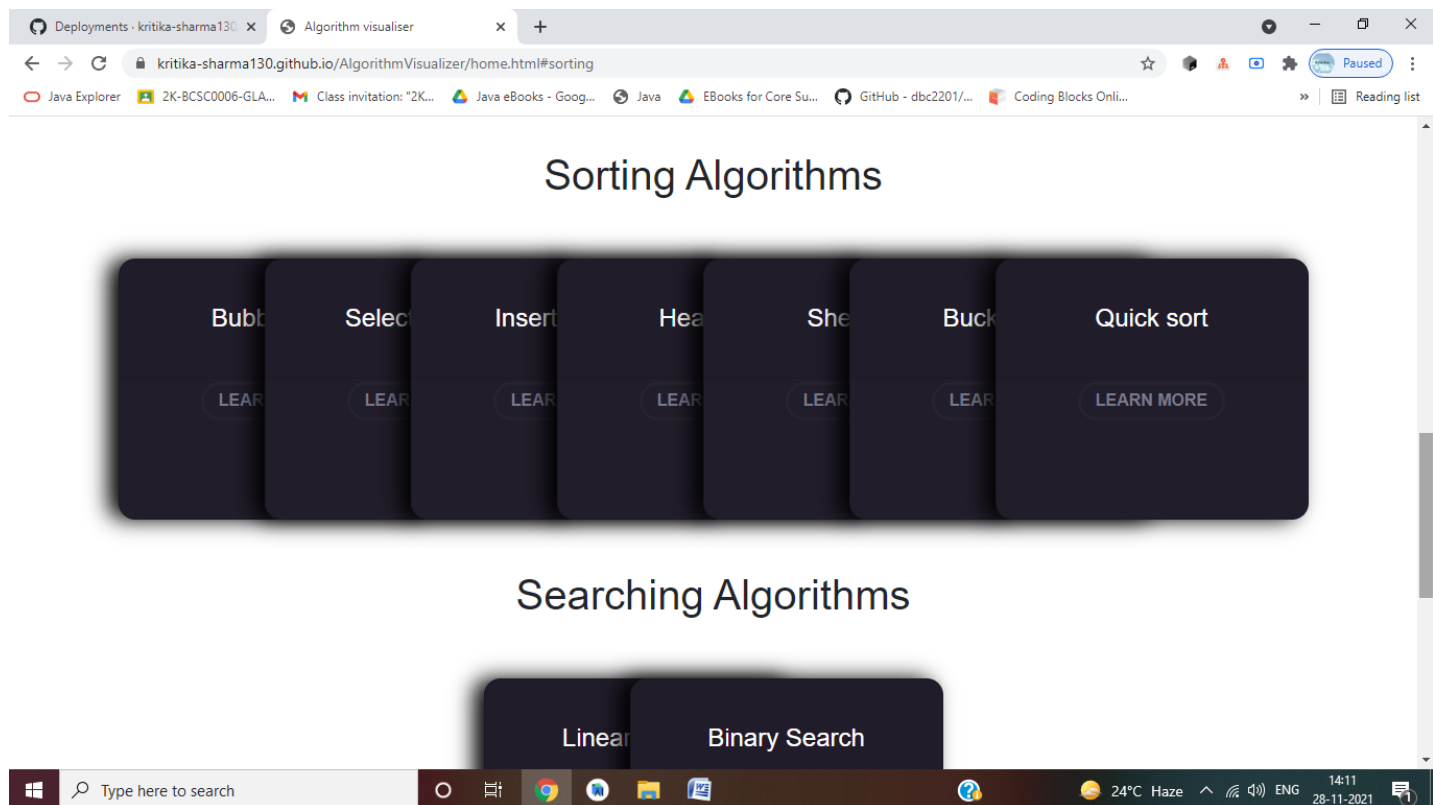
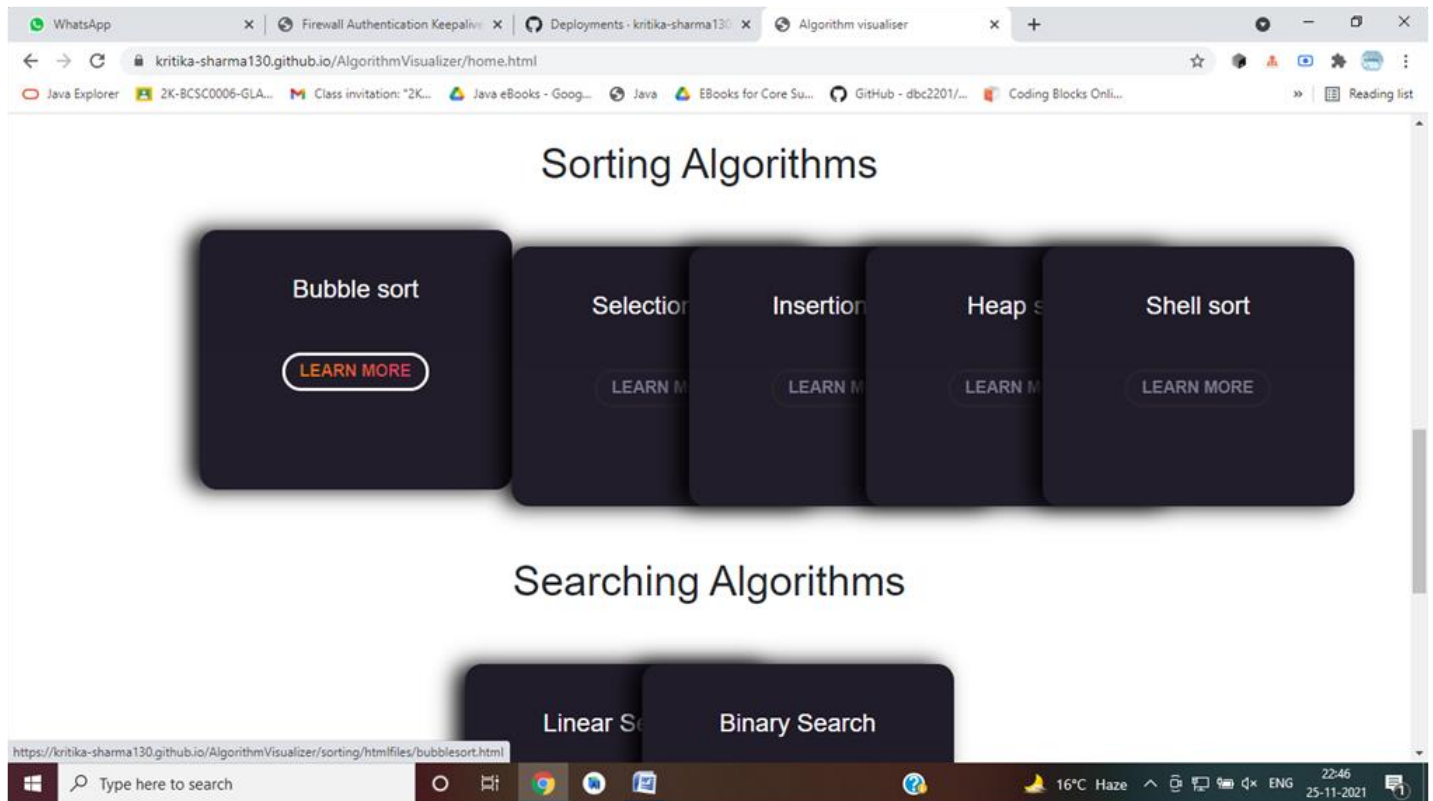
**The picture given above depicts the home page which is the first page visited by the user upon visiting the website. The page displays the theme with “ALGORITHM VISUALIZER” written upon it.**

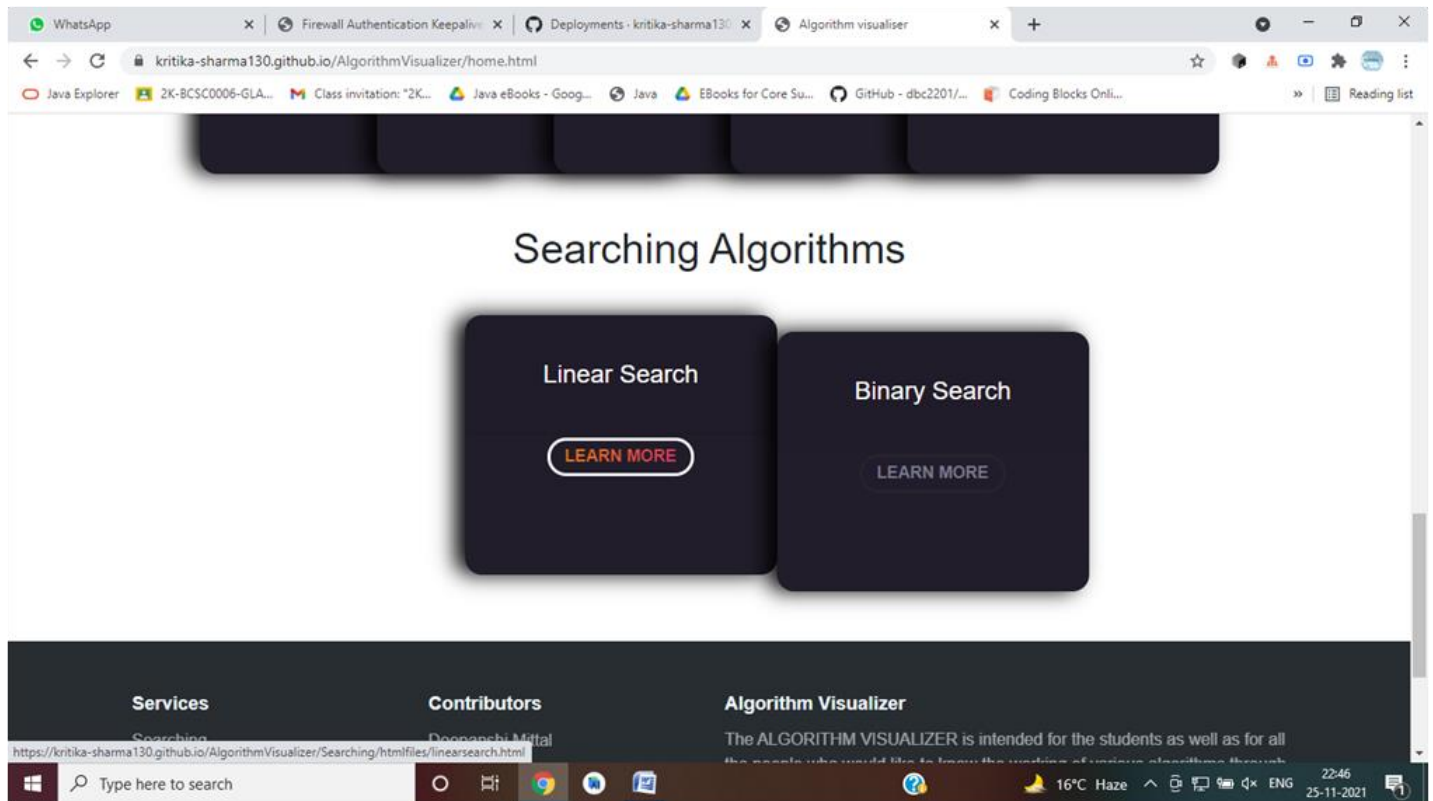
**This is the page which is followed by the homepage. The page describes the vision of the project which describes the use of our project in a real life scenario.**



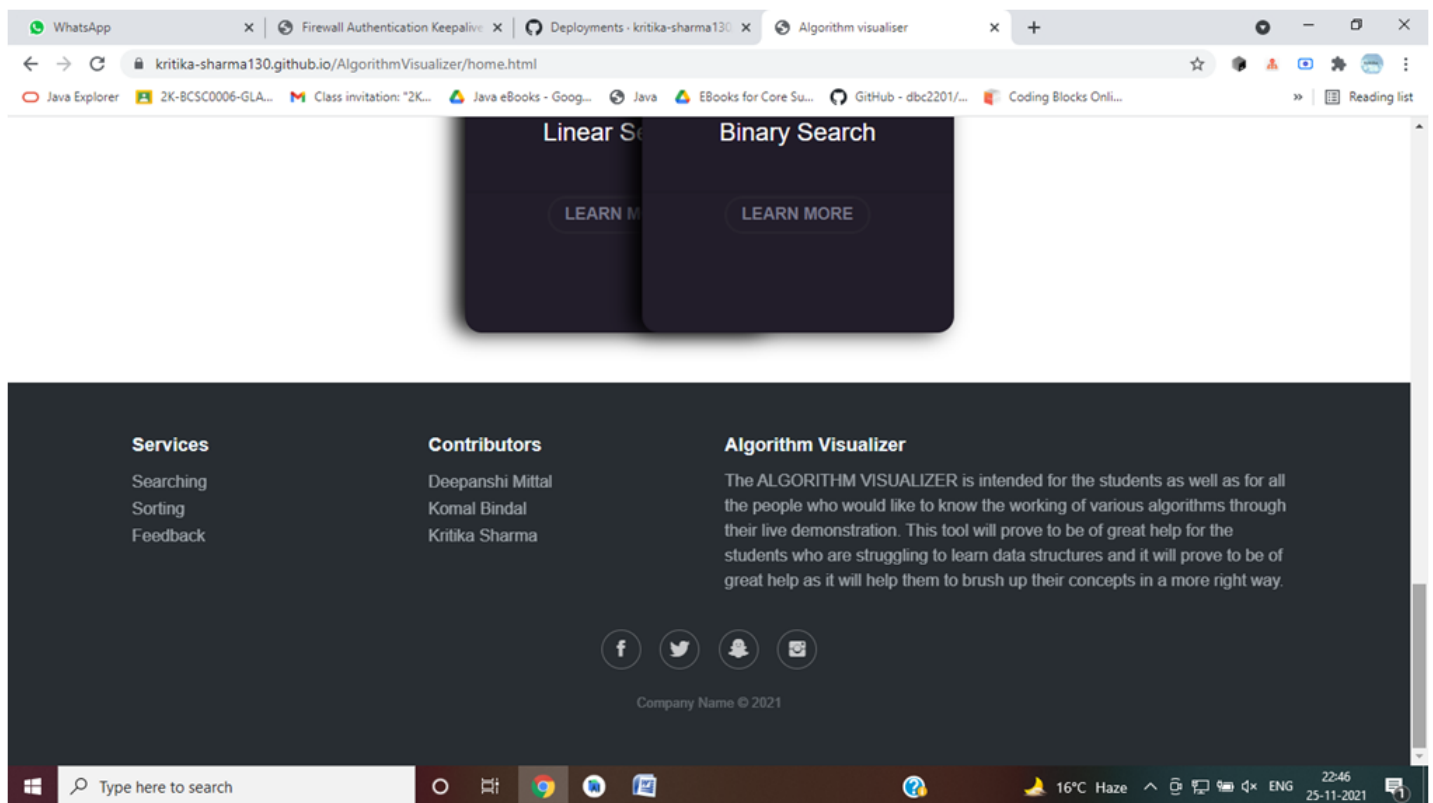


**The page given is followed by the vision page. The page displays the cards which are dynamic in nature. The card on being hovered changes its color and shows transition effects. The learn more is used to learn a particular algorithm technique .The button on being clicked takes us to the page showing the working of the respective algorithm technique.**





The above page depicts the cards for the searching algorithm. The card on being hovered shows transition and on clicking the learn more it takes you to the respective algorithm technique.

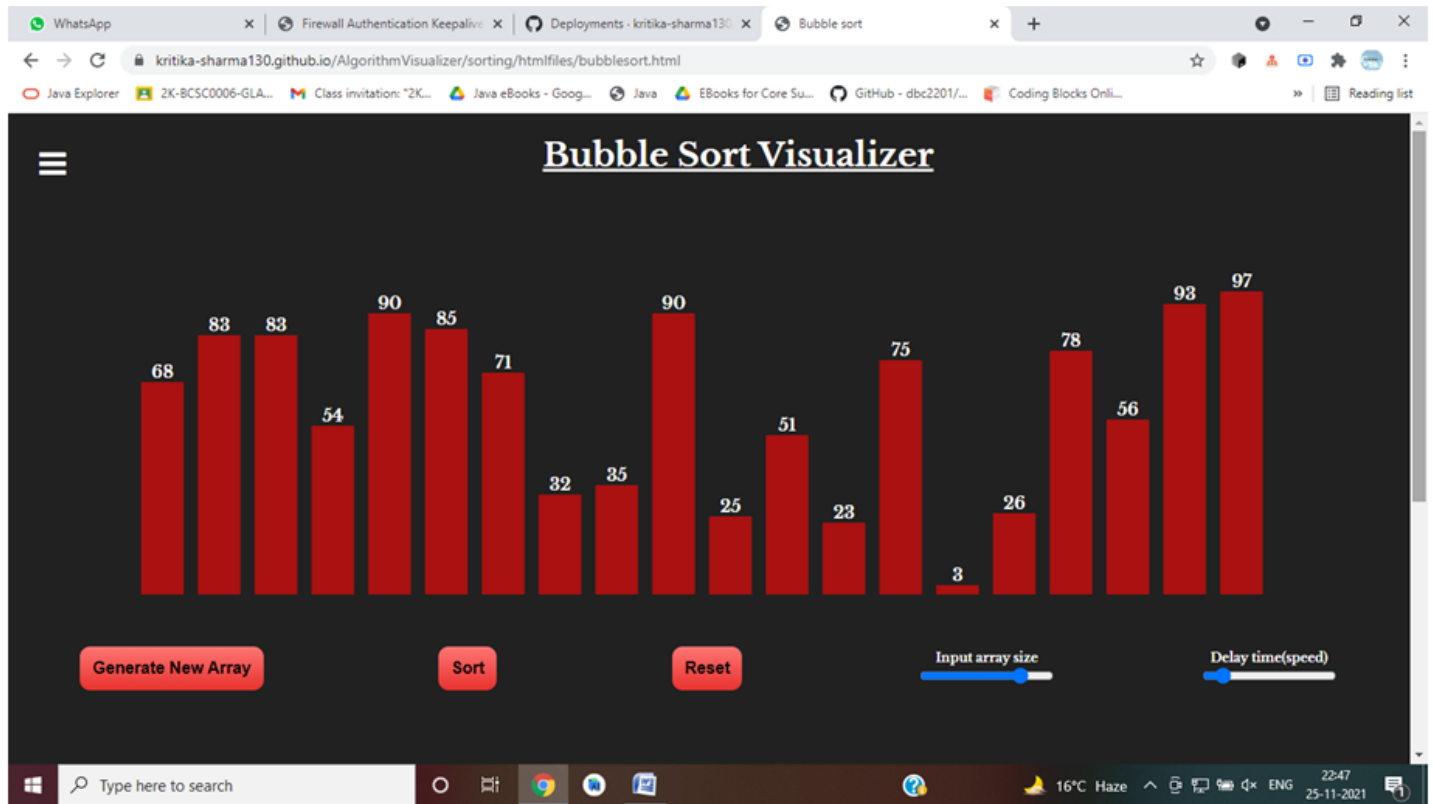


The picture given above depicts the footer section of the page which contains the important information regarding the contributors, services and the basic description of the project. The links in the picture are also dynamic in nature, on being clicked takes us to the respective page .

The screenshot displays a web browser window with multiple tabs. The active tab is titled 'Feedback Form' and shows a URL from 'kritika-sharma130.github.io'. The page content features a 'Feedback Form' with a 5-star rating system. The first three stars are filled yellow, and the word 'Awesome' is displayed below them. A text input field contains the word 'Good', and a 'POST' button is located at the bottom of the form. The browser's taskbar at the bottom includes a search bar, task view icon, and system tray showing the date as 25-11-2021.

The picture given above depicts the feedback section for the user , where the user can share their feedback , which can be used for further improvements of the website.

## SORTING ALGORITHMS:



The picture given above depicts the Bubble Sort Algorithm which falls under the category of SORTING. The picture also contains the Generate New Array button which is used to generate the new array. The Sort button is used to Sort the given array and the Reset button is used to reset to the original order . Apart from that the website also contains two dynamic features which is to change input array size which can change the size of the array and the delay which is used to control the speed of the sorting algorithm technique.

WhatsApp | Firewall Authentication Keepalive | Deployments - kritika-sharma130 | Bubble sort

kritika-sharma130.github.io/AlgorithmVisualizer/sorting/htmlfiles/bubblesort.html

Java Explorer | 2K-BCSC0006-GLA... | Class invitation: "2K... | Java eBooks - Goog... | Java | eBooks for Core Su... | GitHub - dbc2201/... | Coding Blocks Onli... | Reading list

## Bubble Sort

Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order.

Time Complexity:  $O(n^2)$  as there are two nested loops.

C | C++ | Java | Python | Javascript

```
// Optimized java implementation
// of Bubble sort
import java.io.*;

class GFG
{
    // An optimized version of Bubble Sort
    static void bubbleSort(int arr[], int n)
    {
        int i, j, temp;
        boolean swapped;
        for (i = 0; i < n - 1; i++)
        {
            swapped = false;
            for (j = 0; j < n - i - 1; j++)
            {
                if (arr[j] > arr[j + 1])
                {
                    // swap arr[j] and arr[j+1]
                    temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                    swapped = true;
                }
            }
        }
    }
}
```

## Visualizer

Input array size | Delay time(speed)

14°C Haze | 00:11 | 26-11-2021

The sorting algorithm page also contains a hamburger menu which on being clicked gives a complete description regarding the sorting algorithm.

WhatsApp | Firewall Authentication Keepalive | Deployments - kritika-sharma130 | Selection sort

kritika-sharma130.github.io/AlgorithmVisualizer/sorting/htmlfiles/selection.html

Java Explorer | 2K-BCSC0006-GLA... | Class invitation: "2K... | Java eBooks - Goog... | Java | eBooks for Core Su... | GitHub - dbc2201/... | Coding Blocks Onli... | Reading list

## Selection Sort Visualizer

- Unsorted array
- Sorted array
- Selected element
- Current element

Generate New Array | Sort | Reset

16°C Haze | 22:47 | 25-11-2021

The picture given above depicts the selection algorithm .

The screenshot shows a web browser with multiple tabs. The active tab is titled "Selection sort" and displays a page from "kritika-sharma130.github.io/AlgorithmVisualizer/sorting/htmlfiles/selection.html". The page content includes a title "Selection Sort", a description of the algorithm, and a list of two subarrays. A code editor shows a C program for selection sort. To the right, a visualizer displays an array of numbers: 84, 54, 26, 80, 16, 83, 50, 69, 2. A legend indicates: Unsorted array (red), Sorted array (green), Selected element (blue), and Current element (purple). The visualizer also has sliders for "Input array size" and "Delay time(speed)".

### Selection Sort

The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning.

The algorithm maintains two subarrays in a given array.

1. The subarray which is already sorted.
2. Remaining subarray which is unsorted.

In every iteration of selection sort, the minimum element (considering ascending order) from the unsorted subarray is picked and moved to the sorted subarray.

Time Complexity:  $O(n^2)$  as there are two nested loops.  
Auxiliary Space:  $O(1)$

Language: C C++ Java Python Javascript

```
// C program for implementation of selection sort
#include <stdio.h>

void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

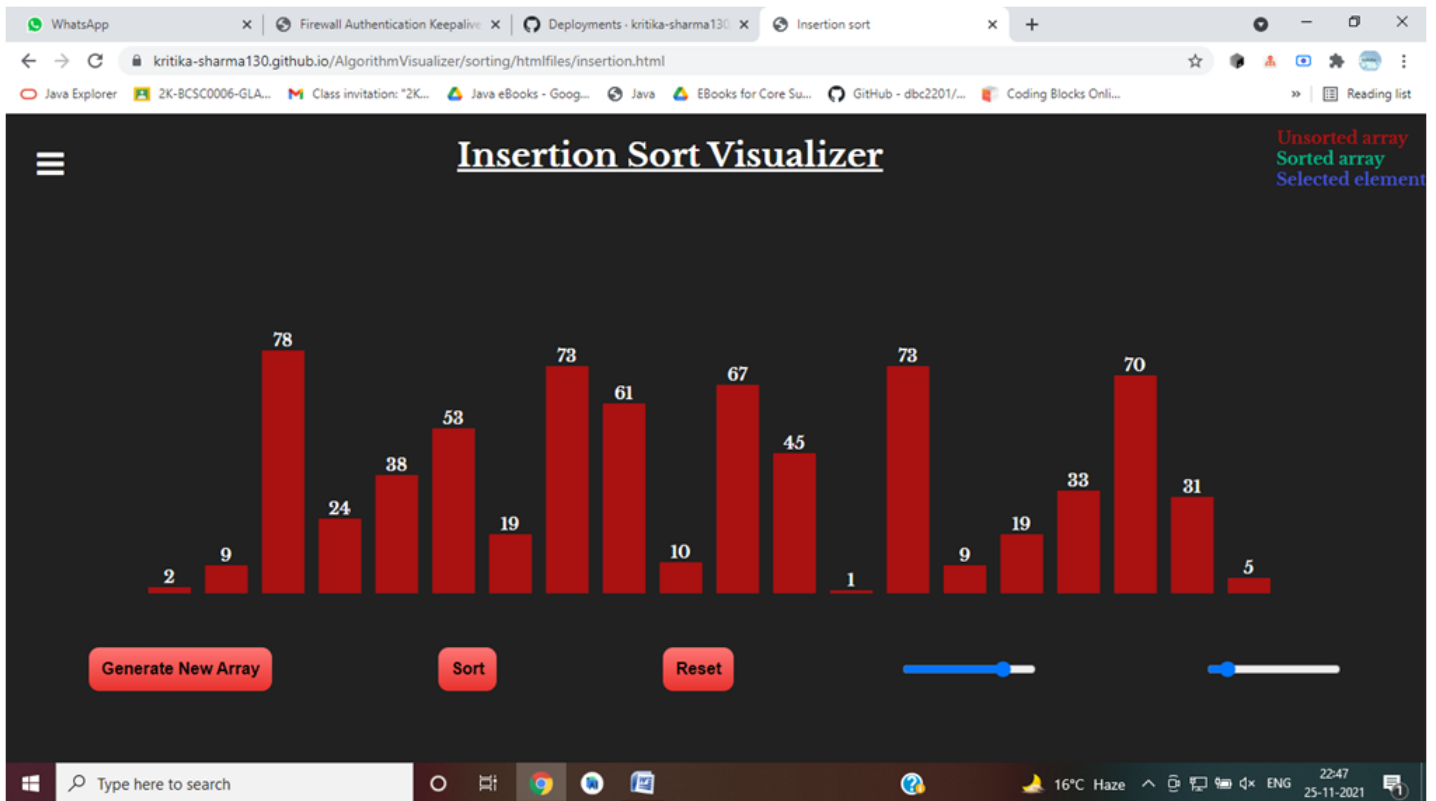
void selectionSort(int arr[], int n)
{
    for (int i = 0; i < n-1; i++)
    {
        int min = i;
        for (int j = i+1; j < n; j++)
        {
            if (arr[j] < arr[min])
                min = j;
        }
        swap(&arr[i], &arr[min]);
    }
}
```

Visualizer: 84, 54, 26, 80, 16, 83, 50, 69, 2

Legend: Unsorted array (red), Sorted array (green), Selected element (blue), Current element (purple)

Input array size: [slider] Delay time(speed): [slider]

The picture given above depicts the menu regarding the Selection Sort Algorithm. It shows the detailed description about the selection sort technique algorithm.



The picture given above depicts the page containing the insertion sort algorithm technique .

**Insertion Sort**

Insertion sort is a simple sorting algorithm that works similar to the way you sort playing cards in your hands. The array is virtually split into a sorted and an unsorted part. Values from the unsorted part are picked and placed at the correct position in the sorted part.

To sort an array of size  $n$  in ascending order:

1. Iterate from  $arr[1]$  to  $arr[n]$  over the array.
2. Compare the current element (key) to its predecessor.
3. If the key element is smaller than its predecessor, compare it to the elements before. Move the greater elements one position up to make space for the swapped element.

Time Complexity:  $O(n^2)$   
Auxiliary Space:  $O(1)$

C C++ Java Python Javascript

```
// C program for insertion sort
#include <stdio.h>
#include <stdlib.h>

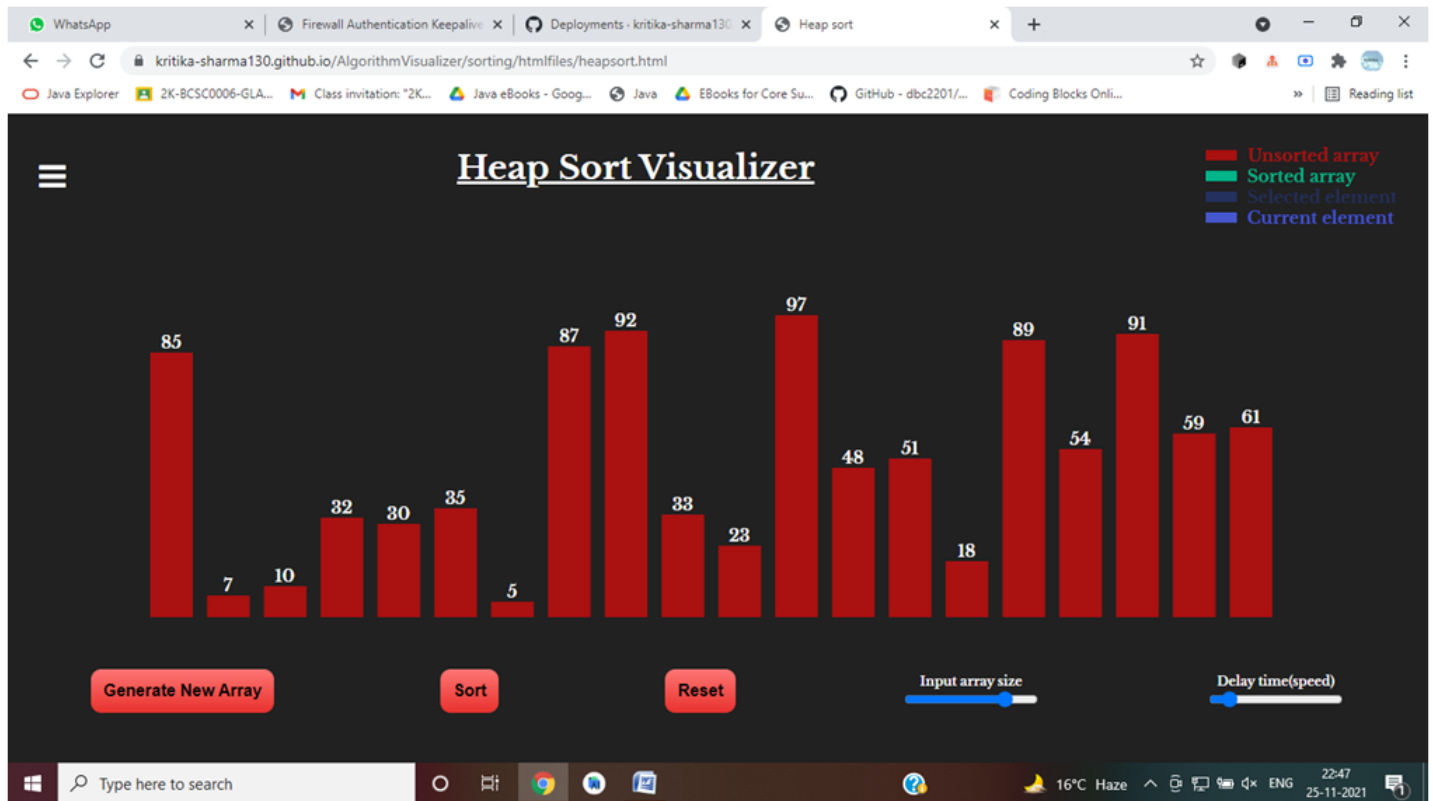
/* Function to sort an array using insertion sort*/
void insertionSort(int arr[], int n)
{
    int i, key, j;
    for (i = 1; i < n; i++) {
        key = arr[i];
        j = i - 1;
        while (j >= 0 && arr[j] > key)
            arr[j + 1] = arr[j];
        arr[j + 1] = key;
    }
}
```

97 28 46 26 97 49 72 89 9

Input array size Delay time(speed)

The picture given above depicts the slider window for the insertion sort technique algorithm.





The picture given above depicts the heap sort algorithm technique.

### Heap Sort

Heap sort is a comparison-based sorting technique based on Binary Heap data structure. It is similar to selection sort where we first find the minimum element and place the minimum element at the beginning. We repeat the same process for the remaining elements.

Time Complexity:  $O(n \log n)$

C#

C++

Java

Python

Javascript

```
// C# program for implementation of Heap Sort
using System;

public class HeapSort {
    public void sort(int[] arr)
    {
        int n = arr.Length;

        // Build heap (rearrange array)
        for (int i = n / 2 - 1; i >= 0; i--)
            heapify(arr, n, i);

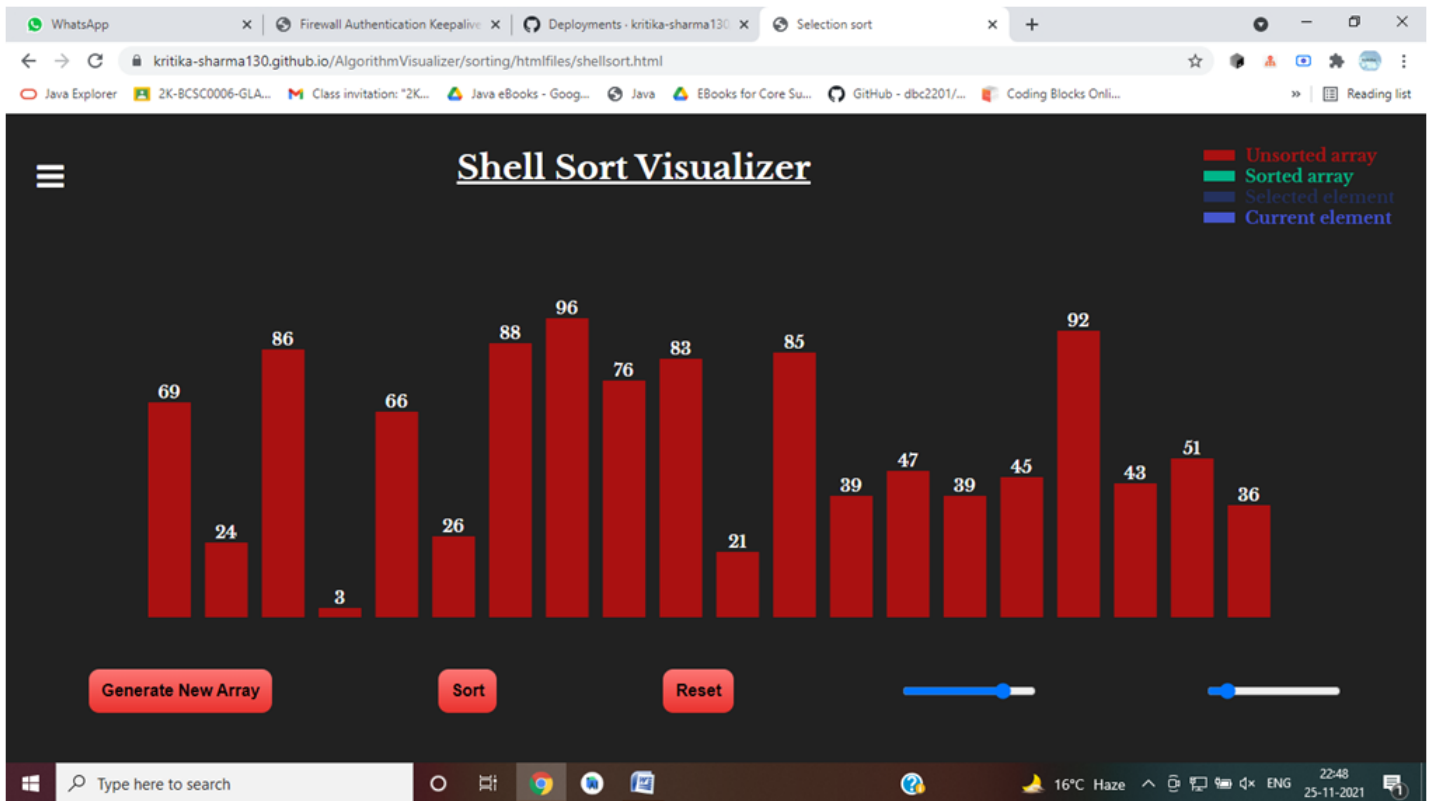
        // One by one extract an element from heap
        for (int i = n - 1; i > 0; i--) {
            // Move current root to end
            int temp = arr[0];
            arr[0] = arr[i];
            arr[i] = temp;

            // call max heapify on the reduced heap
            heapify(arr, i, 0);
        }
    }
}
```

Array values: 38, 55, 84, 12, 56, 23, 18, 65, 76.

Sliders: Input array size, Delay time(speed).

The picture given above depicts the slider window for the heap sort algorithm .



The picture given above depicts the shell Sort Algorithm technique .

### Shell Sort

ShellSort is mainly a variation of Insertion Sort. In insertion sort, we move elements only one position ahead. When an element has to be moved far ahead, many movements are involved. The idea of shellSort is to allow exchange of far items. In shellSort, we make the array h-sorted for a large value of h. We keep reducing the value of h until it becomes 1. An array is said to be h-sorted if all sublists of every h'th element is sorted.

Time complexity of above implementation of shellsort is  $O(n^2)$ .

C#

C++

Java

Python

Javascript

```
// C# implementation of ShellSort
using System;

class ShellSort
{
    /* An utility function to
    print array of size n*/
    static void printArray(int []arr)
    {
        int n = arr.Length;
        for (int i=0; i < n; i++)
        {
            // Do a gapped insertion sort for this gap size.
            // The first gap elements a[0..gap-1] are already
            // in gapped order keep adding one more element
            // until the entire array is gap sorted
            for (int i = gap; i < n; i += 1)
            {
                // add a[i] to the elements that have
            }
        }
    }
}
```

er

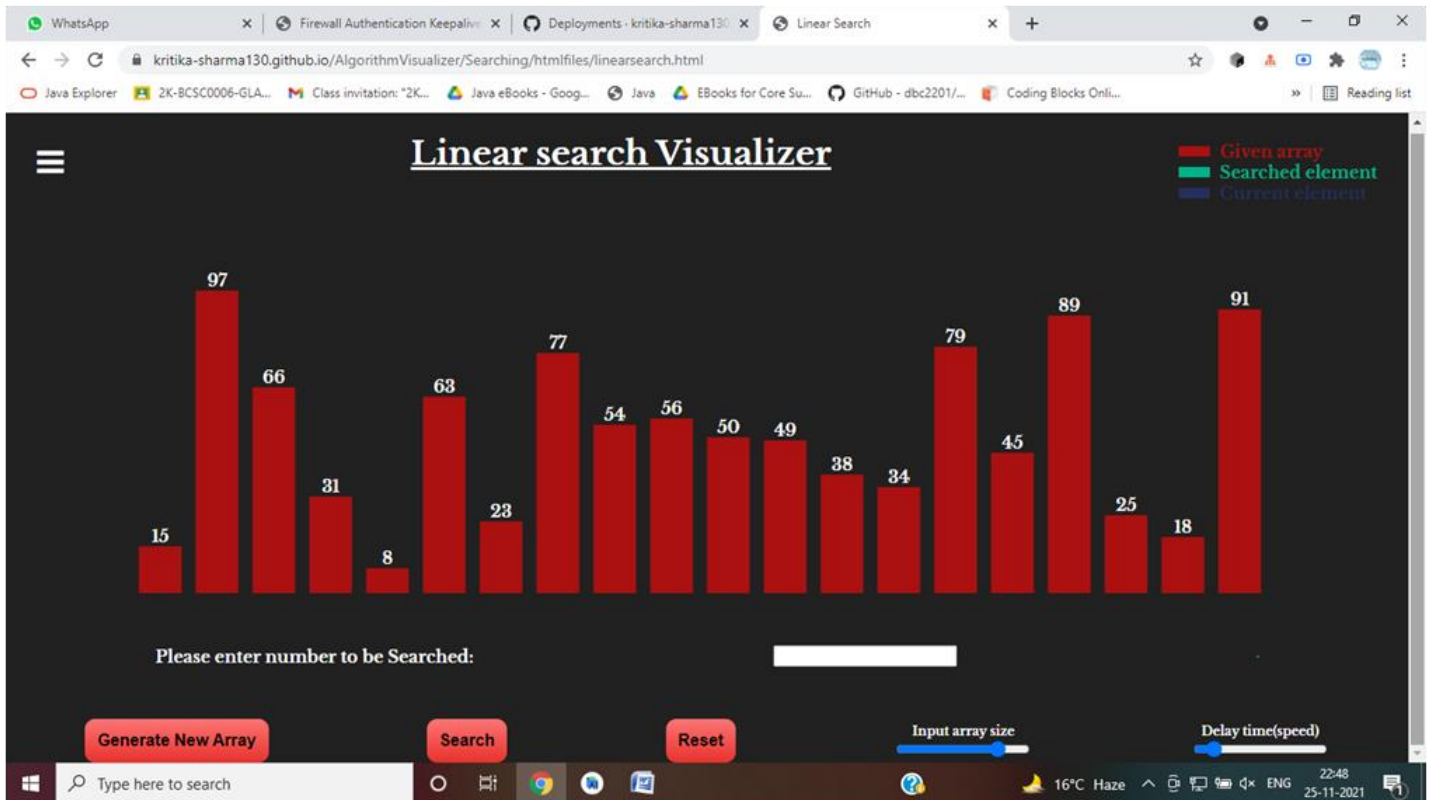
Unsorted array: 20, 3, 26, 81, 70, 85, 20, 40, 30

Sliders: Input array size, Delay time(speed)

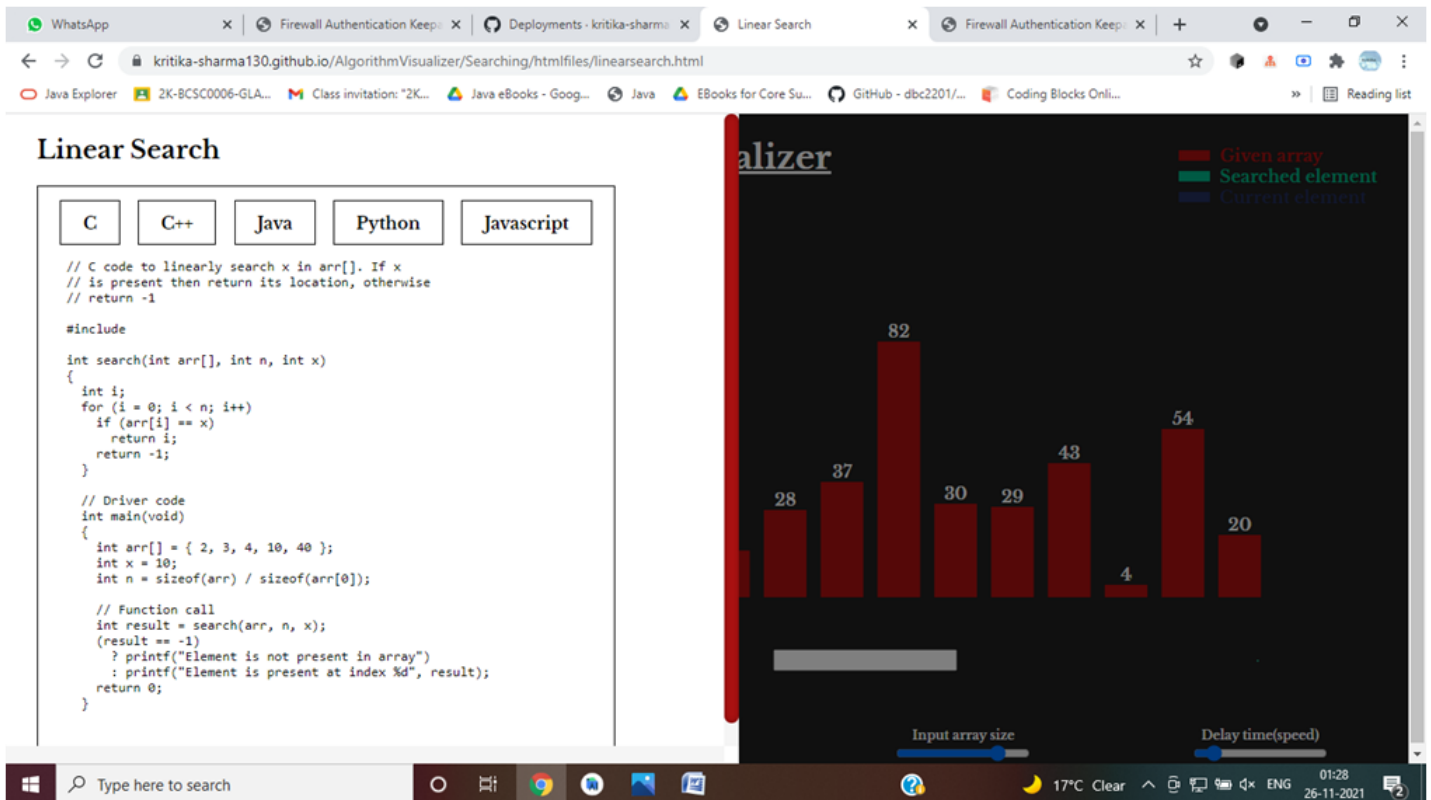
Legend: Unsorted array (red), Sorted array (green), Selected element (blue), Current element (purple)

The picture given above depicts the slider window for the Shell Sort Algorithm technique.

# SEARCHING



The picture given above depicts the linear search algorithm.



The picture given above depicts the representation of the slider for the Linear Search Algorithm.



The picture given above depicts the representation of a Binary Search Algorithm.

WhatsApp | Firewall Authentication Keep: x | Deployments - kritika-sharma x | Binary Search | Firewall Authentication Keep: x | +

kritika-sharma130.github.io/AlgorithmVisualizer/Searching/htmlfiles/binarysearch.html

Java Explorer | 2K-BCSC0006-GLA... | Class invitation: "2K... | Java eBooks - Goog... | Java | eBooks for Core Su... | GitHub - dbc2201/... | Coding Blocks Onli... | Reading list

## Binary Search

C C++ Java Python Javascript

```
// C program to implement recursive Binary Search
#include

// A recursive binary search function. It returns
// location of x in given array arr[l..r] is present,
// otherwise -1
int binarySearch(int arr[], int l, int r, int x)
{
    if (r >= l) {
        int mid = l + (r - l) / 2;

        // If the element is present at the middle
        // itself
        if (arr[mid] == x)
            return mid;

        // If element is smaller than mid, then
        // it can only be present in left subarray
        if (arr[mid] > x)
            return binarySearch(arr, l, mid - 1, x);

        // Else the element can only be present
        // in right subarray
        return binarySearch(arr, mid + 1, r, x);
    }

    // We reach here when element is not
    // present in array
    return -1;
}
```

Visualizer

Given array  
Searched element  
Current element

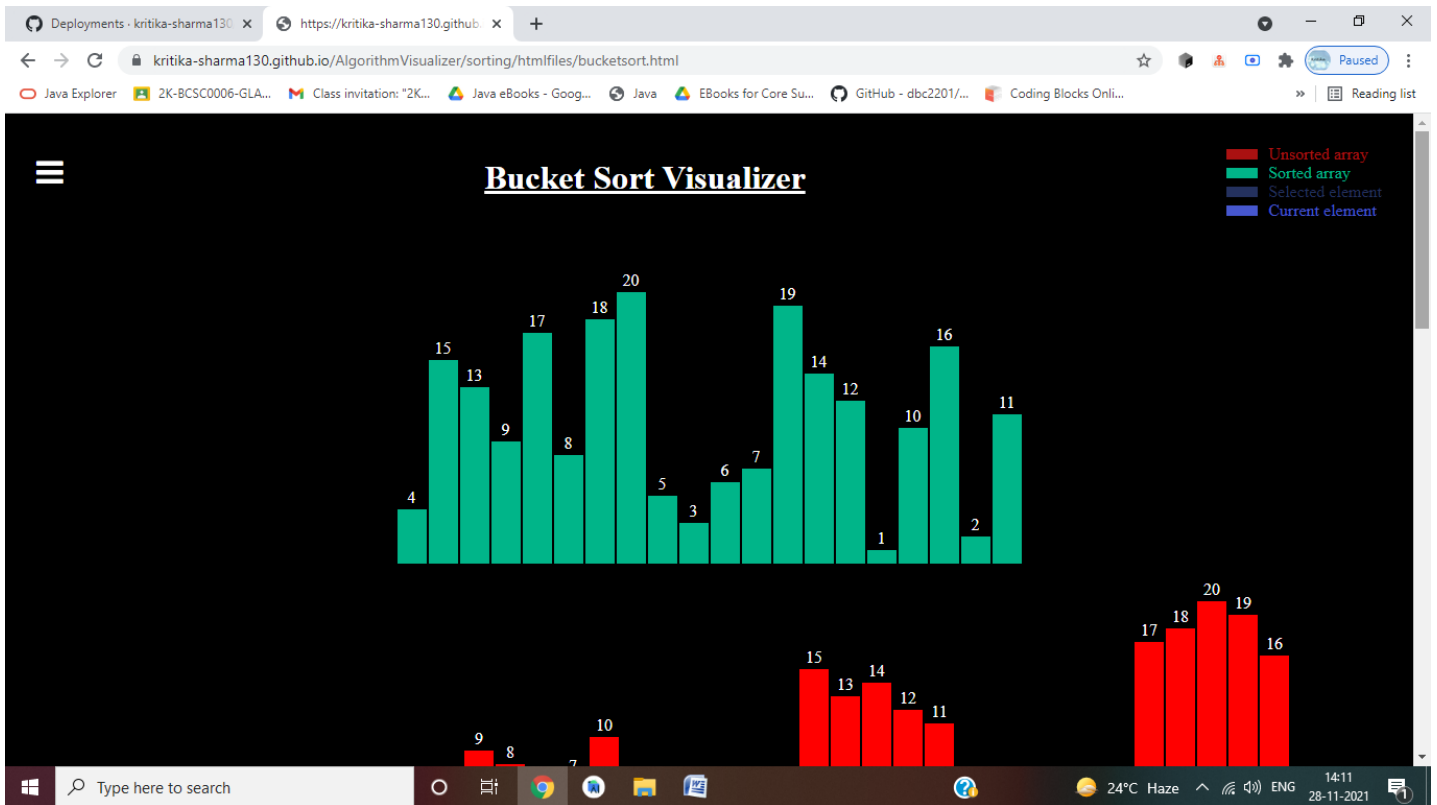
41 45 47 53 72 74 81 97 99

Input array size Delay time(speed)

Type here to search

17°C Clear 01:30 26-11-2021

The picture given above depicts the representation of the slider of binary search algorithm.



The picture given above depicts the bucket sort algorithm technique.

Deployments - kritika-sharma130 x https://kritika-sharma130.github.io x +

kritika-sharma130.github.io/AlgorithmVisualizer/sorting/htmlfiles/bucketsort.html

Java Explorer 2K-BCSC0006-GLA... Class invitation: "2K... Java eBooks - Goog... Java eBooks for Core Su... GitHub - dbc2201/... Coding Blocks Onli... Reading list

### Bucket Sort

Bucket sort is mainly useful when input is uniformly distributed over a range. For example, consider the following problem. Sort a large set of floating point numbers which are in range from 0.0 to 1.0 and are uniformly distributed across the range. How do we sort the numbers efficiently? A simple way is to apply a comparison based sorting algorithm. The lower bound for Comparison based sorting algorithm (Merge Sort, Heap Sort, Quick-Sort .. etc) is  $\Omega(n \log n)$ , i.e., they cannot do better than  $n \log n$ . Can we sort the array in linear time? Counting sort can not be applied here as we use keys as index in counting sort. Here keys are floating point numbers. The idea is to use bucket sort. Following is bucket algorithm.  
Time Complexity:  $O(n)$

C# C++ Java Python Javascript

```
// Java program to sort an array
// using bucket sort
import java.util.*;
import java.util.Collections;

class GFG {

    // Function to sort arr[] of size n
    // using bucket sort
    static void bucketSort(float arr[], int n)
    {
        if (n <= 0)
            return;

        // 1) Create n empty buckets
        @SuppressWarnings("unchecked")
        Vector<Float>[] buckets = new Vector[n];

        for (int i = 0; i < n; i++) {
            buckets[i] = new Vector<Float>();
        }
    }
}
```

Type here to search

24°C Haze 14:11 28-11-2021

The above given picture depicts the slider menu window for the bucket sort algorithm.

Deployments - kritika-sharma130 x Quick Sort Hoare's Partition x +

kritika-sharma130.github.io/AlgorithmVisualizer/sorting/htmlfiles/quickSort/hoarePartition.html

Java Explorer 2K-BCSC0006-GLA... Class invitation: "2K... Java eBooks - Goog... Java eBooks for Core Su... GitHub - dbc2201/... Coding Blocks Onli... Reading list

### Quick Sort Visualizer (Hoare's Partition)

Generate New Array Sort Reset

Input array size Delay time(speed)

Type here to search

24°C Haze 14:12 28-11-2021

The picture given above depicts the Quick sort algorithm technique.



**Quick Sort**

Like Merge Sort, QuickSort is a Divide and Conquer algorithm. It picks an element as pivot and partitions the given array around the picked pivot. There are many different versions of quickSort that pick pivot in different ways.

- Always pick first element as pivot.
- Always pick last element as pivot (implemented below)
- Pick a random element as pivot.
- Pick median as pivot.

The key process in quickSort is partition(). Target of partitions is, given an array and an element x of array as pivot, put x at its correct position in sorted array and put all smaller elements (smaller than x) before x, and put all greater elements (greater than x) after x. All this should be done in linear time.

**Hoare's**

Unsorted array  
Sorted array  
Selected element  
Current element

Input array size  
Delay time(speed)

C# C++ Java Python Javascript

```
// C# implementation of QuickSort
// using Hoare's partition scheme
using System;

class GFG {

    /* This function takes first element as pivot, and
    places all the elements smaller than the pivot on the
    left side and all the elements greater than the pivot
    on the right side. It returns the index of the last
    element on the smaller side*/
    static int partition(int[] arr, int low, int high)
    {
```

The picture given above depicts the slider window for the Quick Sort Algorithm Technique.

Authentication

Users Sign-in method Templates Usage

Search by email address, phone number, or user UID Add user

Identifier	Providers	Created	Signed In	User UID
komal@gmail.com		Nov 28, 2021	Nov 28, 2021	UAcsJLsJDge32YcOf7IUuz1FqE3
dsjaikjd@gmail.com		Nov 28, 2021	Nov 28, 2021	yaPcRgZXz3WEyC6DI3Sr6sAPejw1
msakmk@gmail.com		Nov 28, 2021	Nov 28, 2021	MWlJYoQJmUcRSruPwHTCDonu...
acbc@gmail.com		Nov 28, 2021	Nov 28, 2021	MxC0Jd5yolaAD1GM4CWBz2gez...
kritika.sharma@gmail.com		Nov 26, 2021	Nov 26, 2021	F0uVSPiOktbHRBub9xihqOSJ9rp2
acder@gmail.com		Nov 26, 2021	Nov 26, 2021	ors3Bl0eiXa27OsY6DBqRvdi0N2
abc@gmail.com		Nov 25, 2021	Nov 28, 2021	nmQWvsXvGfVjotQcVfj0c3Q2gq2

The above picture depicts the firebase login activity for the project.



# **CHAPTER - 6**

## **TESTING**

Once source code has been generated, software must be tested to uncover as many errors as possible before delivery. It is very important to work the system successfully and achieve high quality of software. Testing include designing a series of test cases that have a high likelihood of finding errors by applying software-testing techniques.

System testing makes logical assumptions that if all the parts of the system are correct, the goal will be successfully achieved. The system should be checked logically. Validations and cross checks should be there. Avoid duplications of record that cause redundancy of data.

In other Words, Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. It is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

We have tested our website the website is fully functional with proper responsiveness.

There are different types of testing some of them are listed below:

### **6.1 Installation Testing:**

There are two types of apps on an Android device i.e., Pre-installed applications and the applications which are installed later by the user.

For both of the above, installation testing is carried out by our teammates. It is ensuring smooth installation of the application without ending up in errors, partial installation etc.

### **6.2 Unit Testing**

It focuses on smallest unit of software design. In this we test an individual unit or groups of inter related units. It is often done by programmer by using sample input and observing its corresponding outputs. In this testing technique we are primarily focuses on

- Loop methods and function is working fine or not.
- Misunderstood or incorrect Arithmetic precedence
- Incorrect Initialization

## **CHAPTER -7**

### **CONCLUSION**

The ALGORITHM VISUALIZER is intended for the students as well as for all the people who would like to know the working of various algorithms through their live demonstration. This tool will prove to be of great help for the students who are struggling to learn data structures and it will prove to be of great help as it will help them to brush up their concepts in a more right way. This application has wide range of scope in the upcoming era.

### **REFERENCES**

#### **1. Introduction to HTML,CSS AND JAVASCRIPT:**

<https://www.beta-labs.in/>

<https://www.w3schools.com/>