ALGOLEARN

A project report submitted by

RESHMA M U (GTATSCS003)

JOMOL THOMAS (GTATSCS018)

SANDHYA V (GTATSCS022)

QULIYET JACK H A (GTATSCS021)

Under the guidance of
DR. JEEVAMOL JOY
ASSISTANT PROFESSOR
DEPARTMENT OF COMPUTER SCIENCE

For the award of the Degree of

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

(Calicut University)

2021-2022

Submitted to

Department of Computer Science
Sri. C. Achuthamenon Government College, Thrissur
Kuttanellur, Kerala

DECLARATION

We wish to state that the work embodied in this project titled ***ALGOLEARN*** *forms our own contribution to the project work carried out under the guidance of Dr.Jeevamol Joy , Assistant professor , Department of Computer Science, Sri. C. Achuthamenon Government College, Kuttanellur, Thrissur . We hereby declare that this work is submitted to University of Calicut in partial fulfillment of the requirement for the award of the degree of Bachelor of Computer Science and this submission contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text*

Reshma M U	**(GTATSCS003)**
Jomol Thomas	**(GTATSCS018)**
Sandhya V	**(GTATSCS018)**
QuliyetJack H A	**(GTATSCS021)**

Place

Date:

CONTENTS

ALGOLEARN

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 ALGOLEARN (Algorithm Visualization Website)

All students who have to deal with algorithms in their studies may not be able to understand the algorithms and their logic properly. It is a famous talk that a picture equals to thousand words. If we can describe Algorithms and their logic through some pictures or Animations many students can solve their difficulty in understanding the algorithms properly.

This project concentrates on making visualizations of some basic algorithms so that students can easily understand the basics of programming. As an Algorithm is a bridge to coding, Understanding Algorithms clearly will help to concrete their confidence in coding.

1.2 Features of Existing Systems

There are existing systems of Algorithm visualization. They are showing a general predefined structure of visualization. During the analysis phase, the existing system was studied, currently, the existing system contains sorting, searching, graph, and tree algorithms.

1.3 Limitations of Existing Systems

The existing system is not making a friendly approach to the beginner. It is just made for a general audience it is not specifically useful for a beginner. And it does not have provisions for receiving inputs from the user.

1.4 Area and Category of the Project Work

This project will be helpful mainly for the students in the field of computer science as a beginner and also helpful to general students. This project will also be helpful for tutors in teaching algorithms.

Chapter 2

PROBLEM DEFINITION & METHODOLOGY

2.1 Problem Definition

This project will make a visualization of basic Algorithms. This Project is intended to understand basic algorithms and concretes their base in coding.

2.2 Objectives

Through this project, we can help students who want to concrete their field in the field of computer science. And will be helpful to teachers also. They can use this to share knowledge with others. Whenever they will have any doubts about the basics, they can revisit it easily as visualization as this project provides a simple visualization of Algorithms.

2.3 Motivation

Innovations are the solutions to some problems. In this Arise of covid scenario, both teachers and students find difficulty in the process of learning. The teachers who find difficulty in teaching and students who find difficulty in learning are our biggest motivations. The computer science students' aroused covid scenario learning problems can be solved through these visualization tools.

2.4 Methodology

Our project is a web-based animation tool. user can select the desired algorithm and can give inputs. They can see the visualizations of selected algorithms. There will be a small box that shows the description of the animation process.

We use

- HTML
- CSS
- Java Script

We decided to include Algorithms of:

- Bubble sort
- Insertion sort
- Selection sort
- Binary search
- Linear search

2.5 Scope

As this project is a visualization of basic algorithms this will have a great scope in the future. we can include more visualization of algorithms and later this can be developed into apps and can modify accordingly.

Chapter 3

ANALYSIS

3.1 Requirement Analysis

Requirement analysis is the process of defining user expectations for a new software being built or modified.

Home Page: it is the page which viewed when the website opens. It contains headings, a description of the site, and a list of algorithms. Each element in the algorithm list has a link to a new page that shows the visualization

Animation window to display animation according to algorithm logic

Control buttons to control animation.

It should work on hardware systems such as laptops, tablets, touch phones, and so on.

3.2 Existing System

An algorithm visualization system is an existing system in the form of websites and apps.

Some examples of websites are: -

- Visual Algo - Visualizing data structures and algorithms through animation.
- Algorithm Visualizer - Visualize algorithm from code
- Data Structure Visualization - Visualize data structures

Some examples of apps are: -

- Algorithm Visualizer
- Algorithm Learn Easy
- Algorithms
- Data Structures and Algorithms.

3.3 Proposed System

We proposed to implement visualization of sorting algorithms, and searching algorithms. It will have options for creating arrays and a small description of the processes happening in the animation.

3.4 Requirement Specification

3.4.1 Functional Requirements

User: The person who views the website and controls the functions

User input: Functional mechanism to control the movement of the website.

The selection of links, Control buttons are considered as user input that controls the movement.

Animation Window: A window box to display the animation of the algorithm.

Control Buttons: It is the buttons for controlling the animation like set and reset.

Selection button: It is the button for selecting an algorithm.

Algorithms List: It helps the user to select an algorithm. When clicking on the list item user can move the page to the animation page.

The list items are: -

- Sorting Algorithms
- Searching Algorithms

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.
Implemented sorting algorithms are:

- Bubble sort
- Insertion sort
- Selection sort

Bubble sort

Bubble sort, sometimes referred to as sinking sort, is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. the pass through the list is repeated until the list is sorted. This simple algorithm performs poorly in real-world use and is used primarily as an educational tool. The algorithm which is a comparison sort, is named for the way smaller or larger elements "bubble" to the top of the list. Bubble sort has a worst-case and average complexity of O(n^2) when n is the number of items being sorted.

Insertion sort

Insertion sort is a simple sorting algorithm that builds the final sorted array one item at a time. It is much less efficient on large lists than more advanced algorithms. Insertion sort iterates, consuming one input element each repetition, and grows a sorted output list. At each iteration, insertion sort removes one element from the input data, finds the location it belongs within the sorted list, and inserts it there. It repeats until no input elements remains there.

Selection sort

Selection sort is an in-place comparison 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. Selection sort is noted for its simplicity and has a performance advantage over more complicated algorithms.

The algorithm divides the input list into two parts: a sorted sub-list of items which is built up from left to right at the front of the list and a sub-list of the remaining unsorted items that occupy the rest of the list. Initially the sorted sub-list is empty and unsorted sub-list is the entire input list. The algorithm proceeds by finding the smallest in the unsorted sub-list, exchanging it with the leftmost unsorted element, and moving the sub-list boundaries one element to the right.

Searching Algorithms

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:

1. **Sequential Search**: In this, the list or array is traversed sequentially and every element is checked. For example, Linear Search.
2. **Interval Search**: These algorithms are specifically designed for searching in sorted data structures. These types of search 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.

This project implements linear search and binary search.

Linear search

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.

Linear search is usually very simple to implement and is practical when the list has only a few elements, or when performing a single search in an unordered list. A linear search runs in at the worst linear time and makes at most n comparisons where n is the length of the list.

Binary search

Binary search work on sorted arrays. Binary search begins by comparing an element in the middle of the array within the target value. If the target value matches the element, its position in the array is returned. If the target value is less than the element, the search continues in the lower half of the array. If the target value is greater than the element, the search continues in the upper half of the array. By doing this, the algorithm eliminates the half in which the target value cannot lie in each iteration. Binary search is faster than linear search except for small arrays.

3.4.2 Non-functional Requirements

A product needs to work properly and have distinguishable features. With nonfunctional requirements, you can build a product with unique properties. Knowing examples of nonfunctional requirements and how they work in an application can help you design a system that meets the needs of your end-users.

Performance

The created system should work properly without failure and gives a user-friendly approach.

Reliability

Reliability refers to the extent to which an experiment, test, or measuring procedure yields the same results on repeated trials. The product that we created must be reliable.

Availability

Availability is defined as the probability that the system is operating properly when it is requested for use. In other words, availability is the probability that a system is not failing or undergoing a repair action when it needs to be used.

Security

System security refers to the technical innovations and procedures applied to the hardware and operating systems to protect against deliberate or accidental damage from a defined threat. In contrast, Data security is the protection of data from modification and destruction the security features are considered while developing the system. To avoid the errors and omissions that may lead to security problems.

System integrity refers to the proper functioning of hardware and programs, Appropriate physical security, and safety against external threats. A threat to a computer system is an event that adversely affects one or more assets or resources, which make up the system.

An event can be any of the following:

- Interruption of communication
- Destruction of hardware
- Modification of the software
- Removal of programs
- Disclose information

There are many methods for handling a threat:

- Avoid it by altering the design
- Threat retention
- Threat reduction (Frequency of occurrence of a threat is reduced)

In comparison, Data integrity makes sure that data do not differ from their original form and have not been accidentally or intentionally.

Maintainability

Maintainability is defined as the probability that a failed component or system will be restored or repaired to a specified condition within a specified period when maintenance is performed by prescribed procedures.

Portability

Portability in a high-level computer programming language is the usability of the same software in different environments. the pre-requirement for portability is the generalized abstraction between the application logic and system interfaces. When software with the same functionality is produced per several computing platforms portability is the key issue per development cost reduction.

3.4.3 Hardware Requirements

The selection of hardware configuration is a very important task related to software development. The processor should be powerful to handle entire operations.

- Processor: intel core Duo 2.0ghz or more.
- RAM: minimum 128 MB or faster
- Hard Disk: minimum 80 GB
- Monitor: normal LCD monitor
- Keyboard: standard
- Mouse: normal

3.4.4 Software Requirements

The major element in building a system is the selection of compactable software. The software requirement of the system on which the project was developed is as follows:

Operating system: Windows 7 or more
Front End : HTML, CSS, JavaScript
Web Browser: Google Chrome

3.5 Feasibility Study

A feasibility study is a system proposal according to workability, impact on the operation, ability to meet the user needs and efficient use of resources. An important outcome of the preliminary outcome is the determination that the requested system is feasible. All projects are feasible when given unlimited resources and infinite time. It is both necessary and prudent to evaluate the feasibility of a project at the earliest possible time. A feasibility study is not warranted for systems in which economic justification is obvious, technical risk is low, few legal problems are expected, and no reasonable alternatives exist. An estimate is made of whether the identified user needs may be satisfied using current software and hardware technologies.

The three key considerations involved in feasibility analysis are

- Technical feasibility
- Operational feasibility
- Economic feasibility

The system is feasible because the user can be easy to use & control electrical equipment. A feasibility study is carried out to select the best system that meets performance requirements. The main aim of the feasibility study activity is to determine whether it would be financially and technically feasible to develop the product. The feasibility study activity involves the analysis of the problem and collection of all relevant information relating to the product such as the different data items which would be input to the system, the processing required to be carried out on these data, the output data required to be produced by the system as well as various constraints on the behavior of the system.

3.5.1 Technical feasibility

Feasibility deals with hardware and software requirements it deals with the scope was whether the work for the project is done with the equipment, The outcome of the study was positive. In the proposed system, data can easily store and managed. A study of function, performance, and constraints may improve the ability to create an acceptable system. Technical feasibility is frequently the most difficult area to achieve in the storage of the product engineering process.

Considering that the normally associated with the technical feasibility include

- Development
- Resource availability
- Technology
- Technical

3.5.2 Operational feasibility

The proposed system is beneficial only if it can be turned into an information system that will meet the organization's operating requirements. The purpose of the operational feasibility study is to determine whether the new system will be used if it is developed and implemented. Here are questions that will help test the operational feasibility of a project:

- Is there sufficient support for the projects from management?
- Are current business methods acceptable to the users?
- Have the users been involved in the planning and development of the project?

Simply stated this test of feasibility asks if the system will work when it is developed and installed. And whether there will be resistance from users that will undermine the possible application benefits. This feasibility study is carried out by a small group of people who are familiar with information system techniques and are skilled in system analysis and design processes.

Proposed projects are beneficial only if they can be turned into the information system that can be turned into information systems that will meet the operating requirements of the organization. This test of feasibility asks if the system will work when it is developed and installed.

3.5.3 Economic feasibility

Economic feasibility is a kind of cost-benefit analysis of the examined project, which assesses whether it is possible to implement it. This term means the assessment and analysis of the project’s potential to support the decision-making process by objectively and rationally identifying its strengths, weakness, opportunities, and risks associated with it, the resources that will be needed to implement the project, and an assessment of its chances of success. It consists of: -

- Market analysis
- Economic analysis
- Technical analysis
- Strategic analysis

Chapter 4

Design

4.1 Introduction.

The most creative and challenging phase of the system life cycle is the system design. The term design describes a final system and the process by which it is developed. Design is the phase that indicates the final system. It is the solution, the translation of requirements in ways of meeting them. In this phase, the following elements were designed namely dataflow, data stores, processes, and procedures. Firstly, the logical design was done where the outputs, inputs, databases, and procedures were formulated in a manner that meet the project requirements. After logical design physical construction of the system is done.

4.2 Modularity Criteria

Modularity is a property of a system that measures the degree to which a system's components may be separated and recombined.
Modular design is an approach used to design a product or system.it can break down into different independent modules.

The modules use here are:

Home.html

It is the home page of the website it provides links to all other pages.

Bubblemain.html

Here animation of bubble sort is shown. A text box for accepting array size, and buttons for generating array and sorting arrays will be available here. Also, algorithms and source codes of the bubble sort will be given.

insert.html

Here animation of the insertion sort is shown. A text box for accepting array size, and buttons for generating arrays and sorting arrays will be available here. Also, algorithms and source codes of the insertion sort will be given.

selectionmain.html

Here animation of the selection sort is shown. A text box for accepting array size, and buttons for generating arrays and sorting arrays will be available here. Also, algorithms and source codes of the selection sort will be given.

mainbinary.html

Here animation of the binary search is shown. A text box for accepting array size, and buttons for generating array and searching elements will be available here. Also, algorithms and source codes of the binary search will be given.

mainlinear.html

Here animation of linear search is shown. A text box for accepting array size, and buttons for generating array and searching elements will be available here. Also, algorithms and source codes of the linear search will be given.

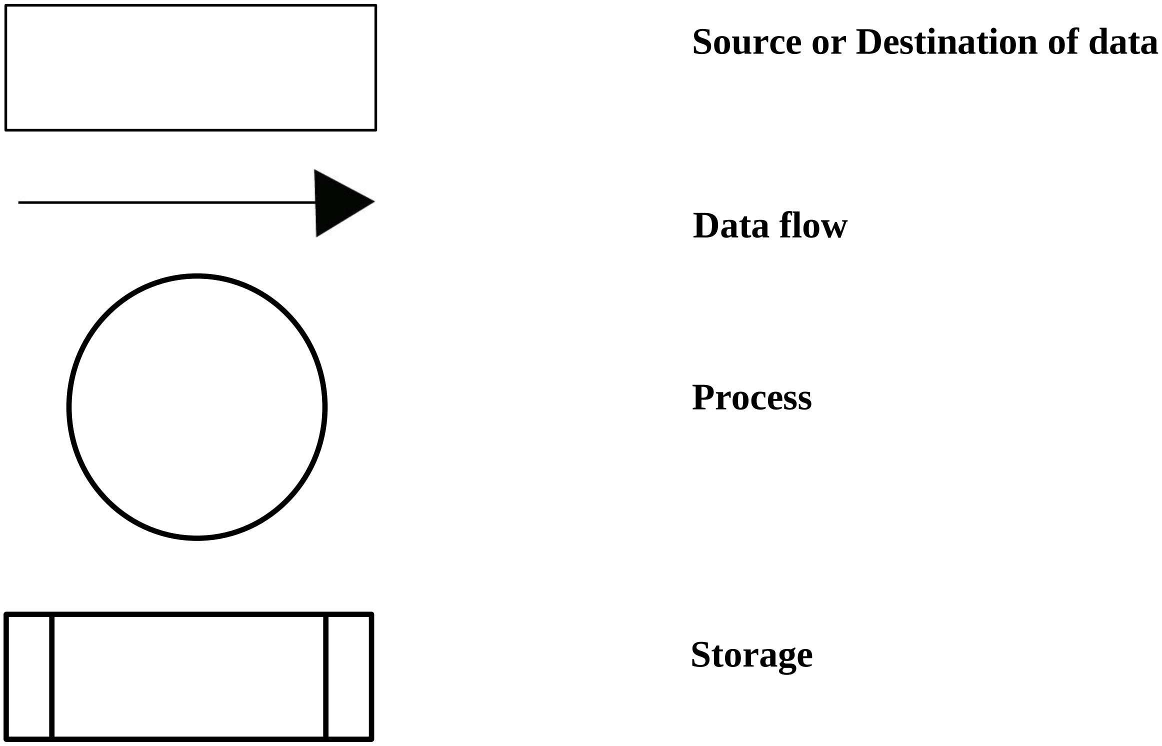
4.3 Architecture Diagrams/DFD

A data Flow Diagram (DFD) is one of the best ways of documenting the entire functionality of the system. For the system, which will have some data flows in and have some processing inside, and then some data flow out from the system can be documented or represented effectively using data flow diagrams. The data flow diagrams are a diagrammatic representation of the system, which has input, process, and outputs. Once any system is represented using a data flow diagram, we can identify the following things easily:

- Various entities interacting with the system are identified.
- Flow of data from one entity to another is identified.
- The various processes involved in the interaction of two or more entities in the system are pointed out.
- The various data stores, which hold the data in between the processes, are identified.

Four basic symbols are used to construct data flow diagrams. They are symbols that represent data sources, data flows, and data storage. The points at which data are transformed are represented by enclosed figures, usually circles, which are called nodes.

Data flow diagram symbols: -



Steps to Construct Data Flow Diagrams:

Four steps are commonly used to construct a DFD

❖ The process should be named and numbered for easy reference. Each name should be representative of the process.

❖ The destination of a flow is from top to bottom and from left to right.

❖ When a process is exploded into lower-level details they are numbered.

❖ The names of data stores, sources, and destinations are written in capital letters.

Rules for constructing a Data Flow Diagram:

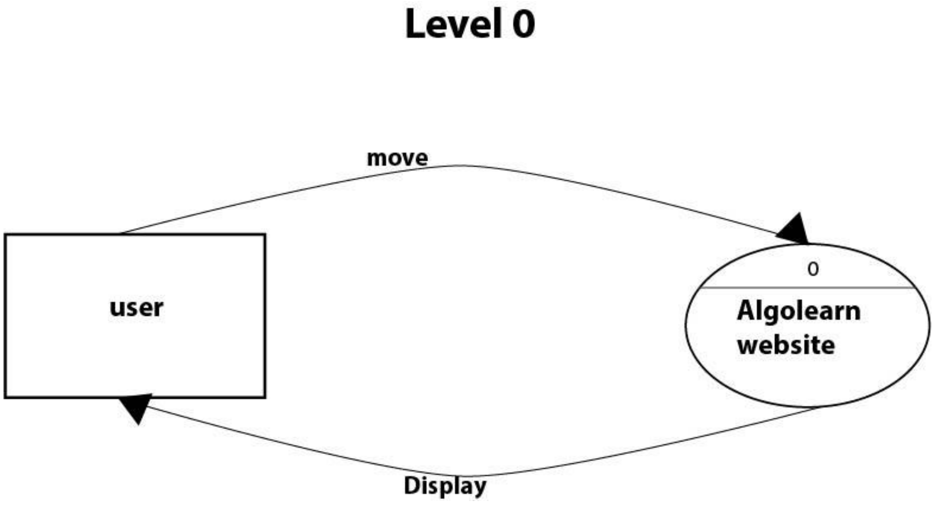
❖ Arrows should not cross each other.

❖ Squares, circles, and files must bear names.

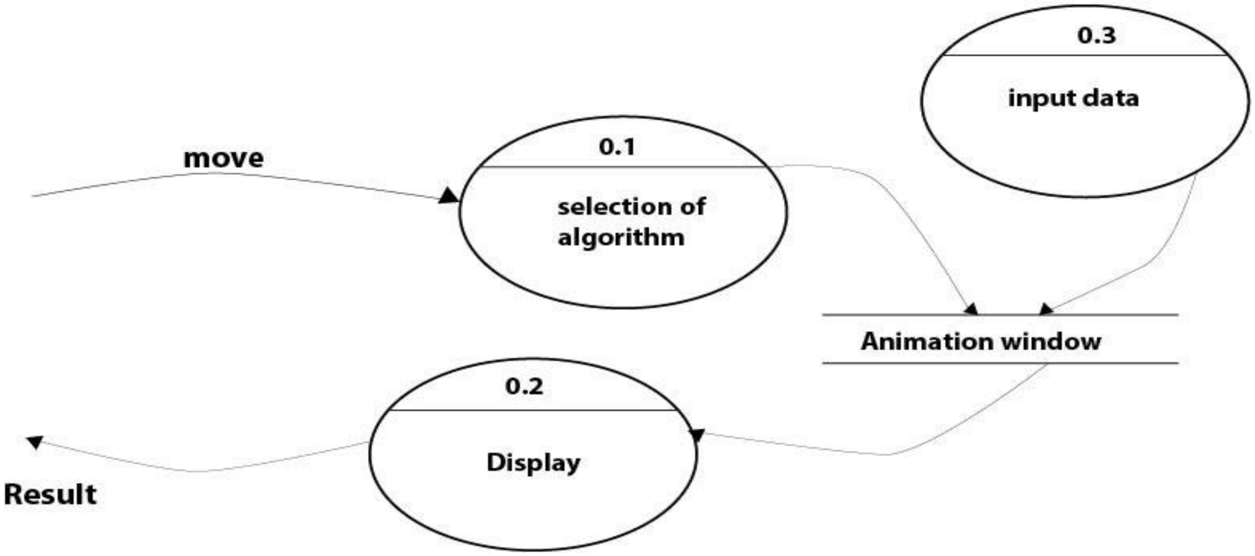
❖ Decomposed data flow squares and circles can have the same names.

❖ Draw all data flow around the outside of the diagram.

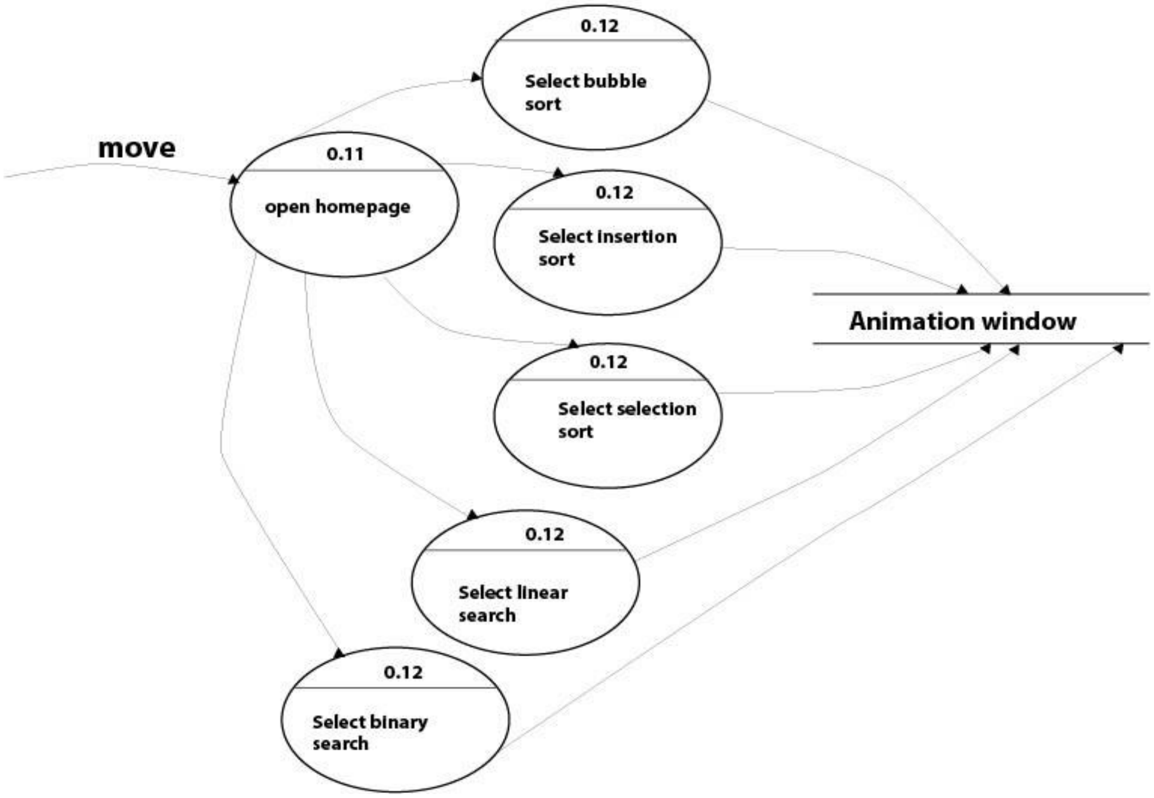
Data flow diagrams of our system



Level 1



Level 2



Chapter 5

IMPLEMENTATION

5.1 Introduction

Implementation includes all those activities that take place to convert from the old system to the new. The old system consists of manual operations, which are operated in a very different manner from the proposed new system. Proper implementation is essential to provide a reliable system to meet the requirements of the organizations. An improper installation may affect the success of the computerized system.

5.2 Tools/Scripts for Implementation

HTML

The Hyper-Text 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 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.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. The inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), the former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997. A form of HTML, known as HTML5, is used to display video and audio, primarily using the element, in collaboration with JavaScript.

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. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. 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 enables the. CSS file to be cached to improve the page load speed between the pages that share the file and its formatting.

JAVASCRIPT

JavaScript often abbreviated JS is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. Over 97% of websites use JavaScript on the client-side for web page behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to

execute the code on the user's device. JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object orientation, and first-class functions. It is multi-paradigm, supporting 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).
Key points on JavaScript are:

- ➢ JavaScript code is typically embedded in the HTML
- ➢ Used to make web pages more interactive
- ➢ It executes on our web browser window
- ➢ Interpreted and run by the client 's browser
- ➢ The file stored using the extension. js
- ➢ JavaScript code is case sensitive
- ➢ White space between words and tabs is ignored
- ➢ Line breaks are ignored except within a statement
- ➢ JavaScript statements end with a semi-colon (;).

Google chrome

A web browser is application software that allows users to find, access, display, and view websites. for the testing and working of our system, we need a web browser. Most of the people use google chrome as a web browser. Google chrome is a cross-platform web browser developed by Google. Google Chrome features a minimalistic user interface, with its user-interface principles later being implemented into other browsers.

Visual studio code

Visual studio code also commonly referred to as vs code is a source-code editor made by Microsoft for Windows, Linux, and macOS. Its features include support for debugging, syntax highlighting, intelligent code competition, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.
Vs code can be used for a variety of programming languages including Java, JavaScript, Go, Node js, Python, and C++. Vs code includes basic support for most of the common programming languages.

5.4 Coding

Home.html

```
<html>
  <head>
      <title>algolearn</title>
      <style>
         #t1 { color: rgb(10, 1, 1);

            margin-top: 40px;
            height:40px;
            width: 250px;
            padding-top: 10px;
            background: rgb(98, 189, 228);
         }
         #t2 {
            margin-left: 20%;
            margin-top: 40px;

            padding-top: 10px;

         }
         t3{
            color: black;
            margin-top: 40px;
            height:40px;
            width: 250px;
            padding-top: 10px;
         }
         h1{
             font-size: 80;
         }
         h2{
             font-size: 40;
         }h3{
             font-size: 30;
         }
         button{
             background:rgb(98, 189, 228);
             width: 200;
             font-size:20;

         }

      </style>
   </head>
<body style="  background-color:rgb(98, 189, 228);">
   <center><h1>ALGOLEARN</h1></center>
   <table id="t2">
       <tr > <td id="t3"> <h3>Understanding algorithm is a better way to understand data structures...
       </h3>
        <br>
             <h2>  It helps to learn algorithm visually...! </h2>
           </td>
        <td width="10%"></td>
       <td>
        <span>
           <div id="t1">
              <a href="bubblemain.html"><button><font color="black">Bubble Sort</font></button></a>
           </div>
            <div id="t1">
               <a href="insert.html"><button><font color="black">Insertion Sort</font></button></a>
           </div>
           <div id="t1">
               <a href="selectionmain.html"><button><font color="black">Selection Sort</font></button></a>
           </div>
           <div id="t1">
          <a href="mainlinear.html"><button><font color="black">Linear Search</font></button></a>
           </div>
           <div id="t1">
                <a href="mainbinary.html"><button><font color="black">Binary Search</font></button> </a>
           </div>

        </span>
       </td>
       </tr>
      </table>

</body>
</html>
```

Bubblemain.html

```
<html>
<head>
    <style>
        iframe{
            border: 0px;
        }
        #div1{background-color: rgb(98, 189, 228);
        color: rgb(192, 125, 79);

        }
        #myselect{
           background-color:black;
        color:azure;
        }
        #txt{ background-color:black;
        color:azure;
        width: 100%;
        height: 500px;
        overflow: scroll;

        }
        button:hover {
  background-color: rgb(122, 116, 117) !important;
  opacity: .5;
}

button:focus {
  background-color: rgb(121, 107, 109);
}

    </style>
</head>
<body>
<div id="div1">
    <h1><center><font color="black">ALGOLEARN</font></center></h1>
    <div align="right"><a href="home.html" ><button style="background-color: rgb(98, 189, 228);"><font color="black" size="3">Back to Home </font></button></a></div>
    <div> <iframe src="bubblesort (1).html" frameborder="1" height="700" width="99.7%"></iframe></div>



    <button value="Algorithm" onclick="displayal()">Algorithm</button>
    <button value="C" onclick="displayc()">C</button>
    <button  value="JAVA" onclick="displayj()">JAVA</button>
    <button  value="C++" onclick="displaycp()">C++</button><br>
        <textarea id="txt" readonly></textarea>
    <script>
        var buttons = $('button');
buttons.click(function() {
  buttons.css('background-color', 'snow');
  $(this).css('background-color', 'pink');
});
        function displayal(){
           var cd="Step 1. Look at the first number in the list.\n\nStep 2. Compare the current number with the next number. \n\nStep 3. Is the next number smaller than the current number? If so,swap
```

```
the two numbers around. If not, do not swap. \n\nStep 4. Move to the next number along in the list and make this the current number. \n\nStep 5. Repeat from step 2 until the last number in the list has been reached. \n\nStep 6. If any numbers were swapped, repeat again from step 1. \n\nStep 7. If the end of the list is reached without any swaps being made, then the list is ordered and the algorithm can stop.";
            document.getElementById("txt").value=cd;
        }
        function displayc()
        {
            var cd="//THIS IS C PROGRAMMING \n #include <stdio.h>\n//perform the bubble sort\nvoid bubbleSort(int array[], int size) \n// loop to access each array element\nfor (int step = 0; step < size - 1; ++step) {\n// loop to compare array elements\nfor (int i = 0; i < size - step - 1; ++i) {\n// compare two adjacent elements\n // change > to < to sort in descending order\nif (array[i] > array[i + 1]) {\n// swapping occurs if elements\n // are not in the intended order\nint temp = array[i];\narray[i] = array[i + 1];\n array[i + 1] = temp; }\n}\n}\n}\n// print array\nvoid printArray(int array[], int size) {\nfor (int i = 0; i < size; ++i) {\nprintf("+"%"+"d, array[i]);\n}\nprintf("+"\ "+"n"+");\n}\nint main() {\nint data[] = {-2, 45, 0, 11, -9};\n// find the array's length\nint size = sizeof(data) / sizeof(data[0]);\nbubbleSort(data, size);\nprintf("+"Sorted Array in Ascending Order:\ n"+");\nprintArray(data, size);\n}";
            document.getElementById("txt").value=cd;

        }
        function displayj(){
            var cd="//THIS IS JAVA PROGRAMMING \n import java.util.Scanner; \n class BubbleSort { \n public static void main(String []args) { \n int n, c, d, swap; \n  Scanner in = new Scanner(System.in); \n System.out.println(\"Input number of integers to sort\"); \n  n = in.nextInt(); \n int array[] = new int[n]; \n System.out.println(\"Enter \" + n + \" integers\"); \n  for (c = 0; c < n; c++) \n array[c] = in.nextInt(); \n for (c = 0; c < ( n - 1 ); c++) \n  { \n  \tfor (d = 0; d < n - c - 1; d++) \n { \n if (array[d] > array[d+1])  /* For descending order use < */ \n  {  \n swap      = array[d]; \n array[d]    = array[d+1]; \n array[d+1] = swap; \n } \n } \n } \n System.out.println(\"Sorted list of numbers\"); \n for (c = 0; c < n; c++) \n System.out.println(array[c]); \n } \n }";
            document.getElementById("txt").value=cd;
        }
        function displaycp(){
            var cd="//THIS IS Cpp PROGRAMMING \n #include <stdio.h> \n int main() \n { \n int arr[50], num, x, y, temp;  \n   printf(\"Please Enter the Number of Elements you want in the array: \"); \n scanf("+"%"+"d, &num); \n       printf(\"Please Enter the Value of Elements: \"); \n for(x = 0; x < num; x++) \n scanf("+"%"+"d, &arr[x]); \n for(x = 0; x < num - 1; x++) \n {   \n     for(y = 0; y < num - x - 1; y++) \n {    \n      if(arr[y] > arr[y + 1]) \n { \n   temp = arr[y]; \n arr[y] = arr[y + 1]; \n arr[y + 1] = temp; \n } \n  } \n  }  \n printf(\"Array after implementing bubble sort: \"); \n for(x = 0; x < num; x++) \n {  \n printf("+"%"+"d, arr[x]); \n  } \n  return 0; \n }";
            document.getElementById("txt").value=cd;
        }

    </script>
</div>

</body>
</html>
```

Insert.html

```
<html>
<head>
   <style>
      iframe{
         border: 0px;
      }
      #div1{background-color: rgb(98, 189, 228);
      color: rgb(192, 125, 79);

      }
      #myselect{
         background-color:black;
      color:azure;
      }
      #txt{ background-color:black;
      color:azure;
      width: 100%;
      height: 500px;
      overflow: scroll;

      }
      button:hover {
  background-color: rgb(122, 116, 117) !important;
  opacity: .5;
}

button:focus {
  background-color: rgb(121, 107, 109);
}

   </style>
</head>
<body>
<div id="div1">
   <h1><center><font color="black">ALGOLEARN</font></center></h1>
   <div align="right"><a href="home.html" ><button style="background-color: rgb(98, 189, 228);"><font color="black" size="3">Back to Home </font></button></a></div>
   <div> <iframe src="insertionsort(1).html" frameborder="1" height="700" width="99.7%"></iframe></div>



   <button value="Algorithm" onclick="displayal()">Algorithm</button>
   <button value="C" onclick="displayc()">C</button>
   <button  value="JAVA" onclick="displayj()">JAVA</button>
   <button  value="C++" onclick="displaycp()">C++</button><br>
      <textarea id="txt" readonly></textarea>
   <script>
```

```
        var buttons = $('button');
buttons.click(function() {
  buttons.css('background-color', 'snow');
  $(this).css('background-color', 'pink');
});
        function displayal(){
            var cd="Step 1. If the element is the first element, assume that it is already sorted. Return
1. \n Step 2. Pick the next element, and store it separately in a key. \n Step 3. Now, compare the
key with all elements in the sorted array. \n Step 4. If the element in the sorted array is smaller
than the current element, then move to the next element. Else, shift greater elements in the array
towards the right. \n Step 5. Insert the value. \n Step 6. Repeat until the array is sorted. \n ";
            document.getElementById("txt").value=cd;
        }
        function displayc()
        {
            var cd="//THIS IS C PROGRAMMING \n/* C Program to sort an array in ascending order
using Insertion Sort */ \n #include <stdio.h> \n int main() \n { \n int n, i, j, temp; \n int arr[64]; \n
printf(\"Enter number of elements"+"\ "+"n"+"\"); \n scanf("+"%"+"d, &n); \n printf(\"Enter
"+"%"+"d integers "+"\ "+"n"+" \", "+"n"+"); \n for (i = 0; i < n; i++) \n { \n scanf("+"%"+"d,
&arr[i]); \n } \n for (i = 1 ; i <= n - 1; i++) \n { \n j = i; \n while ( j > 0 && arr[j-1] > arr[j]) \n { \n
temp = arr[j]; \n arr[j]    = arr[j-1]; \n arr[j-1] = temp; \n j--; \n } \n } \n printf(\"Sorted list in
ascending order:"+"\ "+"n"+"\"); \n for (i = 0; i <= n - 1; i++) \n { \n printf("+"%"+"d"+"\
"+"n"+"\", arr[i]); \n } \n return 0; \n } \n ";
            document.getElementById("txt").value=cd;

        }
        function displayj(){
            var cd="//THIS IS JAVA PROGRAMMING \n import java.util.Scanner; \n /* Class InsertionSort */ \n public class InsertionSort \n { \n /* Insertion Sort function */ \n public static void
sort( int arr[] ) \n { \n int N = arr.length; \n int i, j, temp; \n for (i = 1; i< N; i++) \n { \n j = i; \n
temp = arr[i]; \n while (j > 0 && temp < arr[j-1]) { \n arr[j] = arr[j-1]; \n j = j-1; \n } \n arr[j] =
temp; \n } \n } \n /* Main method */ \n public static void main(String[] args) \n { \n Scanner scan
= new Scanner( System.in );    \n System.out.println(\""+"\ "+"n"+"Insertion Sort Test"+"\
"+"n"+"\"); \n int n, i; \n /* Accept number of elements */ \n System.out.println(\"Enter number of
integer elements\"); \n n = scan.nextInt(); \n /* Create integer array on n elements */ \n int arr[] =
new int[ n ]; \n /* Accept elements */ \n System.out.println(\""+"\ "+"n"+"Enter "+"n"+" integer
elements  \"); \n for (i = 0; i < n; i++) \n arr[i] = scan.nextInt(); \n  /* Call method sort */ \n sort(arr);
\n /* Print sorted Array */ \n System.out.println(\""+"\ "+"n"+"Elements after sorting \"); \n for (i
= 0; i < n; i++) \n System.out.print(arr[i]+\" \"); \n  System.out.println(); \n} \n } \n ";
            document.getElementById("txt").value=cd;
        }
        function displaycp(){
            var cd="//THIS IS Cpp PROGRAMMING \n #include<iostream> \n using namespace std;
\n void display(int *array, int size) { \n for(int i = 0; i<size; i++) \n cout << array[i] << \" \"; \n
cout << endl; \n } \n void insertionSort(int *array, int size) { \n int key, j; \n for(int i = 1; i<size;
i++) { \n key = array[i];//take value \n j = i; \n while(j > 0 && array[j-1]>key) { \n array[j] =
array[j-1]; \n j--; \n } \n array[j] = key;   //insert in right place \n } \n } \n int main() { \n int n; \n
cout << \"Enter the number of elements: \"; \n cin >> n; \n int arr[n];    //create an array with given
number of elements \n cout << \"Enter elements: \" << endl; \n for(int i = 0; i<n; i++) { \n cin >>
arr[i]; \n } \n cout <<\"Array before Sorting: \"; \n display(arr, n); \n insertionSort(arr, n); \n cout
<< \"Array after Sorting: \"; \n display(arr, n); \n } \n";
            document.getElementById("txt").value=cd;
        }

    </script>
</div>

</body>
</html>
```

selectionmain.html

```
<html>
<head>
    <style>
        iframe{
            border: 0px;
        }

        #div1{background-color: rgb(98, 189, 228);
        color: rgb(192, 125, 79);

        }
        #myselect{
            background-color:black;
        color:azure;
        }
        #txt{ background-color:black;
        color:azure;
        width: 100%;
        height: 500px;
        overflow: scroll;

        }
        button:hover {
  background-color: rgb(122, 116, 117) !important;
  opacity: .5;
}

button:focus {
  background-color: rgb(121, 107, 109);
}

    </style>
</head>
<body>
<div id="div1">
    <h1><center><font color="black">ALGOLEARN</font></center></h1>
    <div align="right"><a href="home.html" ><button style="background-color: rgb(98, 189,
228);"><font color="black" size="3">Back to Home </font></button></a></div>
    <div> <iframe src="selectionsort.html" frameborder="1" height="750"
width="99.7%"></iframe></div>



    <button value="Algorithm" onclick="displayal()">Algorithm</button>
    <button value="C" onclick="displayc()">C</button>
    <button  value="JAVA" onclick="displayj()">JAVA</button>
    <button  value="C++" onclick="displaycp()">C++</button><br>

        <textarea id="txt" readonly></textarea>
    <script>
        var buttons = $('button');
buttons.click(function() {
  buttons.css('background-color', 'snow');
  $(this).css('background-color', 'pink');
});

        function displayal(){
            var cd="Step 1. Read ARR \nStep 2. Repeat step 3 to 6 for I=0 to N-1 \nStep 3. Set
MIN=ARR[I] and Set LOC=I \nStep 4. Repeat step 5 for J=I+1 to N \nStep 5. If MIN>ARR[J],
then \n (a) Set MIN=ARR[J] \n (b) Set LOC=J \n [End of if] \n [End of step 4 loop] \nStep 6.
Interchange ARR[I] and ARR[LOC] using temporary variable \n [End of step 2 outer loop] \nStep
7. Exit";
            document.getElementById("txt").value=cd;
        }
        function displayc()
        {
            var cd="//THIS IS C PROGRAMMING \n  /** C Program to Implement Selection Sort \n
*/ \n #include <stdio.h> \n void selectionSort(int arr[], int size); \n void swap(int *a, int *b); \n /**
Selection sort function \n */ \n void selectionSort(int arr[], int size) \n { \n int i, j; \n for (i = 0 ;  i
< size;i++) \n { \n for (j = i ; j < size; j++) \n { \n if (arr[i] > arr[j]) \n swap(&arr[i], &arr[j]); \n }
\n } \n } \n /* Function to swap two variables */ \n void swap(int *a, int *b) \n { \n int temp; \n
temp = *a; \n *a = *b; \n *b = temp; \n } \n /* \n * Main Function \n */ \n int main() \n { \n int
array[10], i, size; \n printf(\"How many numbers you want to sort: \"); \n scanf("+"%"+"d,
"+"&"+"size); \n printf(\""+"\ "+"n"+"Enter %d numbers\", size); \n printf("+"\ "+"n"+"); \n for (i
= 0; i < size; i++) \n scanf("+"%"+"d, &array[i]); \n  selectionSort(array, size); \n  printf(\""+"\
"+"n"+"Sorted array is \"); \n for (i = 0; i < size;i++) \n        printf("+"%"+"d , array[i]); \n  return
0; \n     } ";
            document.getElementById("txt").value=cd;

        }
        function displayj(){
            var cd="//THIS IS JAVA PROGRAMMING \n import java.util.Scanner; \n public class
CodesCracker \n{ \n public static void main(String[] args) \n { \n int tot, i, j, count, small, index=0,
x; \n Scanner scan = new Scanner(System.in); \n System.out.print(\"Enter the Size of Array: \");
\n tot = scan.nextInt(); \n int[] arr = new int[tot];  \n System.out.print(\"Enter \" +tot+ \" Elements
for the Array: \"); \n for(i=0; i<tot; i++) \n arr[i] = scan.nextInt(); \n for(i=0; i<(tot-1); i++) \n { \n
count=0; \n small = arr[i]; \n for(j=(i+1); j<tot; j++) \n { \n if(small>arr[j]) \n { \n small = arr[j];
\n count++; \n index = j; \n } \n  } \n if(count!=0) \n { \n x = arr[i]; \n arr[i] = small; \n arr[index]
= x; \n } \n }\n System.out.println(\"\nThe new sorted array is: \"); \n for(i=0; i<tot; i++) \n
System.out.print(arr[i]+ \" \"); \n } \n }";
      document.getElementById("txt").value=cd;
    }
    function displaycp(){
      var cd="//THIS IS Cpp PROGRAMMING \n #include<iostream> \n using namespace std;
\n int main() \n { \n int i,j,n,loc,temp,min,a[30]; \n cout<<\"Enter the number of elements:\"; \n
cin>>n; cout<<\""+"\ "+"n"+"Enter the elements"+"\ "+"n"+"\"; \n for(i=0;i<n;i++) \n {  cin>>a[i];
\n   } \n for(i=0;i<n-1;i++) \n {  \n min=a[i]; \n loc=i; \n for(j=i+1;j<n;j++) \n { \n if(min>a[j]) \n
{ \n min=a[j]; \n loc=j; \n } \n } \n temp=a[i]; \n a[i]=a[loc]; \n a[loc]=temp; \n } \n cout<<\""+"\
"+"n"+"Sorted list is as follows"+"\ "+"n"+"\"; \n for(i=0;i<n;i++) \n { \n cout<<a[i]<<\" \"; \n }
\n return 0; \n }";
      document.getElementById("txt").value=cd;
    }

  </script>
</div>

</body>
</html>
```

Mainlinear.html

```
<html>
<head>
  <style>
    iframe{
      border: 0px;
    }
    #div1{background-color: rgb(98, 189, 228);
    color: rgb(192, 125, 79);

    }
    #myselect{
      background-color:black;
    color:azure;
    }
    #txt{ background-color:black;
    color:azure;
    width: 100%;
    height: 500px;
    overflow: scroll;

    }
    button:hover {
 background-color: rgb(122, 116, 117) !important;
 opacity: .5;
```

```
}

button:focus {
  background-color: rgb(121, 107, 109);
}

    </style>
</head>
<body>
<div id="div1">
    <h1><center><font color="black">ALGOLEARN</font></center></h1>
    <div align="right"><a href="home.html" ><button style="background-color: rgb(98, 189,
228);"><font color="black" size="3">Back to Home </font></button></a></div>
    <div> <iframe src="linearsearch.html"   height="700" width="99.7%"></iframe></div>



    <button value="Algorithm" onclick="displayal()">Algorithm</button>
    <button value="C" onclick="displayc()">C</button>
    <button  value="JAVA" onclick="displayj()">JAVA</button>
    <button  value="C++" onclick="displaycp()">C++</button><br>
        <textarea id="txt" readonly></textarea>
    <script>
        var buttons = $('button');
buttons.click(function() {
  buttons.css('background-color', 'snow');
  $(this).css('background-color', 'pink');
});
        function displayal(){
            var cd="Linear_Search(a, n, val) // 'a' is the given array, 'n' is the size of given array, 'val'
is the value to search  \n Step 1: set pos = -1  \n Step 2: set i = 1  \n Step 3: repeat step 4 while i <=
n  \n Step 4: if a[i] == val \n  set pos = i \n  print pos \n  go to step 6  \n [end of if]  \n set ii = i + 1
\n[end of loop]  \nStep 5: if pos = -1 \n   print \"value is not present in the array \" \n [end of if]
\nStep 6: exit  ";
            document.getElementById("txt").value=cd;
        }
        function displayc()
        {
            var cd="//THIS IS C PROGRAMMING \n * C program to input N numbers and store them
in an array.\n * Do a linear search for a given key and report success\n * or failure.\n */\n#include
<stdio.h>\nvoid main()\n{   int num;\n   int i,  keynum, found = 0;\n printf(\"Enter the number of
elements \");\n scanf(\""+"%"+"d\", &num); \n int array[num];\n printf(\"Enter the elements one
by one "+"\ "+"n"+"\");\n for (i = 0; i < num; i++)\n   {\n scanf(\""+"%"+"d\" &array[i]);\n }\n
printf(\"Enter the element to be searched \");\n scanf(\""+"%"+"d\", &keynum);\n /*  Linear search
begins */\n for (i = 0; i < num ; i++) \n {\n if (keynum == array[i] )\n  {\n  found = 1;\n break; \n
}\n }\n  if (found == 1) \n     printf(\"Element is present in the array at position "+"%"+"d\",i+1);
\n else\n printf(\"Element is not present in the array"+"\ "+"n"+"\");\n }";
         document.getElementById("txt").value=cd;
        }
        function displayj(){
            var cd="//THIS IS JAVA PROGRAMMING \n import java.util.Scanner; \n class
LinearSearchExample2 \n { \n public static void main(String args[]) \n { \n int c, n, search,
array[]; \n Scanner in = new Scanner(System.in); \n System.out.println(\"Enter number of elements\");
\n n = in.nextInt(); \n array = new int[n]; \n System.out.println(\"Enter those \" + n + \"
elements\"); \n for (c = 0; c < n; c++) \n array[c] = in.nextInt(); \n System.out.println(\"Enter
value to find\"); \n search = in.nextInt(); \n for (c = 0; c < n; c++) \n { \n if (array[c] == search)
/* Searching element is present */ \n { \n System.out.println(search + \" is present at location \"
+ (c + 1) + \".\"); \n break; \n } \n } \n if (c == n) /* Element to search isn't present */ \n
System.out.println(search + \" isn't present in array.\"); \n } \n }";
            document.getElementById("txt").value=cd;
        }
        function displaycp(){
            var cd="//THIS IS Cpp PROGRAMMING \n #include < iostream >\n using namespace
std;\n void linearSearch(int a[], int n) {\n int temp = -1; \n for (int i = 0; i < 5; i++) {\n if (a[i] ==
n) { \n cout << \"Element found at position: \" << i + 1 << endl;\n temp = 0;\n break;\n }\n }\n
if (temp == -1) { \n cout << \"No Element Found\" << endl; \n}\n }\n int main() {\n int arr[5];\n
cout << \"Please enter 5 elements of the Array\" << endl;\n for (int i = 0; i < 5; i++) { \n cin >>
arr[i];\n}\n cout <<\"Please enter an element to search\" << endl;\n int num; \n cin >> num; \n
linearSearch(arr, num);\n return 0;\n}";
            document.getElementById("txt").value=cd;
        }

    </script>
</div>

</body>
</html>
```

Mainbinary.html

```
<html>
<head>
    <style>
        iframe{
            border: 0px;
        }
        #div1{background-color: rgb(98, 189, 228);
        color: rgb(192, 125, 79);

        }
        #myselect{
            background-color:black;
        color:azure;
        }
        #txt{ background-color:black;
        color:azure;
        width: 100%;
        height: 500px;
        overflow: scroll;

        }
        button:hover {
  background-color: rgb(122, 116, 117) !important;
  opacity: .5;
}

button:focus {
  background-color: rgb(121, 107, 109);
}

    </style>
</head>
<body>
<div id="div1">
    <h1><center><font color="black">ALGOLEARN</font></center></h1>
    <div  align="right"><a  href="home.html"  ><button  style="background-color:  rgb(98,  189,
228);"><font color="black" size="3">Back to Home </font></button></a></div>
    <div>        <iframe        src="binarysearch.html"        frameborder="1"        height="700"
width="99.7%"></iframe></div>



    <button value="Algorithm" onclick="displayal()">Algorithm</button>
    <button value="C" onclick="displayc()">C</button>
    <button  value="JAVA" onclick="displayj()">JAVA</button>
    <button  value="C++" onclick="displaycp()">C++</button><br>
        <textarea id="txt" readonly></textarea>
    <script>
        var buttons = $('button');
buttons.click(function() {
  buttons.css('background-color', 'snow');
  $(this).css('background-color', 'pink');
});
        function displayal(){
          var cd="Mainfunction_algorithm \n 1  Start\n 2          Read array size in k\n 3          Set
n=k-1\n 4        Declare an array arr[k] \n 5    Read k elements to arr[]  and  sort  arr[]\n 6   Read
number x to search\n 7           Set result = binary(arr,n-1,0,x)\n 8     If result=-1 then print element
not found\n 9   Else then print element present at position \"result\"  \n 10   Stop \n  binary()_algo-
rithm \n 1        Start \n 2         If max>=min \n  Then set mid=(max+min)/2 \n  If (arr[mid]==r) \n
Return  mid  \n   Else  if  (arr[mid]>r)\n  Return  binary(arr,mid-1,min,r)   \n   Else   \n    Return  bi-
nary(arr,max,mid+1,r)  \n 3      Return -1\n 4 Stop.";
          document.getElementById("txt").value=cd;
       }
       function displayc()
       {
          var cd="//THIS IS C PROGRAMMING \n #include <stdio.h>\n int main()\n {\n  int n,x,i,k;
\n printf(\"Enter maximum number of elements :\");\n scanf(\""+"%"+"d\","+"&"+"k);\n  n=k-1;\n
int arr[k];\n printf(\"Enter numbers in ascending order:\"); \n for(i=0;i<k;i++){\n
scanf(\""+"%"+"d\",&arr[i]);\n }\n printf(\"Enter a number to search :\");\n
scanf(\""+"%"+"d\",&x);\n int result=binary(arr,n-1,0,x);\n if(result==-1)\n {\n printf(\"Element
not present \"); \n }\n else{ printf(\"Element is present at index "+"%"+"d\",result); }\n return
0;\n }\n int binary(int arr[],int max,int min,int r)\n {\n if(max>=min)\n {\n int
mid=(max+min)/2;\n if(arr[mid]==r) \n { return mid; }\n else if(arr[mid]>r) \n { return binary(arr,mid-1,min,r); } \n else { return binary(arr,max,mid+1,r); }\n }\n return -1;\n }";
        document.getElementById("txt").value=cd;

      }
      function displayj(){
    var cd="//THIS IS JAVA PROGRAMMING \n import java.util.*;\n public class BinarySearchExample{ \n public static void binarySearch(int arr[], int first, int last, int key){ \n int
mid = (first + last)/2; \n while( first <= last ){ \n if ( arr[mid] < key ){ \n first = mid + 1; \n
}else if ( arr[mid] == key ){ \n System.out.println(\"Element is found at index: \" + mid); \n
break; \n }else{ \n last = mid - 1; \n } \n mid = (first + last)/2; \n } \n if ( first > last ){ \n
System.out.println(\"Element is not found!\"); \n } \n } \n public static void main(String args[]){
\n int a,b,i;\n Scanner val=new Scanner(System.in);\n System.out.println(\"Enter the size of
array: \");\n a=val.nextInt();\n int arr[]=new int[a] ; \n System.out.println(\"Enter the elements for
the first array\"); \n for(i=0;i<a;i++) \n { arr[i]=val.nextInt(); } \n System.out.println(\"Enter number to search : \"); \n b=val.nextInt();\n int key = b; \n int last=arr.length-1; \n binarySearch(arr,0,last,key); \n } \n }";
          document.getElementById("txt").value=cd;
       }
       function displaycp(){
          var cd="//THIS IS Cpp PROGRAMMING \n #include <iostream> \n #include <conio.h>
\n using namespace std; \n int main () \n { \n // declaration of the variables and array \n int
arr[100], st, mid, end, i, num, tgt; \n cout << \" Define the size of the array: \" << endl; \n cin >>
num; // get size \n // enter only sorted array \n cout << \" Enter the values in sorted array either
ascending or descending order: \" << endl; \n // use for loop to iterate values \n for (i = 0; i < num;
i++) \n { \n cout << \" arr [\" << i << \"] = \"; \n cin >> arr[i]; \n } \n // initialize the
starting and ending variable's values \n st = 0; \n end = num - 1; // size of array (num) - 1 \n //
define the item or value to be search \n cout <<\" Define a value to be searched from sorted array:
\" << endl; \n cin >> tgt; \n // use while loop to check 'st', should be less than equal to 'end'. \n
while ( st <= end) \n { n\ // get middle value by splitting into half \n mid = ( st + end ) / 2; \n
/* if we get the target value at mid index, print the position and exit from the program. */ \n if
(arr[mid] == tgt) \n { \n cout << \" Element is found at index \" << (mid + 1); \n exit (0); //
use for exit program the program \n } \n // check the value of target element is greater than the
mid element' value \n else if ( tgt > arr[mid]) \n { st = mid + 1; // set the new value for st
variable \n } \n // check the value of target element is less than the mid element' value \n else
if ( tgt < arr[mid]) \n { \n end = mid - 1; // set the new value for end variable \n } \n } \n
cout << \" Number is not found.\" << endl; \n return 0; \n}";
          document.getElementById("txt").value=cd;
       }
   </script>
</div>
</body>
</html>
```

Bubblesort(1).html

```
<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="UTF-8" />
   <meta name="viewport" content=
      "width=device-width, initial-scale=1.0" />
<style>
* {
   margin: 0px;
   padding: 0px;
   box-sizing: border-box;
}
#alerttxt{ background-color:black;
      color:azure;
      width: 500px;
      height: 50px;
      overflow: scroll;

      }
.header {
   font-size: 20px;
   text-align: center;
}

#array {
   background-color: white;
   height: 413px;
   width: 598px;
   margin: auto;
   position: relative;
   margin-top: 64px;
}

.block {
   width: 28px;
   background-color: #6b5b95;
   position: absolute;
   bottom: 0px;
   transition: 0.2s all ease;
}

.block_id {
   position: absolute;
   color: black;
   margin-top: -20px;
   width: 100%;
   text-align: center;
```

```
}
</style>
</head>

<body>
   <br />
   <center>
   <p class="header"><h2><b>BUBBLE SORT</b></h2></p>
   <br>
   <div >
         <form name=form1>
            Enter the size of Array
         <input type="text" name="numsize" size=10>
         &nbsp;&nbsp;&nbsp;&nbsp;
         <input type="button" value="Generate Array" onclick="generatearray();">
         &nbsp;&nbsp;
         <input type="button" value="Sort" onclick="BubbleSort();">
       <input type="button" value="reset" onClick="window.location.reload();">
         </form>
   </div>
   <br><br>
   <div><form name="tt">

      <input type="text" id="alerttxt" readonly>
   </form>
   <table><tr><td style="color: #131766;"><b>Unsorted element will be shown in purple</b></td><td style="background-color: #6b5b95;width:20px;height:20px;"></td></tr></table>
   <table><tr><td style="color: #131766;"><b>Compared elements will be shown in red</b></td><td style="background-color: #FF4949;width:20px;height:20px;"></td></tr></table>
   <table><tr><td style="color: #131766;"><b>Sorted element will be shown in green</b></td><td style="background-color: #13CE66;width:20px;height:20px;"></td></tr></table>

   </div>
</center>
   <div id="array"></div>

<script>
var container = document.getElementById("array");
// Function to generate the array of blocks
function generatearray() {

   var ssize=document.form1.numsize.value;
   var arr=[];

  for(var i=0;i<ssize;i++)
    {  arr.push(prompt('Enter the Array Elements '+(i+1) )) ;  }
   for (var i = 0; i < ssize; i++) {

      // Return a value from 1 to 100 (both inclusive)
      var value = parseInt(arr[i]);

      // Creating element div
      var array_ele = document.createElement("div");

      // Adding class 'block' to div
      array_ele.classList.add("block");

      // Adding style to div
      array_ele.style.height = `${value * 3}px`;
      array_ele.style.transform = `translate(${i * 30}px)`;

      // Creating label element for displaying
      // size of particular block
      var array_ele_label = document.createElement("label");
      array_ele_label.classList.add("block_id");
      array_ele_label.innerText = value;

      // Appending created elements to index.html
      array_ele.appendChild(array_ele_label);
      container.appendChild(array_ele);
   }
}

// Promise to swap two blocks
function swap(el1, el2) {
   return new Promise((resolve) => {

      // For exchanging styles of two blocks
      var temp = el1.style.transform;
      el1.style.transform = el2.style.transform;
      el2.style.transform = temp;

      window.requestAnimationFrame(function() {

         // For waiting for .25 sec
         setTimeout(() => {
            container.insertBefore(el2, el1);
            resolve();
         }, 5000);
      });
   });
}

// Asynchronous BubbleSort function
async function BubbleSort(delay = 5000) {
   var blocks = document.querySelectorAll(".block");

// BubbleSort Algorithm
   for (var i = 0; i < blocks.length; i += 1) {

      for (var j = 0; j < blocks.length - i - 1; j += 1) {

         var artx="\touter loop (i="+i+")      \n inner loop (j="+j+")    \n compare at index j and j+1";
         document.getElementById("alerttxt").value=artx;
         // To change background-color of the
         // blocks to be compared


         blocks[j].style.backgroundColor = "#FF4949";
         blocks[j + 1].style.backgroundColor = "#FF4949";
         // To wait for 1 sec
         await new Promise((resolve) =>
            setTimeout(() => {
                resolve();
            }, delay)
         );

         console.log("run");
         var value1 = Number(blocks[j].childNodes[0].innerHTML);
         var value2 = Number(blocks[j + 1]
                     .childNodes[0].innerHTML);

         // To compare value of two blocks
         if (value1 > value2) {
            var artr="\touter loop (i="+i+")      \n inner loop (j="+j+")    \nvalue at j > j+1, swap";
            document.getElementById("alerttxt").value=artr;
            await swap(blocks[j], blocks[j + 1]);
            blocks = document.querySelectorAll(".block");
         }
// Changing the color to the previous one

         blocks[j].style.backgroundColor = "#6b5b95";
         blocks[j + 1].style.backgroundColor = "#6b5b95";
      }

      //changing the color of greatest element
      //found in the above traversal
      blocks[blocks.length - i - 1]
            .style.backgroundColor = "#13CE66";
   }

}
```

```
// Calling generatearray function
generatearray();

// Calling BubbleSort function
BubbleSort();
</script>
</body>

</html>
```

Insertionsort(1).html

```
<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="UTF-8" />
   <meta name="viewport" content=
      "width=device-width, initial-scale=1.0" />
<style>
* {
   margin: 0px;
   padding: 0px;
   box-sizing: border-box;
}
#alerttxt{ background-color:black;
      color:azure;
      width: 500px;
      height: 50px;
      overflow: scroll;

      }
.header {
   font-size: 20px;
   text-align: center;
}

#array {
   background-color: white;
   height: 413px;
   width: 598px;
   margin: auto;
   position: relative;
   margin-top: 64px;
}

.block {
   width: 28px;
   background-color: #6b5b95;
   position: absolute;
```

```
    bottom: 0px;
    transition: 0.2s all ease;
}

.block_id {
    position: absolute;
    color: black;
    margin-top: -20px;
    width: 100%;
    text-align: center;
}
</style>
</head>

<body>
    <br />
    <center>
    <p class="header"><h2><b>INSERTION SORT</b></h2></p>
    <br>
    <div >
            <form name=form2>
                Enter the size of Array
            <input type="text" name="numsize" size=10>
            &nbsp;&nbsp;&nbsp;&nbsp;
            <input type="button" value="Generate Array" onclick="generatearray();">
            &nbsp;&nbsp;
            <input type="button" value="Sort" onclick="InsertionSort();">
         <input type="button" value="reset" onClick="window.location.reload();">
            </form>
    </div>
    <br><br><br>
    <div><form name="tt">

        <input type="text" id="alerttxt" readonly>
    </form>
    </div>
    <table><tr><td style="color: #131766;"><b>Unsorted element will be shown in purple</b></td><td style="background-color: #6b5b95;width:20px;height:20px;"></td></tr></table>
    <table><tr><td style="color: #131766;"><b>compared elements will be shown in red</b></td><td style="background-color:#FF4949;width:20px;height:20px;"></td></tr></table>
    <table><tr><td style="color: #131766;"><b>inner compared elements will be shown in pink</b></td><td style="background-color:#FF0089;width:20px;height:20px;"></td></tr></table>

</center>
    <div id="array"></div>
```

```
<script>
var container = document.getElementById("array");
// Function to generate the array of blocks
function generatearray() {

    var ssize=document.form2.numsize.value;
    var arr=[];

  for(var i=0;i<ssize;i++)
     {  arr.push(prompt('Enter the Array Element '+(i+1) )) ;  }
    for (var i = 0; i < ssize; i++) {

        // Return a value from 1 to 100 (both inclusive)
        var value = parseInt(arr[i]);

        // Creating element div
        var array_ele = document.createElement("div");

        // Adding class 'block' to div
        array_ele.classList.add("block");

        // Adding style to div
        array_ele.style.height = `${value * 3}px`;
        array_ele.style.transform = `translate(${i * 30}px)`;

        // Creating label element for displaying
        // size of particular block
        var array_ele_label = document.createElement("label");
        array_ele_label.classList.add("block_id");
        array_ele_label.innerText = value;

        // Appending created elements to index.html
        array_ele.appendChild(array_ele_label);
        container.appendChild(array_ele);
    }
}

// Promise to swap two blocks
function swap(el1, el2) {
    return new Promise((resolve) => {

        // For exchanging styles of two blocks
        var temp = el1.style.transform;
        el1.style.transform = el2.style.transform;
        el2.style.transform = temp;

        window.requestAnimationFrame(function() {

            // For waiting for 1 sec
            setTimeout(() => {
                container.insertBefore(el2, el1);
                resolve();
            }, 5000);
        });
    });
}

// Asynchronous InsertionSort function
async function InsertionSort(delay = 5000) {
    var blocks = document.querySelectorAll(".block");
    var small;
//  InsertionSort Algorithm
    for (var i = 0; i < blocks.length; i += 1) {
        var artx="\touter loop (i="+i+")      \n compare at index i and i+1";
        document.getElementById("alerttxt").value=artx;
            blocks[i].style.backgroundColor = "#FF4949";
            blocks[i+1].style.backgroundColor = "#FF4949";

            // To wait for 1 sec
            await new Promise((resolve) =>
                setTimeout(() => {
                    resolve();
                }, delay)
            );
            console.log("run");
            var value1 = Number(blocks[i].childNodes[0].innerHTML);
            var value2 = Number(blocks[i + 1]
                        .childNodes[0].innerHTML);
            if (value1 > value2) {
                var artr="\touter loop (i="+i+")       \n swap index i and i+1";
                document.getElementById("alerttxt").value=artr;
                await swap(blocks[i], blocks[i + 1]);
                blocks = document.querySelectorAll(".block");
            }
            blocks[i].style.backgroundColor = "#6b5b95";
           // blocks[i + 1].style.backgroundColor = "#6b5b95";
        for (var j = i; j > 0 ; j -= 1) {
            var artt="\touter loop (i="+i+")      inner loop (j="+j+")    \n compare at index j-1 and j";
            document.getElementById("alerttxt").value=artt;
            // To change background-color of the
            blocks[j-1].style.backgroundColor = "#FF0089";
            blocks[j].style.backgroundColor = "#FF0089";
            // blocks to be compared
            // To wait for 1 sec
            await new Promise((resolve) =>
                setTimeout(() => {
                    resolve();
                }, delay)
```

```
      );

      console.log("run");
      var value1 = Number(blocks[j-1].childNodes[0].innerHTML);
      var value2 = Number(blocks[j]
            .childNodes[0].innerHTML);

      // To compare value of two blocks
      if (value1 > value2) {
        var artp="\touter loop (i="+i+")      inner loop (j="+j+")     \n swap index j-1 and j";
       document.getElementById("alerttxt").value=artp;
        await swap(blocks[j-1], blocks[j]);
        blocks = document.querySelectorAll(".block");
      }
// Changing the color to the previous one
      blocks[j].style.backgroundColor = "#6b5b95";
      blocks[j -1].style.backgroundColor = "#6b5b95";
    }

    //changing the color of greatest element
    //found in the above traversal
   // blocks[blocks.length - i - 1].style.backgroundColor = "#13CE66";
  }

  blocks[i-1].style.backgroundColor = "#6b5b95";
}


// Calling generatearray function
generatearray();

// Calling BubbleSort function
InsertionSort();
</script>
</body>

</html>
```

Selectionsort.html

```
<!DOCTYPE html>
<html lang="en">

 <!-- head -->
 <head>
  <meta charset="UTF-8" />
  <meta name="viewport"
     content="width=device-width, initial-scale=1.0" />
  <meta http-equiv="X-UA-Compatible" content="ie=edge" />

  <title>Sorting Visualizer</title>
 <style>
.mySlides {
  display: none;
}
.head {
  margin-top: 20px;
  margin-right: 20vw;
  margin-left: 20vw;
  text-align: center;
  font-size: 30px;
  background-color: #6f459e;
  color: white;
  border-radius: 19px;
  font-weight: bolder;
}
.data-container {
  width: 600px;
  height: 384px;
  position: relative;
  margin: 0 auto;
}

.bar {
  width: 28px;
  position: absolute;
  left: 0;
  bottom: 0;
  background-color: rgb(0, 183, 255);
  transition: 0.2s all ease;
}

.bar__id {
  position: absolute;
  margin-top: -20px;
  width: 100%;
  text-align: center;
}
.btn1 {
  padding: 12px;
  font-weight: bolder;
  background-color: #6f459e;
  border-radius: 10px;
  color: white;
  font-size: 16px;
  border: white;
  margin-left: 37vw;
  margin-top: 4vw;
  margin-right: 1vw;
}
```

```
.btn2 {
   padding: 12px;
   font-weight: bolder;
   background-color: #6f459e;
   border-radius: 10px;
   color: white;
   font-size: 16px;
   border: white;
}
#alerttxt{ background-color:black;
      color:azure;
      width: 500px;
      height: 50px;
      overflow: scroll;

      }
#array {
   background-color: white;
   height: 450px;
   width: 598px;
   margin: auto;
   position: relative;
   margin-top: 64px;
}
</style>
  </head>
<body >
  <br />
  <center>
  <p class="header"><H2><b>SELECTION SORT</b></H2></p>

  <div> <font color="black">Note 1: Here i,j,small represent index value<br>
  Note 2: Here index value is start from 0</font></div>
  <div >
        <form name=form1>
           Enter the size of Array
        <input type="text" name="numsize" size=10>
        &nbsp;&nbsp;&nbsp;&nbsp;
        <input type="button" value="Generate Array" onclick="generatebars();">
        &nbsp;&nbsp;
        <input type="button" value="Sort" onclick="SelectionSort();">
      <input type="button" value="reset" onClick="window.location.reload();">
        </form>
  </div>
  <br>
    <div><form name="tt">

      <input type="text" id="alerttxt" readonly>
    </form>
    <table><tr><td style="color: #131766;"><b>Unsorted element will be shown in
blue</b></td><td style="background-color: rgb(0, 183,
255);width:20px;height:20px;"></td></tr></table>
    <table><tr><td style="color: #131766;"><b>Compared elements will be shown in
red</b></td><td style="background-color: red;width:20px;height:20px;"></td></tr></table>
    <table><tr><td style="color: #131766;"><b>Sorted element will be shown in
green</b></td><td style="background-color: rgb(49, 226,
13);width:20px;height:20px;"></td></tr></table>

    </div>
  </center>
  <div id="array"></div>
  </body>

  <script>
var container = document.getElementById("array");

// function to generate bars
function generatebars() {
  var ssize=document.form1.numsize.value;
    var arr=[];
  //for loop to generate 20 bars
  for (let i = 0; i < ssize; i += 1)
  {  arr.push(prompt('Enter the Array Elements '+(i+1) )) ;  }

  for (let i = 0; i < ssize; i += 1) {
      var value = parseInt(arr[i]);
    // To create element "div"
    const bar = document.createElement("div");

    // To add class "bar" to "div"
    bar.classList.add("bar");

    // Provide height to the bar
    bar.style.height = `${value * 3}px`;

    // Translate the bar towards positive X axis
    bar.style.transform = `translateX(${i * 30}px)`;
 // To create element "label"
    const barLabel = document.createElement("label");

    // To add class "bar_id" to "label"
    barLabel.classList.add("bar_id");

    // Assign value to "label"
    barLabel.innerText = value;

    // Append "Label" to "div"
    bar.appendChild(barLabel);
    // Append "div" to "data-container div"
    container.appendChild(bar);
  }
}
  // asynchronous function to perform "Selection Sort"
async function SelectionSort(delay = 5000) {
  let bars = document.querySelectorAll(".bar");
  // Assign 0 to min_idx
   var min_idx = 0;
   for (var i = 0; i < bars.length; i += 1) {
    var artx=" \touter loop (i="+i+")    small= "+i;
            document.getElementById("alerttxt").value=artx;
    // Assign i to min_idx
    min_idx = i;

    // Provide darkblue color to the ith bar
    bars[i].style.backgroundColor = "#118280";
    //pause
    await new Promise((resolve) =>
        setTimeout(() => {
          resolve();
        }, 5000)
      );
    for (var j = i + 1; j < bars.length; j += 1) {
      var artb="\touter loop (i="+i+")    inner loop (j="+j+")  small= "+min_idx;
            document.getElementById("alerttxt").value=artb;
      // Provide red color to the jth bar
      bars[j].style.backgroundColor = "red";

      //  pause
      await new Promise((resolve) =>
        setTimeout(() => {
          resolve();
        }, 5000)
      );
   // To store the integer value of jth bar to var1
      var val1 = parseInt(bars[j].childNodes[0].innerHTML);

      // To store the integer value of (min_idx)th bar to var2
      var val2 = parseInt(bars[min_idx].childNodes[0].innerHTML);

      // Compare val1 & val2
      var artc="\touter loop (i="+i+")    inner loop (j="+j+")     check (value at j< value at small):is
"+val1+"<"+val2;
        document.getElementById("alerttxt").value=artc;
        await new Promise((resolve) =>
        setTimeout(() => {
          resolve();
        }, 5000)
      );
      if (val1 < val2) {
        var artc="\touter loop (i="+i+")   inner loop (j="+j+")   small= "+j;
            document.getElementById("alerttxt").value=artc;
        if (min_idx !== i) {
          // Provide skyblue color to the (min-idx)th bar
          bars[min_idx].style.backgroundColor = "  rgb(24, 190, 255)";
         }
        min_idx = j;
      } else {
        var artc="\touter loop (i="+i+")   inner loop (j="+j+")   small= "+min_idx;
            document.getElementById("alerttxt").value=artc;
        // Provide skyblue color to the jth bar
        bars[j].style.backgroundColor = "  rgb(24, 190, 255)";
      }
    }
    await new Promise((resolve) =>
        setTimeout(() => {
          resolve();
        }, 5000)
      );
      var arty="\touter loop (i="+i+")   inner loop (j="+j+")   swap "+min_idx+" and "+i;
      document.getElementById("alerttxt").value=arty;
   // To swap ith and (min_idx)th bar
    var temp1 = bars[min_idx].style.height;
    var temp2 = bars[min_idx].childNodes[0].innerText;
    bars[min_idx].style.height = bars[i].style.height;
    bars[i].style.height = temp1;
    bars[min_idx].childNodes[0].innerText = bars[i].childNodes[0].innerText;
    bars[i].childNodes[0].innerText = temp2;

    // Provide skyblue color to the (min-idx)th bar
    bars[min_idx].style.backgroundColor = "  rgb(24, 190, 255)";

    // Provide lightgreen color to the ith bar
    bars[i].style.backgroundColor = " rgb(49, 226, 13)";
    await new Promise((resolve) =>
        setTimeout(() => {
          resolve();
        }, 5000)
      );
  }
 }

// Call "generatebars" function
generatebars();

</script>
```

```
</html>
```

linearsearch.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content=
    "width=device-width, initial-scale=1.0" />
<style>
* {
  margin: 0px;
  padding: 0px;
  box-sizing: border-box;
}
#alerttxt{ background-color:black;
    color:azure;
    width: 500px;
    height: 50px;
    overflow: scroll;

    }
.header {
  font-size: 20px;
  text-align: center;
}

#array {
  background-color: white;
  height: 400px;
  width: 598px;
  margin: auto;
  position: relative;
  margin-top: 64px;
}

.block {
  width: 28px;
  background-color: #6b5b95;
  position: absolute;
  bottom: 0px;
  transition: 0.2s all ease;
}

.block_id {
  position: absolute;
  color: black;
  margin-top: -20px;
```

```
   width: 100%;
   text-align: center;
}
</style>
</head>

<body>
   <br />
   <center>
   <p class="header"><h2><b>LINEAR SEARCH</b></h2></p>
   <br>
   <div >
         <form name=form1>
            Enter the size of Array
         <input type="text" name="numsize" size=10>
         &nbsp;&nbsp;&nbsp;&nbsp;
         <input type="button" value="Generate Array" onclick="generatearray();">
         &nbsp;&nbsp;&nbsp;&nbsp;
         Enter number to search
         <input type="text" name="srchnum" size=10>
         &nbsp;&nbsp;&nbsp;&nbsp;
         <input type="button" value="search" onclick="Linearsearch();">&nbsp;&nbsp;
       <input type="button" value="reset" onClick="window.location.reload();">
         </form>
   </div>
   <br><br><br>
   <div><form name="tt">

      <input type="text" id="alerttxt" readonly>
   </form>
   </div>
   <table><tr><td style="color: #131766;"><b>Founded element will be shown in dark blue</b></td><td style="background-color: #131766;width:20px;height:20px;"></td></tr></table>
   </center>
   <div id="array"></div>

<script>
var container = document.getElementById("array");
// Function to generate the array of blocks
function generatearray() {

   var ssize=document.form1.numsize.value;
   var arr=[];

 for(var i=0;i<ssize;i++)
    {  arr.push(prompt('Enter the Array Elements '+(i+1) )) ;  }
   for (var i = 0; i < ssize; i++) {
      // Return a value from 1 to 100 (both inclusive)
      var value = parseInt(arr[i]);

      // Creating element div
      var array_ele = document.createElement("div");

      // Adding class 'block' to div
      array_ele.classList.add("block");

      // Adding style to div
      array_ele.style.height = `${value * 3}px`;
      array_ele.style.transform = `translate(${i * 30}px)`;

      // Creating label element for displaying
      // size of particular block
      var array_ele_label = document.createElement("label");
      array_ele_label.classList.add("block_id");
      array_ele_label.innerText = value;

      // Appending created elements to index.html
      array_ele.appendChild(array_ele_label);
      container.appendChild(array_ele);
   }
}


// Asynchronous Linearsearch function
async function Linearsearch(delay = 5000) {
   var blocks = document.querySelectorAll(".block");
   var searchnum= document.form1.srchnum.value;
   var flag=2;
// Linearsearch Algorithm
   for (var i = 0; i < blocks.length; i += 1) {
          var lopp=1;
          var val = Number(blocks[i].childNodes[0].innerHTML);
          var artx=" \t \t loop (i="+i+")      \n check : is "+val+" is equal to "+searchnum+" ";
         document.getElementById("alerttxt").value=artx;
         // To change background-color of the
         // blocks to be compared
         blocks[i].style.backgroundColor = "#FF4949";
         // To wait for 2 sec
         await new Promise((resolve) =>
            setTimeout(() => {
                resolve();
            }, delay)
         );

         console.log("run");
         //var value1 = Number(blocks[i].childNodes[0].innerHTML);
         // To compare value of two block with user value
         if (val==searchnum) {
            blocks[i].style.backgroundColor = "#131766";
            flag=1;
            var chk=" \t \t loop (i="+i+")      \n found "+searchnum+" at position "+i;
            document.getElementById("alerttxt").value=chk;

            blocks = document.querySelectorAll(".block");
            await new Promise((resolve) =>
            setTimeout(() => {
                resolve();
            }, delay)
         );
         break;
         }
         else{flag=0;}

      blocks[i].style.backgroundColor = "#6b5b95";
   }

     if(flag==0)
     {
      var chk=" \t \t loop (i="+i+")      \n not found" ;
      document.getElementById("alerttxt").value=chk;

      blocks = document.querySelectorAll(".block");
    }



}


// Calling generatearray function
generatearray();

// Calling BubbleSort function
Linearsearch();
</script>
</body>

</html>
```

binarysearch.html

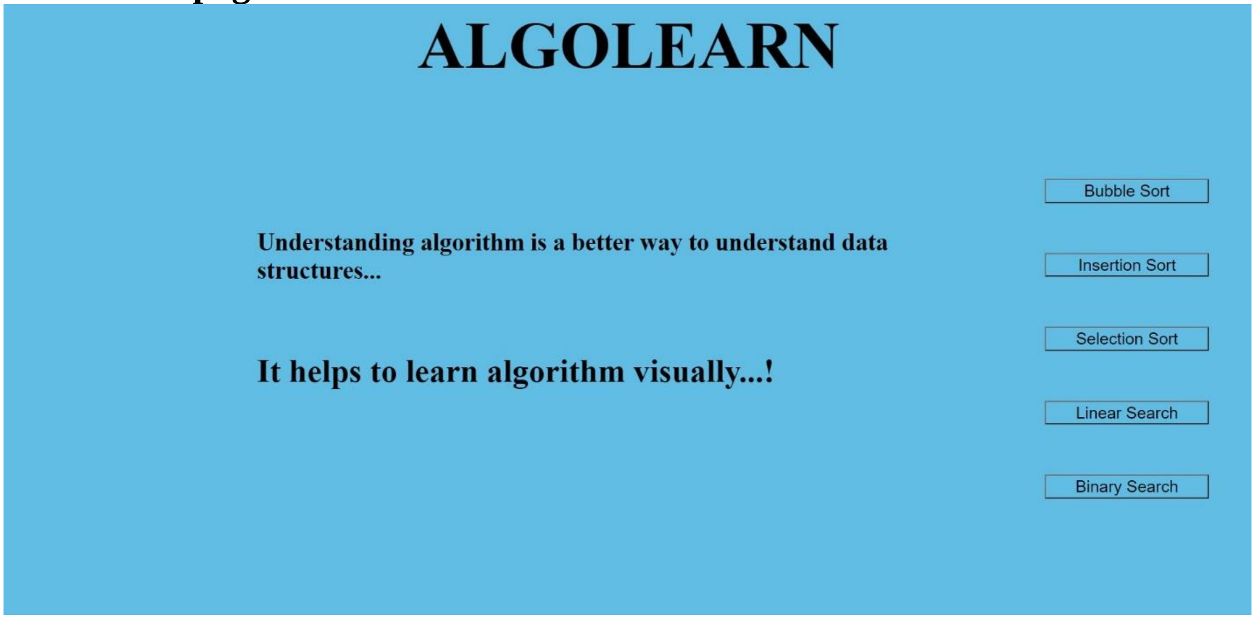
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport"
          content="width=device-width, initial-scale=1.0" />
    <style>
        * {
  margin: 0px;
  padding: 0px;
  box-sizing: border-box;
}

.header {
  font-size: 35px;
  text-align: center;
}
#alerttxt1{ background-color:black;
        color:azure;
        width: 500px;
        height: 50px;
        overflow: scroll;

        }
#alerttxt{ background-color:black;
        color:azure;
        width: 500px;
        height: 50px;
        overflow: scroll;

        }

#array {
  background-color: white;
  height: 305px;
  width: 598px;
  margin: auto;
  position: relative;
  margin-top: 64px;
}
.block {
  width: 28px;
  background-color: #6b5b95;
  position: absolute;
  bottom: 0px;
  transition: 0.2s all ease;
}

.block_id {
  position: absolute;
  color: black;
  margin-top: -20px;
  width: 100%;
  text-align: center;
}
    </style>
  </head>

  <body>
    <br />
    <p class="header">Binary Search</p>


    <br /><br />
    <div style="text-align: center">
        Enter the size of Array<input type="text" id="numsize" size=10>
        &nbsp;&nbsp;&nbsp;&nbsp;
        <input type="button" value="Generate Array" onclick="generatearray();">
        &nbsp;&nbsp;&nbsp;&nbsp;
        <label for="fname">
          Enter number to Search
        </label>
        <input type="text" id="fname" name="fname" />
               &nbsp;&nbsp;&nbsp;&nbsp;
        <button id="btn" onclick="BinarySearch()">Search</button>
        &nbsp;&nbsp;
     <input type="button" value="reset" onClick="window.location.reload();">
        <br />
        <br />
        <br><br><br>
          <div>
            <div><b>The numbers you have entered will be sorted before searching</b></div><br>
            <form name="tt">
        <input type="text" id="alerttxt1" readonly><br>
        <input type="text" id="alerttxt" readonly><br>
         </form> <center><table><tr><td style="color: #131766;"><b>Founded element will be
shown in dark blue</b></td><td style="background-color:
#131766;width:20px;height:20px;"></td></tr></table></center>
          </div>

        <div id="text"></div>
      </div>
      <div id="array"></div>
      <script>
      var container = document.getElementById("array");
 // Function to generate the array of blocks
 function generatearray() {
   var ssize = document.getElementById("numsize").value;
   // Creating an array
   var arr = [];

   // Filling array with random values
   for (var i = 0; i < ssize; i++) {
     // Return a value from 1 to 100 (both inclusive)
     arr.push(prompt('Enter the Array Elements '+(i+1) )) ;
    // var val = Number(Math.ceil(Math.random() * 100));
    // arr.push(val);
   }

   // Sorting Array in ascending order
   arr.sort(function (a, b) {
     return a - b;
   });
   for (var i = 0; i < ssize; i++) {
    var value = arr[i];

    // Creating element div
    var array_ele = document.createElement("div");

    // Adding class 'block' to div
    array_ele.classList.add("block");

    // Adding style to div
    array_ele.style.height = `${value * 3}px`;
    array_ele.style.transform = `translate(${i * 30}px)`;

    // Creating label element for displaying
    // size of particular block
    var array_ele_label = document.createElement("label");
    array_ele_label.classList.add("block_id");
    array_ele_label.innerText = value;

    // Appending created elements to index.html
    array_ele.appendChild(array_ele_label);
    container.appendChild(array_ele);
  }
}
// Asynchronous BinarySearch function
async function BinarySearch(delay = 5000) {
    var ssiz = document.getElementById("numsize").value;
  var blocks = document.querySelectorAll(".block");
  var output = document.getElementById("text");

  //Extracting the value of the element to be searched
  var num = document.getElementById("fname").value;

  //Colouring all the blocks violet
  for (var i = 0; i < blocks.length; i += 1) {
    blocks[i].style.backgroundColor = "#6b5b95";
  }
```

```
document.getElementById("alerttxt1").value=artx;
 await new Promise((resolve) =>
     setTimeout(() => {
       resolve();
     }, 5000)
   );
 // BinarySearch Algorithm
 var start = 0;
 var end = ssiz-1;
 var flag = 0;
 while (start <= end) {

   //Middle index
   var mid = Math.floor((start + end) / 2);
   var artw=" start= "+start+"   end= "+end+"   mid= "+mid+"  : while (start <= end)";
   document.getElementById("alerttxt1").value=artw;
   blocks[start].style.backgroundColor = "#0000ff";
   blocks[end].style.backgroundColor = "#00ff00";
   blocks[mid].style.backgroundColor = "#FF4949";

   //Value at mid index
   var value = Number(blocks[mid].childNodes[0].innerHTML);
   await new Promise((resolve) =>
     setTimeout(() => {
       resolve();
     }, 5000)
   );
   var artx=" check value at middle position , is  it equals  or  less than  or  greater than  "+num;
   document.getElementById("alerttxt").value=artx;
    //Current element is equal to the element
   //entered by the user
   await new Promise((resolve) =>
     setTimeout(() => {
       resolve();
     }, 5000)
   );
   if (value == num) {
       var artx=" value at middle position is equals "+num;
   document.getElementById("alerttxt").value=artx;
     output.innerText = "Element Found at index "+mid;
    blocks[mid].style.backgroundColor = "#131766";
     flag = 1;
     break;
   }
   //Current element is greater than the element
   //entered by the user
   if (value > num) {
   var artx=" value at middle position is greater than "+num+" set end = mid-1";
   document.getElementById("alerttxt").value=artx;
```

```
    blocks[end].style.backgroundColor = "#6b5b95";
      end = mid - 1;
     blocks[mid].style.backgroundColor = "#6b5b95";
      flag=2;
    } else {
        var artx="value at middle position is less than "+num +" set start = mid+1";
       document.getElementById("alerttxt").value=artx;
       blocks[start].style.backgroundColor = "#6b5b95";
      start = mid + 1;
     blocks[mid].style.backgroundColor = "#6b5b95";
      flag=2;
    }
    await new Promise((resolve) =>
      setTimeout(() => {
        resolve();
      }, 5000)
    );
  }
  if (flag == 2) {
    output.innerText = "Element Not Found";
     }
     await new Promise((resolve) =>
      setTimeout(() => {
        resolve();
      }, 5000)
    );
     var artx=" ";
     document.getElementById("alerttxt").value=artx;
}

// Calling generatearray function
//generatearray();
      </script>
    </body>
  </html>
```

5.5 Screen shots

5.5.1 Home page

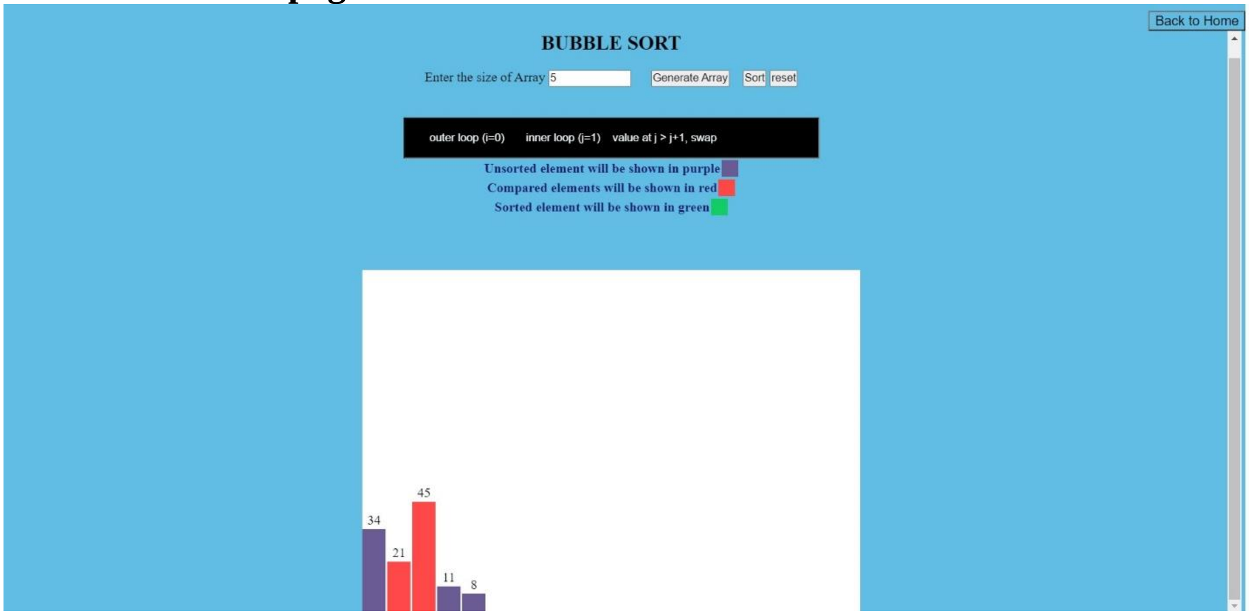


ALGOLEARN

Understanding algorithm is a better way to understand data structures...

It helps to learn algorithm visually...!

5.5.1 bubblemain page



BUBBLE SORT

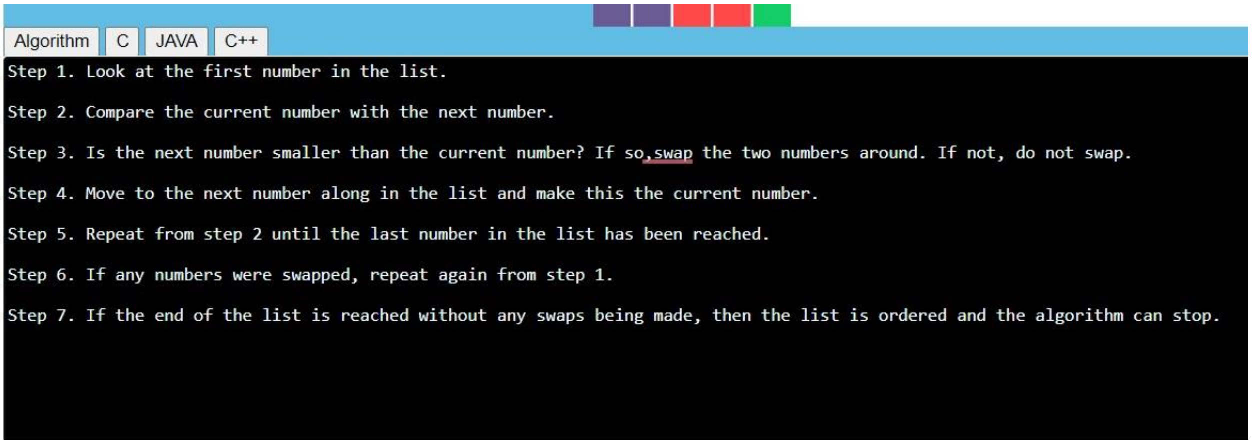
Enter the size of Array 5

outer loop (i=0) inner loop (j=1) value at j > j+1, swap

Unsorted element will be shown in purple🟪

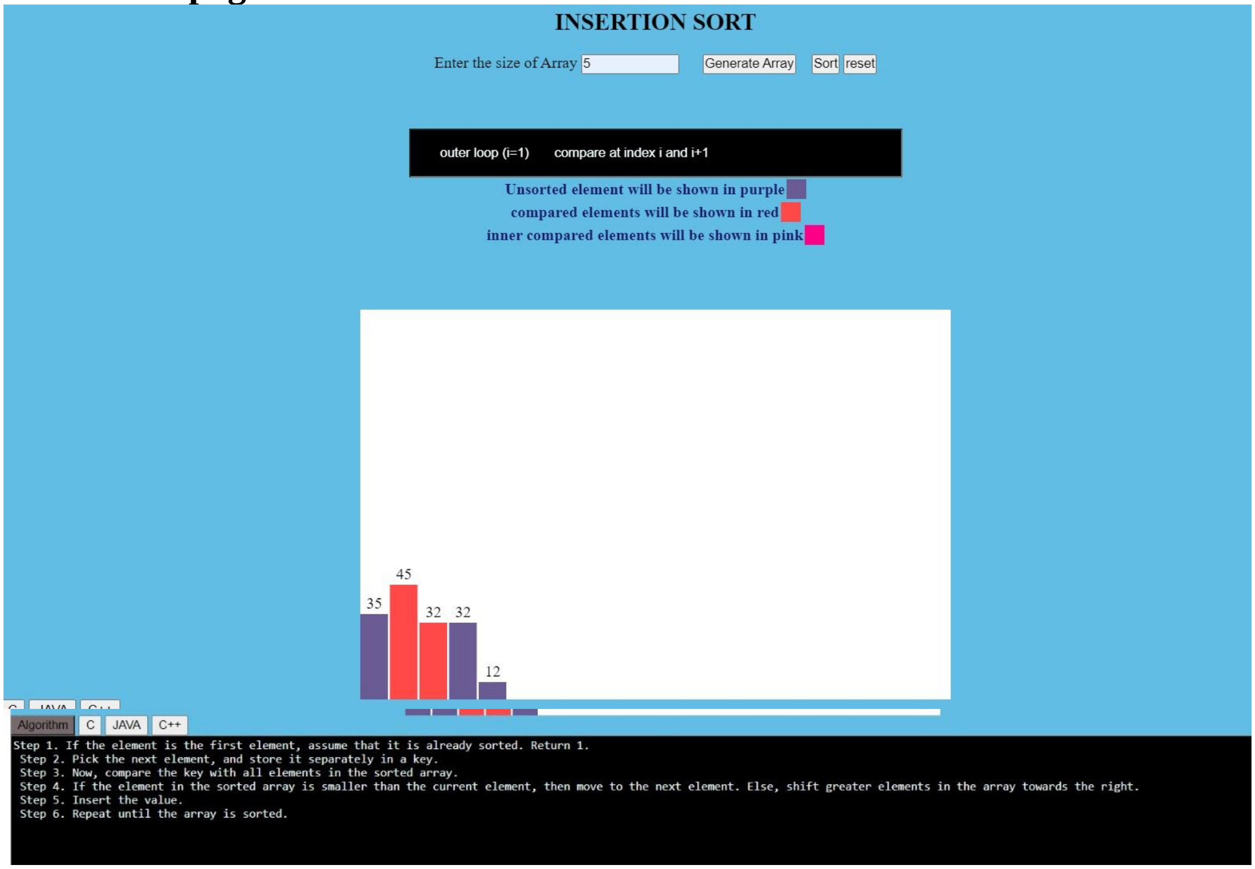
Compared elements will be shown in red🟥

Sorted element will be shown in green🟩



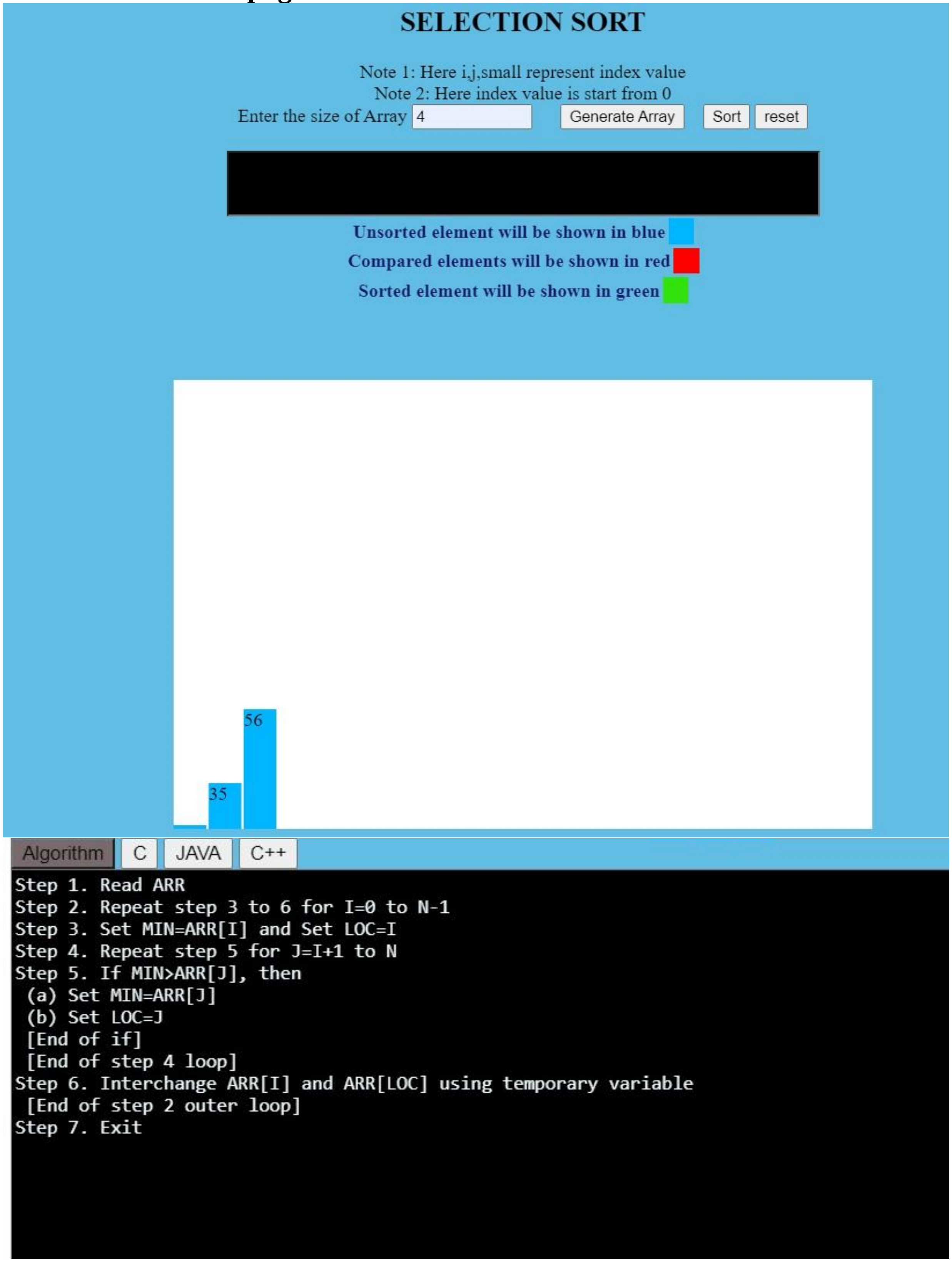
```
Step 1. Look at the first number in the list.

Step 2. Compare the current number with the next number.

Step 3. Is the next number smaller than the current number? If so,swap the two numbers around. If not, do not swap.

Step 4. Move to the next number along in the list and make this the current number.

Step 5. Repeat from step 2 until the last number in the list has been reached.

Step 6. If any numbers were swapped, repeat again from step 1.

Step 7. If the end of the list is reached without any swaps being made, then the list is ordered and the algorithm can stop.
```

5.5.1 insert page


INSERTION SORT
Enter the size of Array 5
Generate Array
Sort
reset
outer loop (i=1)
compare at index i and i+1
Unsorted element will be shown in purple
compared elements will be shown in red
inner compared elements will be shown in pink
45
35
32
32
12
Algorithm
C
JAVA
C++

```
Step 1. If the element is the first element, assume that it is already sorted. Return 1.
 Step 2. Pick the next element, and store it separately in a key.
 Step 3. Now, compare the key with all elements in the sorted array.
 Step 4. If the element in the sorted array is smaller than the current element, then move to the next element. Else, shift greater elements in the array towards the right.
 Step 5. Insert the value.
 Step 6. Repeat until the array is sorted.
```

5.5.1 Selectionmain page



SELECTION SORT

Note 1: Here i,j,small represent index value

Note 2: Here index value is start from 0

Enter the size of Array: 4

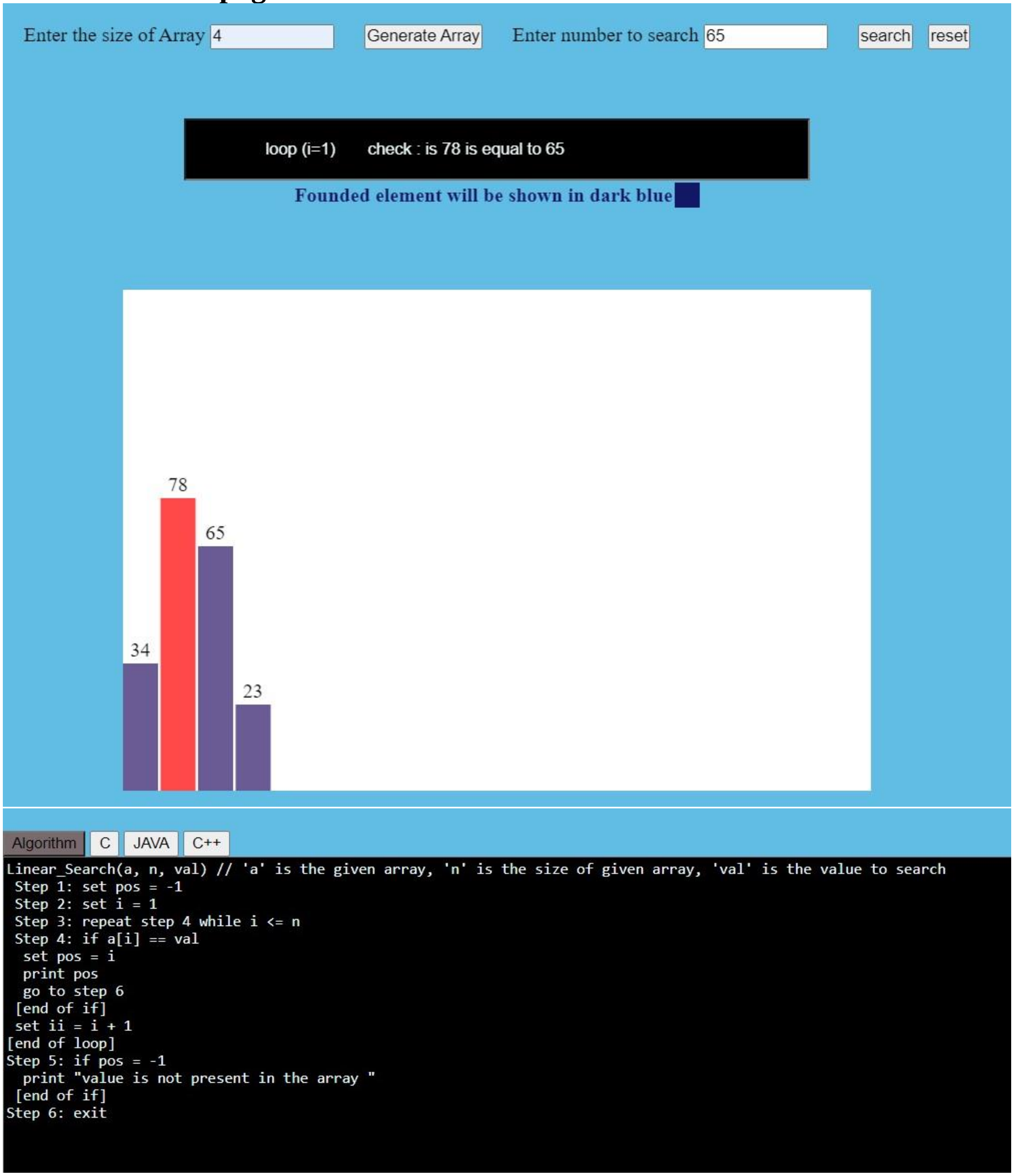
Unsorted element will be shown in blue 🟦

Compared elements will be shown in red 🟥

Sorted element will be shown in green 🟩

```
Step 1. Read ARR
Step 2. Repeat step 3 to 6 for I=0 to N-1
Step 3. Set MIN=ARR[I] and Set LOC=I
Step 4. Repeat step 5 for J=I+1 to N
Step 5. If MIN>ARR[J], then
 (a) Set MIN=ARR[J]
 (b) Set LOC=J
 [End of if]
 [End of step 4 loop]
Step 6. Interchange ARR[I] and ARR[LOC] using temporary variable
 [End of step 2 outer loop]
Step 7. Exit
```

5.5.1 linearmain page



Enter the size of Array 4

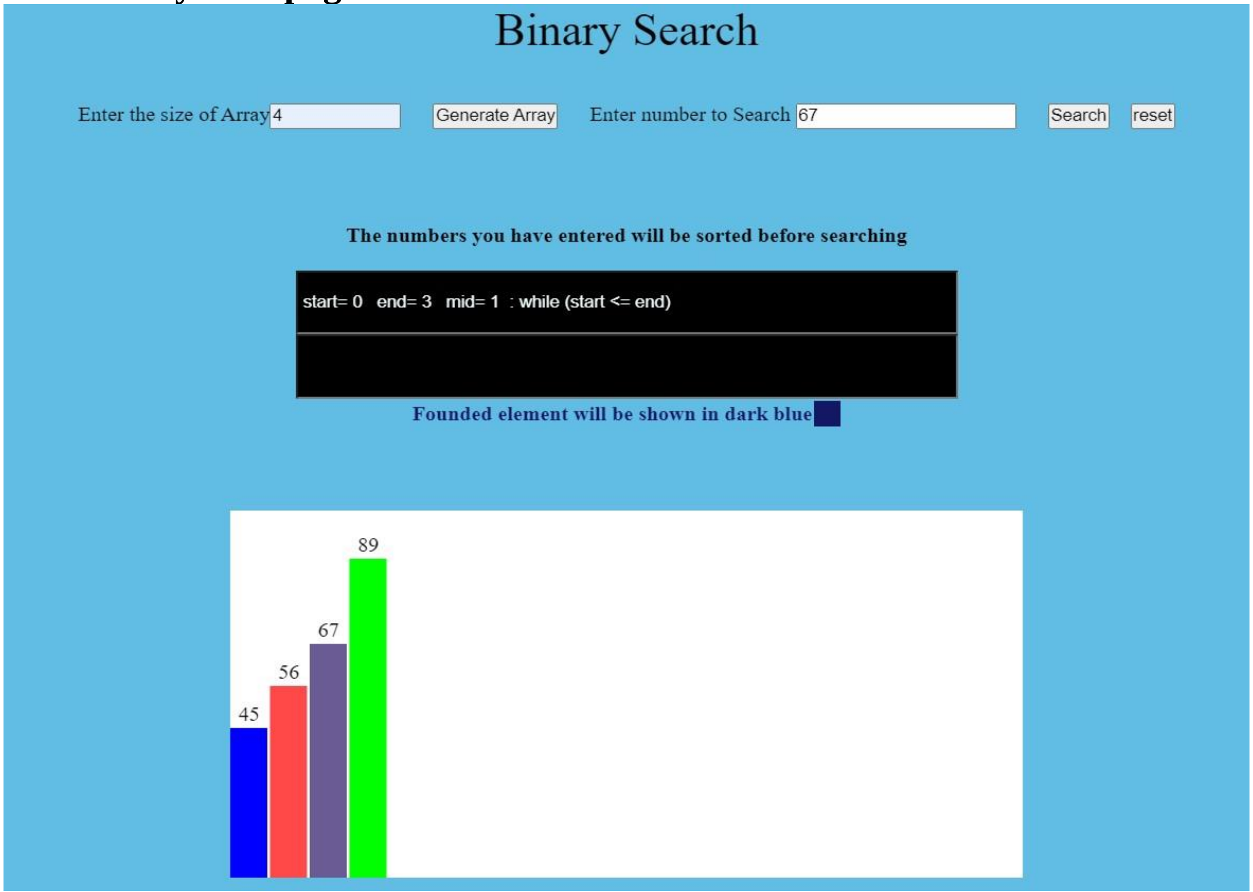
Enter number to search 65

loop (i=1) check : is 78 is equal to 65

Founded element will be shown in dark blue ■

```
Linear_Search(a, n, val) // 'a' is the given array, 'n' is the size of given array, 'val' is the value to search
 Step 1: set pos = -1
 Step 2: set i = 1
 Step 3: repeat step 4 while i <= n
 Step 4: if a[i] == val
  set pos = i
  print pos
  go to step 6
 [end of if]
 set ii = i + 1
[end of loop]
Step 5: if pos = -1
  print "value is not present in the array "
 [end of if]
Step 6: exit
```

5.5.1 binarymain page



Binary Search

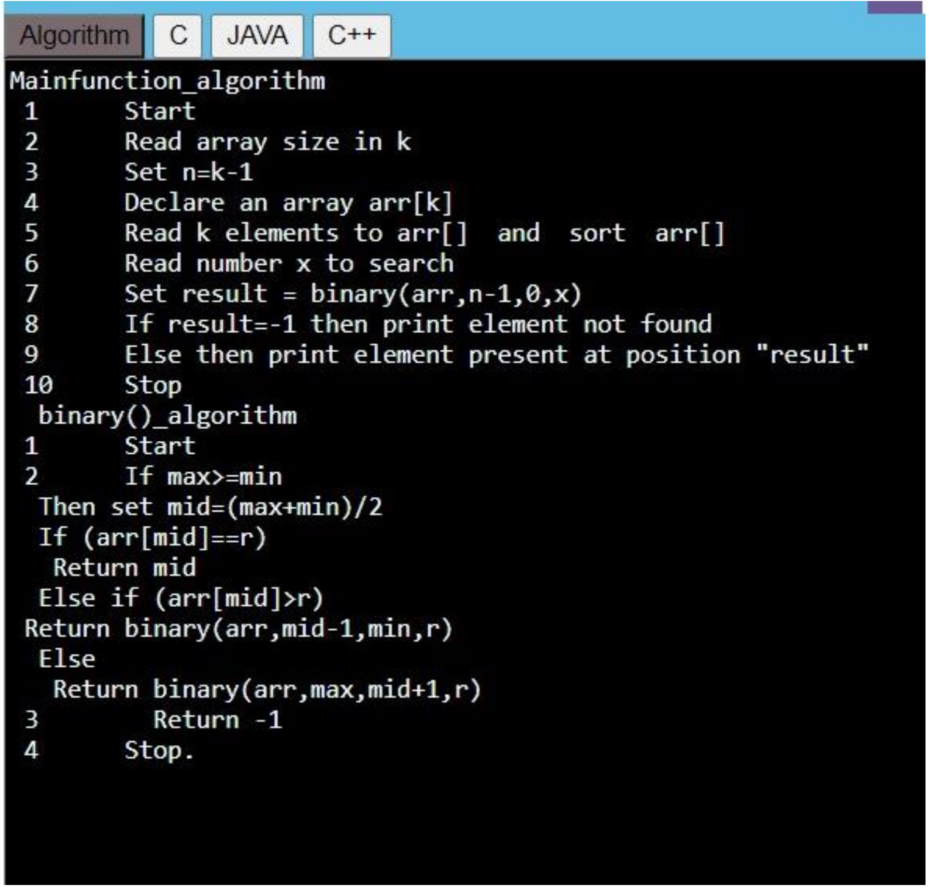
Enter the size of Array 4

Enter number to Search 67

The numbers you have entered will be sorted before searching

start= 0 end= 3 mid= 1 : while (start <= end)

Founded element will be shown in dark blue



```
Mainfunction_algorithm
 1      Start
 2      Read array size in k
 3      Set n=k-1
 4      Declare an array arr[k]
 5      Read k elements to arr[]  and  sort  arr[]
 6      Read number x to search
 7      Set result = binary(arr,n-1,0,x)
 8      If result=-1 then print element not found
 9      Else then print element present at position "result"
 10     Stop
  binary()_algorithm
 1      Start
 2      If max>=min
  Then set mid=(max+min)/2
  If (arr[mid]==r)
   Return mid
  Else if (arr[mid]>r)
 Return binary(arr,mid-1,min,r)
  Else
   Return binary(arr,max,mid+1,r)
 3        Return -1
 4      Stop.
```

Chapter 6

TESTING

6.1Introduction

System testing is a critical aspect of Software Quality Assurance and represents the ultimate review of specification, design, and coding. Testing is the process of executing a program with the intent of finding an error. A good test has a probability of finding an as-yet-undiscovered error. The purpose of testing is to identify and correct bugs in the developed system. Nothing is complete without testing. Testing is vital to the success of the system. The entire testing process can be divided into different phases:

- Unit testing
- Integration testing

6.2 Testing Methodologies adopted

6.2.1 Unit Testing

The first level of testing is unit testing. This test focuses on each module in the system. This is also known as “Module Testing”. To check whether each module in the software is proper and it gives desired outputs to the given inputs. All validations and conditions are tested in the module in the first unit. The goal is to test the internal logic of the module. In unit testing, each unit is tested during the programming stage itself, in the proposed system we tested the following.
in the proposed system we tested the following:

- Response of the system to inputs.
- Animation of the selected item.

6.2.2 Integration Testing

The modules are integrated to form a complete software package. The purpose of integration is to verify functional, performance, and reliability requirements placed on major design items. This testing is conducted based on modules. The integration testing is performed to detect design errors by focusing on testing the interconnection between modules. The objective is to take the unit-tested modules combined and tested as a whole.
in the proposed system we tested the following:

- Response of the system to inputs.
- Animation of the selected item.
- Linking between all the pages.

6.2.3 System testing

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements.

In system testing, integration testing passed components are taken as input. System testing detects defects within both the integrated units and the whole system. system testing tests the behavior and design of the system and the expectations of the customer. system testing is a black-box testing.

System testing is performed on the entire system in the context of either functional requirement specification (FRS) or system requirement specification (SRS) or both. System testing is usually carried out by a team that is independent of the development team to measure the quality of the system unbiased.

These are the list of types of system testing: -

- Usability testing
- Load testing
- Regression testing
- Recovery testing
- Migration testing
- Functional testing
- Security testing
- Stress testing
- Performance testing
- Deployment testing
- Hardware/software testing

6.3 Test Plan & Test Cases

The primary objectives of test case design methods are to drive a set of tests that have of highest likelihood of uncovering the defects. To accomplish this objective, two categories of test case design techniques are used. Black box testing and White box testing.

Black box testing

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing such as unit testing, integration testing, system testing, and acceptance testing. It is sometimes referred to as specification-based testing. The tester is aware of what the software is supposed to do but is not aware of how it does it.

White-box testing

White-box testing is a method of software testing that tests the internal structures or workings of an application, as opposed to its functionality. In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases. The tester requires an in-depth knowledge of the source code being tested. The tester chooses inputs to exercise paths through the code and determines the expected outputs. This is analogous to testing nodes in a circuit. White-box testing can be applied at the unit, integration, and system levels of the software testing process.

Test cases

6.3.1 bubble main page

Sl No	Test Case	Expected result	Observed result	Pass/Fail
1	Enter the size of the array in the textbox and click on generate array button	It should prompt for accepting the entered number of arrays	Prompt appeared	Pass
2	Do not Enter the size of the textbox array in the textbox and click on generate array button	Prompt for accepting numbers does not happen	No prompt appeared	Pass

6.3.2 Insert page

Sl No	Test Case	Expected result	Observed result	Pass/Fail
1	Enter the size of the array in the textbox and click on generate array button	It should prompt for accepting the entered number of arrays	Prompt appeared	Pass
2	Do not Enter the size of the array in the textbox and click on generate array button	Prompt for accepting numbers does not happen	No prompt where appeared	Pass

6.3.3 selectionmain page

Sl No	Test Case	Expected result	Observed result	Pass/Fail
1	Enter the size of the array in the textbox and click on generate array button	It should prompt for accepting the entered number of arrays	Prompt appeared	Pass

2	Do not Enter the size of the textbox array in the textbox and click on generate array button	Prompt for accepting numbers does not happen	No prompt where appeared	Pass

6.3.4 mainlinear page

Sl No	Test Case	Expected result	Observed result	Pass/Fail
1	Enter the size of the array in the textbox and click on generate array button	It should prompt for accepting the entered number of arrays	Prompt appeared	Pass
2	Do not Enter the size of the array in the textbox and click on generate array button	Prompt for accepting numbers does not happen	No prompt appeared	Pass

6.3.5 mainbinary page

Sl No	Test Case	Expected result	Observed result	Pass/Fail
1	Enter the size of the array in the textbox and click on generate array button	It should prompt for accepting the entered number of arrays	Prompt appeared	Pass
2	Do not Enter the size of the array in the textbox and click on generate array button	Prompt for accepting numbers does not happen	No prompt appeared	Pass

Chapter 7

CONCLUSION

Conclusion

This project has an important role among computer science students and teachers. Both can use this as an educational tool and understand basic algorithms clearly.

The option we created for generating an array by the user itself can help the user to understand how the process of the algorithm is taking place as they are the one who is deciding the inputs.

Moreover, the process of each time of sorting is displayed in a small window. We also provide algorithms and codes in c, c++, and java. since we can conclude our project as a complete educational tool for a better understanding of sorting and searching algorithms.

Future Work

Every creation is subjected to modification according to the need of that time. Our system can also be developed and can be expanded further in the future. Major future works we noted are: -

- Develop a pause and resume button for an animation, the user can manage the speed of animation and get time for thinking in each step the of algorithm
- include other data structure algorithms such as a tree, graph, stack, queue, and their animations to help users to learn those algorithms.

References

[1] HTML 5 Blackbook, Dream Tech Press,2016 Edition.
[3] Horowitz and Sahani, “Fundamentals of Data Structures”, Galgotia Publication Pvt. Ltd.,
NewDelhi.
[3] JavaScript Tutorial, Available at https://youtu.be/3mjwtu4_0uk/
[4] Web designing on https://www.w3schools.com/
[5] HTML 5 and CSS Tutorials at https://www.geeksforgeeks.org/