

MINI PROJECT

(SESSION-2020-2021)

ALGORITHM VISUALIZER

Report



Institute of Engineering & Technology

Team Members

SIDDHARTH SINGH

(181500710)

KAVYA SINGH

(181500319)

SPARSH GUPTA

(181500723)

HARSHIT JAISWAL

(181500255)

SURYANSHU GUPTA

(181500740)

Supervised By

MR.NEERAJ KHANNA

TECHNICAL TRAINER

Department of Computer Engineering



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

Declaration

I hereby declare that the work which is being presented in the Mini Project “**Algorithm Visualizer**”, in partial fulfillment of the requirements for Mini project Lab is an authentic record of my own work carried under the supervision of **Mr. Neeraj Khanna, Technical Trainer.**

Siddharth Singh

Kavya Singh

Sparsh Gupta

Harshit Jaiswal

Suryanshu Gupta



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

Certificate

This is to certify that the project entitled “Coaching Management Application” carried out in Mini Project – I Lab is a bonafide work done by **Siddharth Singh** (181500710), **Kavya Singh** (181500319), **Sparsh Gupta** (181500723), **Harshit Jaiswal** (181500255) and **Suryanshu Gupta** (181500740) and is submitted in partial fulfillment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

Signature of Supervisor:

Name of Supervisor: Mr. Anand Gupta

Date: 05/04/2019

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Mini Project undertaken during B. Tech. Third Year. This project in itself is an acknowledgement to the inspiration, drive and technical assistance contributed to it by many individuals. This project would never have seen the light of the day without the help and guidance that we have received.

Our heartiest thanks to Dr. (Prof). Anand Singh Jalal, Head of Dept., Department of CEA for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal.

We owe a special debt of gratitude to Mr. Neeraj Khanna, Technical Trainer, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. He has showered us with all his extensively experienced ideas and insightful comments at virtually all stages of the project & has also taught us about the latest industry-oriented technologies.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Siddharth Singh

Kavya Singh

Sparsh Gupta

Harshit Jaiwal

Suryanshu Gupta

Table of Contents

Declaration

Certificate

Acknowledgments

Abstract

1. Introduction

Motivation and Overview

Objective

Hypothesis

Methodology

Summary

2. Software Requirement Analysis

Define the problem

Define the modules and their functionalities (SRS)

3. Tools Used

Web Browser

HTML

CSS

Javascript

React Js

Node Js

Softwares used

4. Deployment

Github

GitBash

5. Reason for Selecting this Project

Software Design

. Screenshots

References

Introduction :

Algorithms and data structures form one cornerstone of an undergraduate computer science education. A technique for improving instruction in this critical area is to include algorithm and data structure visualizations. Analysis and design of algorithms is a great challenge for computer science student's. We selected this project because we were fascinated by sorting algorithms, and we wanted to visualize them in action. This Project is a fully automatic visualization system with step-by-step explanations and comparison of sorting algorithms. Design principles and technical structure of the visualization system as well as its practical implications and educational benefits are presented and discussed.

Objective:

Our projects focus on how a person can learn the working of an algorithm by not only learning its source code but also by observing how it is working. Students who are studying in colleges or schools have privilege of having teachers who can teach them basics of algorithms but there are also many who dreams of learning to code and get placed in big tech giants without having Computer Science background. When some of these people try to understand certain algorithms, they puzzled themselves in the lengthy code but if at the same time they were able visualize that very code it will be easier for them to learn. Many a time experienced teachers also find it difficult to teach a certain algorithm on a teaching board but in today's era of technology and the facilities provided by the institutions when get combined with our project we can also solve that very problem.

Hypothesis:

Our project focuses on algorithms and targets both teaching as well as tech industries so the future scope of our project seems to be endless. We all hope that every developer will contribute to our open source project. Regarding the time provided our project initially focuses on some basic sorting algorithms such as merge sort, selection sort and few others. In future we aim to add as many algorithms as possible, first we would like to cover all sorting

algorithms and after that our next target will be to work on algorithms which work on graphs or mainly path finding algorithms such as Dijkstra's Algorithm and Best-First-Search.

Methodology (Including a summary of the project):

I was fascinated by sorting algorithms, and I wanted to visualize them in action. Algorithm visualizer project consists of sorting methods like Bubble, heap, merge and quicksort (We would try to include a few more methods). The main aim of our project is to visualize the sorting of an unsorted array having random numbers but including duplicate values by JavaScript on React. These arrays would be converted into bars by CSS (Cascading Style Sheets) techniques. This project will consist of various technologies like CSS (Cascading Style Sheets), HTML (Hypertext Markup Language), JavaScript and this all would be done through React. We would provide a slider button in our project so that the user can make the array bigger or smaller as per his/her convenience. In addition to that, sorting speed could also be controlled in our project. The user just has to click the sort button and click and visualize the algorithm. While sorting we included animations like colour pattern and used different colours for unsorted elements, already ordered elements, sorted elements etc. The important point of this project is to show the sorting algorithm process graphically since graphically sorting can be shown simply by swapping the lines.

CSS will play a major role in our project as it will help us to create the visual animations. All the source code for the algorithms will be written in JavaScript. Since, our project being a web application therefore we are using React.js which is a library for JavaScript.

What contribution would the project make and where? :

The contribution that an open-source project creates is endless, the only constraint which limits our imagination. The project on our team is working and has an enormous amount of contribution. "Everybody should learn to program a computer, because it teaches you how to think." - Steve Jobs. Our project aims at this very thought of Steve Jobs, not only normal people but sometimes many CS graduates or people associated with the technology industry fail to understand how

an algorithm works. “Visualizing information can give us a very quick solution to problems. We can get clarity or the answer to a simple problem very quickly.” David McCandless. Our projects focus on how a person can learn the working of an algorithm by not only learning its source code but also by observing how it is working. Students who are studying in colleges or schools have privilege of having teachers who can teach them basics of algorithms but there are also many who dreams of learning to code and get placed in big tech giants without having Computer Science background. When some of these people try to understand certain algorithms, they puzzled themselves in the lengthy code but if at the same time they were able visualize that very code it will be easier for them to learn. Many a time experienced teachers also find it difficult to teach a certain algorithm on a teaching board but in today's era of technology and the facilities provided by the institutions when get combined with our project we can also solve that very problem.

“Scientists study the world as it is; engineers create the world that has never been.” —Theodore von Karman, Hungarian-American mathematician, aerospace engineer, and physicist. Our projects will make a small contribution to this very thinking but yes we are proud of the contributions which our project will make in shaping future engineers.

Scope for extension into a major project:

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers. Our scope here is the higher education in the field of computer science. So within the paper, we discuss the extension of standard methods of teaching algorithms, using the whiteboard or slides, with the algorithm visualizations. According to them they can be used to attract students' attention during the lecture, explain concepts in visual terms, encourage a practical learning process, and facilitate better communication between students and instructors. Interactive algorithm visualizations allow students to experiment and explore the ideas with respect to their individual needs.

Summary: This project would help one to understand the sorting algorithms in depth. We used React JS, CSS, JavaScript, HTML etc to make this project.

Contribution summary:

Suryanshu has completed the sorting algorithms

Kavya and siddharth has completed the ui

Sparsh and harshit has completed the deployment section

DEFINITIONS

ALGORITHM VISUALIZER

A web based application used to demonstrate the working of various sorting algorithms techniques.

Analysis and design of algorithms is a great challenge for computer science student's .We selected this project because we were fascinated by sorting algorithms, and we wanted to visualize them in action

Algorithms and data structures form one cornerstone of an undergraduate computer science education. A technique for improving instruction in this critical area is to include algorithm and data structure visualizations. Thus our Mini project titled as Algorithm visualizer would help the beginners to visualize the basic algorithms and get a better understanding of the underlying algorithms.

In making this project we have used many technologies like React JS,JS,CSS,HTML etc. We have make our project from scratch So without any further delay I would like to show u the live demo of our project

UI:

Our application consists of a webpage from where users can select different sorting algorithms which we want to visualize or want to understand how they work.User can also select the size of array and also select the speed of algorithm visualization.

VISUAL STUDIO_:

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for source control systems (like Subversion and Git) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer).

Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C,[6] C++, C++/CLI, Visual Basic .NET, C#, F#,JavaScript, TypeScript, XML, XSLT, HTML, and CSS. Support for other languages such as Python,Ruby, Node.js, and M among others is available via plug-ins. Java (and J#) were supported in the past.

Microsoft Visual Studio is an [integrated development environment](#) (IDE) from [Microsoft](#). It is used to develop [computer programs](#), as well as [websites](#), [web apps](#), [web services](#) and [mobile apps](#). Visual Studio uses Microsoft software development platforms such as [Windows API](#), [Windows Forms](#), [Windows Presentation Foundation](#), [Windows Store](#) and [Microsoft Silverlight](#). It can produce both [native code](#) and [managed code](#).

Visual Studio includes a [code editor](#) supporting [IntelliSense](#) (the [code completion](#) component) as well as [code refactoring](#). The [integrated debugger](#) works both as a source-level debugger and a machine-level debugger. Other built-in tools include a [code profiler](#), forms designer for building [GUI](#) applications, [web](#)

designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for source control systems (like Subversion and Git) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer).

Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C,[6] C++, C++/CLI, Visual Basic .NET, C#, F#,[7] JavaScript, TypeScript, XML, XSLT, HTML, and CSS. Support for other languages such as Python,[8] Ruby, Node.js, and M among others is available via plug-ins. Java (and J#) were supported in the past.

The most basic edition of Visual Studio, the Community edition, is available free of charge.

The currently supported Visual Studio version is 2019.

WEB BROWSER :

A **web browser** (commonly referred to as a **browser**) is a software application for accessing information on the World Wide Web. Each individual web page, image, and video is identified by a distinct Uniform Resource Locator (URL), enabling browsers to retrieve these resources from a web server and display them on the user's device.

A web browser is not the same thing as a search engine, though the two are often confused.[1][2] For a user, a search engine is just a website, such as google.com, that stores searchable data about other websites. But to connect to a website's server and display its web pages, a user needs to have a web browser installed on their device.[3]

The most popular browsers are Chrome, Firefox, Safari, Internet Explorer, and Edge

TOOLS USED

HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web application. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

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

HTML Elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as `` and `<input/>` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

HTML code ensures the proper formatting of text and images so that your Internet browser may display them as they are intended to look. Without HTML, a browser would not know how to display text as elements or load images or other elements. HTML also provides a basic structure of the page, upon which Cascading Style Sheets are overlaid to change its appearance. One could think of HTML as the bones (structure) of a web page, and CSS as its skin (appearance).

CSS (Cascading Style Sheets)

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

Advantage Of CSS

CSS saves time – You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.

Pages load faster – If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.

Easy maintenance – To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.

Superior styles to HTML – CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

Multiple Device Compatibility – Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.

Global web standards – Now HTML attributes are being deprecated and it is being recommended to use CSS. So it's a good idea to start using CSS in all the HTML pages to make them compatible with future browsers.

JAVA SCRIPT (JS)

JavaScript (/ˈdʒɑːvəˌskrɪpt/),[8] often abbreviated as **JS**, is a high-level, interpreted programming language that conforms to the ECMAScript specification. It is a programming language that is characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web.[9] JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it,[10] and major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has APIs for working with text, arrays, dates, regular expressions, and the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities. It relies upon the host environment in which it is embedded to provide these features.

Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

The terms *Vanilla JavaScript* and *Vanilla JS* refer to JavaScript not extended by any frameworks or additional libraries. Scripts written in Vanilla JS are plain JavaScript code.[11][12]

Although there are similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design. JavaScript was influenced by programming languages such as Self and Scheme.[13]

FONT AWESOME

Font Awesome is a font and icon toolkit based on CSS and LESS. It was made by Dave Gandy for use with Twitter Bootstrap, and later was incorporated into the BootstrapCDN. Font Awesome has a 20% market share among those websites

which use third-party Font Scripts on their platform, ranking it second place after Google Fonts.[2]

Font Awesome 5 was released on December 7, 2017 with 1,278 icons[3]. Version 5 comes in two packages: Font Awesome Free and the proprietary Font Awesome Pro (available for a fee). The free versions (all releases up to 4 and the free version for 5) are available under SIL Open Font License 1.1, Creative Commons Attribution 4.0, and MIT License.

DEVELOPMENT TOOL-

REACT JS

React (also known as **React.js** or **ReactJS**) is an [open-source, front end, JavaScript library](#)^[3] for building [user interfaces](#) or UI components. It is maintained by [Facebook](#) and a community of individual developers and companies.^{[4][5][6]} React can be used as a base in the development of [single-page](#) or mobile applications. However, React is only concerned with rendering data to the [DOM](#), and so creating React applications usually requires the use of additional libraries for [state management](#) and routing.^{[7][8]} [Redux](#)^[9] and React Router^[10] are respective examples of such libraries.

Advantages of React js

React has many other reasons for being popular and in demand. Few of the reasons are mentioned below:

1. It facilitates the overall process of writing components
2. It boosts productivity and facilitates further maintenance
- 3 It ensures faster rendering
- 4 It guarantees stable code
- 5 It is SEO friendly
- 6 It comes with a helpful developer toolset
- 7 There is React Native for mobile app development
- 8 It is focused and easy-to-learn
- 9 It is backed by a strong community
- 10 It is used by both Fortune 500 companies and innovative startups

Main Components of React js

Components

React code is made of entities called components. Components can be rendered to a particular element in the [DOM](#) using the React DOM library. When rendering a component, one can pass in values that are known as "props"^[11]:

```
ReactDOM.render(<Greeter greeting="Hello World!" />,
document.getElementById('myReactApp'));
```

The two primary ways of declaring components in React is via functional components and class-based components.

Functional components

Functional components are declared with a function that then returns some JSX.

```
const Greeting = (props) => <div>Hello, {props.name}!</div>;
```

Class-based components

Class-based components are declared using [ES6](#) classes.

```

class ParentComponent extends React.Component {
  state = { color: 'green' };
  render() {
    return (
      <ChildComponent color={this.state.color} />
    );
  }
}

```

Virtual DOM

Another notable feature is the use of a virtual [Document Object Model](#), or virtual DOM. React creates an [in-memory](#) data-structure cache, computes the resulting differences, and then updates the browser's displayed DOM efficiently.^[12] This process is called **reconciliation**. This allows the programmer to write code as if the entire page is rendered on each change, while the React libraries only render subcomponents that actually change. This selective rendering provides a major performance boost. It saves the effort of recalculating the CSS style, layout for the page and rendering for the entire page.

JSX

JSX, or JavaScript [XML](#), is an extension to the JavaScript language syntax.^[13] Similar in appearance to HTML, JSX provides a way to structure component rendering using syntax familiar to many developers. React components are typically written using JSX, although they do not have to be (components may also be written in pure JavaScript). JSX is similar to another extension syntax created by Facebook for [PHP](#) called [XHP](#).

An example of JSX code:

```

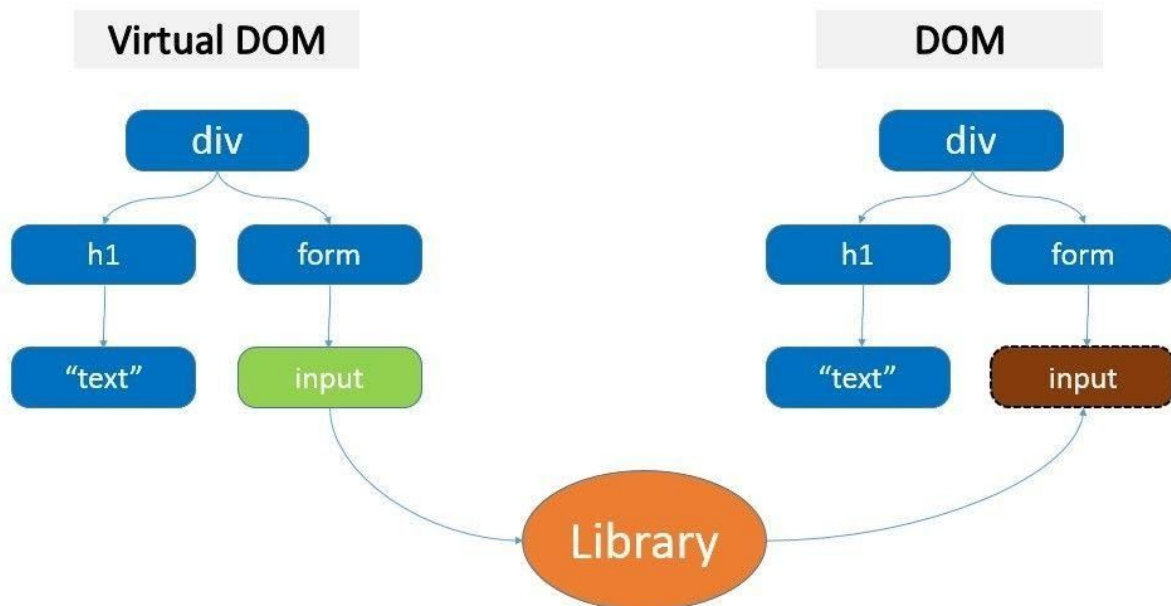
class App extends React.Component {
  render() {
    return (
      <div>
        <p>Header</p>
        <p>Content</p>
        <p>Footer</p>
      </div>
    );
  }
}

```

React hooks

Hooks are functions that let developers "hook into" React state and lifecycle features from function components.^[21] They make code readable and easily understandable. Hooks don't work inside classes — they let you use React without classes.^[22]

React provides a few built-in Hooks like `useState`,^[23] `useContext`, `useReducer` and `useEffect`^[24] to name a few. They are all stated in the Hooks API Reference.^[25] `useState` and `useEffect`, which are the most used, are for controlling states and side effects respectively in React Components.



NODE JS

Node.js provides a JavaScript Environment which allows the user to run their code on the server (outside the browser). Node package manager i.e. npm allows the user to choose from thousands of free packages (node modules) to download

Advantages Of Node.js

- Open source JavaScript Runtime Environment
- Single threading – Follows a single threaded model.
- Data Streaming
- Fast – Built on Google Chrome's JavaScript Engine, Node.js has a fast code execution.
- Highly Scalable
- Initialize a Node.js application by typing running the below command into the command window. Accept the standard settings. "npm init"

GitBash

Git Bash is a source control management system for Windows. It allows users to type Git commands that make source code management easier through versioning and commit history. **Bash** is a Linux-based command line (that has been ported over to Windows) while **Shell** is a native Windows command line.

Git Bash is not a GUI software, it is a command-line prompt. You will only use Git Bash to write and run commands on the terminal.

GitHub

GitHub is a for-profit company that offers a cloud-based Git repository hosting service. Essentially, it makes it a lot easier for individuals and teams to use Git for version control and collaboration.

GitHub's interface is user-friendly enough so even novice coders can take advantage of Git. Without GitHub, using Git generally requires a bit more technical savvy and use of the command line.

GitHub is so user-friendly, though, that some people even use GitHub to manage other types of projects – [like writing books](#).

Additionally, anyone can sign up and host a public code repository for free, which makes GitHub especially popular with open-source projects.

As a company, GitHub makes money by selling hosted private code repositories, as well as other business-focused plans that make it easier for organizations to manage team members and security. We utilize Github extensively at Kinsta to manage and develop internal projects.

Deployment:

Deployment on GitHub Pages Deploying to GitHub Pages is automatic. Once it's set up, deploying happens whenever you push your local changes to your remote, GitHub-hosted repository. Head to GitHub Pages' setup instructions and follow the steps exactly to get your main GitHub Pages page setup.

Since we have deployed our react app on github so we first we have to add a homepage on package.json file and then install gh-pages by using few commands.

Steps for deployment:

Step 1: Add homepage to package.json#

The step below is important!

If you skip it, your app will not deploy correctly.

Open your package.json and add a homepage field for your project: "homepage": "https://myusername.github.io/my-app", or for a GitHub user page: "homepage": "https://myusername.github.io", or for a custom domain page: "homepage": "https://mywebsite.com", Create React App uses the homepage field to determine the root URL in the built HTML file.

Step 2: Install gh-pages and add deploy to scripts in package.json#

Now, whenever you run npm run build, you will see a cheat sheet with instructions on how to deploy to GitHub Pages.

To publish it at https://myusername.github.io/my-app, run:

```
npm install --save gh-pages
```

Alternatively you may use yarn:

```
yarn add gh-pages
```

Add the following scripts in your package.json:

```
"scripts": {  
  + "predeploy": "npm run build",  
  + "deploy": "gh-pages -d build",  
  "start": "react-scripts start",  
  "build": "react-scripts build",
```

The predeploy script will run automatically before deployment.

If you are deploying to a GitHub user page instead of a project page you'll need to make one additional modification:

Tweak your package.json scripts to push deployments to master:

```
"scripts": {  
  "predeploy": "npm run build",  
  - "deploy": "gh-pages -d build",  
  
  + "deploy": "gh-pages -b master -d build",  
}
```

Step 3: Deploy the site by running `npm run deploy`

Then run:

```
npm run deploy
```

Step 4: For a project page, ensure your project's settings use `gh-pages`

Finally, make sure GitHub Pages option in your GitHub project settings is set to use the `gh-pages` branch:

Step 5: Optionally, configure the domain

You can configure a custom domain with GitHub Pages by adding a CNAME file to the public/ folder.

Your CNAME file should look like this:

Requirements:

Following are the hardware and the software requirements for our project:

a) Hardware:

- ✓ Laptop/Desktop
- ✓ 1.8 GHz or faster processor. Quad-core or better recommended
- ✓ 4 GB of RAM
- ✓ Hard disk space: Minimum of 800MB up to 210GB of available space
- ✓ Video card that supports a minimum display resolution of 720p (1280 by 720)

b) Software:

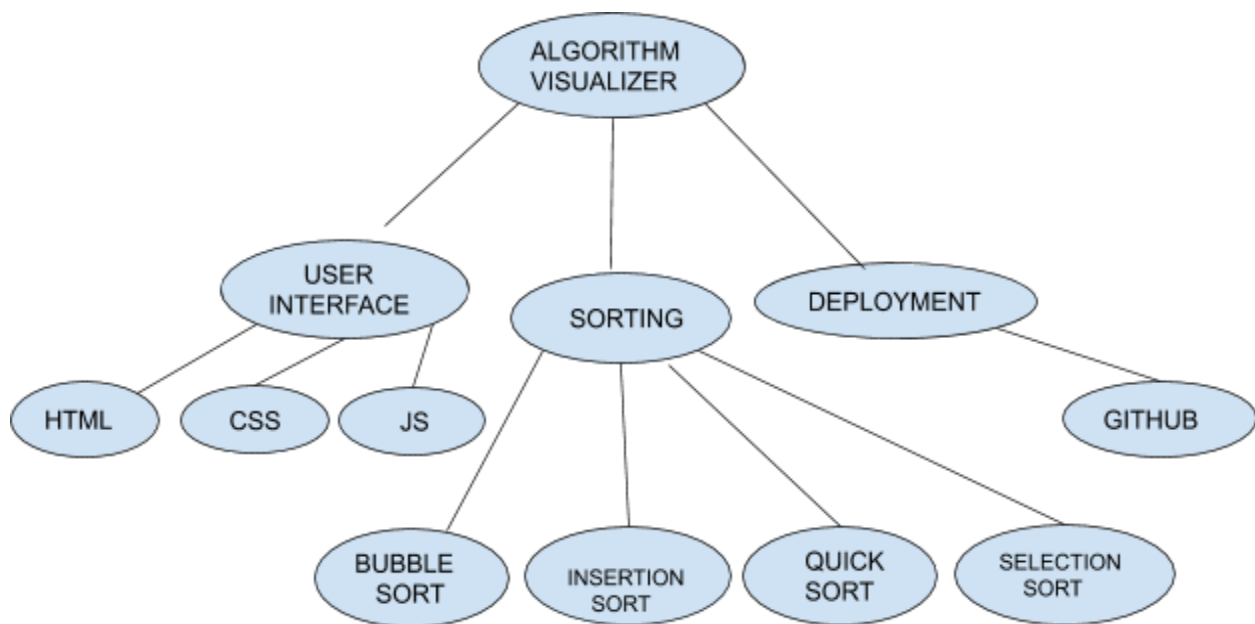
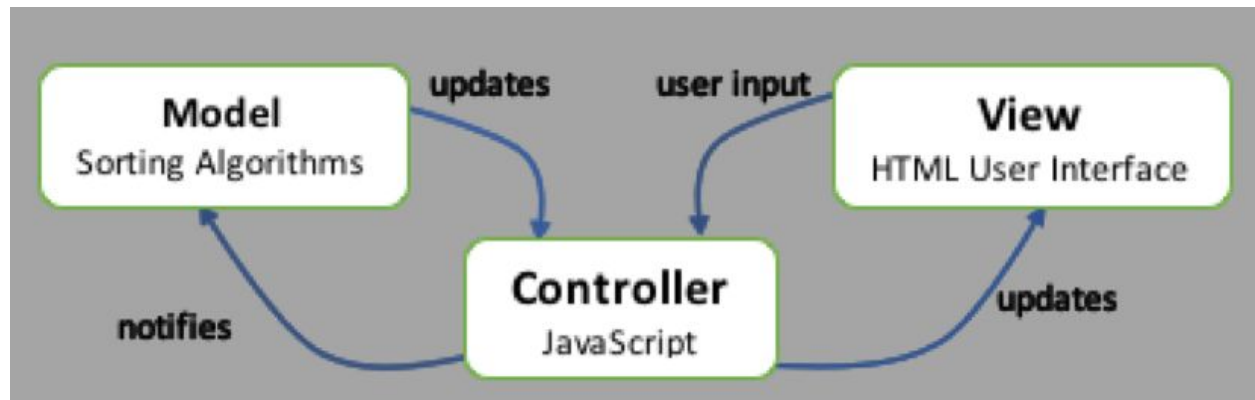
- Windows 8.1 and above
- Visual Studio
- Web Browser
- Node js for hosting server on localhost

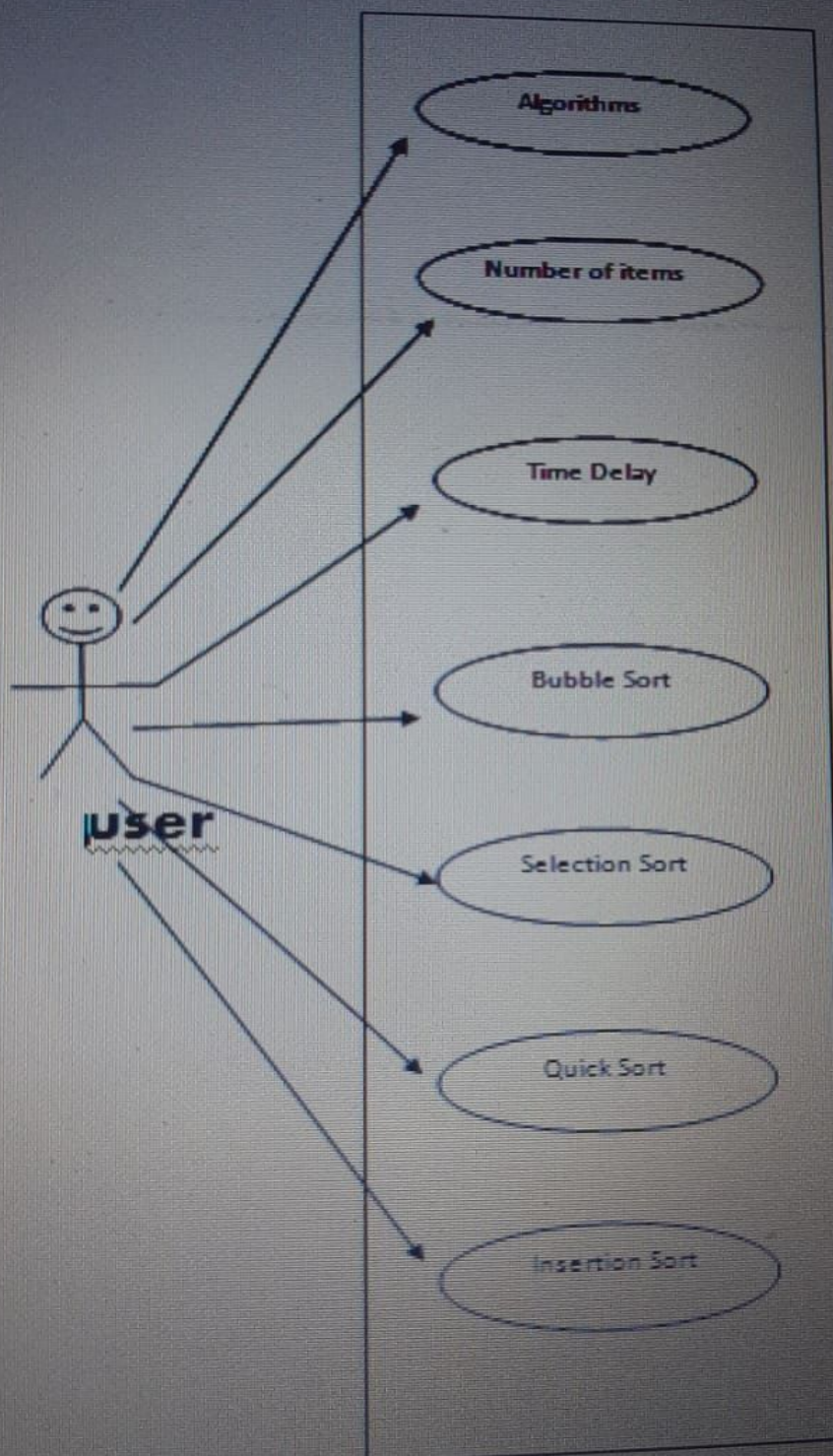
Reason for selecting the topic

Analysis and design of algorithms is a great challenge for computer science student's .We selected this project because we were fascinated by sorting algorithms, and we wanted to visualize them in action.This Project is a fully automatic visualization system.

Design principles and technical structure of the visualization system as well as its practical implications and educational benefits are presented and discussed.

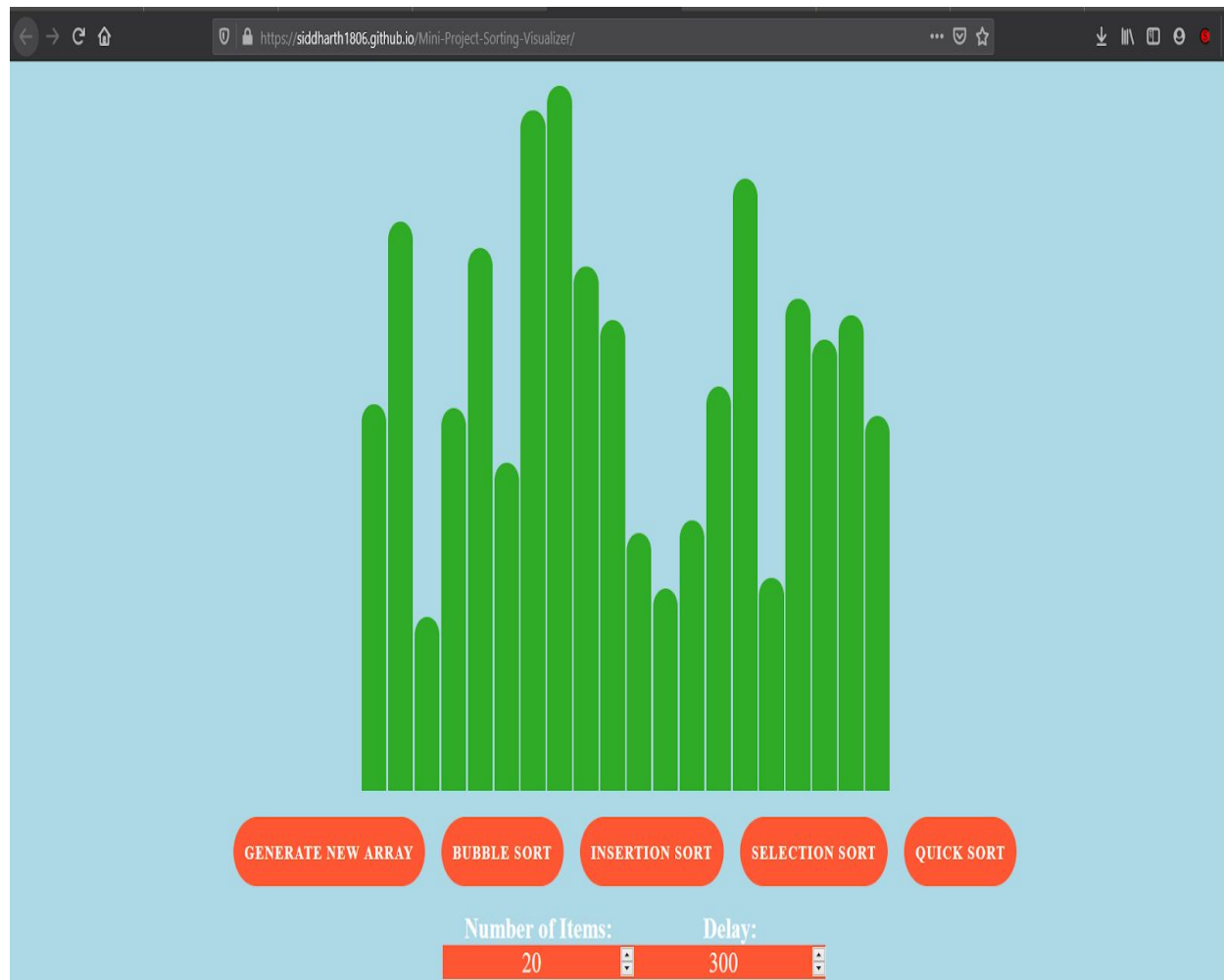
Software Design





Implementation and user interface

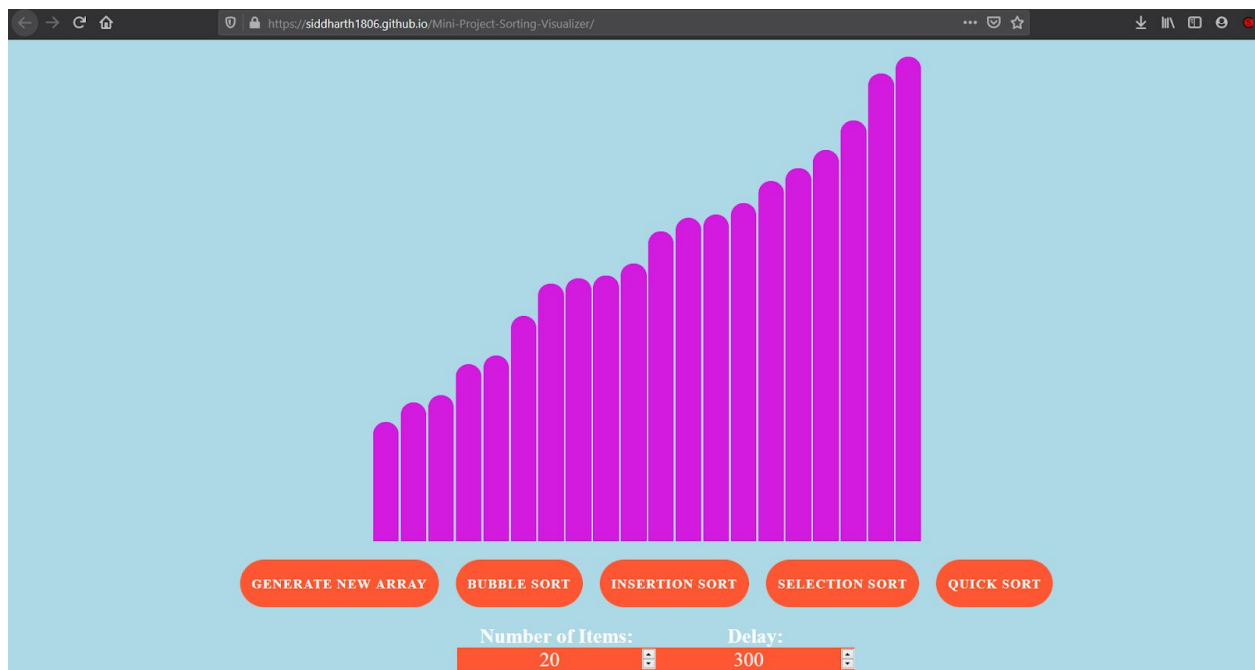
Figure 1 : UI of the project.



a. The figure shows the array before any sorting begins. Here, green color shows that the input data is unsorted.



b. The intermediate process when we have selected the Bubble sort. The yellow color shows the current elements which are getting compared.



c. The process when the array has been sorted.

App.js

```
Mini project > sorting > src > JS App.js > [0] default
1  import React from 'react';
2
3  import SortingVisualizer from './SortingVisualizer/SortingVisualizer.jsx'
4
5  function App() {
6    return (
7      <div>
8        <SortingVisualizer />
9      </div>
10   );
11 }
12
13 export default App;
```

index.js

```
Mini project > sorting > src > JS index.js
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import App from './App';
4
5
6  ReactDOM.render(<App />, document.getElementById('root'));
7
```

Sorting Visualizer:

SortingVisualizer.css

Contains the styling of the page of all the buttons and bars.

```
# SortingVisualizer.css X
Mini project > sorting > src > SortingVisualizer > # SortingVisualizer.css > html
1  {
2    background: lightblue;
3  }
4
5
6  #centerdiv{
7    margin: 0 auto;
8    white-space: nowrap;
9  }
10
11  .centerdivKeepWidth{
12    margin: 0 auto;
13    text-align: center;
14  }
15
16  #itemsDiv{
17    height: 42vw;
18    position: relative;
19  }
20
21  .array-item{
22    display: inline-block;
23    border-radius: 15px 15px 0px 0px;
24    margin: 0px 0px 2px 2px;
25    transition: height 0.3s;
26    position: sticky;
27    top: 999999px;
28  }
29
30  label{
31    color: white;
32    font-size: x-large;
33    font-weight: 700;
34  }
35
36  #buttonsDiv{
37    height: 30%;
38    margin: 20px 0px 20px 0px;
```

```
file Edit Selection View Go Run Terminal Help
SortingVisualizer.css - Mini-Project-Sorting-Visualizer-main - Visual Studio Code

# SortingVisualizer.css X
Mini project > sorting > src > SortingVisualizer > # SortingVisualizer.css > html
41  button{
42    font-family: inherit;
43    font-size: inherit;
44    cursor: pointer;
45    padding: 15px 10px;
46    transition: all 0.2s;
47    display: inline-block;
48    text-transform: uppercase;
49    letter-spacing: 1px;
50    font-weight: 700;
51    outline: none;
52    position: relative;
53    color: #fff;
54    background: #ff5733;
55    border: 4px solid #ff5733;
56    border-radius: 60px;
57  }
58  button:hover {
59    background: transparent;
60    color: #ff5733;
61  }
62
63  div button, div input, div label{
64    margin: 0px 10px;
65  }
66
67
68  .flexDiv{
69    display: flex;
70    margin: 0px 35%;
71  }
72
73  .numberLabel{
74    display: block;
75  }
76
77  .numberInput{
78    all: unset;
```

SortingVisualizer.jsx

```
file Edit Selection View Go Run Terminal Help
SortingVisualizer.jsx - Mini-Project-Sorting-Visualizer-main - Visual Studio Code

SortingVisualizer.jsx X
Mini project > sorting > src > SortingVisualizer > SortingVisualizer.jsx > ...

39 generateNewArray(){
40     abort = true;
41     this.resetArray();
42 }
43
44 async SortArray(algo){
45     let sortedArrayAnim = algo(this.state.arrayToSort);
46     let arrayToSort = this.state.arrayToSort;
47     let prevChanged = this.state.prevChanged;
48
49     for (let index = 0; index < sortedArrayAnim.length; index++) {
50         if(this.abort){
51             console.log(abort);
52             return null;
53         }
54         const [i,j] = sortedArrayAnim[index];
55
56         let temp = arrayToSort[i];
57         arrayToSort[i] = arrayToSort[j];
58         arrayToSort[j] = temp;
59
60         prevChanged.push(i,j);
61
62         if(index == sortedArrayAnim.length - 1){
63             prevChanged.push(arrayToSort.length + 1, arrayToSort.length + 1);
64             this.setState({prevChanged});
65         }
66
67         this.setState({ arrayToSort,prevChanged });
68         await sleep(this.state.delay);
69     }
70 }
71
72 async selectionSort(){
73     let sortedArrayAnim = SelectionSort(this.state.arrayToSort);
74     let arrayToSort = this.state.arrayToSort;
75     let prevChanged = this.state.prevChanged;
76
```

```
file Edit Selection View Go Run Terminal Help
SortingVisualizer.jsx - Mini-Project-Sorting-Visualizer-main - Visual Studio Code

SortingVisualizer.jsx X
Mini project > sorting > src > SortingVisualizer > SortingVisualizer.jsx > ...

71
72 async selectionSort(){
73     let sortedArrayAnim = SelectionSort(this.state.arrayToSort);
74     let arrayToSort = this.state.arrayToSort;
75     let prevChanged = this.state.prevChanged;
76
77     //loop through all the animations
78     for (let index = 0; index < sortedArrayAnim.length; index++) {
79         const [i,j, swap] = sortedArrayAnim[index];
80
81         //change array
82         if(swap){
83             let temp = arrayToSort[i];
84             arrayToSort[i] = arrayToSort[j];
85             arrayToSort[j] = temp;
86         }
87
88         prevChanged.push(i,j);
89
90         if(index == sortedArrayAnim.length - 1){
91             prevChanged.push(arrayToSort.length + 1, arrayToSort.length + 1);
92             this.setState({prevChanged});
93         }
94
95         this.setState({ arrayToSort, prevChanged });
96
97         await sleep(this.state.delay);
98     }
99 }
100
101 handleItemsInputChange(event){
102     event.persist();
103     this.setState({numberOfItems : event.target.value}, () => {
104         this.resetArray();
105         console.log(event.target.value + " - " + this.state.numberOfItems + " - arraySize: " + this.state.arrayToSort.length);
106     });
107 }
108 }
```



```
SortingVisualizer.jsx X
Mini project > sorting > src > SortingVisualizer > SortingVisualizer.jsx > ...
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152

getColor(index){
    let prevChanged = this.state.prevChanged;
    if(prevChanged.includes(index)){
        if(index == prevChanged[prevChanged.length - 1] || index == prevChanged[prevChanged.length - 2]){
            return CHANGED_COLOR;
        }
        else{
            return AFTER_CHANGE_COLOR;
        }
    }
    else{
        return NORMAL_COLOR;
    }
}

render() {
    const {arrayToSort} = this.state;
    let widthValue = 40 / this.state.numberOfItems;
    return (
        <div className="main-div" id="centerdiv">
            <div className="centerdivKeepWidth" id="itemsDiv">
                {arrayToSort.map((heightValue, idx) => (
                    <div className="array-item" key={idx} style={{height: `${heightValue / 25}vw`, width: `${widthValue}vw`, backgroundColor: this.getColor(idx)}}>
                ))}
            </div>
            <div className="centerdivKeepWidth" id="buttonsDiv">
                <button onClick={() => this.generateNewArray()}>Generate new array</button>
                <button onClick={() => this.sortArray(BubbleSort)}>Bubble Sort</button>
                <button onClick={() => this.sortArray(InsertionSort)}>Insertion Sort</button>
                <button onClick={() => this.selectionSort()}>Selection Sort</button>
                <button onClick={() => this.sortArray(GetQuickSortAnimation)}>Quick Sort</button>
            </div>
        </div>
    );
}
```

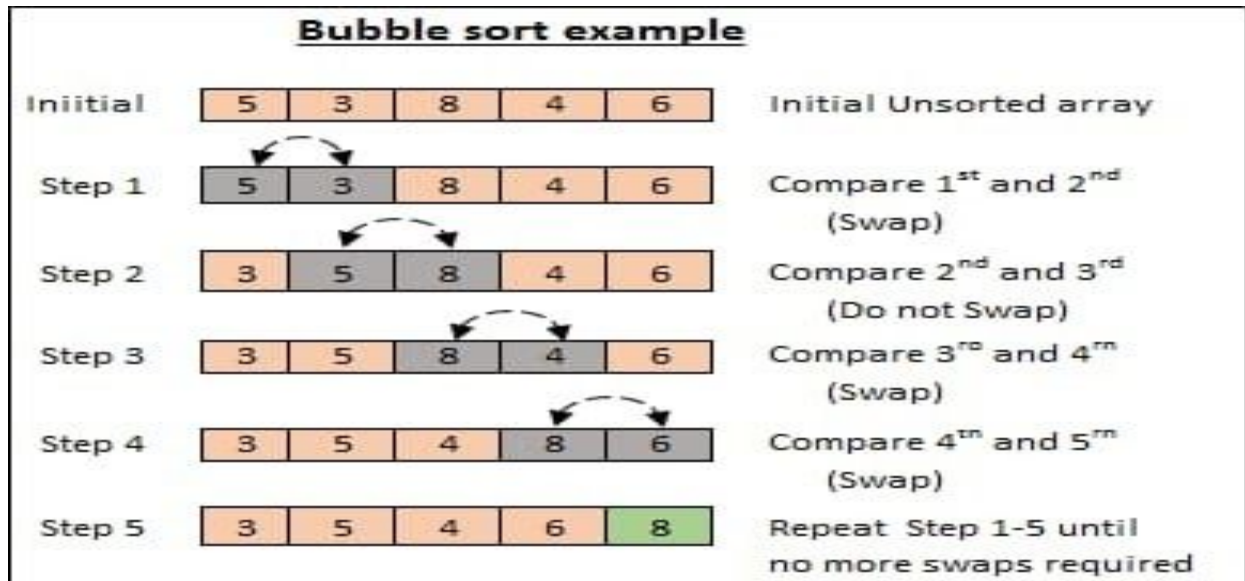
```
File Edit Selection View Go Run Terminal Help
SortingVisualizer.jsx - Mini-Project-Sorting-Visualizer-main - Visual Studio Code
SortingVisualizer.jsx X
Mini project > sorting > src > SortingVisualizer > SortingVisualizer.jsx > ...
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170

render() {
    const {arrayToSort} = this.state;
    let widthValue = 40 / this.state.numberOfItems;
    return (
        <div className="main-div" id="centerdiv">
            <div className="centerdivKeepWidth" id="itemsDiv">
                {arrayToSort.map((heightValue, idx) => (
                    <div className="array-item" key={idx} style={{height: `${heightValue / 25}vw`, width: `${widthValue}vw`, backgroundColor: this.getColor(idx)}}>
                ))}
            </div>
            <div className="centerdivKeepWidth" id="buttonsDiv">
                <button onClick={() => this.generateNewArray()}>Generate new array</button>
                <button onClick={() => this.sortArray(BubbleSort)}>Bubble Sort</button>
                <button onClick={() => this.sortArray(InsertionSort)}>Insertion Sort</button>
                <button onClick={() => this.selectionSort()}>Selection Sort</button>
                <button onClick={() => this.sortArray(GetQuickSortAnimation)}>Quick Sort</button>
            </div>
            <div className="flexDiv">
                <div className="centerdivKeepWidth">
                    <label className="numberLabel">Number of Items: </label>
                    <input className="numberInput" type="number" min="5" max="1500" onChange={(event) => this.handleItemsInputChange(event)} defaultValue={this.state.numberOfItems}>
                </div>
                <div className="centerdivKeepWidth">
                    <label className="numberLabel">Delay: </label>
                    <input className="numberInput" type="number" min="1" max="100" onChange={(event) => this.handleDelayInputChange(event)} defaultValue={this.state.delay}>
                </div>
            </div>
        </div>
    );
}

RandomIntBetweenRange(min, max){
    return Math.floor(Math.random() * (max - min + 1)) + min;
}
```

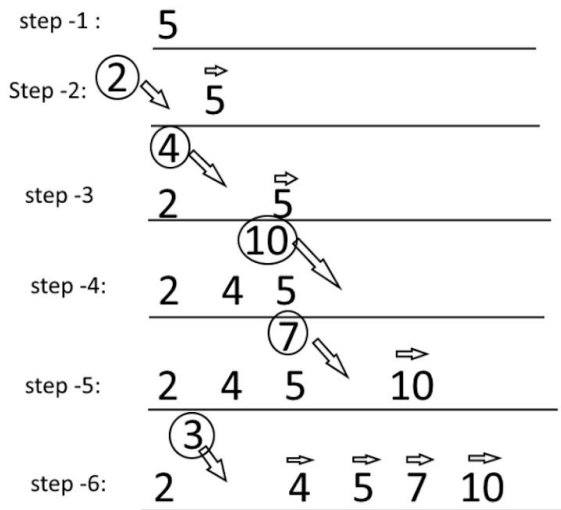
SortingAlgorithms:

Bubblesort.js



```
JS BubbleSort.js X
Mini project > sorting > src > SortingAlgorithms > JS BubbleSort.js > BubbleSort
1 export default function BubbleSort(array){
2   let arr = array.slice(0);
3   let animations = []
4   while(true){
5     let hasChanged = false;
6     for (let i = 0; i < arr.length; i++) {
7       if(arr[i] > arr[i + 1]){
8         let temp = arr[i];
9         arr[i] = arr[i + 1];
10        arr[i + 1] = temp;
11        animations.push([i, i + 1]);
12        hasChanged = true;
13      }
14    }
15    if(!hasChanged) break;
16  }
17  return animations;
18 }
19
20
21 }
```

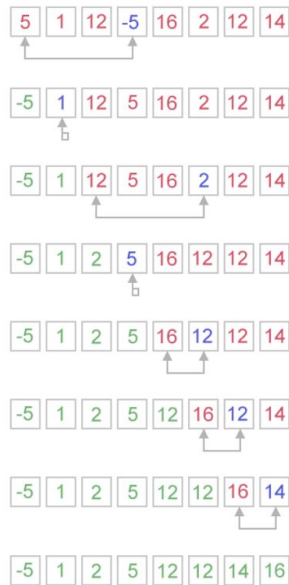
Insertionsort.js



```
Go Run Terminal Help InsertionSort.js - Mini-Project-Sorting-Visualizer-main - Visual Studio Code

JS InsertionSort.js X
Mini project > sorting > src > SortingAlgorithms > JS InsertionSort.js > InsertionSort
1 export default function InsertionSort(array){
2   let arr = array.slice(0);
3   let animations = [];
4   for (let index = 0; index < arr.length; index++) {
5
6     if(index != 0){
7       let currentIndex = index
8       while(arr[currentIndex] < arr[currentIndex - 1]){
9         if(currentIndex == 0){ break; }
10        let temp = arr[currentIndex];
11        arr[currentIndex] = arr[currentIndex - 1];
12        arr[currentIndex - 1] = temp;
13        animations.push([currentIndex, currentIndex - 1]);
14        currentIndex--;
15      }
16    }
17  }
18  return animations;
19 }
```

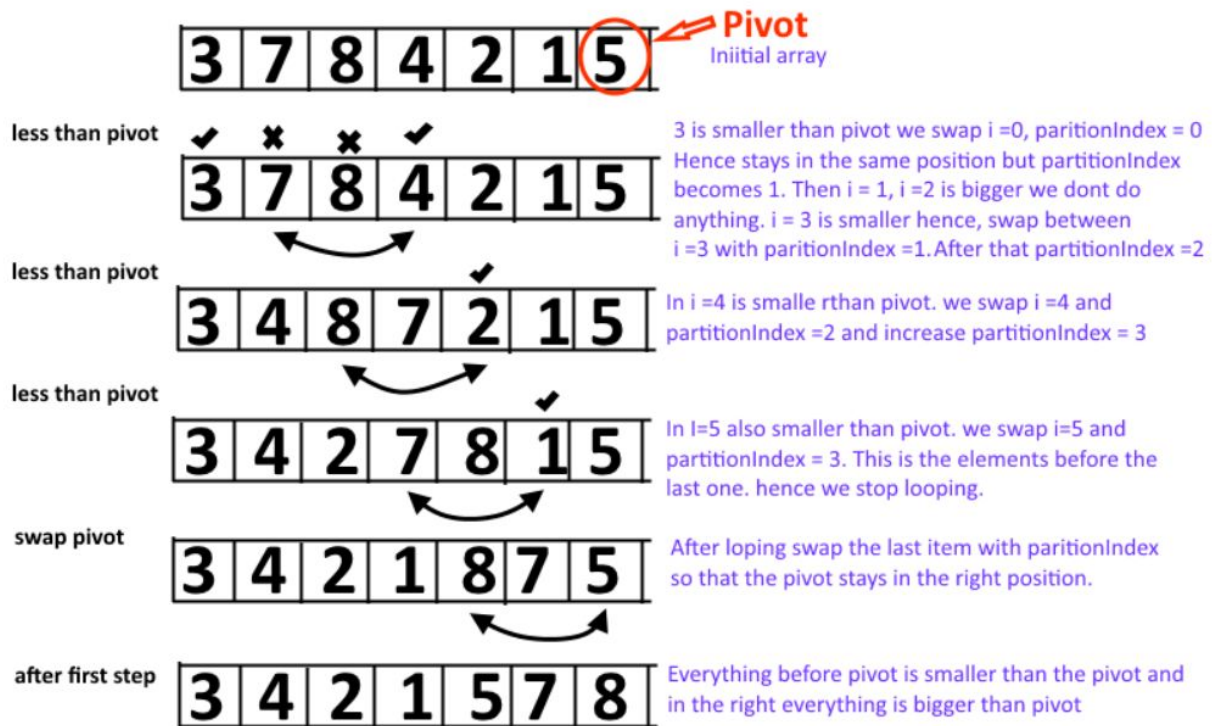
Selectionsort.js



```
Go Run Terminal Help SelectionSort.js - Mini-Project-Sorting-Visualizer-main - Visual Studio Code

JS SelectionSort.js X
Mini project > sorting > src > SortingAlgorithms > JS SelectionSort.js > SelectionSort
1 export default function SelectionSort(arr){
2   let array = arr.slice(0);
3   let animations = [];
4   for (let i = 0; i < array.length; i++) {
5     let minIndex = i;
6     for (let j = i; j < array.length; j++) {
7       if(array[j] < array[minIndex]){
8         minIndex = j;
9         animations.push([i, minIndex, false]);
10      }
11    }
12
13    let temp = array[i];
14    array[i] = array[minIndex];
15    array[minIndex] = temp;
16    animations.push([i, minIndex, true]);
17  }
18
19  return animations;
20 }
```

Quicksort.js



```
Go Run Terminal Help QuickSort.js - Mini-Project-Sorting-Visualizer-main - Visual Studio Code
JS QuickSort.js X
Mini project > sorting > src > SortingAlgorithms > JS QuickSort.js > ...
1
2 let animations = [];
3 export default function GetQuickSortAnimation(array){
4     animations = [];
5     let arr = array.slice(0);
6     QuickSort(arr, 0, arr.length - 1);
7     return animations;
8 }
9
10 function QuickSort(arr, low, high){
11
12     if(low < high){
13         let pi = Partition(arr, low, high);
14         QuickSort(arr, low, pi - 1);
15         QuickSort(arr, pi + 1, high);
16     }
17 }
18
19
20 function Partition(arr, low, high){
21     let i = low - 1; //smaller element
22     for (let index = low; index <= high - 1; index++) {
23         if( arr[index] <= arr[high]){
24             i++;
25             //swap
26             let temp = arr[i];
27             arr[i] = arr[index];
28             arr[index] = temp;
29             animations.push([i, index]);
30         }
31     }
32     let temp = arr[i + 1];
33     arr[i + 1] = arr[high];
34     arr[high] = temp;
35     animations.push([i + 1, high]);
36     return i + 1;
37 }
```

package.json

```
package.json X
Mini project > sorting > package.json > ...
1 {
2   "name": "sorting",
3   "version": "0.1.0",
4   "private": true,
5   "dependencies": {
6     "@testing-library/jest-dom": "^5.11.6",
7     "@testing-library/react": "^11.2.2",
8     "@testing-library/user-event": "^12.2.2",
9     "react": "^17.0.1",
10    "react-dom": "^17.0.1",
11    "react-scripts": "4.0.1",
12    "web-vitals": "^0.2.4"
13  },
14  "scripts": {
15    "start": "react-scripts start",
16    "build": "react-scripts build",
17    "test": "react-scripts test",
18    "eject": "react-scripts eject"
19  },
20  "eslintConfig": {
21    "extends": [
22      "react-app",
23      "react-app/jest"
24    ]
25  },
26  "browserslist": {
27    "production": [
28      ">0.2%",
29      "not dead",
30      "not op_mini all"
31    ],
32    "development": [
33      "last 1 chrome version",
34      "last 1 firefox version",
35      "last 1 safari version"
36    ]
37  }
}
```


Certificates:



Certificate of Completion

*This is to certify that **Siddharth Singh**
successfully completed 15.5 total hours of **React**
Certification Training (beginner to expert level)
2020 online course on Nov. 18, 2020*

Uplatz Training
Uplatz Training, Instructor

&



Certificate no: UC-d01f4f8f-6c70-4208-aa32-65d717f6614e
Certificate url: ude.my/UC-d01f4f8f-6c70-4208-aa32-65d717f6614e
Version 3

#BeAble

Certificate of Completion

This is to certify that **Sparsh gupta** successfully
completed 2 total hours of **JavaScript Course:
Complete Guide (Step by Step)** online course on
Sept. 11, 2020

Fatah Gabriel
Fatah Gabriel, Instructor

&



Certificate no: UC-0c23e323-d477-433e-a7fa-b12355c#1b9
Certificate url: ude.my/UC-0c23e323-d477-433e-a7fa-b12355c#1b9

#BeAble

Certificate of Completion

*This is to certify that **SURYANSHU Gupta**
successfully completed 15.5 total hours of **React**
Certification Training (beginner to expert level)
2020 online course on Nov. 25, 2020*

Uplatz Training
Uplatz Training, Instructor

&



Certificate no: UC-45975f4c-efce-476a-b753-a731765782a9
Certificate url: ude.my/UC-45975f4c-efce-476a-b753-a731765782a9
Version 3

#BeAble

References :

Udemy

<https://www.udemy.com/courses/search/?src=ukw&q=mern+stack>

Stackoverflow

<https://stackoverflow.com/>

React js Documentation

<https://reactjs.org/docs/getting-started.html>

<https://khan4019.github.io/front-end-Interview-Questions/sort.html#quickSort>

W3school

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