"ALGORITHMS VISUALIZER"

A

Project Report

submitted

in partial fulfillment

for the award of the Degree of

Bachelor of Technology

in Department of Computer Science and Engineering

Project Mentor:
Name: Ankit Kumar
Designation: Assistant Prof.-1

Submitted By :
Ayush Agrawal
17ESKCS043

Department of Computer Science and Engineering
Swami Keshvanand Institute of Technology, M & G, Jaipur
Rajasthan Technical University, Kota
Session 2020-2021

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that Mr. Ayush Agrawal, a student of B.Tech(Computer Science & Engineering) VII semester has submitted his Project Report entitled ”Algorithm Visualizer” under my guidance.

Mentor

Ankit Kumar

Assistant Prof.-1

Signature

Acknowledgement

It is my pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behavior, and acts during the course of study.

I express my sincere gratitude to Dr. Mukesh Kumar Gupta, HOD, (Department of Computer Science Engineering) for providing me an opportunity to work in a consistent direction and providing all necessary means to complete my presentations and report thereafter.

I would like to thank my esteemed supervisor Mr. Ankit Kumar, Department of Computer Science & Engineering, Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur for his valuable suggestions, keen interest, constant encouragement, incessant inspiration and continuous help throughout this work. His excellent guidance has been instrumental in making this work a success.

I express my sincere heartfelt gratitude to all the staff members of Department of Computer Science & Engineering who helped me directly or indirectly during this course of work.

I would like to give a special thanks to my parents Mr. Ashesh Vaish and Mrs. Neha Vaish for giving me support, motivation and blessings. A special thank goes to all my friends for their support in completion of this work.

AYUSH AGRAWAL
17ESKCS043

Contents

List of Figures

Chapter 1

Introduction

1.1 OVERVIEW

The project is an algorithm visualizer. That is, the project is designed in order to show the working of algorithms behind the scenes with help of animations. So far, Sorting and Path-Finding algorithms of graphs have been implemented in the project.

1.2 MOTIVATION

It is very difficult to understand how an algorithm is working behind the scenes, and to analyze it on paper is very difficult.

Thus as a student of Computer Science I found a need to develop a visulizer that can graphically show how an algorithm works in an easy way. This need motivated me to choose and build this project.

1.3 OBJECTIVES

- To develop an easy to understand and minimal UI.
- To integrate algorithms that are difficult to analyze on paper.
- To implement animations in such a way that they are smooth and easy to understand.

1.4 POTENTIAL AUDIENCE

- Anyone interested in learning algorithms.
- Someone facing difficulty in understanding the flow of algorithms.
- Someone who has interest in algorithms.

Chapter 2

KNOWLEDGE BASE

2.1 ALGORITHMS

2.1.1 Selection Sort

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

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

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

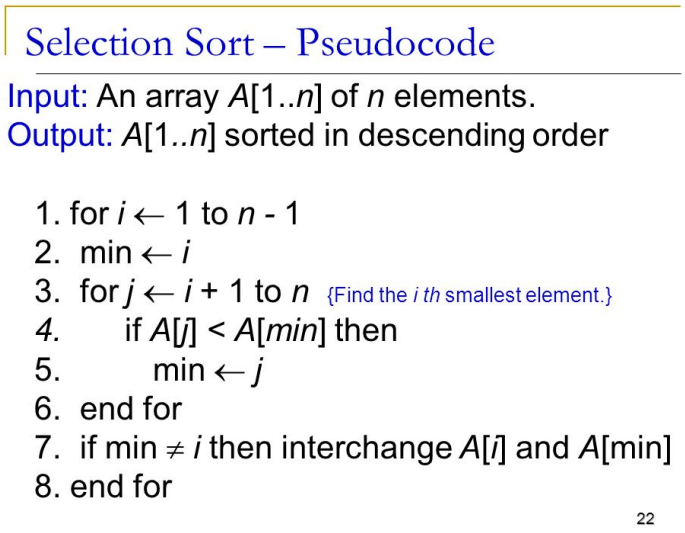

Figure 2.1: Selection Sort

2.1.2 Bubble Sort

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

```
FOR iPass = 1 to Length(ArrayToSort) - 1
    FOR i = 0 to Length(ArrayToSort) - 1 - iPass
        IF ArrayToSort(i) > ArrayToSort(i + 1) THEN
            Temp = ArrayToSort(i)
            ArrayToSort(i) = ArrayToSort(i + 1)
            ArrayToSort(i + 1) = Temp
        END IF
    NEXT i
NEXT iPass
```

Figure 2.2: Bubble Sort

2.1.3 Breadth First Search

Breadth First Traversal (or Search) for a graph is similar to Breadth First Traversal of a tree (See method 2 of this post). The only catch here is, unlike trees, graphs may contain cycles, so we may come to the same node again. To avoid processing a node more than once, we use a boolean visited array.

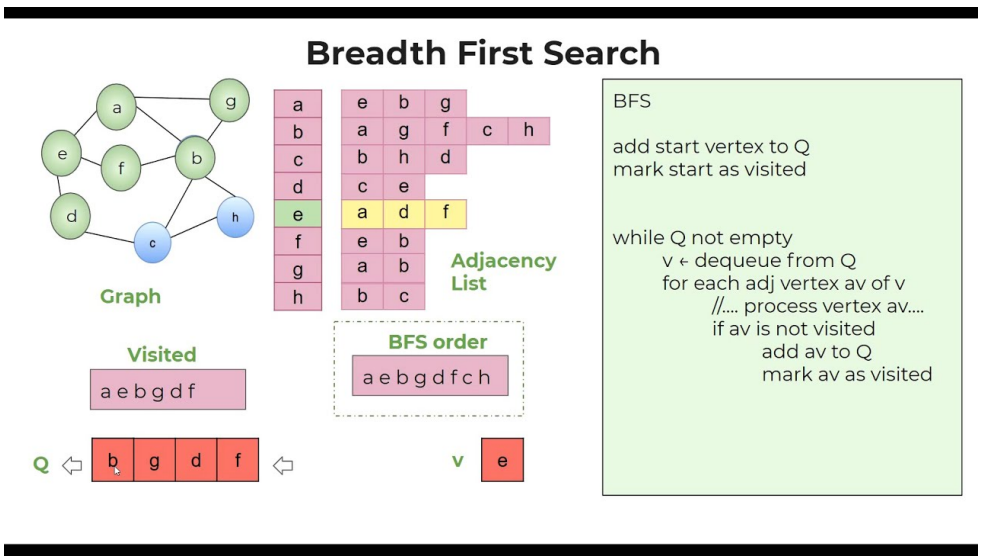

Figure 2.3: BFS

2.1.4 Depth First Search

Depth First Traversal (or Search) for a graph is similar to Depth First Traversal of a tree. The only catch here is, unlike trees, graphs may contain cycles, a node may be visited twice. To avoid processing a node more than once, use a boolean visited array.

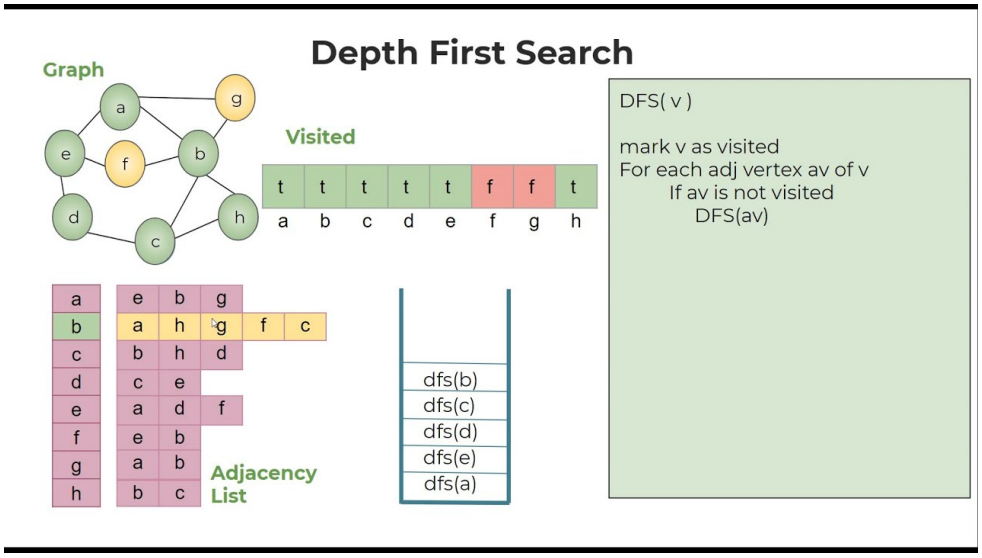

Figure 2.4: DFS

Chapter 3

Methodology

3.1 SDLC

Figure 3.1: SDLC

Software Development Life Cycle, or SDLC, widely used in the software industry as a framework to define tasks performed in each step of the development process. The goal of SDLC is to provide the best product with the resources given.

3.2 AGILE DEVELOPMENT MODEL

Agile Development Model is one of the two most popular SDLC models, with the other being the Waterfall Development Model. In opposition to the Waterfall method, the agile method focusses on process adaptability and customer satisfaction by rapid delivery of working software product.

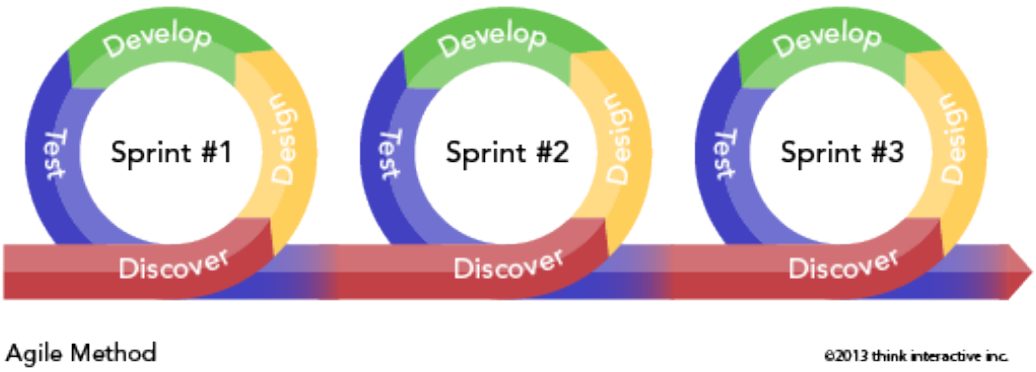

Figure 3.2: Agile Model

The Agile model is an incremental process model with multiple iterations. Each iteration consists of the fundamental SDLC phases with minor modifications. These phases are planning, designing, building, testing, reviewing and launching the product. A typical iteration can last from one to three weeks. Every iteration involves cross-functional teams working simultaneously in their own areas. (Tutorials Point)

Agile uses an adaptive approach where there is no detailed planning. It only clarifies on what feature need to be developed next and what task is required to complete that feature. Development teams can adapt to a product with dynamic requirements. The product is tested, reviewed by peers and customers, thus greatly reduce the risks of any major failures. (Tutorials Point)

The rationale for selecting Agile as the development methodology is due to its flexible nature. The Buddify CMS project has a dynamic requirement as new functions can be added at any time. The company can also shift feature prioritization causing the development team to change tasks.

Chapter 4

Requirements

The following section defines the requirements for the project. This is the Defining stage of the SDLC.

Extensibility is a design principle that takes future growth into consideration with each implementation. In software engineering, not everything can be designed in advance. Embracing this, extensibility design emphasizes properties that minimize the effort required to modify or extend the system. The three most important properties of extensibility are modifiability, scalability, and maintainability.

Modifiability is the ease with which a software system can be modified. Modifiability can be determined by how each functionality is separated and architecturally organized, and by how coding techniques are applied. A system has high modifiability when changes to the system require as few changes in each related component as possible. (Bass et al. 2003).

Maintainability is the effort required to locate and fix an error in an operational software. Maintainability is similar to Modifiability in definition. The essential difference between them is that maintainability is also takes the correction of bugs into consideration whereas modification is not.

Scalability as defined by Bondi is "the ability of a system to expand in a chosen dimension without major modifications to its architecture" (Bondi 2000). Scalability can be measured by how the system can add new functionalities or new components with minimal effort, and without having to undertake a major redesign.

Chapter 5

Introduction to Project / Modules

5.1 ABOUT WEB DEVELOPMENT

Web development is the building and maintenance of websites; it's the work that happens behind the scenes to make a website look great, work fast and perform well with a seamless user experience. Web developers, or 'devs', do this by using a variety of coding languages. The languages they use depends on the types of tasks they are preforming and the platforms on which they are working. Web development skills are in high demand worldwide and well paid too – making development a great career option. It is one of the easiest accessible higher paid fields as you do not need a traditional university degree to become qualified.
The field of web development is generally broken down into front-end (the user-facing side) and back-end (the server side).

5.1.1 Front-End Development

A front-end dev takes care of layout, design and interactivity using HTML, CSS and JavaScript. They take an idea from the drawing board and turn it into reality. What you see and what you use, such as the visual aspect of the website, the drop down menus and the text, are all brought together by the front-end dev, who writes a series of programmes to bind and structure the elements, make them look good and add interactivity. These programmes are run through a browser.

5.1.2 Back-End Development

The back-end developer engineers what is going on behind the scenes. This is where the data is stored, and without this data, there would be no front-end. The back-end of the web consists of the server that hosts the website, an application for running it and a database to contain the data. The back-end dev uses computer

programmes to ensure that the server, the application and the database run smoothly together. This type of dev need to analyse what a company's needs are and provide efficient programming solutions. To do all this amazing stuff they use a variety of server-side languages, like PHP, Ruby, Python and Java.

5.1.3 Full Stack Development

If both Frontend and Backend development appeal to you, you could consider becoming a Full-Stack Developer. Full-stackers take care of both the front-end and the back-end, and need to know how the web works on all levels, in order to determine how the client- and server-sides will relate. Naturally working up to this level of expertise will take longer, as there is more to learn.

5.2 BASIC WEB DEVELOPMENT COMPONENTS

5.2.1 HTML

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

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

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. 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), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

5.2.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML.[1] 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, and reduce complexity and repetition in the structural content.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device. The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

5.2.3 SASS

Sass (short for syntactically awesome style sheets) is a style sheet language initially designed by Hampton Catlin and developed by Natalie Weizenbaum. After its initial versions, Weizenbaum and Chris Eppstein have continued to extend Sass with SassScript, a simple scripting language used in Sass files.

Sass is a preprocessor scripting language that is interpreted or compiled into Cascading Style Sheets (CSS). SassScript is the scripting language itself. Sass consists of two syntaxes. The original syntax, called ”the indented syntax,” uses a syntax similar to Haml. It uses indentation to separate code blocks and newline characters to separate rules. The newer syntax, ”SCSS” (Sassy CSS), uses block formatting like that of CSS. It uses braces to denote code blocks and semicolons to separate

lines within a block. The indented syntax and SCSS files are traditionally given the extensions .sass and .scss, respectively.

CSS3 consists of a series of selectors and pseudo-selectors that group rules that apply to them. Sass (in the larger context of both syntaxes) extends CSS by providing several mechanisms available in more traditional programming languages, particularly object-oriented languages, but that are not available to CSS3 itself. When SassScript is interpreted, it creates blocks of CSS rules for various selectors as defined by the Sass file. The Sass interpreter translates SassScript into CSS. Alternatively, Sass can monitor the .sass or .scss file and translate it to an output .css file whenever the .sass or .scss file is saved.

5.2.4 JavaScript

JavaScript often abbreviated as JS, is a high-level, interpreted scripting language that conforms to the ECMAScript specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it, 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.

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. The JSON serialization format, used to store data structures in files or transmit them across networks, is based on JavaScript

5.3 REACTJS

Figure 5.1: ReactJS

React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It's 'V' in MVC. ReactJS is an open-source, component-based front end library responsible only for the view layer of the application. It is maintained by Facebook. React uses a declarative paradigm that makes it easier to reason about your application and aims to be both efficient and flexible. It designs simple views for each state in your application, and React will efficiently update and render just the right component when your data changes. The declarative view makes your code more predictable and easier to debug.

5.3.1 Why React?

React's popularity today has eclipsed that of all other front-end development frameworks. Here is why:

- **Easy creation of dynamic applications:** React makes it easier to create dynamic web applications because it requires less coding and offers more functionality, as opposed to JavaScript, where coding often gets complex very quickly.
- **Improved performance:** React uses Virtual DOM, thereby creating web applications faster. Virtual DOM compares the components' previous states and updates only the items in the Real DOM that were changed, instead of updating all of the components again, as conventional web applications do.
- **Reusable components:** Components are the building blocks of any React application, and a single app usually consists of multiple components. These components have their logic and controls, and they can be reused throughout the application, which in turn dramatically reduces the application's development time.
- **Unidirectional data flow:** React follows a unidirectional data flow. This means that when designing a React app, developers often nest child components within parent components. Since the data flows in a single direction, it becomes easier to debug errors and know where a problem occurs in an application at the moment in question.
- **Small learning curve:** React is easy to learn, as it mostly combines basic HTML and JavaScript concepts with some beneficial additions. Still, as is the case with other tools and frameworks, you have to spend some time to get a proper understanding of React's library.
- **It can be used for the development of both web and mobile apps:** We already know that React is used for the development of web applications, but that's not all it can do. There is a framework called React Native, derived from React itself, that is hugely popular and is used for creating beautiful mobile applications. So, in reality, React can be used for making both web and mobile applications.
- **Dedicated tools for easy debugging:** Facebook has released a Chrome extension that can be used to debug React applications. This makes the process of debugging React web applications faster and easier.

5.3.2 Features of ReactJS

JSX - JavaScript Syntax Extension

JSX is a syntax extension to JavaScript. It is used with React to describe what the user interface should look like. By using JSX, we can write HTML structures in the same file that contains JavaScript code. This makes the code easier to understand and debug, as it avoids the usage of complex JavaScript DOM structures.

Virtual DOM

React keeps a lightweight representation of the "real" DOM in the memory, and that is known as the "virtual" DOM (VDOM). Manipulating real DOM is much slower than manipulating VDOM because nothing gets drawn on the screen. When the state of an object changes, VDOM changes only that object in the real DOM instead of updating all of the objects.

DOM (Document Object Model) treats an XML or HTML document as a tree structure in which each node is an object representing a part of the document. When the state of an object changes in a React application, VDOM gets updated. It then compares its previous state and then updates only those objects in the real DOM instead of updating all of the objects. This makes things move fast, especially when compared to other front-end technologies that have to update each object even if only a single object changes in the web application.

Performance

React uses VDOM, which makes the web applications run much faster than those developed with alternate front-end frameworks. React breaks a complex user interface into individual components, allowing multiple users to work on each component simultaneously, thereby speeding up the development time.

Extensions

React goes beyond simple UI design and has many extensions that offer complete application architecture support. It provides server-side rendering, which entails rendering a normally client-side only web application on the server, and then sends a fully rendered page to the client. It also employs Flux and Redux extensively in web application development. Finally, there is React Native, a popular framework derived from React, used to create cross-compatible mobile applications.

One-way Data Binding

React's one-way data binding keeps everything modular and fast. A unidirectional data flow means that when a developer designs a React app, they often nest child components within parent components. This way, a developer knows where and when an error occurs, giving them better control of the whole web application.

Debugging

React applications are easy to test due to a large developer community. Facebook even provides a small browser extension that makes React debugging faster and easier.

5.4 GIT-VERSION CONTROL

Git is a free and open source distributed version control system designed to handle all small and large projects. Git is a very essential tool for working in industry and collaborating. It is even important to work with open-source software projects and for making open-source contributions.

The course guides us on using git from both command line and graphical user interface. It starts from the very beginning to initialize the repository and goes all the way to making changes, making commits, branching, merging, resolving merge conflicts and working with remote repositories. The course also gives the knowledge of forking a remote repository, i.e. working on an open source project and contributing to it.

5.5 OVERVIEW OF PROJECT/MODULES

Project name is **Algorithm Visualizer** for visualizing the working of Sorting and Graph's Path Finding Visualizer. The project is built with following tchnologies:

1. ReactJS - For User Interface and Visualisation
2. Git - For version Control
3. Github Pages - For deployment of the project

The project has 2 parts **Sorting Algorithm Visualizer** and **Graph's PathFinding Algorithm Visualizer**.

5.5.1 Sorting Algorithm Visualizer

Project live at: https://ayushag99.github.io/Sorting-Visualizer/

Sorting Algorithm Visualizer is a project implemented in ReactJS to visualize the various sorting algorithms. So far Selection sort, Bubble Sort, Quicksort and Heap Sort are implemented and algorithms like Shell Sort and Radix Sort are yet to be implemented.

Working

As per the working of the project goes, it shows an array consisting of integer values arranged randomly, that is to be sorted. This array is showed graphically using bars with height in proportion to the values they consist of. Each bar in itself is a ReactJS component and their height are assigned when the first time the component loads in ratio of screen height, array element value and value of the maximum and minimum element in the array.

Further, there is a control bar at the bottom of the page to start, stop, resume, Forward and Rewind the animation playing. When the play button is pressed the animation starts playing. While playing the animation the elements being accessed by the algorithm are shown in red colour and the elements on which certain operation is being performed is being shown in blue colour for better understanding.

There are modules that stores the algorithms, once a particular algorithm is selected, that algorithm runs over the array and return a list of all the moves/swaps done. This list is received by the toolbar that then performs an animation with the same list.

5.5.2 PathFinding Algorithm Visualizer

Project live at: https://ayushag99.github.io/PathFinding-Visualizer/

PathFinding Algorithm Visualizer is a project implemented in ReactJS to visualize the various path finding algorithms. So far Breath First Search(BFS), Depth First Search(DFS), Dijkstra's Algorithms and Bellman-Ford Algorithms are implemented and algorithms like Swamp Search, Bidirectional Dijkstra's, Bidirectional BFS, Greedy BFS and A* Search are yet to be implemented.

Working

As per the working of the project goes, it has 2 parts one is a control bar at the top and another is the animation area. Once the algorithm and speed is selected, the animation can be started. The animation starts from the start node and searches for the end node depending on the algorithm chosen.

The animation area consists of a matrix where every cell in the matrix represents a node. These nodes are connected to 4 nodes(above, below, left and right if exists.). The animation first shows how the it spreads to search for the end point, and once end point is found then it shows the path that is found by the algorithm.

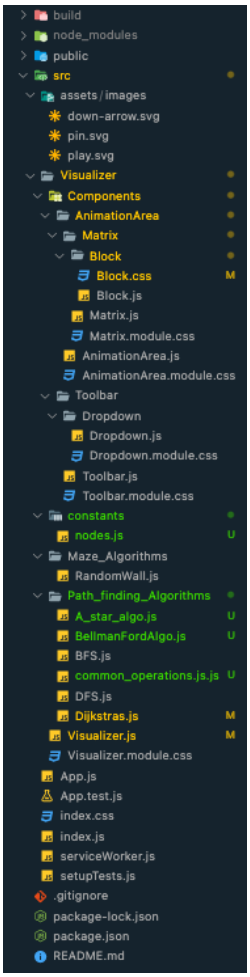

Figure 5.2: Folder Structure(Path-Finding Visualizer)

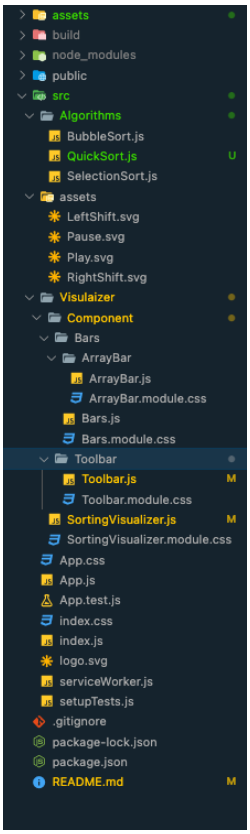

Figure 5.3: Folder Structure(Sorting-Algorithm Visualizer)

Chapter 6

Testing

Testing is the process of verifying that the program is working as intended. In this stage, each feature is checked for defects. Product defects are reported, tracked using Jira and fixed by the person responsible for the tested feature. The testing phase is repeated until the product reaches the standards defined in previous stages.

In this project, Testing is done in two stages: Unit testing during development and after deployed to the Dev server.

6.1 Unit Testing

UNIT TESTING is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected. Unit Testing is done during the development (coding phase) of an application by the developers

In unit Testing, all the algorithms were individually tested against randomly generated test data to ensure the algorithms are logically correct and generate correct output, so the algorithms can be individually correct.

Chapter 7

Results

7.1 RESULTS ACHIEVED

Results achieved after the completion of the web designing project were quite amazing. The resulting web pages were quite great and performed the task of visualizing the algorithms pretty well.

Both the sorting algorithm visualizer and pathfinding visualizer, visualized the algorithms smoothly. Both are very close to the idea with which the project was started. The UI of the project is very simple and easy to understand by the user and could be easily used by someone willing to visualize the algorithms. All the algorithms in spite of the complexity worked fine and came out exactly how they should be.

Moreover, the project gave me a better understanding of the react component lifecycle and how react functions overall. Even the project gave an idea of version control with git and how is git used at industry standard. In all building this project was a great learning experience of how a project is taken from start to end with industry standard tools.

7.2 SCREENSHOTS OF ASSIGNMENT/RESULT ACHIEVED

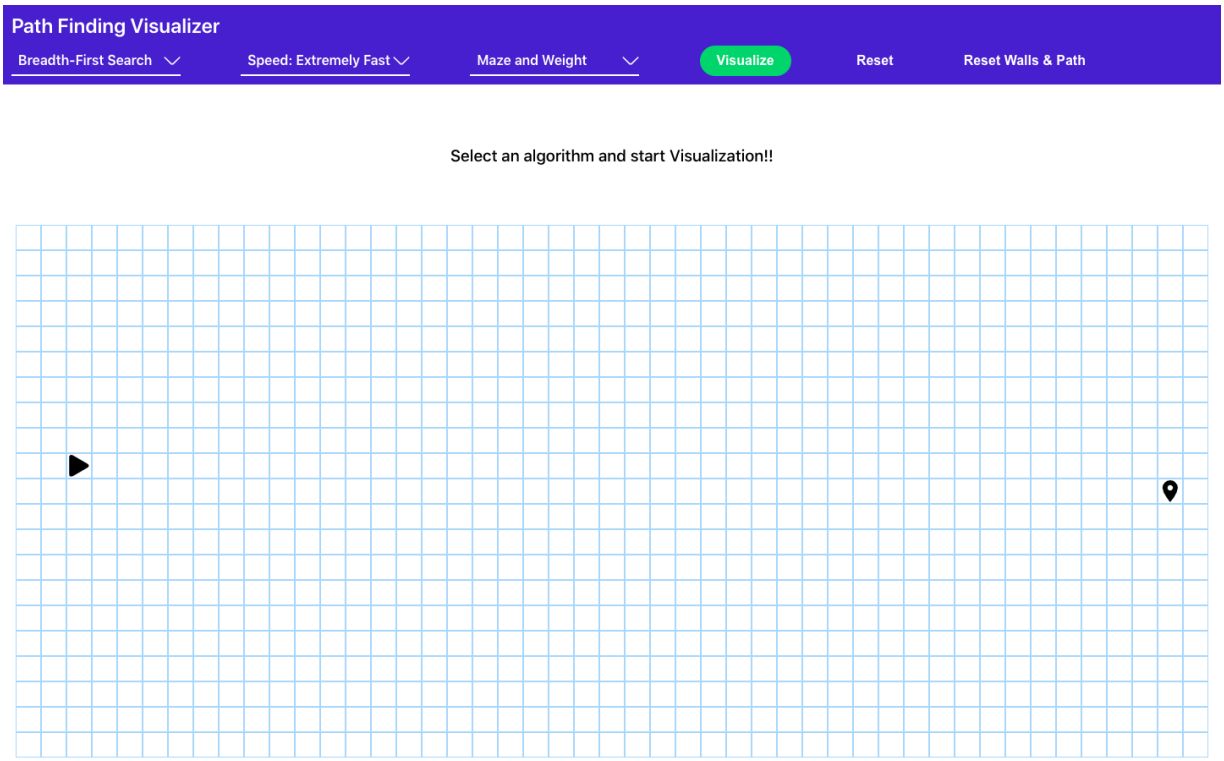

Figure 7.1: Pathfinding Visualizer-Initial View

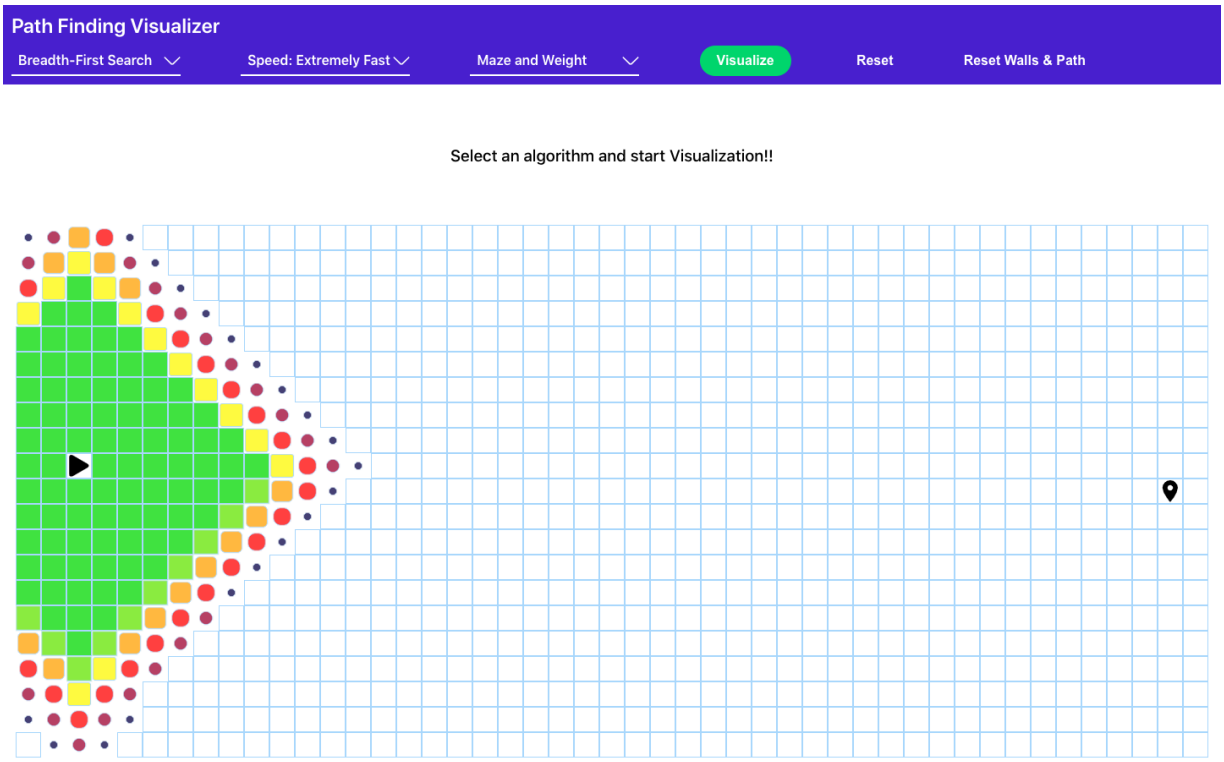

Figure 7.2: Pathfinding Visualizer-BFS algorithm in progress

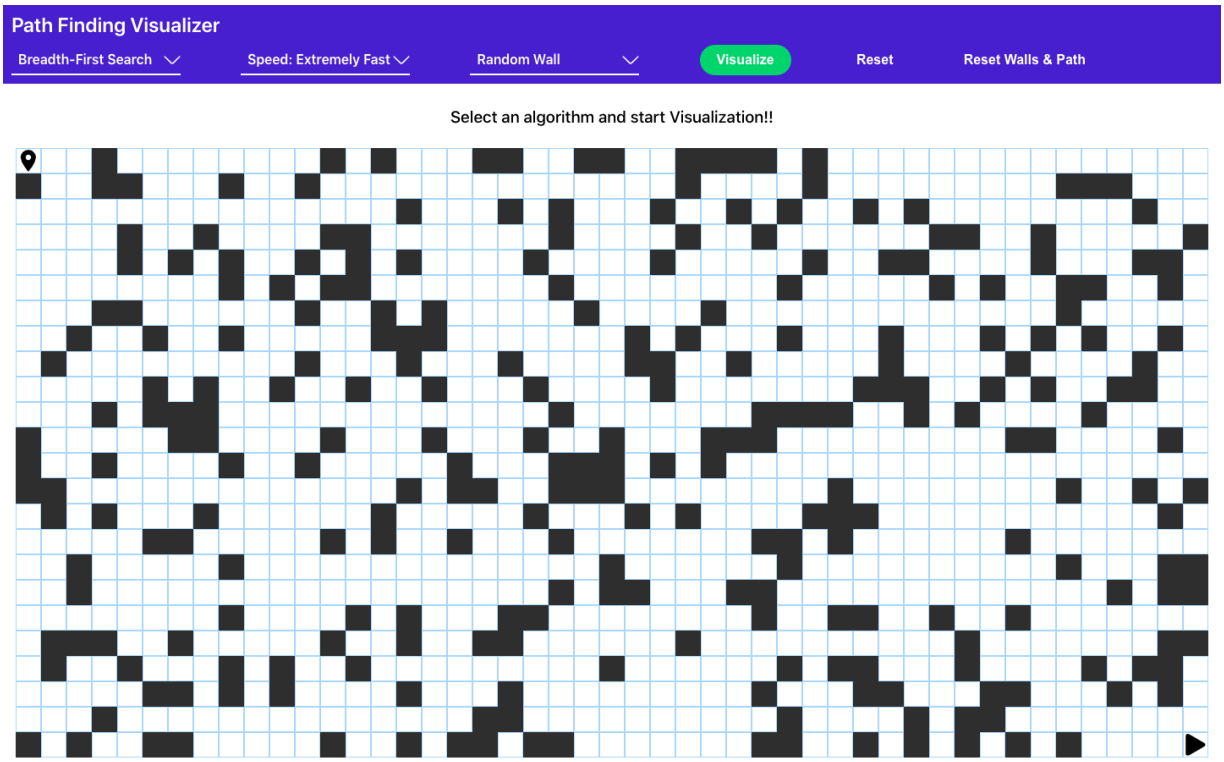

Figure 7.3: Pathfinding Visualizer-Walls initialized

Figure 7.4: Pathfinding Visualizer-BFS algorithm with walls

Figure 7.5: Pathfinding Visualizer-Shortest Path

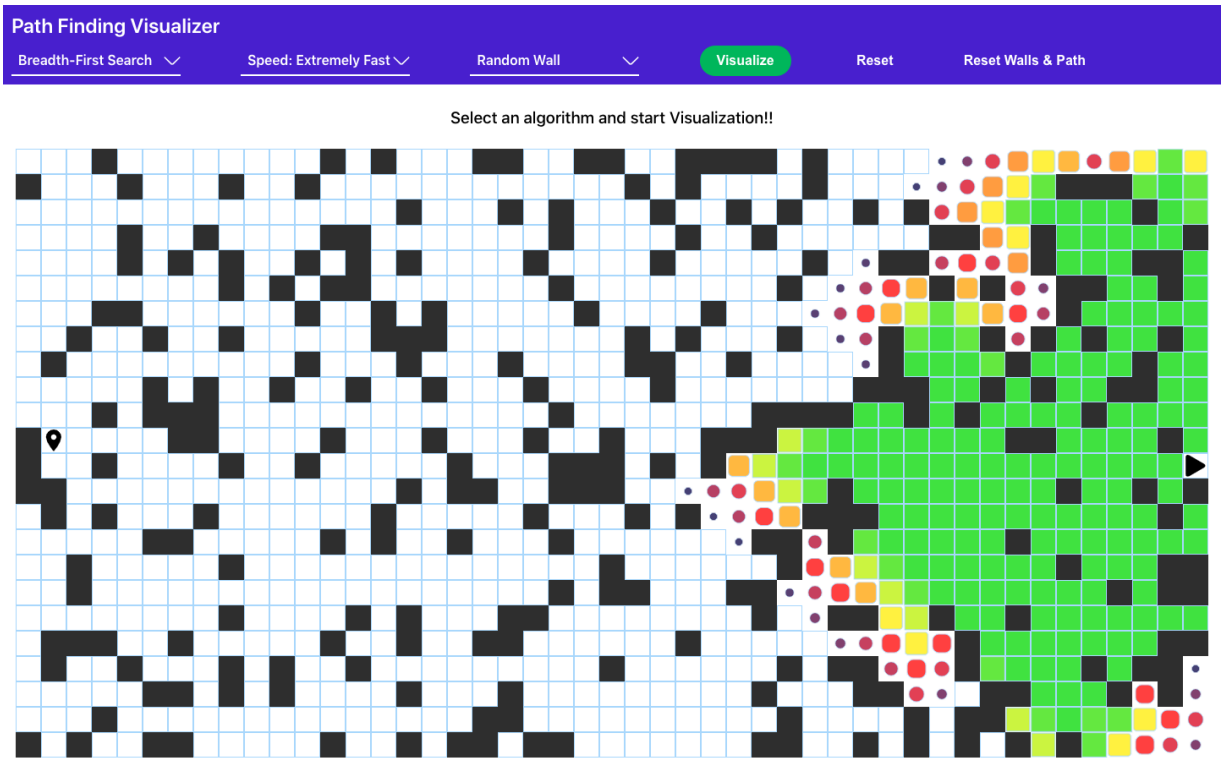

Figure 7.6: Pathfinding Visualizer-BFS Search in major obstruction

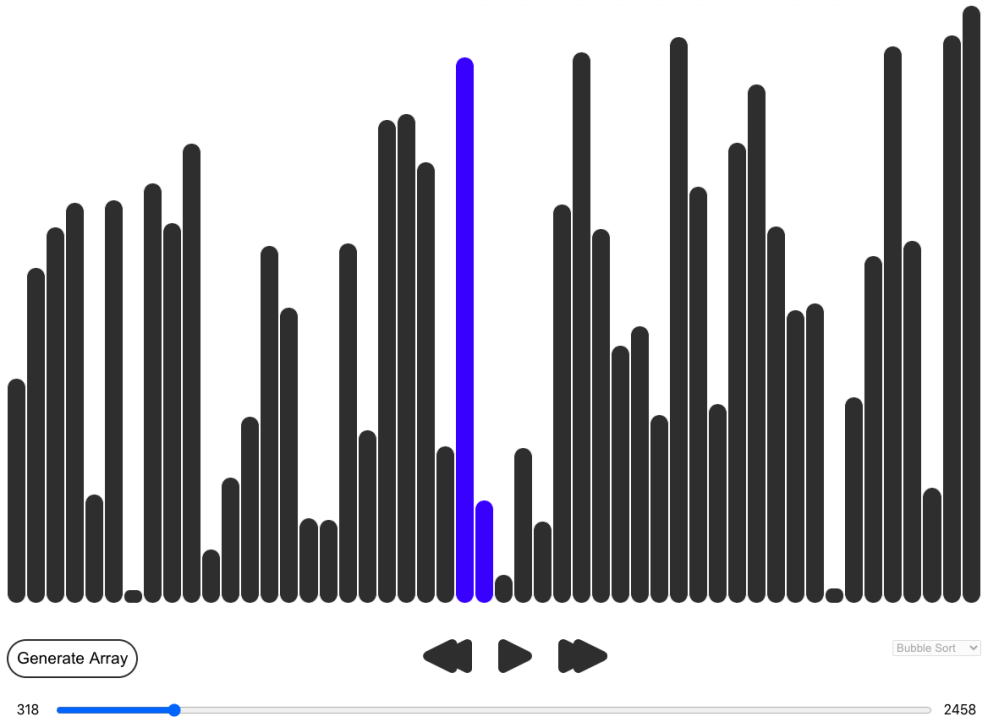

Figure 7.7: Sorting Visualizer-Bubble Sort in Progress-1

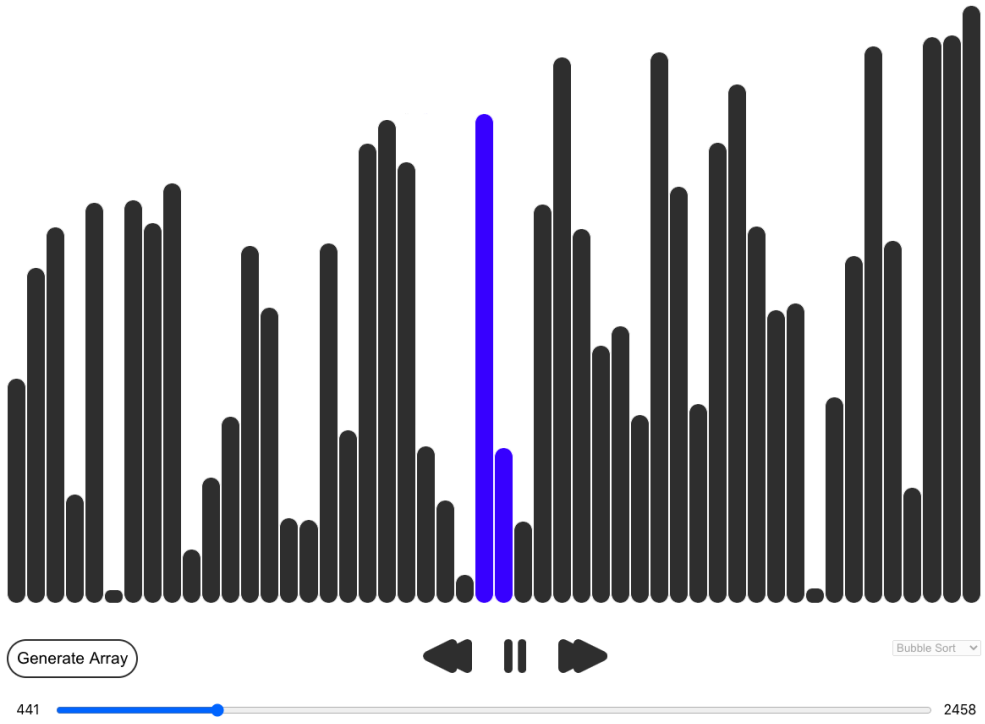

Figure 7.8: Sorting Visualizer-Bubble Sort in Progress-2

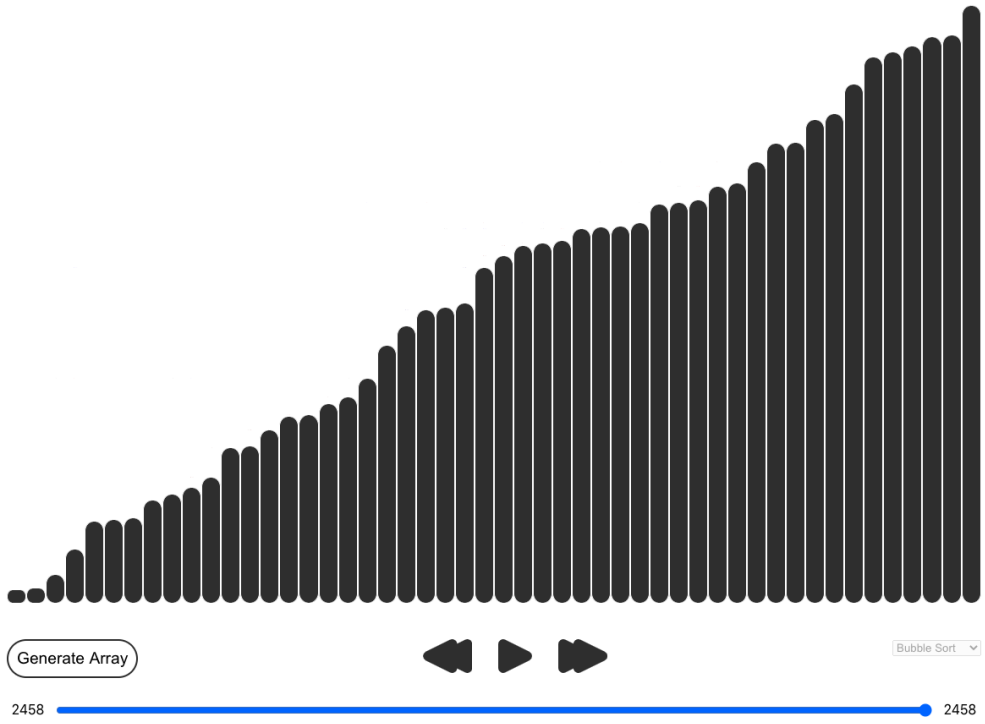

Figure 7.9: Sorting Visualizer-Sorted array by Bubble Sort

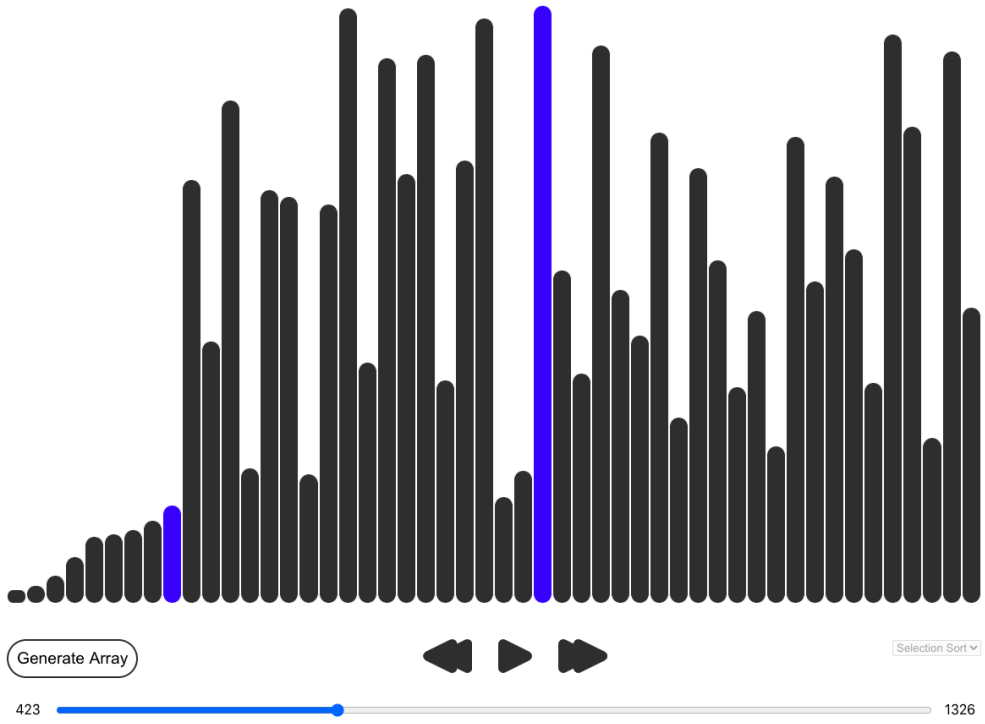

Figure 7.10: Sorting Visualizer-Selection Sort in Progress-1

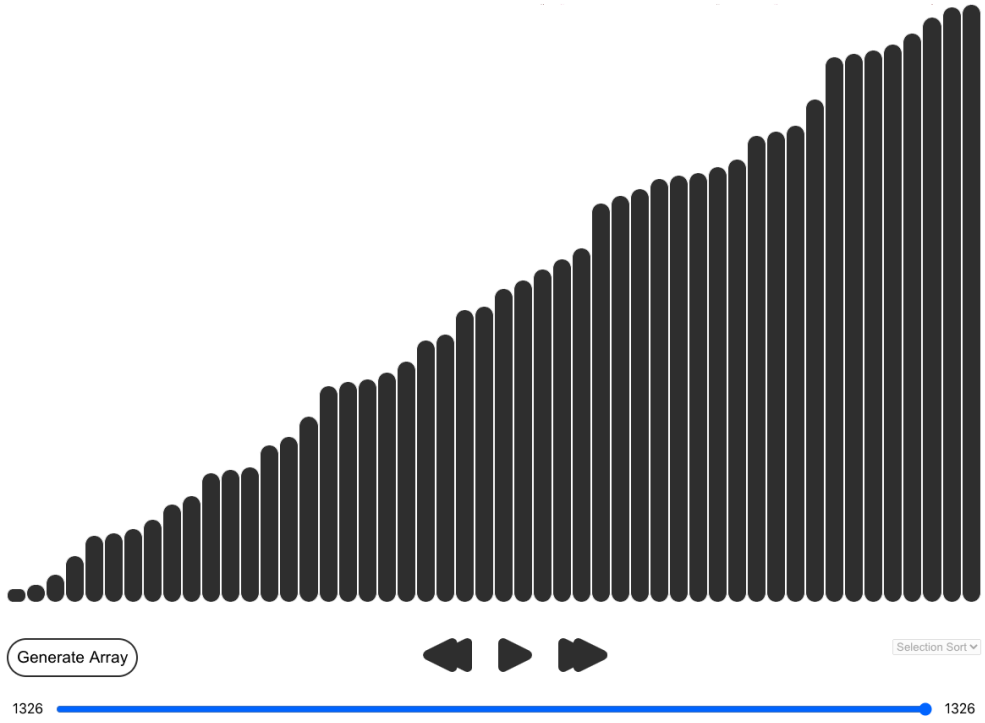

Figure 7.11: Sorting Visualizer-Sorted array by Selection Sort

Chapter 8

Outcomes

8.1 PROBLEMS AND SOLUTIONS

There are a lot of problems in the field of web development because it contains more type markup language, methods, objects, modifiers, and many ways to create code or project. Usually the problem is related to improper aligning of content or undesired shifts due to addition or removal of any content, or due to the results not matching up to the expectations. The solution to these problems are hidden in the field itself, but are noticeable after after getting some experience.

8.1.1 Problems

A major problem that I faced was with the speed of the animation. Since the project involves a lot of animation, there are a lot of elements that are getting updated every microsecond. In ReactJS whenever the state of a component changes it leads to that component getting re-rendered at the virtual DOM level to show the change. With the project especially the path finding visualizer, it becomes difficult to show the animation very fast as the state is huge and re-rendering it takes a lot of time, which makes the animation slow and the animation starts to stutter and fluctuate, resulting in a very bad user experience.

There are some other challenges which are faced in web designing. These challenges are about keeping a website responsive enough so that it can be viewed and accessed on all devices. Sometimes websites take time to load, so in order to provide a better user experience, one of the toughest challenges for web design is to make website load faster. It is a difficult task to manage all the content, features and loading time of the website all at once.

1. **The Problem with CSS:** A major problem with CSS is that, CSS gives val-

ues to properties of an element, sometimes 2 values are contradictory in their implementation leading to failure in creating the required designs. This either leads to loss of alignment of the element or improper positioning of the element, which spoils the overall design of the website

2. **Problem Of Responsive Design:** Responsive Designing leads to a lot of issues with the web pages. It requires the pages to be designed separately for different devices, so that the page can be loaded in more interactive way giving a better feel to the user. In many cases the responsive design ends up in messed up state due to lack of attention payed to designing the web page before actually designing the front-end part.

3. **Improving Loading Speeds:** When the website contains a lot of graphics or it is a responsive website, in these cases a lot of content is to be loaded to the browser, thus it affects the loading time of a web page by a great margin. This creates a very bad impact on the user and leads to a negative publicity.

Clients often encounter issues with their website and turn to the designer/developer to correct them, but when they check the site, it looks fine from their end. Sometimes, these issues are caused by out-of-date web servers or changes made incorrectly by the customer. Time spent detecting the source of a problem means a delay in tackling the problem itself.

8.1.2 Solutions

Really web development is an amazing field that manages to catch my interest very easily. This field is so amazing that it makes me search, brainstorm and do whatever it takes to solve the design related issues. But still there our some conventional solutions build to solve the most common challenges.

1. **Making animations faster**: To solve this we could use React refs that access teh element by conventional javascript way and make performing an operation on it faster.

2. **Graphical Designing:** UI designing is very useful to solve issues of responsive design and css contradictions. Wireframing and UI designing allows the designer to get a perfect idea of what the website should like on different devices with varying sizes. This creates a very rigid idea about the layout, space allocation, positioning, size, colour, and other attributes of different elements

on the website. After this these designs shall be referred constantly while coding the front-end part of the websites to ensure the code following the graphical UI design. This does not solve the problem completely but greatly reduces the possibility of the error.

3. **Increasing the loading speeds:** Including external libraries and scripts increases the loading time of the web page, since an extra request is to be sent by the browser to server demanding the resource and then loading it. Moreover in production version of code-base all comments, spaces and other non-useful stuff should be removed to reduce the size of code to be sent to browser. This process decreases the size of the file by almost 40

These are most common steps which experienced web developers follow to increase their speed and redusce the errors in their work.

Chapter 9

Conclusion

9.1 DISCUSSION

Internet plays important role in the business management. The algorithm visualizer website report is based on a simple tool thaat could make understanding of the algorithms better. This report mainly discusses the management of the project. It is needed to be developed with module formulation. To accomplish all the functionality of website, each module in necessary to completed with full accuracy. This report will give the idea about the procedures of development of the website.

The current work is the initial background report for the website development project. This report aims to provide a critical review of the relevant literature in the web field and also to describe key aspects of the methodology that will be applied throughout the project. This report tries to examine various issues that arise while building a website. It is important to highlight, that websites are not simple software artifacts. Mastering the necessary software skills and tools to build a website does not guarantee its success. For this reason, following the guidance and the advice of my supervising professor, this report focuses on many other challenges that come up through the development process, like performing website evaluation, conducting market research and choosing the right business model for the website proposal. These are the first and most crucial steps that will ensure that the final IT employment website, will be developed according to the requirements of the market and will be tailored to the needs of its users. Further research and more focus will be given on software tools after these parts of the methodology are complete and will be presented in later stages of the project. The decisions on how the website will be build depend on the results of the problem investigation stage, since they will play a major role into describing the specific user requirements for the software. The decision will be based on evaluating case studies of website development, in order

to examine a vast variety of techniques and software tools that have been applied successfully in real projects. Furthermore, further research will be conducted to spot the strong points of various techniques and justify the choice that will be finally made. Finally, an attempt will be made to contact some experienced web developers and ideally some web developers who have worked on similar projects, in order to absorb valuable knowledge from their experience.

9.2 FUTURE SCOPE

Future Scope and scalability of this project is immense. This project could be scaled by adding more and complex algorithms and visulalizing them. So this project could contribute in making understanding of algorithms better for people who prefer visual animations to understand the algorithm better.

Moreover a very important feature which I am working on right now and can become really useful is complexity analysis and comparison. In this feature 2 algorithms can be simultaneously played and compared in terms of their time complexities and advantages and disadvantages in different scenarios.