A

PROJECT WORK REPORT

on

"SIMULATION BASED LEARNING SYSTEM FOR DATA STRUCTURES AND ALGORITHMS"

submitted to the Savitribai Phule University, Pune

in fulfilment of the requirements for the award of the degree

B. E. (INFORMATION TECHNOLOGY)

BY

Mr. KATKHADE DATTATRAYA JANKIRAM	**SEAT NO. B150108547**
Miss. PURANIK RITUJA YOGESH	**SEAT NO. B150108561**
Miss. JAWALE SWAPNALI RAOSAHEB	**SEAT NO. B150108539**
Mr. SHAIKH OWES NOORMOHMMAD	**SEAT NO. B150108568**

Under the Guidance of

Prof. GAWALI A. D.

DEPARTMENT OF INFORMATION TECHNOLOGY

AMRUTVAHINI COLLEGE OF ENGINEERING, SANGAMNER

A/P: GHULEWADI, SANGAMNER, AHMEDNAGAR, PIN - 422608

ACADEMIC YEAR 2021-22

DEPARTMENT OF INFORMATION TECHNOLOGY

AMRUTVAHINI COLLEGE OF ENGINEERING, SANGAMNER

A/P: GHULEWADI, SANGAMNER, AHMEDNAGAR, PIN - 422608

ACADEMIC YEAR 2021-22

CERTIFICATE

This is to certify that Project report entitled

"SIMULATION BASED LEARNING SYSTEM FOR DATA STRUCTURES AND ALGORITHMS"

is submitted as fulfilment of curriculum of the B.E. of Information Technology

by

Mr. KATKHADE DATTATRAYA JANKIRAM	**SEAT NO. B150108547**
Miss. PURANIK RITUJA YOGESH	**SEAT NO. B150108561**
Miss. JAWALE SWAPNALI RAOSAHEB	**SEAT NO. B150108539**
Mr. SHAIKH OWES NOORMOHMMAD	**SEAT NO. B150108568**

Prof. Gawali A. D.
Project Guide

Dr. Markad A. V.
Project Coordinator

Dr. Gunjal B. L.
H.O.D. I.T.

Dr. Venkatesh M. A.
Principal

Savitribai Phule Pune University

CERTIFICATE

This is to certify that

Mr. KATKHADE DATTATRAYA JANKIRAM	**SEAT NO. B150108547**
Miss. PURANIK RITUJA YOGESH	**SEAT NO. B150108561**
Miss. JAWALE SWAPNALI RAOSAHEB	**SEAT NO. B150108539**
Mr. SHAIKH OWES NOORMOHMMAD	**SEAT NO. B150108568**

Students of B.E. Information Technology was examined in Project entitled

"SIMULATION BASED LEARNING SYSTEM FOR DATA STRUCTURES AND ALGORITHMS"

on

/ / 2022

at

DEPARTMENT OF INFORMATION TECHNOLOGY

AMRUTVAHINI COLLEGE OF ENGINEERING, SANGAMNER

YEAR 2021-22

---------- ----------

(Internal Examiner) **(External Examiner)**

Certificate by Guide

This is to certify that

Mr. KATKHADE DATTATRAYA JANKIRAM	**SEAT NO. B150108547**
Miss. PURANIK RITUJA YOGESH	**SEAT NO. B150108561**
Miss. JAWALE SWAPNALI RAOSAHEB	**SEAT NO. B150108539**
Mr. SHAIKH OWES NOORMOHMMAD	**SEAT NO. B150108568**

has completed the Project work entitled "**Simulation Based Learning System for Data Structures and Algorithms"** under my guidance and that, I have verified the work for its originality in documentation, problem statement, literature survey and conclusion presented in project work.

Date - / / 2022 **Prof. Gawali A. D.**

Place - Sangamner **Department of I.T**

AVCOE, Sangamner

ACKNOWLEDGEMENT

Apart from individual efforts, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental throughout the project work.

It is our privilege to express our gratitude towards our project guide **Prof. Gawali A. D.**, for their valuable guidance, encouragement, inspiration, and whole-hearted cooperation throughout the project work. We thank him for being a motivation through all our highs and importantly, our lows.

We deeply express our sincere thanks to our Head of Department **Dr. Gunjal B. L.** for encouraging and allowing us to present the project on the topic "Simulation Based Learning System for Data Structures and Algorithms" and providing us with the necessary facilities to enable us to fulfil our project requirements as best as possible. We take this opportunity to thank all faculty members and staff of the Department of Information Technology, who have directly or indirectly helped our project.

We pay our respects to honourable Principal **Dr. Venkatesh M. A.** for their encouragement. Our thanks and appreciations also go to our family and friends, who have been a source of encouragement and inspiration throughout the project.

Mr. KATKHADE DATTATRAYA JANAKIRAM
Ms. PURANIK RITUJA YOGESH
Ms. JAWALE SWAPNALI RAOSAHEB
Mr. SHAIKH OWES NOORMOHMMAD
AVCOE, Sangamner-422608

ABSTRACT

To understand the data structures and algorithms, we often employ some kind of visualization like drawing it on paper, use of images and videos for understanding purposes. But while going through only pseudo-code and explanation of algorithm and trying to imagine how it works it becomes difficult and frustrating sometimes.

Visualizations are always better as compared to textual data for understanding purposes. The same problem can be resolved using the web application AlgoViz which will visualize data structures and algorithms in the form of animations. This will help teachers and students to identify problematic areas/steps and work on improving them.

Keywords: Visualization, Data Structure, AlgoViz, Pseudo-Code

INDEX

LIST OF FIGURES

LIST OF ABBREVIATIONS

AlgoViz	Algorithm Visualization
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JS	Java Script

Chapter 1

INTRODUCTION

1.1 Background:

In the mathematical and empirical analyses of algorithms, there is yet a third way to study algorithms. It is called ***algorithm visualization*** and can be defined as the use of images to convey some useful information about algorithms. That information can be a visual illustration of an algorithm's operation, of its performance on different kinds of inputs, or of its execution speed versus that of other algorithms for the same problem. To accomplish this goal, algorithm visualization uses graphic elements—points, line segments, two- or three-dimensional bars, and so on—to represent some "interesting events" in the algorithm's operation.

There are two principal variations of algorithm visualization:

1] Static algorithm visualization

2] Dynamic algorithm visualization, also called algorithm animation

Static algorithm visualization shows an algorithm's progress through a series of still images. Algorithm animation, on the other hand, shows a continuous, movie-like presentation of an algorithm's operations. Animation is an arguably more sophisticated option, which, of course, is much more difficult to implement.

Early efforts in the area of algorithm visualization go back to the 1970s. The watershed event happened in 1981 with the appearance of a 30-minute colour sound film titled *Sorting Out Sorting*. This algorithm visualization classic was produced at the University of Toronto by Ronald Baecker with the assistance of D. Sherman [Bae81, Bae98]. It contained visualizations of nine well-known sorting algorithms (more than half of them are discussed later in the book) and provided quite a convincing demonstration of their relative speeds.

1.2 Importance of the project:

There are two principal applications of algorithm visualization: research and education. Potential benefits for researchers are based on expectations that algorithm visualization may help uncover some unknown features of algorithms. For example, one researcher used a visualization of the recursive Tower of Hanoi algorithm in which odd- and even-numbered disks were colored

in two different colors. He noticed that two disks of the same color never came in direct contact during the algorithm's execution. This observation helped him in developing a better non-recursive version of the classic algorithm. To give another example, Bentley and McIlroy mentioned using an algorithm animation system in their work on improving a library implementation of a leading sorting algorithm.

The application of algorithm visualization to education seeks to help students learning algorithms. The available evidence of its effectiveness is decisively mixed. Although some experiments did register positive learning outcomes, others failed to do so. The increasing body of evidence indicates that creating sophisticated software systems is not going to be enough. It appears that the level of student involvement with visualization might be more important than specific features of visualization software. In some experiments, low-tech visualizations prepared by students were more effective than passive exposure to sophisticated software systems.

1.3 Motivation:

Many students find it difficult to understand how data structures work because it requires abstract thinking. Imagining algorithms only with the help of textual data is time-consuming and confusing many times. There are a wide variety of algorithm visualizers which deal with some specific algorithm.

It would be very helpful if there was a one-stop visualization tool of data structures such as arrays, queues, stacks, trees, and graphs for students to manipulate. This motivates us to bring the idea that overcomes this problem by providing them a simple solution.

1.4 Aim:

To design and develop a web-based learning system to visualize animation of computer algorithms like stack, queue, binary search tree, searching and sorting algorithms and to understand the concept of data structure and algorithms it has some test series.

1.4 Objectives:

- To visualize basic data structures stack, queue, and binary search tree

- To visualize sorting and searching algorithms
- To analyses understanding with the help of the test section
- To provide a one-stop solution for visualizing data structures and algorithms
- To enhance traditional learning of complex concepts

1.5 Scope:

The rundown of presently available modules includes visualizations of searching algorithms like linear search, binary search, jump search, and some sorting algorithms like the bubble sort, insertion sort, selection sort, and Quick sort. Visualization for basic data structures like the stack, queue, and binary search tree is also available.

The extension can be made on different visual views on running algorithm or simultaneous comparison of different algorithm visualizations.

We can add some more complex data structures like heap, priority queue, segment tree, etc. The future implementation will include visualization of complex graph algorithms and greedy algorithms like job sequencing, etc. Also, we can implement a system in which the system can check its knowledge opposite to system-generated implementation.

The rundown of presently available modules includes visualizations of searching algorithms like linear search, binary search, jump search, and some sorting algorithms like the bubble sort, insertion sort, selection sort, and Quicksort. Visualization for basic data structures like the stack, queue, and binary search tree is also available.

The extension can be made on different visual views on running algorithm or simultaneous comparison of different algorithm visualizations.

We can add some more complex data structures like heap, priority queue, segment tree, etc. The future implementation will include visualization of complex graph algorithms and greedy algorithms like job sequencing, etc.

Chapter 2

LITERATURE SURVEY

1. "Creating Engaging Online Learning Material with the JSAV JavaScript Algorithm Visualization Library", Data Structures and Algorithms are a central part of Computer Science. Due to their abstract and dynamic nature, they are a difficult topic to learn for many students. To alleviate these learning difficulties, instructors have turned to algorithm visualizations (AV) and AV systems. Research has shown that especially engaging AVs can have an impact on student learning of DSA topics. Until recently, most AV systems were Java-based systems. But, the popularity of Java has declined and is being supplanted by HTML5 and JavaScript content online. In this paper, we present JSAV: the JavaScript AV development library. JSAV goes beyond traditional AV library support for displaying standard data structures components, to provide functionality to simplify creation of AVs on many engagement levels including interactive exercises. We describe the growing body of content created with JSAV and summarize our three years of experience and research results from using JSAV to build content that supports CS education. [1]

2. "Visualization Tool for Tree and Graph Algorithms with Audio Comments", Visualization tool is graphical representation of data in static or dynamic 2D or 3D format. In this paper author presented visualization tool of tree and graph of data structure integrated with audio comments. In this paper authors mainly focused on non-linear data structures because most of the students find it difficult to understand. It includes visualization of different types of trees like AVL trees, Red Black Tree, Heap and different complex graph algorithms like Breadth First Search. [2]

3. A Tool for Data Structure Visualization and User-defined Algorithm Animation", In this paper authors presented the following functionality:

a) Provides complete visualization for the widely used data structures such as an array. Stack, queue, tree, heap, graph, etc.

b) Provides the animation of common operations associated with the data structures, such as inserting an element into and deleting an element from the array, stack, and queue.

c) Provides animation of simple user-defined algorithms. [3]

4. "Visualizing data structures in an e-learning system", In this paper Three main aspects have to be taken into account when selecting techniques suitable for visualizing data structures in an e-learning system: First, general layout and design questions have to be answered. For example, data structures like trees or lists already hint towards layout constraints regarding the order of elements in an image. Second, the use case of an e-learning scenario requires special features like displaying objects that are missing in a data structure because they have been deleted unintentionally. Third, visualizations of data structures are used in the context of algorithms, producing sequences of changed structures over time. Thus, displaying these changes in an appropriate manner is an additional requirement. [4]

Chapter 3

REQUIREMENT AND ANALYSIS

3.1 Technical Specifications

Operating Systems:

Windows, RedHat Linux

Software Requirements:

Visual Studio Code, MongoDB Compass

Languages used for Front-end:

HTML, CSS, JavaScript, p5.js library

Back-end:

Node JS, Express Framework, MongoDB Server, Kubernetes

3.2 Problem Definition:

The basic idea behind the project is to develop data structures and algorithms visualization platform called AlgoViz. This platform is intended to be used as a support tool for the subject data structures and algorithms taught in engineering studies. The application is built using algorithms concepts and object-oriented concepts for developing data structure. The target audience for our application will be students for learning purposes, teachers for teaching purposes, and people who are curious about how the algorithms works.

3.3 Related Theory

To understand the data structures and algorithms, we often employ some kind of visualization like drawing it on paper, use of images and videos for understanding purposes. But

while going through only pseudo-code and explanation of algorithm and trying to imagine how it works it becomes difficult and frustrating sometimes.

Visualizations are always better as compared to textual data for understanding purposes. The same problem can be resolved using the web application AlgoViz which will visualize data structures and algorithms in the form of animations. This will help teachers and students to identify problematic areas/steps and work on improving them.

Chapter 4

DESIGN

4.1 Proposed system Architecture:

The proposed application will serve as a teaching and learning tool with 3 major components.

1] Testing Mode: Data structures and algorithm operations will be demonstrated step by step on random data set generated by the system in which each step is narrated as a status in detail.

2] Learning Mode: Users can give their data as input and users can perform necessary operations to get the correct final output. They can test their knowledge by replicating the algorithm's steps.

3] Analysis Mode: Users can give random tests provided on the platform to test their knowledge about data structures and algorithms.

Including these three modules, the application is using a database server for storing user's session information. This whole setup is deployed on the Kubernetes - an open-source container orchestration system for automating web application deployment, scaling, and management.

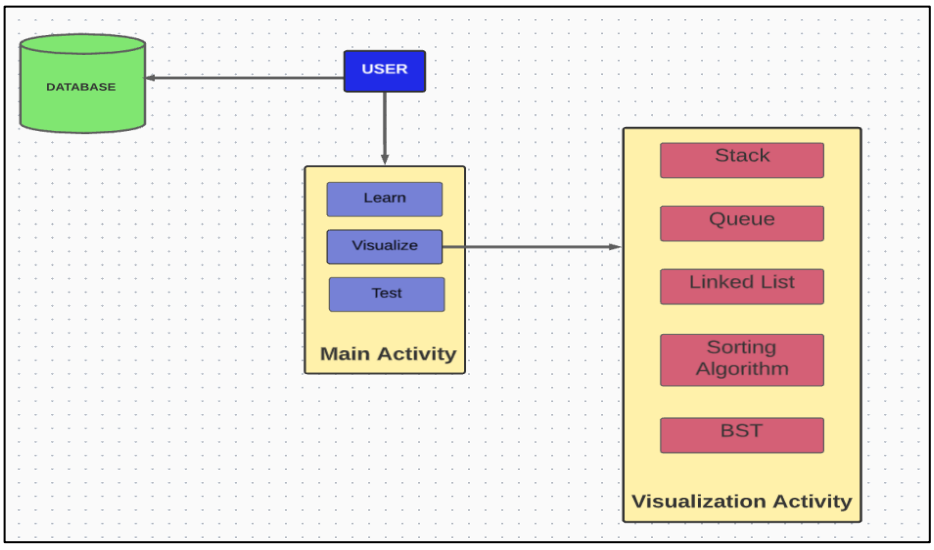


Fig. 4.1 Proposed architecture of AlgoViz

4.2 Data Flow and UML Diagram

4.2.1 Data Flow Diagram:

Data flow diagram maps the flow of information for any process or entity. The DFD diagram has predefined symbols such as rectangle, circle, double arrows with text to show entities, processes, and flow of data within the system. The DFD flow can vary from a simple hand-drawn path to a multilevel in-depth data flow.

Level 0 DFD:

Level 0 DFD shows the simple path of data within the system. The data path shows the flow from server to client. The algorithm visualizer captures the inputs from hardware and server and it shows output/visualizations to the client.

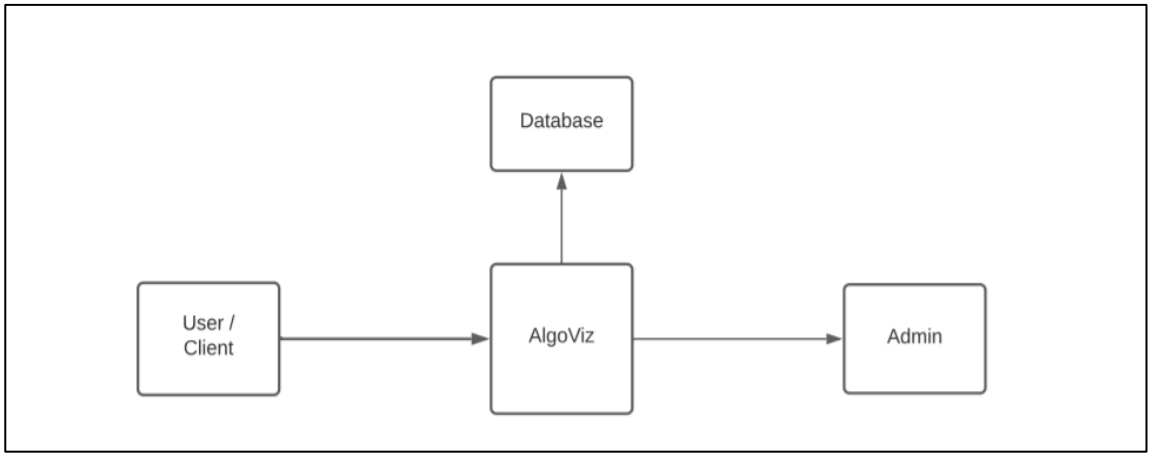


Fig. 4.2 Level 0 DFD

Level 1 DFD:

Level 1 DFD elaborates the level 0. It shows the data flow within the system with more details. In level 1, the entity's database or server sends data to the system. Learning mode and Testing Mode are available at level 1. Desktop systems and web applications display visualizations to the users.

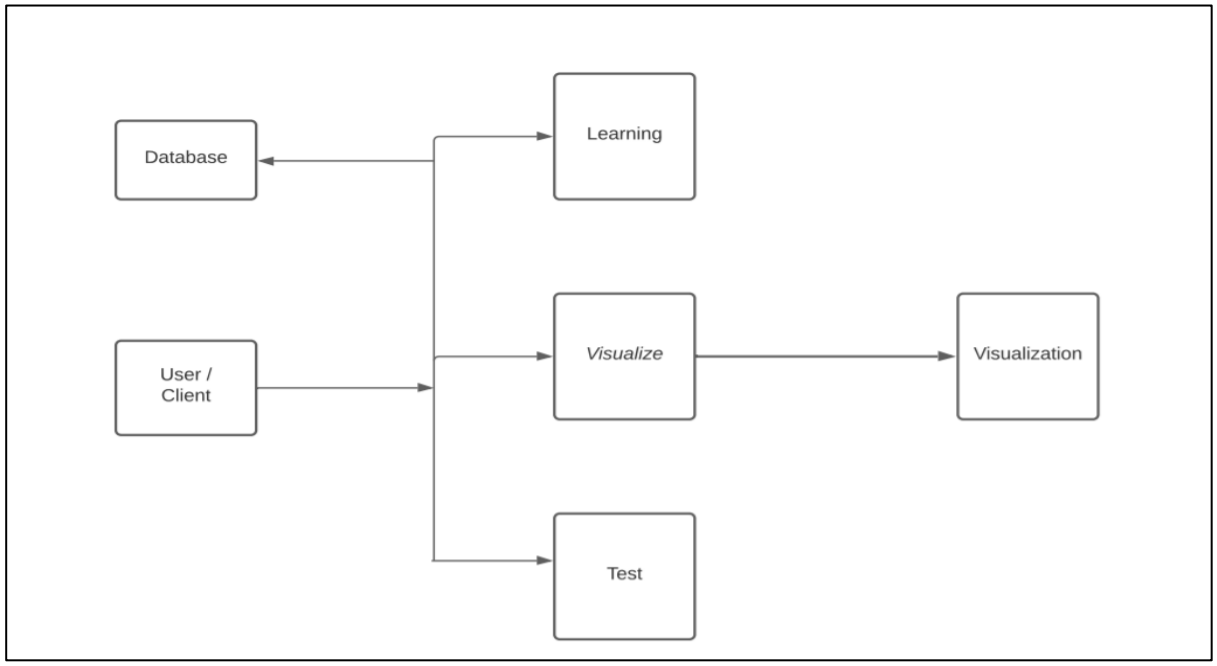


Fig. 4.3 Level 1 DFD

Level 2 DFD:

Level 2 DFD shows the most detailed path of data within the system. It elaborates the level 1 of DFD. The entities involved in level 2 are client, On-screen data, Visualization of data structures, and Algorithms. Firstly, the system does the authentication of the server with help of auth keys.

User login into the system with their credentials. They will access mainly three modes learning mode, test mode, and explore mode. The explore mode is containing the different data structures and algorithms to visualize.

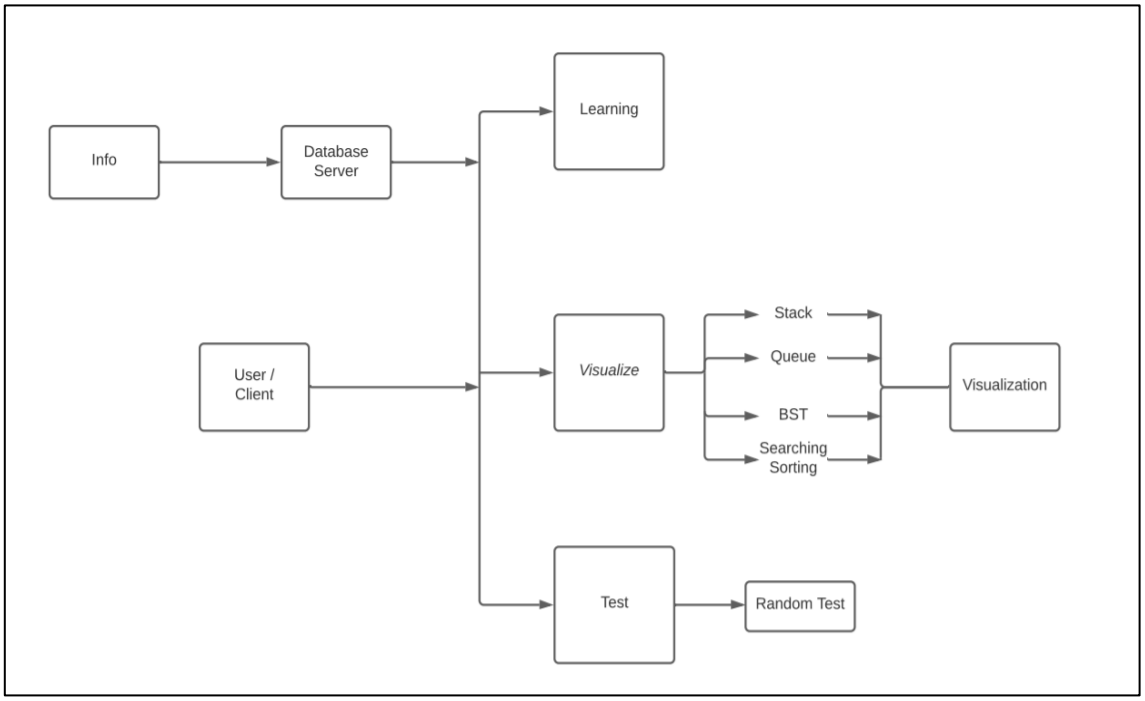


Fig. 4.4 Level 2 DFD

4.2.2 Entity Relationship Diagram (ER Diagram):

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research.

Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

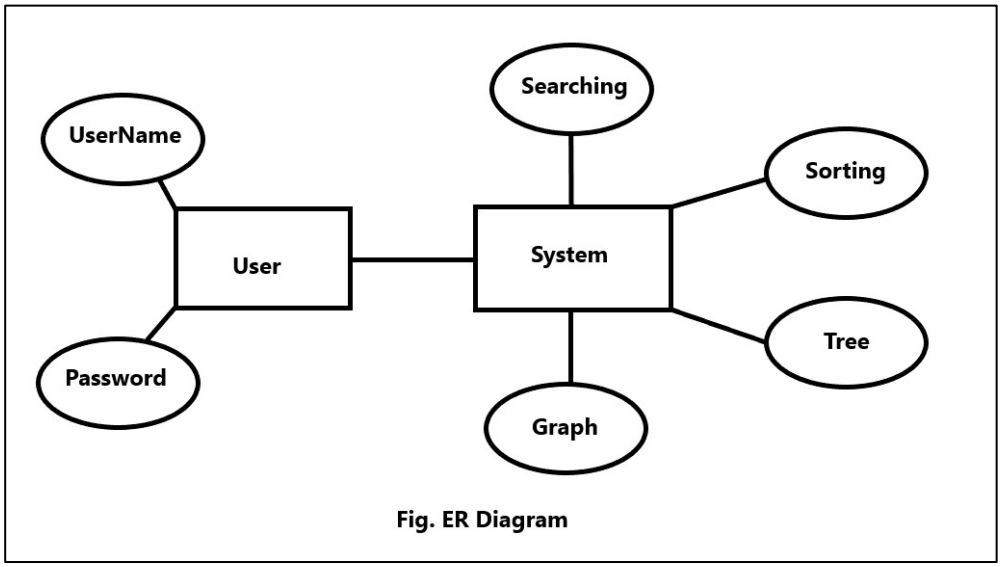


Fig. 4.5 ER Diagram

4.2.3 Use Case Diagram:

A use case diagram is a graphical representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can show the different types of users of a system and the various ways in which they interact with the system. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So, when a system is analyzed to gather its functionality use cases are prepared and actors are identified. The purposes of use case diagrams can be as follows:

- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.
- Show the interaction among the actors.

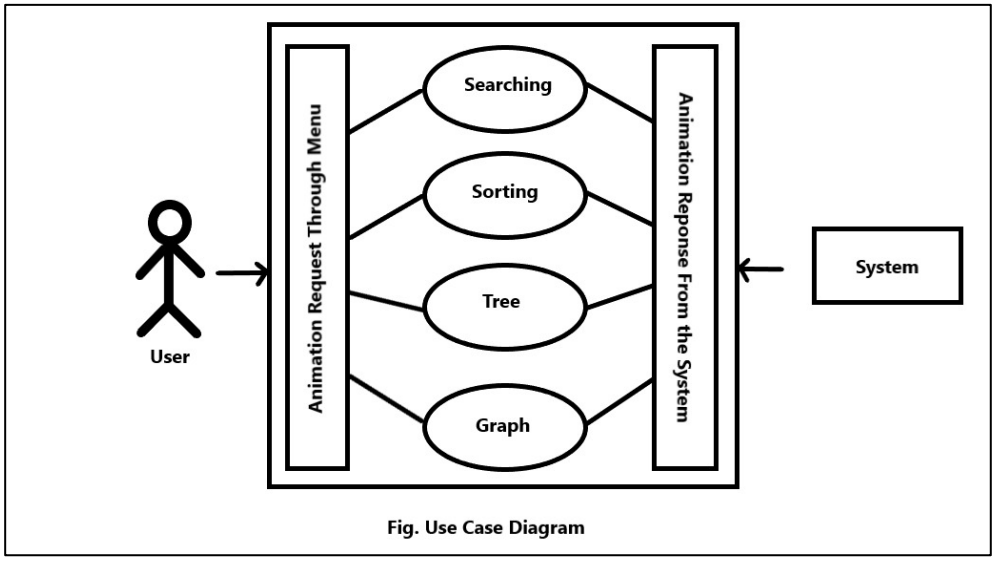


Fig. 4.6 Use Case Diagram

4.2.4 Activity Diagram:

The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

The purpose of an activity diagram can be described as −

- Draw the activity flow of a system.

- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

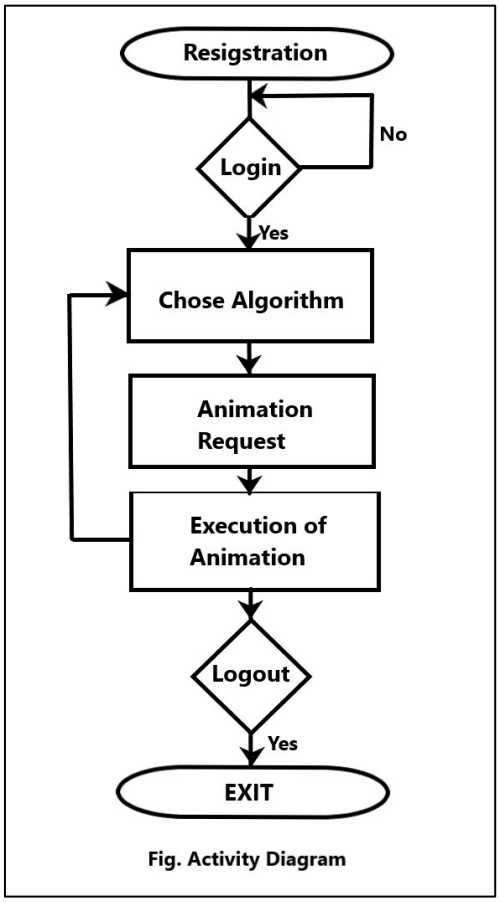


Fig. 4.7 Activity Diagram

4.2.5 Sequence Diagram:

In this project, sequence diagram shows that User and Database are object fields. when user login to the system he gets the slot details which is already stored on database. When he received slot history or details then he books the slot properly and other information for slot booking filled. Then according to near station or availability user booked the slots and his booking for charge his EV Charging will be done. Successfully booked the slots. Admin has rights to add new slots or view booking history or change slots .It represents that when Admin

login he has rights to add new slots, he will checking that no overlapping slots or not. Availability of slots or view slots. Then his work is to solve customers' problems or any queries or enquiries. Then finally Admin also logout from the system.

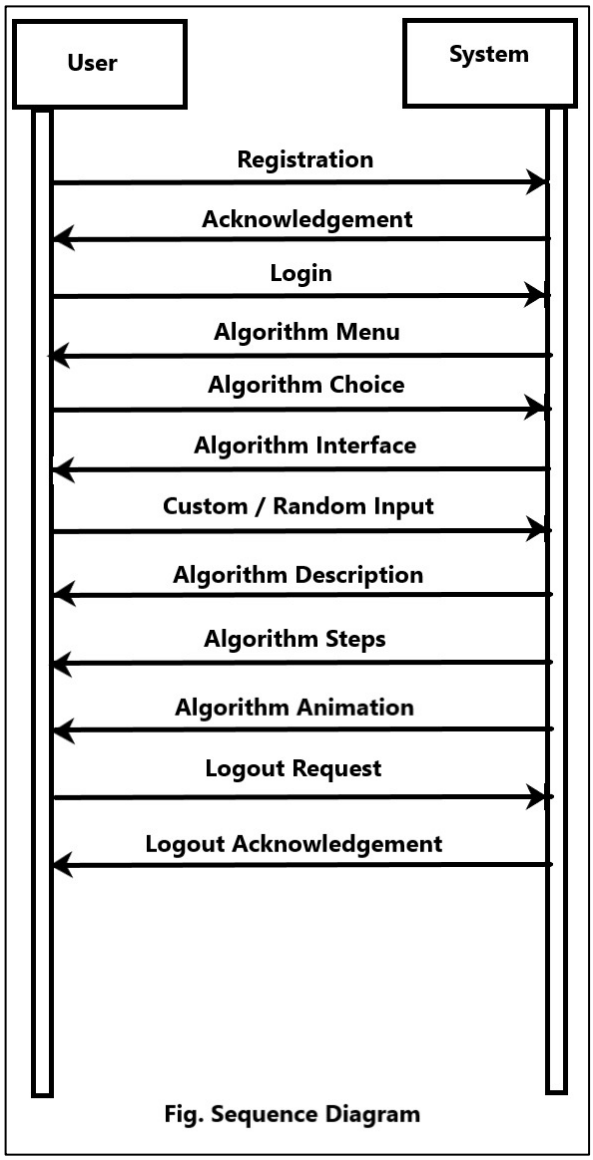


Fig. 4.8 Sequence Diagram

CHAPTER 5
IMPLEMENTATION

5.1 Implementation of AlgoViz:

Software:

Here, we used Visual studio code software for writing programs/codes for webpages in HTML, CSS, and JavaScript for the frontend as well as for manipulations on our website.

Used p5.js to create a canvas on our website to visualize binary search trees. The whole application uses MongoDB server for storing user's session management. This application is deployed on the top of the Kubernetes which is an open-source management tool for management, scaling and load management, etc.

The use of HTML5 (Hypertext Markup Language 5), JavaScript, and CSS combine to form this project's implementation (Cascading Style Sheets). There is only one project file which is an HTML file and contains the code. The only additional piece of code added to the main HTML file is the .m4a sound files to support the sound animation functionality (which are saved as .m4a files). As of now, I only did extensive testing using the Mozilla Firefox browser, and it's the browser of choice in this context.

However, tests done quickly revealed the possibility of Google Chrome and Safari integration. This software uses both object-oriented and functional programming paradigms in how it organizes the code. Before the final phase of development, the design was almost completely functional, where only three objects were used: one to control the canvas that displayed the animation, another to represent a piece of data, or "bar" object (blue rectangle with dynamically changing height and position), and a final one to represent the positions that each bar moved to, or "pos" objects. Although this incorporated several function calls, some instance variables and Boolean values were utilized to keep track of the algorithm picked and when to animation, but this led to a greatly integrated mass of code that was difficult to maintain. Several big refactorings later, the code has now taken on the form of a Model-View-Controller Architecture.

Although, because of its functional character, it possesses a multitude of individualized functions that alter the instance variables and Boolean values, which means it has a multitude of functions that directly alter the View and Controller. The major module in the HTML code between the <script> and </script> tags is known as the global scope. Everything within the framework is able to access the aforementioned variables and methods.

The View:

There are three items on the view: the sort Area, the bar, and the position. These objects operate within the container defined by the <script> and </script> tags in the .html file. This function's space is sometimes referred to as the "main" function, the first function invoked in a program.

Algorithms is the very fundamental about programming which is very important for beginner developers and getting it cleared is really tough for certain people. So this implementation will focus on getting the right thing to large number of people in easy and efficient manner. And here comes the HTML. Website is the easiest and best way of getting to the most of the audience there is always an option of desktop app or mobile application, but those will limit the reach-ability of the idea hence the idea will be implemented using website. Where we will be having website which will be user friendly demonstrating most of the generic algorithms where user can interact and see the algorithm happening infant of them rather than visualize it in dry run or imagining and getting the theory clear. This implementation is broadly divided into 2 steps which is decided based on various factors, considering the facts that all these three steps can be implemented parallel without disturbing the progress of other. Making the development process smooth and fast reaching our end goal is also the important factor which can’t be ignored. So below are the 2 major division of this implementation. A. Setting Up The Environment This step is chosen as the first step because is focuses mainly on the front-end where the focus is more towards UI/UX(User Interface/User Experience) rather than the business login, hence this step includes the steps for getting the right dependencies, building user friendly UI and an integration module which easily fits with the business logic to make the product ready. Detailed review and insight of this step will be further

discussed. B. Designing the visualization logic Any product is a good product only if it has good business logic or back end because if we design a beautiful UI but there is no data to display then it is useless. Hence this step also plays a great role in the final result. Where the main focus is not user driven instead it focuses on developing the visualization function and logic to make the user see what actually is happening in real time by developing a way to implement visualization in real time of different data structures like array, tree, graph etc. More insight about this step is discussed further.

1) Why P5.js Before getting straight into implementation its important to get it clear why P5.js is used in this implementation. P5.js is a client side library which is open source and used for delivering graphical and interactive experiences and this idea also deals with the graphical representation. This is one of the main reason for choosing the library. Another reason for choosing P5.js is that it makes the development process really fast as developing animations and motion can be really tough task and we need to develop our own rendering module hence P5.js act as a great tool to eliminate the rendering part so that the main focus is on visualization logic rather than building the rendering module which can be a mini project in itself. P5.js itself is a great tool for making graphical intensive application really light and fast, so concluding the idea this implementation is build upon the library P5.js. C. Setting Up The Environment As previously discussed this step involves the steps for getting the right dependencies, building user friendly UI and an integration module which easily fits with the business logic to make the product ready. Detailed review and insight of this step will be further discussed. So first we get the simple website layout ready using simple html tags and the most basic html page with empty values.

Chapter 6

RESULT AND DISCUSSION

6.1 Results:

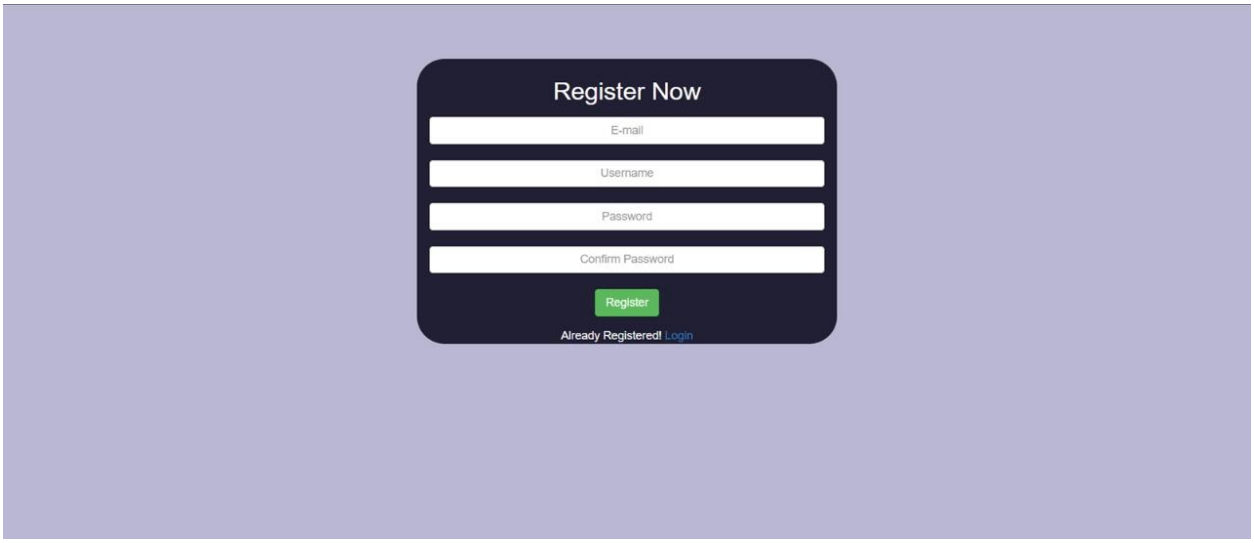
6.1.1 Registration Page:

A signup page (also known as a registration page) enables users and organizations to independently register and gain access to your system. It is common to have multiple signup pages depending on the types of people and organizations you want to register. In this article you will learn about the different types of signup pages, how to configure them and related functionality. Global Administrator access is required to create and modify signup pages

If you chose Multiple Users for the User Signup setting, you will be able to create an organization with more than one user signup as well as more than one set of users.

In this scenario you will see a setting called Multiple User Signup.

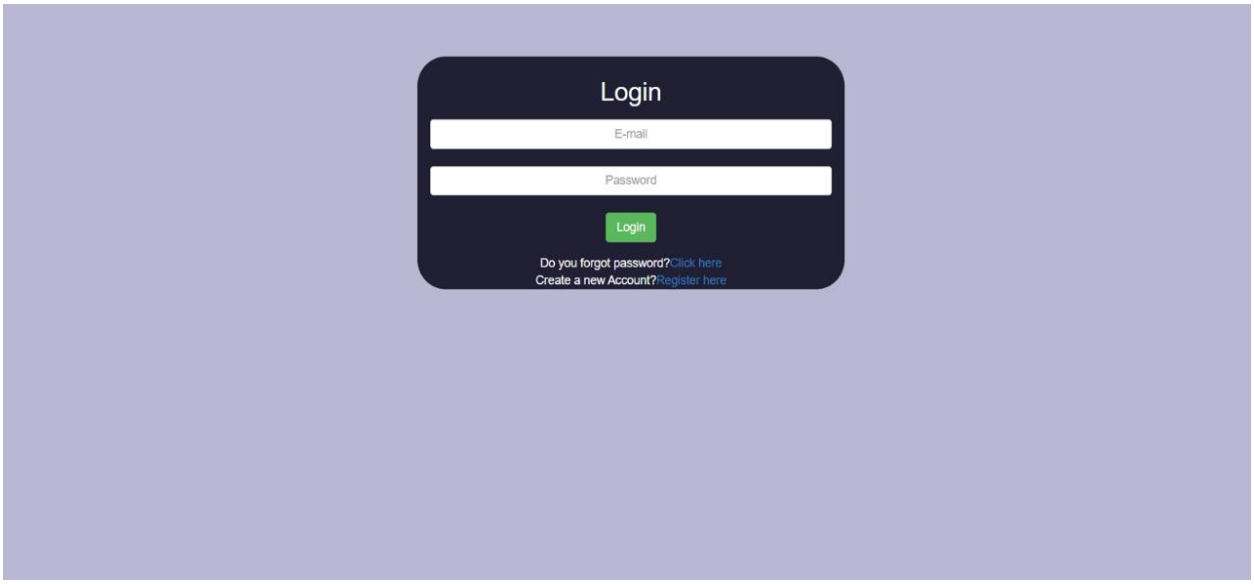
Click the Add Signup Page button to add multiple user signup pages. You can specify a minimum and maximum number of users you want to create per attached signup page as well as specify section and user titles..



6.1.2 Login Page:

A login page is a web page or an entry page to a website that requires user identification and authentication, regularly performed by entering a username and password combination. Logins may provide access to an entire site or part of a website. Logging in not only provides site access for the user, but also allows the website to track user actions and behavior. Logging off a webpage or site may be manual by the user or they can occur automatically when certain conditions (such as closing the page, turning off the computer, a long time delay, etc.) occur.

Some websites use cookies to track users during their logged in sessions. Generally, these cookies will turn off when the user logs out. Those cookies that are automatically deactivated and deleted from the user's computer are calledsession-only cookies. Protective measures that delete and invalidate associations between a user's handle and the session help assure users that logins can happen from any location, including public computers.



6.1.3 Stack Visualizer:

Picture shown above describes the visualization of Stack Operations. On this page user will get two input box one for Length and one for the Range. Length would describe how many elements would there be in the random stack generated and Range would describe the range of values to be filled inside the random stack being created. A Random Stack Generator Button is provided which would generate a random stack as per the Length and the Range given by user as input.

Next card on this frame is the Status card. On this card the current status i.e. the current positions or the step of the stack operation would be displayed such as push, pop, empty, full, etc.

Next card on this frame is the Stack operations card, this card contains the description of the stack operations such as push, pop, top, etc.

Below that is an input box to enter an element value to be inserted into the stack by the user. With that input box two buttons are provided as one for Push i.e. Inserting an element into the stack and the other Pop for removing an element from the Stack.

At last there are two containers showing the dynamic visualization of the stack. First container contains the Push stack i.e. stack with the elements we pushed into the stack. And the other container contains Pop stack i.e. stack with the elements we popped out of the stack.

6.1.4 Queue Visualizer:

Picture shown above describes the visualization of Queue Operations. On this page user will get two input box one for Length and one for the Range. Length would describe how many elements would there be in the random Queue generated and Range would describe the range of values to be filled inside the random Queue being created. A Random Queue Generator Button is provided which would generate a random Queue as per the Length and the Range given by user as input.

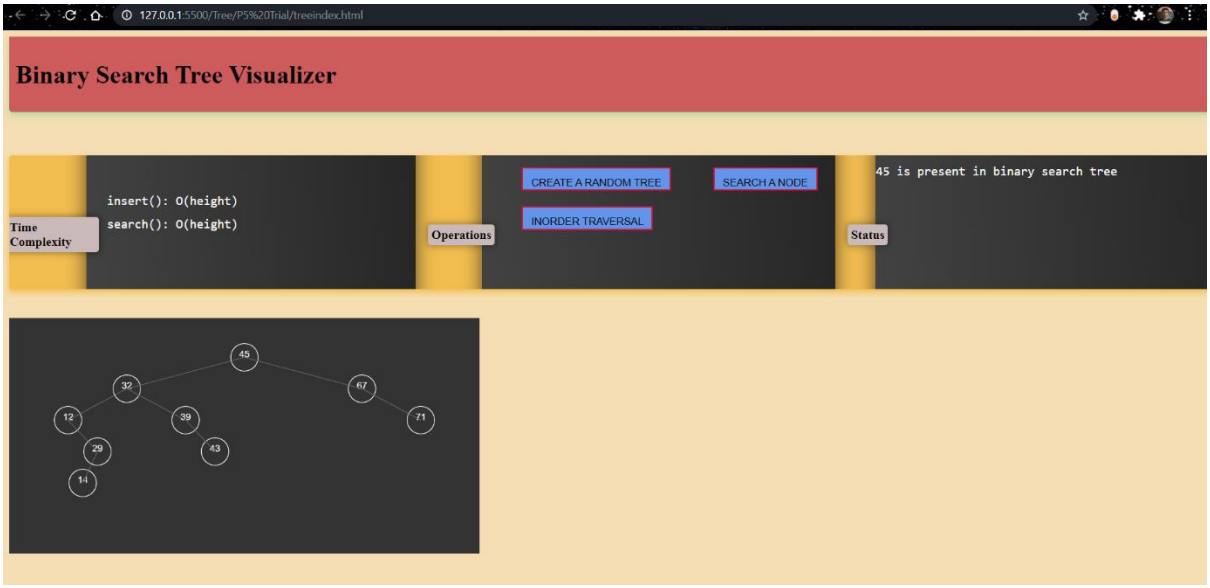
Next card on this frame is the Status card. On this card the current status i.e. the current positions or the step of the Queue operation would be displayed such as Enqueue, Dequeue, empty, full, etc.

Next card on this frame is the Queue operations card, this card contains the description of the stack operations such as Enqueue, Dequeue, Front and Rear.

Below that is an input box to enter an element value to be inserted into the Queue by the user. With that input box two buttons are provided as one for Enqueue i.e. Inserting an element into the Queue and the other Dequeue for removing an element from the Queue.

At last there are two containers showing the dynamic visualization of the Queue. First container contains the Enqueue i.e. Queue with the elements we inserted into the Queue. And the other container contains Dequeue i.e. Queue with the elements we removed out of the Queue.

6.1.5 Binary Search Tree Visualizer:



Now the picture shown above is of the Binary Search Visualizer. This frame of the system contains basically four containers.

First container contains the Time Complexity of the Binary Search Tree.

Second container contains a Create a Random Tree button which when clicked by the user generates a random binary search tree with random values inside it. One more button in this container is Search a node, when user clicks this button, A text box is provided to the user where user can enter an integer value that he wants to search inside the Binary Search Tree and this button would trigger the Binary Search Tree algorithm which would find out the positions of the user-given node inside the Binary Search Tree if it exists. At last this container contains one more button i.e. Inorder Traversal Button. When user clicks this button, the visualization of the previously said Binary Search Tree begins on the visualization screen.

Next container on this frame is the Status frame i.e. a container which displays the current step being performed in the binary search tree or the current status of the binary search tree, the results of the comparison operations being performed in algorithm, etc.

At last this frame contains a visualization container which displays the dynamic animation of the Binary Search Tree with each of it's algorithm steps being performed.

6.1.6 Searching & Sorting Visualizer:

This frame is the core frame of our system, various algorithms of Searching and sorting methods could be accessed through this single frame.

This frame firstly contains an input box where user can enter the integer values and the values would be inserted into the Array after user clicks the Go button next to the said input box.

Then comes the Range pointer, user could scroll this pointer in order to set the Range of values to be inserted into the Random Array. Then there is Length pointer which sets the length of the random array to be generated. At last it contains the Speed pointer, this pointer estimates the speed by which the algorithm steps would be visualized on the screen. Next to it is the Random button by clicking which user could easily generate a random array as per the given Range and the Length given by the same user.

Then below it is shown an algorithm choice bar, which contains various buttons to chose algorithms from. First comes the Sorting algorithms such as Bubble sort, Insertion sort, Quick Sort and Selection sort. Once user clicks on any of the mentioned buttons related Sorting algorithms starts to be executed on the given array and the visualization of the algorithm is displayed on the screen. Then comes an input box, where user could enter an integer value which is to be searched using the searching algorithms inside the Array. Then comes more buttons as Binary Search, Linear Search and at last the Jump Search. By clicking any of the said buttons, user could trigger the Searching algorithm which would search the integer value given by the user inside the array and display the result and the visualization of the algorithms on the screen.

Below that is the main visualization screen and directly under it are three cards as such:

First is the complexity card, this card contains the complexity of each searching and the sorting algorithm and each one is displayed in this card according to the user's choice of the algorithm.

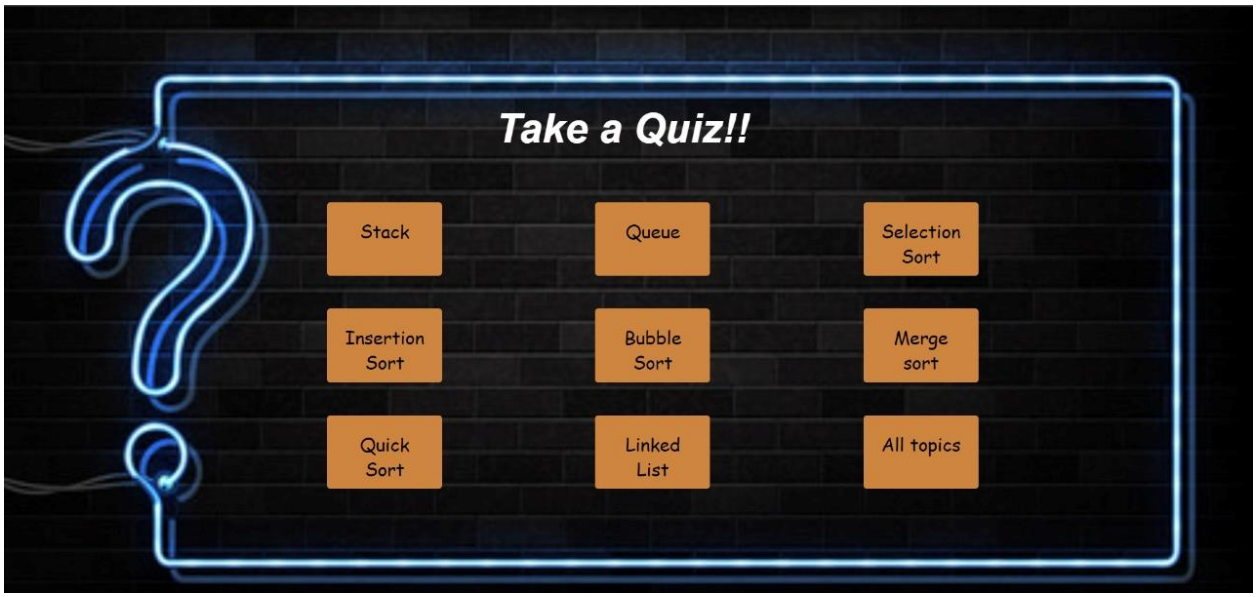
Second is the Algorithm card. This card displays the algorithm steps and highlights the current step being performed.

Next and the very last card on this frame is the status card, this status card contains the current status of the array and displays the result of the algorithm. In addition to that this card also contains the Speed by which the algorithm visualization is being displayed, the Range that user set before generating the random array and the Length of the randomly generated array.

6.1.7 Test Series:

Online test series are mock tests or practises that the entrant chooses to give in order to decide on the preparation of his/her studies.

Online test series works mostly in the form of MCQs (Multiple Choice Questions), where all the candidate has to do is select the right option from the given 4/5 options. The candidate has to enter and select all the valid details asked for, and then give the examination.



6.1.8 Question Frame:

An online test series is the mock test or practice test that is taken up by candidates as a part of studying.

Taking up such mock tests, helps them decide on the further study plan; meaning, it helps them analyse how much study they have done and how much more needs to be done.

They have been replaced by virtual tests and mocks. The new methods to test a person's knowledge is by testing him through the online test series.

Online Test Series is an online assessment tool which will help students to evaluate their preparedness for the final exams.

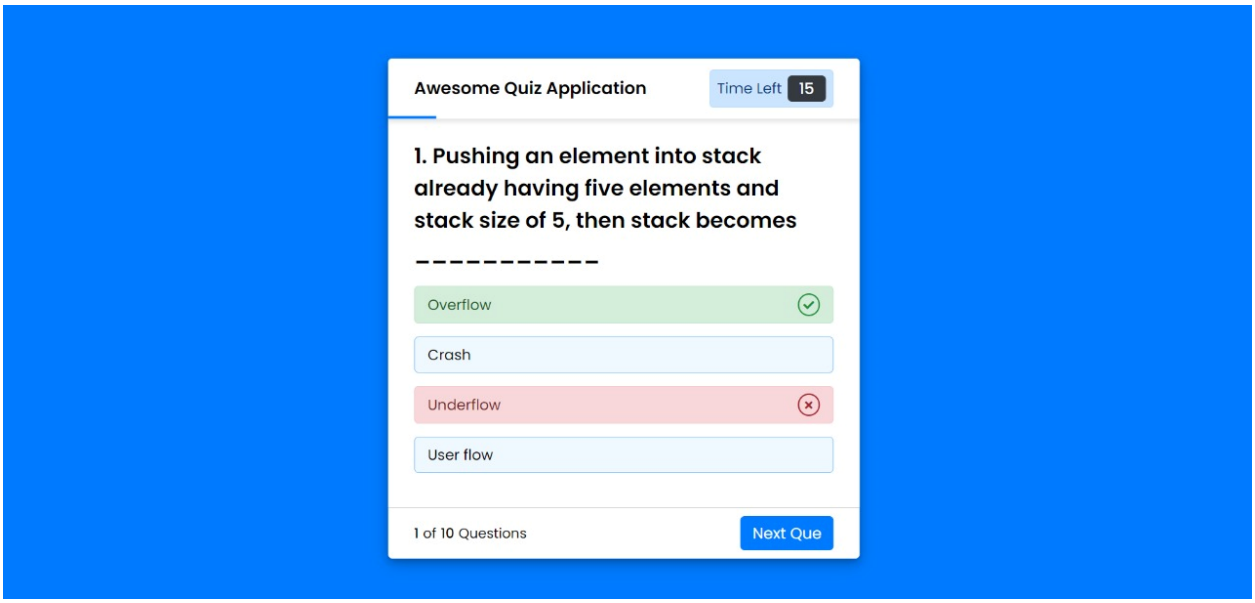
They are the mock exams or say 'Practice tests' that help the students to analyse their study and how much more or less they need to focus on working on the subjects.

Online Test Series is a set of mock exams that are conducted virtually instead of manually.

These set of examinations is taken up to build in the confidence of the candidate to appear for the actual examination.

These examinations are designed such that they are extremely user-friendly and can be taken from any place.

Meaning, a person can have access to the online test series with the means of the internet from any corner of any city.



6.2 Discussion:

Data structures and algorithms operations will be demonstrated step by step on random data set generated by the system in which each step is narrated as a status in detail. Users can give their data as input and users can perform necessary operations to get the correct final output. They can test their knowledge by replicating the algorithm's steps. Users can give random tests provided on the platform to test their knowledge about data structures and algorithms. Including these three modules, the application is using a database server for storing user's session information.

Chapter 7

CONCLUSION

7.1 Conclusion

Algorithm visualization can be seen as a valuable supporting tool, used in addition to standard ways of education in the field of computer science. We believe that it helps to improve the quality of education in the field of computer science. Algorithm visualizations can help to understand the principles, but do not replace the need to implement algorithms by students in a chosen programming language.

This tool can be used as an effective supplement to the traditional classroom education and textbook for Data Structures and Algorithms course.

REFERENCES

[1] Ville Karavirta, Clifford A. Shaffer (2016), *"Creating Engaging Online Learning Material with the JSAV JavaScript Algorithm Visualization Library",* http://jsav.io/ , 2016 IEEE Transactions on Learning Technologies

[2] Ahmad Supli, Norshuhada Shiratuddin, Syamsul Bahrin Zaibon (2016), *"Critical Analysis on Algorithm Visualization Study",* International Journal of Computer Applications

[3] E. Fouh, M. Akbar, and C. Shaffer, "*The role of visualization in computer science education*", Comput. Schools, vol. 29, pp. 95–117, 2012.

[4] Tao Chen', Tarek Sobh' (2001), *"A Tool for Data Structure Visualization and User-Defined Algorithm Animation",* ASEE/IEEE Frontiers in Education Conference

[5] Jauhar Ali (2009). *"A Visualization Tool for Data Structures Course",* IEEE International Conference on Computer Science and Information Technology

[6] S. Hall, E. Fouh, D. Breakiron, M. Elshehaly, and C. Shaffer, "*Evaluating online tutorials for data structures and algorithms courses*", presented at the ASEE Annu. Conf., Atlanta, GA, USA, Jun. 2013, Paper 5951.