Practical approach to learn data structures and algorithms(DSA) using DSA Visualizer.

[1]Shushant Tripathi, [2]Prajwal Sawant, [3]Kaynat Momin, [4]Vedika Patil,
[1,2,3]Student, [4]Assistant professor,
[1]Department of Computer Science,
[1]K. C. College of Engineering and Management Studies & Research, Thane, India.

Abstract: Data Structure and Algorithm Visualization is an important parts of algorithm design. Traditionally, analysis of Data Structure and Algorithms is theoretical and mathematical. As a result, it takes a lot of time, is challenging to study, and lacks comprehension of how an issue is implemented in real life. This restricts the breadth and depth of conceptual frameworks for undertakings. Most abstract ideas and procedures, like the procedural encoding of algorithms and data structures, are difficult for individuals to understand. Using visualisations to help make abstract ideas more apparent will help them understand topics better.

This project presents the design, implementation and visualizations for the data structures and algorithms that occupies a unique niche between the following two domains. In the first domain, called algorithm theory, in this, the theory related to the algorithm is implemented. In the second domain, called algorithm simulation, the framework for exploring and understanding algorithms and data structures is exhibited.

Keywords: Data Structures, Algorithms, Visualization, Real-Life Implementation.

INTRODUCTION:

Data structures and algorithms are essential to computer science students and aid in job placement. The base is made up of algorithms and data structures. Data Structures and Algorithms are essential to achieving the objective, and one must be proficient in problem-solving using programming languages to be a part of the software development industry [1]. A newcomer to computer science, or more specifically, a newcomer to data structures, lacks knowledge of what data structures and algorithms are, how they operate in code, and how they are applied in the real world. We use data structures and algorithms every day, whether we are aware of them or not.[4]

We powerfully believe that the logic is relatively simple: An algorithmic programme uses some information to operate. (the input variables, and also the output data).[2] Any explicit field of engineering typically has its unique way of visualising data. For example, graphs and trees are typically represented as circles joined by lines, while number sequences may be shown as collections of vertical bars.[2,3] There are also standard ways to represent matrices, vectors, real functions, etc. Sometimes an algorithmic animation is produced by executing the algorithmic programme slowly or in stages, and smoothly modifying the visual representation of the data on the screen.

An expert user can observe how the algorithmic programme progresses, while a novice user only sees visual things moving and changing their forms and colours makes them easier to grab the working of the data structures and algorithm. The answer we offer is to consider not just what the algorithmic programme is doing but also why it is behaving the way it is.[7]

We must make sure that we create programmes that visualise data structures and algorithms along with their actual implementation as app development takes centre stage. To assist students and teachers in visualising data structures and algorithms with their actual execution, we have designed and provided an idea, data structures, and algorithm.[5,6]

REVIEW OF LITERATURE:

1. S. Ghadge., et al., 2022, this review focuses on the way they have developed a website which builds programs that visualize data structures and algorithms along with real-life implementation but this website has an input limitation that is the user can only enter up to 14 values in the stack. It provides the animation of graph data structures such as BFS and DFs. The user interface does not seem to be user-friendly.
2. Gururbaxani R., et al., 2022 this study describes an e-learning application for the visualisation of Pathfinder, Prime Numbers, Sorting Algorithms, N Queen, Convex Hull, and Binary Search Game. For instance, the animation tool would display information as a bar while sorting, and after selecting a data ordering and method, the user may either run an automatic animation or progress through it at their own pace

PROPOSED SYSTEM:

- Our proposed system, consist of different data structures and algorithm which are stack, queue, liked list, graph and KMP algorithm, N-Queens, and sorting algorithms (Bubble sort, selection sort and insertion sort) respectively.
- Once the user opens the application and clicks on a particular data structure or algorithm then he/she will navigate to the theory containing that topic. Directly navigating the user to the simulation page makes it difficult to understand if they do not have some basic knowledge of that specific topic.
- On the simulation page, users have the flexibility to enter values of their own choice and perform animation according to their will.
- The end user will be able to visualise all the operations of that particular data structure or algorithm. For instance, in a linked list they can add the nodes at a specific position in the linked list such as at the end, or beginning. Moreover, they can also delete the nodes from a linked list.

- For making the software more user-friendly we have also implemented a sound feature which gives vocal notification whenever any task is performed by the end user.
- Once the student is done with the simulation part they can test their knowledge by giving the test in the MCQ form.
- In the test section, students have some time to solve the question. Time is not the same for all types of questions instead it is based on the complexity of the question.
- If users select the wrong answer they can see the correct answer along with an explanation. After completion of the test, they can see their scorecard.
- The GUI is done using XML and some external libraries are harjot croller, varunest sparkbutton, denzcoskun ImageSlideShow.
    - The backend part is implemented using Java and Kotlin programming languages.

PROJECT FLOWCHART:

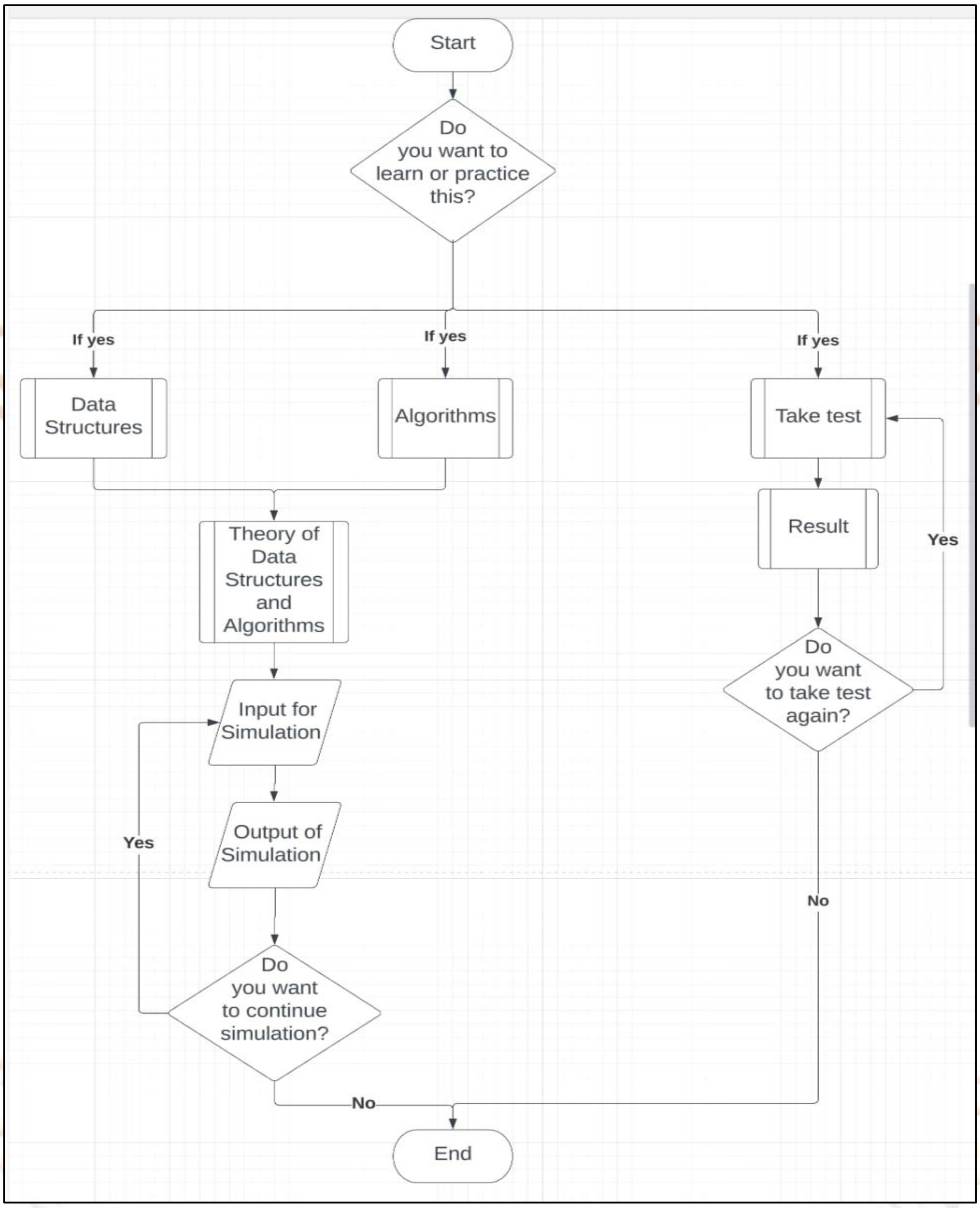

Start
Do you want to learn or practice this?
If yes
If yes
If yes
Data Structures
Algorithms
Take test
Theory of Data Structures and Algorithms
Result
Yes
Do you want to take test again?
Input for Simulation
Output of Simulation
Yes
No
Do you want to continue simulation?
No
End

Figure No. 1. Project flowchart

RESULTS:

We effectively see different Data Structures and Algorithms and comprehend how they function with the help of this Android app. This software, which is interactive and colourful, will aid in the learning and understanding of various Data Structures and Algorithms by students and teachers.

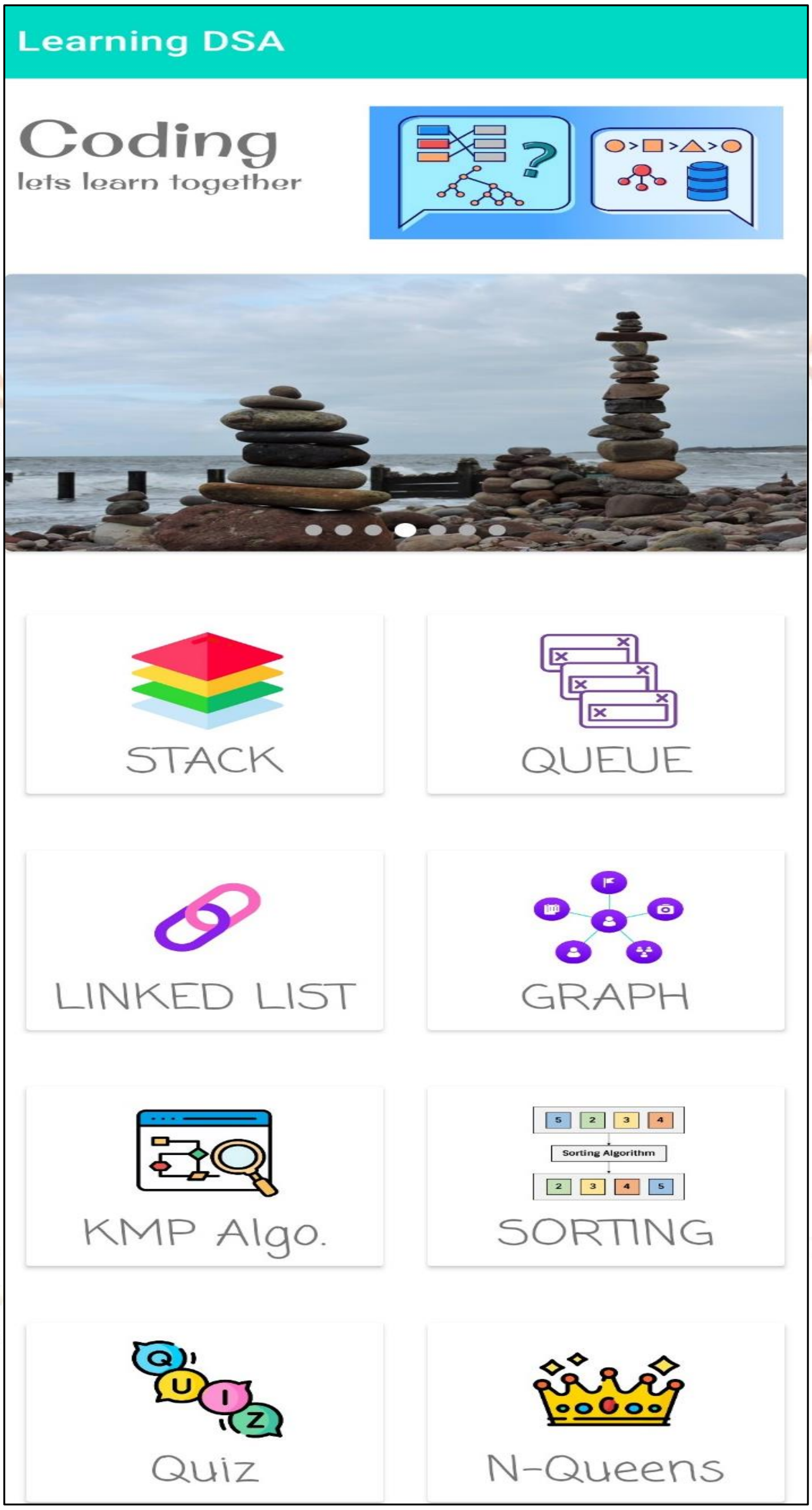
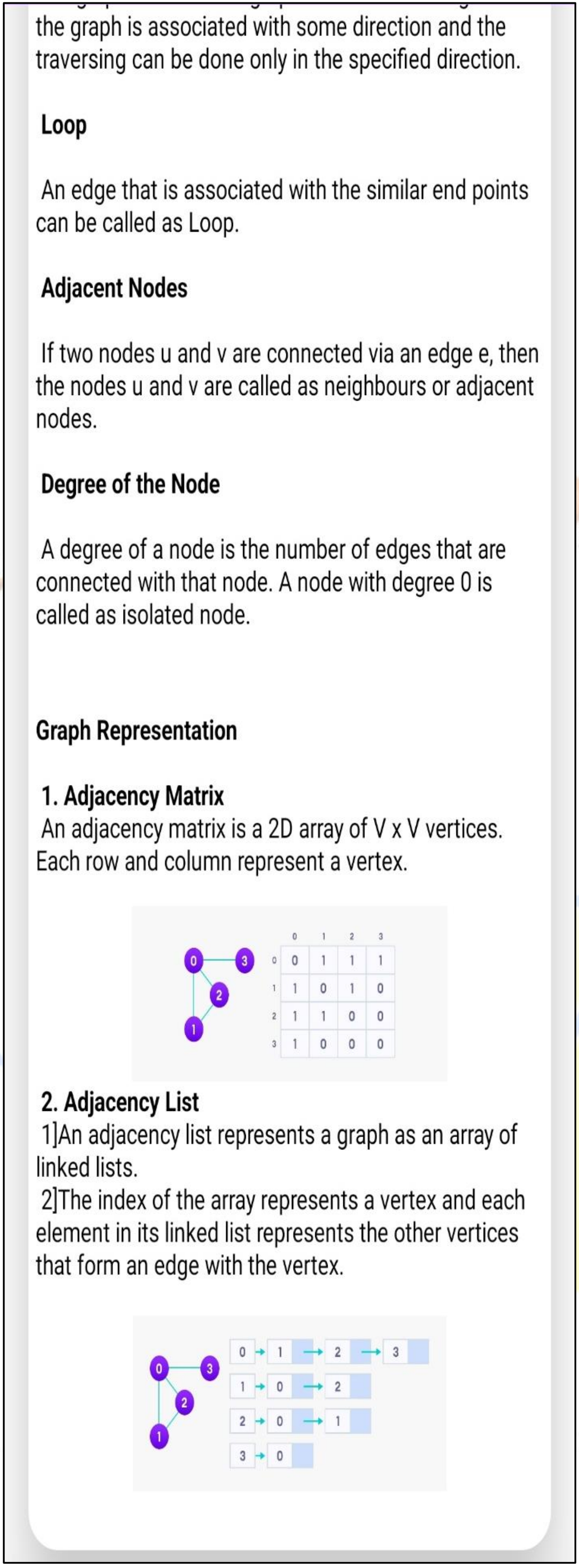


Figure No. 2. Home page of Android application



the graph is associated with some direction and the traversing can be done only in the specified direction.

Loop

An edge that is associated with the similar end points can be called as Loop.

Adjacent Nodes

If two nodes u and v are connected via an edge e, then the nodes u and v are called as neighbours or adjacent nodes.

Degree of the Node

A degree of a node is the number of edges that are connected with that node. A node with degree 0 is called as isolated node.

Graph Representation

1. Adjacency Matrix

An adjacency matrix is a 2D array of V x V vertices. Each row and column represent a vertex.

	0	1	2	3
0	0	1	1	1
1	1	0	1	0
2	1	1	0	0
3	1	0	0	0

2. Adjacency List

1]An adjacency list represents a graph as an array of linked lists.

2]The index of the array represents a vertex and each element in its linked list represents the other vertices that form an edge with the vertex.

0 → 1 → 2 → 3

1 → 0 → 2

2 → 0 → 1

3 → 0

Figure No. 3. Theoretical page of application

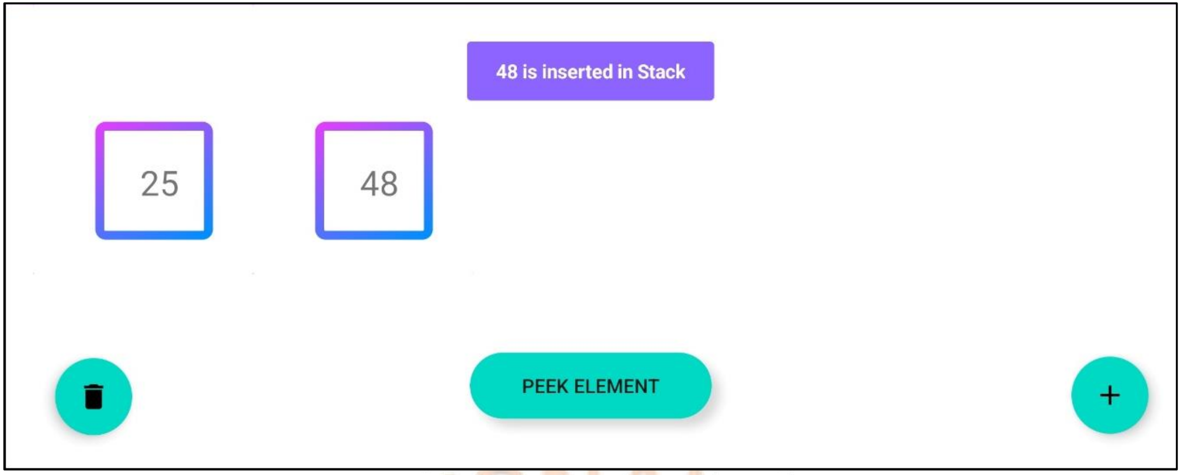
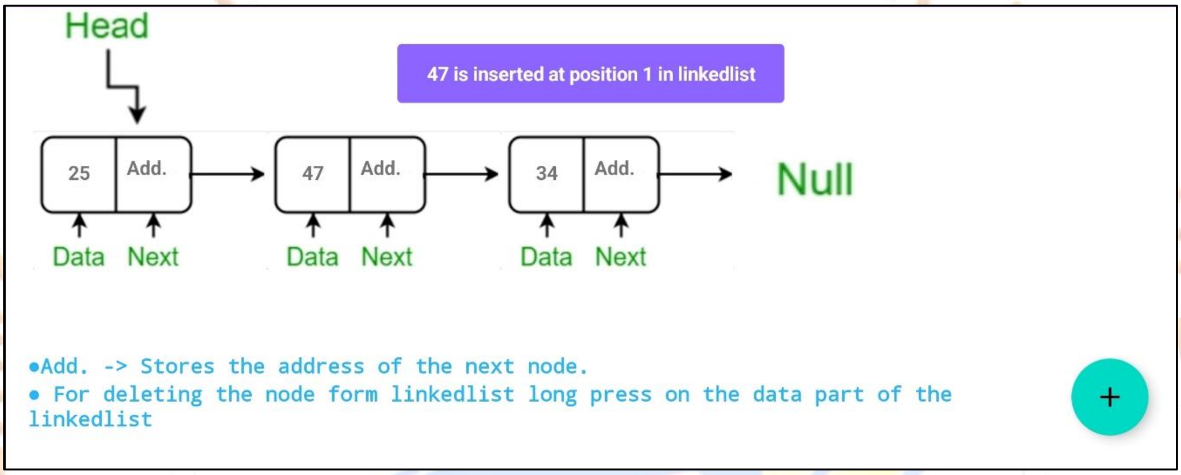

48 is inserted in Stack

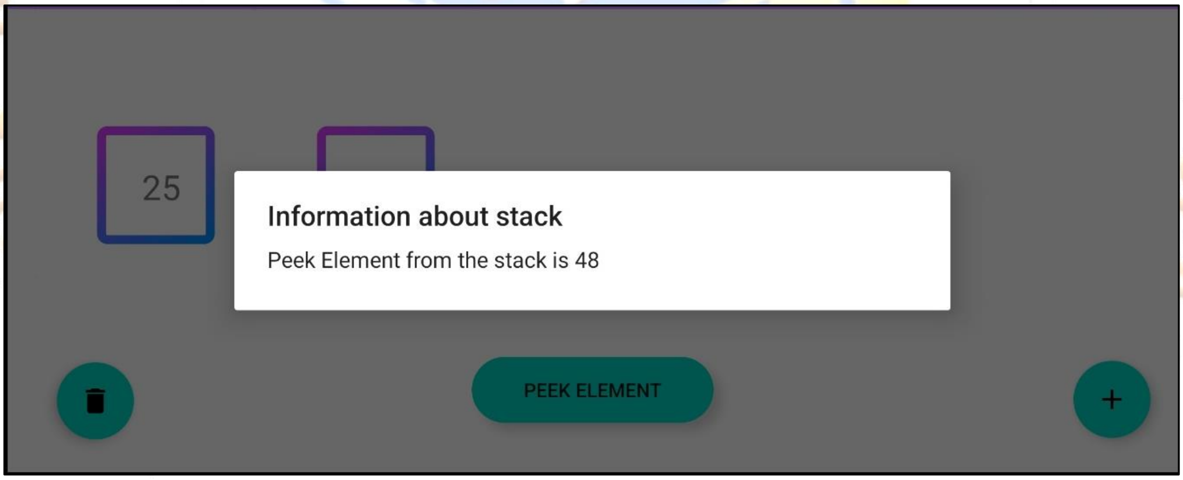
Figure No. 4. Stack data structure


47 is inserted at position 1 in linkedlist

●Add. -> Stores the address of the next node.

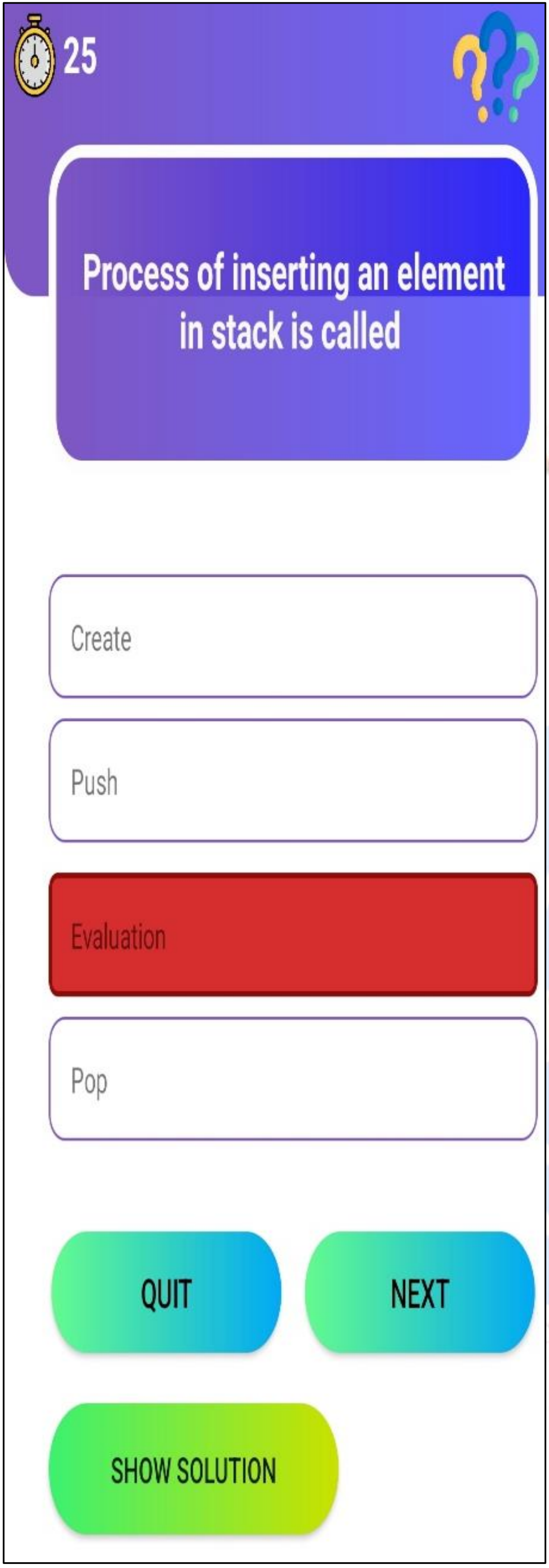
● For deleting the node form linkedlist long press on the data part of the linkedlist

Figure No. 5. Linked list data structure


Information about stack

Peek Element from the stack is 48

Figure No. 6. Peak element of stack



Process of inserting an element in stack is called

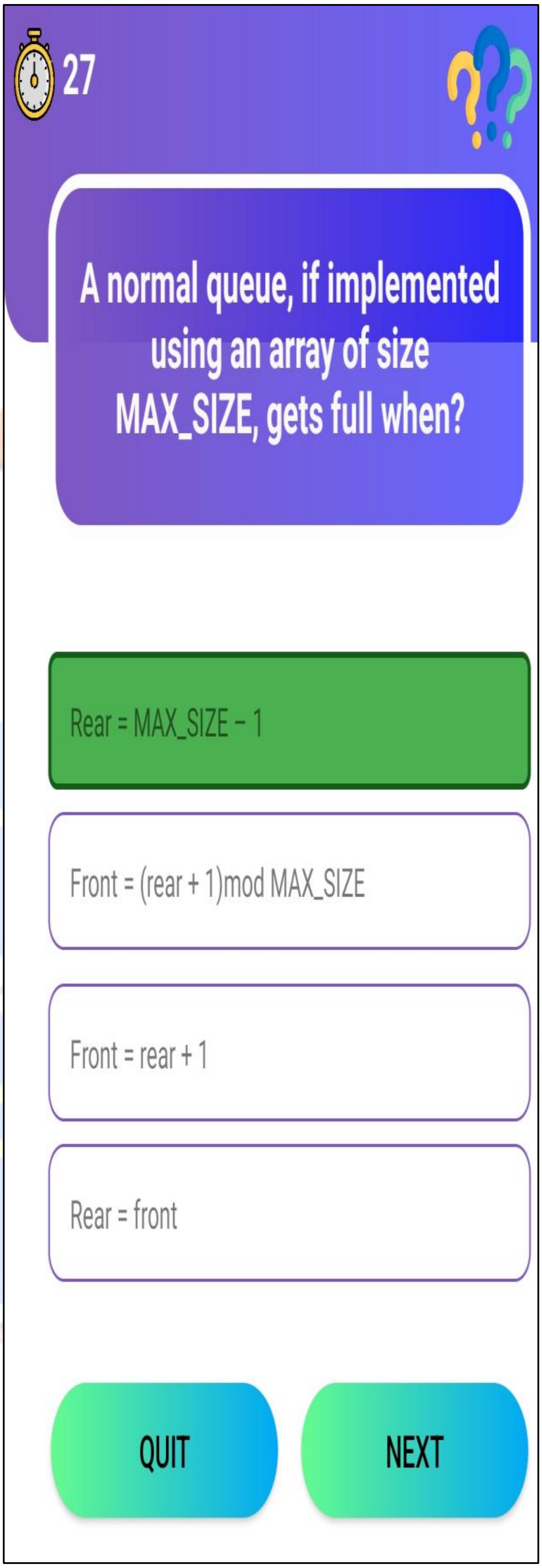
Create

Push

Evaluation

Pop

Figure No. 7. Wrong solution picked



A normal queue, if implemented using an array of size MAX_SIZE, gets full when?

Rear = MAX_SIZE – 1

Front = (rear + 1)mod MAX_SIZE

Front = rear + 1

Rear = front

Figure No. 8. Right solution picked

Fig. No. 4. This image shows the animation of the stack data structure along with the notification when any operation is performed. In this case, the user has inserted the value in the stack.

Fig. No. 5. This image shows the animation of the linked list data structure along with the notification when any operation is performed. In this case, the user has inserted the value at position 1 in the linked list.

Fig. No. 6. In this image end-user has clicked on the peek element button as the topmost element in the stack is 48 so it displays the message “Peek Element from the stack is 48” in the popup window.

Fig. No. 7. This figure shows that the user has selected the wrong answer. Once the user has selected the wrong answer the selected one becomes red and the “SHOW SOLUTION” button is also displayed to show the explanation of why the selected answer is wrong. This feature makes the application more user-friendly.

Fig. No. 8. In this image end-user selected the right answer. So the selected one is displayed in green colour.

CONCLUSION:

To conclude, in our project, we use data structures, algorithms, and app development to build a platform for students, teachers, and anyone else who wants to visualise and comprehend data structures, and algorithms, and also learn about practical examples of those DSA for a better understanding.

REFERENCE:

1. Ghadge S, Mane V. A SURVEY PAPER ON DATA STRUCTURE AND ALGORITHM VISUALIZATION.
2. Slavomır Simonak (2016), “Algorithm Visualizations as a Way of Increasing the Quality in Computer Science Education”, IEEE 14th International Symposium on Applied Machine Intelligence and Informatics.
3. Kumari A, Mittal M, Jha V, Sahu A, Kumar M, Sangwan N, Bohra N. Algorithm visualization-modern web-based visualization of sorting and searching algorithms.
4. Juan Lin (2020), “Data Structure Visualization on the Web”, IEEE International Conference
5. https://ieeexplore.ieee.org/Xplore/home.jsp
6. https://www.springer.com/in
7. https://www.geeksforgeeks.org