



## Data Structure & Algorithm

# DATA STRUCTURES VISUALIZATION WEBSITE

**Submitted To:**

Amna Wajid

**Proposal Submission:**

24/11/2025

**Submitted By:**



M Zeewaqr



M Abdullah



M Mustaqeem

## 1. Project Overview

This project proposes the development of an interactive web-based platform designed to help beginner programmers understand how fundamental data structures work internally. The website will be built using **React** and **JavaScript** (frontend only), and will provide visual, step-by-step animations for operations such as insertion, deletion, traversal, and updating within key data structures.

The platform aims to bridge the gap between theoretical understanding and practical implementation by visually demonstrating pointer movements, node creation, and structural changes. Additionally, the system will generate relevant code snippets in **C++** for students to copy and use in their own programs.

---

## 2. Problem Statement

Many new programmers struggle to visualize how data structures operate internally, especially concepts involving pointers such as linked lists, stacks, queues, and trees. Traditional textbook-style explanations often lack dynamic demonstrations, leading to misunderstandings or incorrect implementation of algorithms. A visual, interactive educational tool can significantly enhance comprehension by illustrating each step of the operation.

---

## 3. Project Objectives

1. To design a React-based website that visually demonstrates fundamental data structure operations.
2. To help students understand internal working mechanisms such as node creation, pointer updates, and element shifting.
3. To offer step-by-step animations for operations like **insertAtHead**, **insertAtTail**, **deleteHead**, **deleteTail**, **push**, **pop**, **enqueue**, **dequeue**, etc.
4. To generate corresponding **C++** code snippets dynamically so users can copy and practice.
5. To develop a semi-advanced educational platform suitable for portfolios and CV enhancement.

## 4. Proposed Features

### Core Data Structures (must include):

- Arrays
- Stack
- Queue
- Linked List (SLL, optional: DLL)
- Tree (Binary Tree / BST)

### Optional Future Expansions (time-permitting):

- Graph Visualization
- Hash Table Visualizer

### Functional Features:

- Users select a data structure from the homepage.
- A list of available operations appears (e.g., Insert at Head, Delete Node, Push, Pop).
- User enters a value (e.g., 5) and chooses an operation.
- The GUI displays step-by-step animation:
  - Node creation
  - Pointer redirection
  - Head/tail movement
  - Array element shifting
  - Stack/queue front/back updates
- A small panel shows the C++ code for that specific operation.
- Speed control for animations (slow/medium/fast).
- Reset and replay functionality.

## 5. Tools and Technologies

- **Frontend Framework:** React.js
- **Programming Language:** JavaScript (ES6+)
- **Styling:** Tailwind CSS / CSS Modules (recommended for clean design)
- **Animation:** React Spring / Framer Motion
- **Code Snippet Generator:** Custom JS logic + syntax highlighting
- **Backend:** *Not required* (static website)
- **Deployment Suggestion:** GitHub Pages / Vercel

*(No backend is needed since it is an educational visualization platform, and all data is generated dynamically in the browser.)*

---

## 6. Expected Outcome

A polished, semi-advanced educational platform capable of teaching the internal mechanics of essential data structures using visual animations and real-time code generation. The final output will be a portfolio-worthy website demonstrating both front-end technical skills and strong understanding of DSA concepts.

---

## 7. Timeline

**Project Proposal Submission:**

**24 November 2025**

**Final Project Submission:**

**1st Week of January 2026**

---

## 8. Conclusion

This project will help new programmers understand the inner workings of key data structures through interactive visualizations, making learning more intuitive and effective. By combining modern web development technologies with foundational algorithmic concepts, the project not only fulfills academic goals but also serves as a strong portfolio piece demonstrating both technical and conceptual mastery.

---