

## 3507 - Final Project:

### Interactive Web Platform for Learning Data Structures and Algorithms

Develop a comprehensive web-based learning platform designed for computer engineering students to explore, understand, and visualize fundamental data structures and algorithms. The platform should provide interactive, step-by-step demonstrations that help users grasp both the logic and internal mechanics of each operation.

## Required Features

### 1. Data Structure Visualizations

Include fully interactive modules for:

- **AVL Trees** (insert, delete, search, rotations)
- **Binary Search Trees**
- **Linked Lists** (singly, doubly, and circular)
- **Dynamic and Static Lists**
- **Stacks and Queues** (optional enhancement)

Each module should:

- Display structure changes in real time.
- Highlight key operations (e.g., rotations, pointer updates).
- Allow user-controlled input (insert, delete, search).

### 2. Sorting Algorithm Visualizations

Implement animated visualizations of:

- **Bubble Sort**
- **Quick Sort**
- **Merge Sort**
- **Selection Sort / Insertion Sort**

Animations should:

- Show comparisons and swaps step by step.
- Offer controls such as play, pause, restart, and speed adjustment.

### 3. Searching Algorithms

Add interactive demonstrations for:

- **Linear Search**
- **Binary Search**

Visuals should highlight the element currently being inspected to reinforce algorithm flow.

## 4. Front Page (Dashboard)

Design a clear and intuitive landing page that:

- Lists all available algorithms and data structures.
- Provides short descriptions and learning objectives for each module.
- Allows users to select a topic and immediately begin interacting with it.

## 5. User Interaction & Step-by-Step Feedback

For every module:

- When a user inserts or deletes a number, the structure must update dynamically.
- Each step of the algorithm should be visually explained (e.g., “comparing nodes,” “performing left rotation,” “swapping elements”).

## 6. Additional Improvement

- Allow users to generate random datasets for testing.
- Provide performance metrics (time complexity, number of comparisons, number of swaps).
- Include a “learning mode” and “challenge mode.”
- Add color coding to highlight active nodes, pointers, or array elements.

**Good Luck...**