Designing and Implementation

Algorithm Visualizer

WebApp

Team Members

Venkatesh A

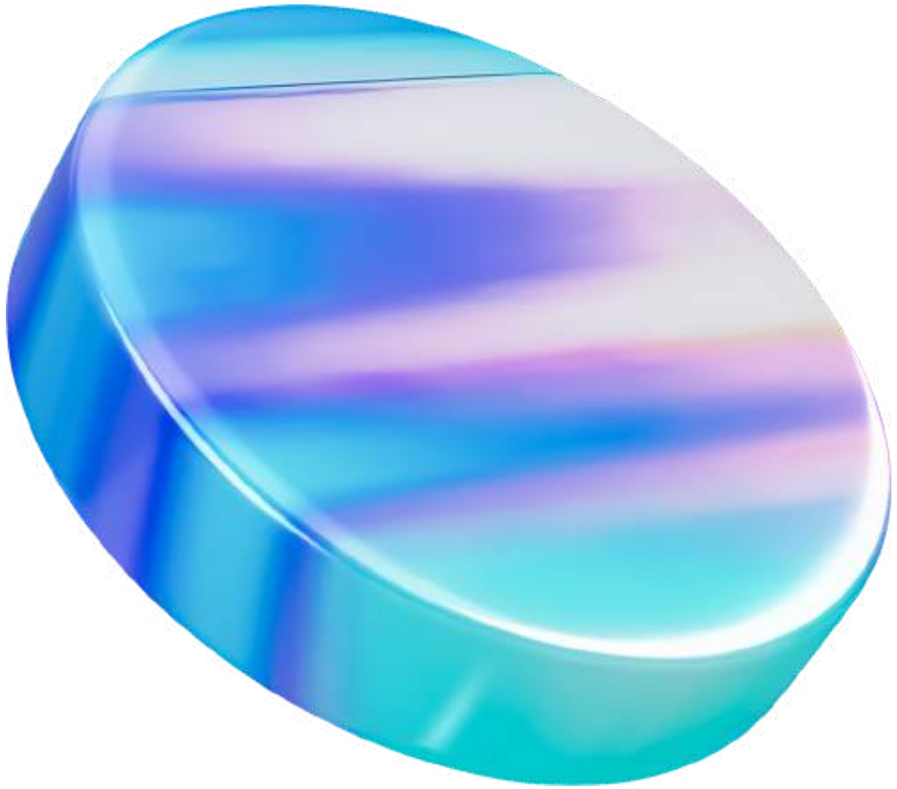
Barathraj B

Chithukati Balaji

Mohamed Akram

Project Guide

Dr C Anbuananth



Motivation

Why Algorithm Visualizer?

- Understanding algorithms can be challenging.
- Traditional text-based explanations often fail to capture dynamic processes.
- Inspired by struggles faced by peers, this project bridges the gap with a visual and interactive tool.

Project Goal

- Enhance understanding of algorithm execution through visual interaction.
- Provide an accessible, web-based platform for students and developers.
- Make learning algorithms intuitive, interactive, and fun.

Tools and Technologies

- **Frontend:** HTML, CSS, JavaScript, ReactJS
- **Backend (if needed):** Node.js
- **Development Environment:** Neovim IDE, Vite

Key Modules

User Interface Module

- Home Page
- Algorithm Selection
- Data Input
- Visualization Display

Algorithm Implementation Module

- Sorting Algorithms
- Pathfinding Algorithm

Interactive Features

1. Real-time visualization of algorithm execution
2. Step-by-step transitions with animations
3. Controls: Start, Pause, Restart, Step Forward, Step Backward
4. Modify datasets dynamically to explore different scenarios

How It Works

User Interaction:

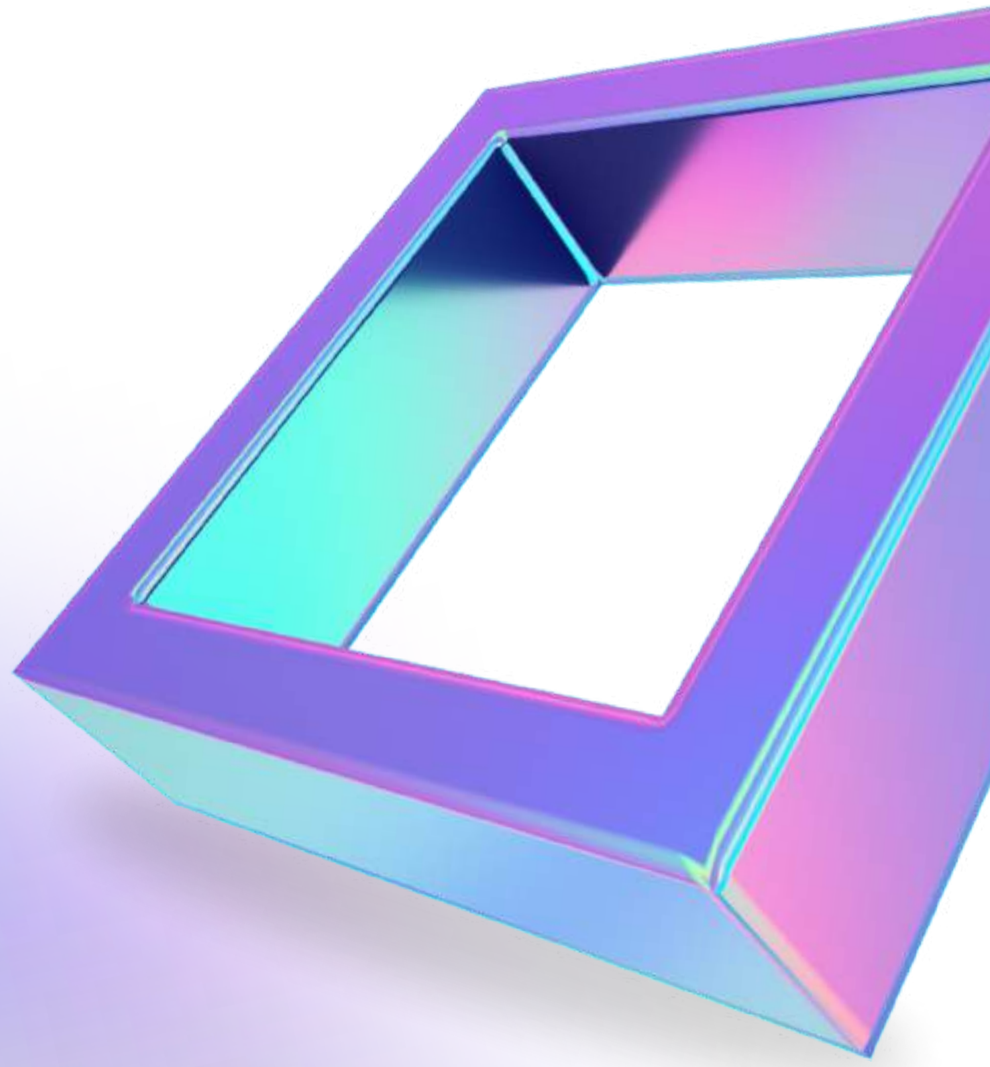
- Select category and algorithm.
- Input custom data or use predefined datasets.

Visualization:

- Algorithm logic is executed and visualized in real-time.
- Observe each step with detailed animations.

Interactive Controls:

- Adjust speed, pause, or step through execution.



Final Thoughts

- **Impact:** Bridges the gap between theory and understanding.
- **Benefits:** Interactive, engaging, and user-friendly platform.
- **Future Scope:** Expand to more algorithms and add gamification for better learning.

Thank You