**Synopsis**

**Name / Title of the project**: Algorithm Visualizer

**Team Members:**

1. SURYANSHU GUPTA(181500740).
2. SIDDHARTH SINGH(181500710).
3. KAVYA SINGH(181500319).
4. SPARSH GUPTA(181500723).
5. HARSHIT JAISWAL(181500255).

**The problem statement**: Algorithms and data structures form one cornerstone of an undergraduate computer science education. A technique for improving instruction in this critical area is to include algorithm and data structure visualizations.

**Reason for selecting the topic**: Analysis and design of algorithms is a great challenge for computer science student’s .We selected this project because we were fascinated by sorting algorithms, and we wanted to visualize them in action. This Project is a fully automatic visualization system with step-by-step explanations and comparison of sorting algorithms. Design principles and technical structure of the visualization system as well as its practical implications and educational benefits are presented and discussed.

**Objectives of the project**: The main objective of this project is to help beginners to be able to visualize the basic algorithms and get a better understanding of the underlying algorithms.

**Literature Survey / Feasibility Study**: This Project can help

* Instructors to illustrate algorithm operations in a lecture.
* Students about fundamental algorithms in a computer science course.
* Students learn about the basic operations of an abstract data type in a computer science laboratory.

**Future Scope:**

Our project focuses on algorithms and targets both teaching as well as tech industries so the future scope of our project seems to be endless. We all hope that every developer will contribute to our open source project. Regarding the time provided our project initially focuses on some basic sorting algorithms such as merge sort, selection sort and few others.

In future we aim to add as many algorithms as possible, first we would like to cover all sorting algorithms and after that our next target will be to work on algorithms which work on graphs or mainly path finding algorithms such as Dijkstra’s Algorithm and Best-First-Search.

**Methodology (Including a summary of the project):**

I was fascinated by sorting algorithms, and I wanted to visualize them in action. Algorithm visualizer project consists of sorting methods like Bubble, heap, merge and quicksort (We would try to include a few more methods).The main aim of our project is to visualize the sorting of an unsorted array having random numbers but including duplicate values by JavaScript on React. These arrays would be converted into bars by CSS (Cascading Style Sheets ) techniques .This project will consist of various technologies like CSS(Cascading Style Sheets ) ,HTML(Hypertext Markup Language) , JavaScript and this all would be done through react. We would provide a slider button in our project so that the user can make the array bigger or smaller as per his/her convenience. In addition to that, sorting speed could also be controlled in our project .The user just has to click the sort button click and visualize the algorithm. While sorting we included animations like colour pattern and used different colours for unsorted elements, already ordered elements, sorted elements etc. The important point of this project is to show the sorting algorithm process graphically since graphically sorting can be shown simply by swapping the lines.

CSS will play a major role in our project as it will help us to create the visual animations. All the source code for the algorithms will be written in JavaScript. Since, our project a web application therefore we are using react js which is a library for JavaScript.

***Summary***: This project would help one to understand the sorting algorithms in depth. We used React JS, CSS, JavaScript, HTML etc. to make this project.

**Hardware & Software to be used:**

Software: VS Code

Technologies: React JS, CSS, JavaScript, HTML

Hardware:

· RAM:- 4.00GB

· Processor:- Intel(R)Core(TM) i3-4005U CPU @ 1.70GHz

**Testing technologies to be used:**

Mainly two technologies used for testing in React

**Running tests-:**

Create React App uses [Jest](https://jestjs.io/) as its test runner. To prepare for this integration, we did a [major revamp](https://jestjs.io/blog/2016/09/01/jest-15.html) of Jest so if you heard bad things about it years ago, give it another try. Jest is a Node-based runner. This means that the tests always run in a Node environment and not in a real browser. This lets us enable fast iteration speed and prevent flakiness.

**Debugging tests:**

After opening that link, the Chrome Developer Tools will be displayed. Select inspect on your process and a breakpoint will be set at the first line of the react script (this is done to give you time to open the developer tools and to prevent Jest from executing before you have time to do so). Click the button that looks like a "play" button in the upper right hand side of the screen to continue execution.

**What contribution would the project make and where? :**

The contribution that an open-source project creates is endless, the only constraint which limits is our imagination. The project on our team is working and has an enormous amount of contribution. “Everybody should learn to program a computer, because it teaches you how to think.” - Steve Jobs. Our project aims at this very thought of Steve Jobs, not only normal people but sometimes many CS graduates or people associated with the technology industry fail to understand how an algorithm works. “Visualizing information can give us a very quick solution to problems. We can get clarity or the answer to a simple problem very quickly.” David McCandless. Our projects focus on how a person can learn the working of an algorithm by not only learning its source code but also by observing how it is working. Students who are studying in colleges or schools have privilege of having teachers who can teach them basics of algorithms but there are also many who dreams of learning to code and get placed in big tech giants without having Computer Science background. When some of these people try to understand certain algorithms, they puzzled themselves in the lengthy code but if at the same time they were able visualize that very code it will be easier for them to learn. Many a time experienced teachers also find it difficult to teach a certain algorithm on a teaching board but in today's era of technology and the facilities provided by the institutions when get combined with our project we can also solve that very problem.

“Scientists study the world as it is; engineers create the world that has never been.” —Theodore von Karman, Hungarian-American mathematician, aerospace engineer, and physicist. Our projects will make a small contribution to this very thinking but yes we are proud of the contributions which our project will make in shaping future engineers.

**Scope for extension into a major project:**

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers. Our scope here is the higher education in the field of computer science. So within the paper, we discuss the extension of standard methods of teaching algorithms,

using the whiteboard or slides, with the algorithm visualizations. According to they can be used to attract students’ attention during the lecture, explain concepts in visual terms, encourage a practical learning process, and facilitate better communication between students and instructors. Interactive algorithm visualizations allow students to experiment and explore the ideas with respect to their individual needs.