# # INTRODUCTION

The Algorithm Visualizer is a powerful web-based application designed to demystify the intricate workings and flow of algorithms and programs. Developed using the Bottle framework and incorporating a JavaScript library bundle, this application seamlessly integrates within a Django website to provide an immersive learning experience.

This project specifically caters to new computer science students who may find themselves perplexed by the complexities of algorithms and program execution. By leveraging a unique terminal interface, the Algorithm Visualizer accepts Python3 or higher version code and generates a step-by-step visual representation of the code's output. Users have the ability to manually navigate through each step, gaining invaluable insights into how algorithms unfold and memory is allocated for different code components.

Also, apart from the terminal the website also provides a unique way for its users to become contributors and work behind the development of the project by providing their own algorithms and snippets of code which could be published directly on the website after admin's approval and then those contributions could be accessed and visualized by the terminal.

This application serves as an invaluable tool for instructors and teachers as well. It empowers them to elucidate complex concepts, such as algorithmic processes and code workflows, with the aid of real-time visualizations. By leveraging the Algorithm Visualizer, educators can provide a dynamic and interactive learning environment that enhances students' understanding and engagement.

The aim of this project is to empower computer science students, both novice and experienced, to grasp the intricate workings of algorithms and code execution. Through the Algorithm Visualizer, we strive to alleviate the turmoil often associated with these concepts and provide a user-friendly platform for exploring and comprehending algorithms.

### # WHAT IT IS?

- A website – with its featured application being the terminal that visualizes programs, specifically algorithms – it basically visualizes programs i.e., instructions written in a programming language and as we know that – algorithms are just the subset of instructions

### # PROBLEMS FACED:

- Complete understanding of algorithms and their step-by-step work flow including memory allocation, traversal, last executed command and the going to be executed command
- Lack of adequate resources to study from or too much information present on the internet(infobesity) leads to confusion.
- Abstraction posed by algorithms, of how things are happening behind the written code
- Complexity: the inability of being able to calculate the steps involved in the piece of code leads to invalid calculations of algorithmic complexities.

### # SOLUTIONS POSED BY ALGORITHM VISUALIZER

- The code snippets are interpreted line by line and the number of steps is calculated based on which the graphical visualization is shown. Also, any related outputs required in the code are also shown in an output window given alongside the visualization.
- The website comprises of most famously known algorithms and user-friendly code snippets contributed by different users of the application and the administration which gives a real-time visualization and there's no need of going anywhere apart from the website.
- The users of the website may also contribute to the project by becoming contributors simply by registering themselves and making their code contributions which can be approved by the administration and subsequently be published on the website.

### # REQUIREMENTS

- The users of the application must have a basic prior knowledge of programming and should be aware of things such as syntax of a programming language.
- If the user is also a contributor, then they must take into account the conditions on which their code will be accepted by the administration which are also clearly stated alongside the posting section such as, the code snippet must execute within a 1000 number of steps and should not be recursing infinitely, the code should be free of syntax errors, etc.

### # SCOPE OF THE PROJECT

- As clearly stated, the algorithm visualizer is restricted to just one programming language which is python3 and above versions. So, one advancement in this would be generalize it for more than one programming languages.

- The code that we input in the terminal should execute within 1000 steps which is because the visualization may reach out the limits of the web page and would eventually create an overhead for the browser which will make it slow. So, solving this problem should be a priority in the future.
- A major increment to this project would be to provide the documentation of each and every famously known and studied algorithm and data structure implementation alongside its visual interpretation with the help of algorithm visualizer all in one place.

**# THINGS I LEARNT:**

- Integrated workflow of the things I studied in web engineering and database management which is the use of frameworks to integrate static web pages, creating their urls and routing their request which comes under both frontend backend. Creating the database and fetching and storing data into it upon POST and GET requests. Learning the complete structure of how the work flow happens within a website and how different web development models are implemented such as agile, waterfall etc.
- Design and modelling of a software before its implementation in the SRS(software requirement specification) I built in 5th semester.