iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** XI **Month of publication:** November 2021

DOI: https://doi.org/10.22214/ijraset.2021.39093

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue XI Nov 2021- Available at www.ijraset.com

An Approach to Develop Web-Based Application for Simulation and Visualization of Operating System Algorithms

Kartik Garg[1], Dhaval Maniyar[2], Harshit Gupta[3], Kirti Rathore[4], Krupi Saraf[5]
[1, 2, 3, 4]*Student, Department of Computer Science and Engineering, Acropolis Institute of Technology and Research, Indore, Madhya Pradesh, India.*
[*5]*Professor, Department of Computer Science and Engineering, Acropolis Institute of Technology and Research, Indore, Madhya Pradesh, India*

***Abstract:** Learning about algorithms and different system concepts related to programming is a difficult task and often becomes complex for students to grasp. If ever a concept is being learned using visualization techniques it becomes easy to learn and remember. There are many Algorithm visualizers built for it, though students may interpret them in a wrong way and that is why it should be appropriate and correct. Based on the previous works this paper describes an effective way to learn the various complex algorithms of the operating system in which we have considered various levels of interactivity, that includes zero interactivity, partial interactivity, and complete interactivity that will be discussed further in detail.[10]*
***Keywords:** Algorithm, Visualization, Operating System, Simulator, Programming, FCFS, SJF, etc*

I. INTRODUCTION

This document is a temp Learning different algorithms and various processes and developing a concrete understanding of them is still a difficult and complex task for many students especially newbies and even for teachers to teach that concept in a more effective manner. An algorithm describes a process that is both Abstract and dynamic.[8]

Operating Systems is one of the most important fundamental subjects of Computer Science containing lots of algorithms that become hard to learn and understand their exact workflow. If the process is properly visualized in detail with proper use of resources, then it would become a very easy task for everyone to understand it more effectively.

Simply going through the animated videos does not guarantee much deeper understanding as compared to the fully interactive and simple interface that allows everyone to go through the algorithm step by step by performing and seeing that algorithm working at their pace of understanding. In this paper, we present an idea to visualize the different algorithms of the operating system with the help of our web-based application OSUM that will help the students to learn the concepts easily, quickly and also get a deeper knowledge about it.[9]late. For questions on paper guidelines, please contact us via e-mail.

II. LITERATURE REVIEW

A. *Hansen S., Narayanan H. N., Hegarty M., Designing educationally effective algorithm visualizations, Journal of Visual Languages and Computing, 13, 291-317, 2002.*

(Designing Educationally Effective Algorithm Visualizations | Request PDF) talks about various visualizing algorithms made by different people but compares their effectiveness and provides the reasons as to why they are not much successful. It discusses in detail about the points which will result in high pedagogical value and provides a clear understanding of the algorithm and helps to learn the algorithms more effectively. It describes the methods that can be used in visualization which will help in successful algorithm visualization with a high degree of successful learning. It also suggests the inclusion of user interaction with the software with the aim to increase the user thinking and self-explanation.[1]

B. *Vrachnos E, Jimoyiannis A. Dave: A dynamic algorithm visualization environment for novice learners. In2008 Eighth IEEE International Conference on Advanced Learning Technologies 2008 Jul 1 (pp. 319-323). IEEE.*

(PDF) DAVE: A Dynamic Algorithm Visualization Environment for Novice Learners it describes the development of various algorithm simulation environment which helps the students to understand and learn various complex problems very easily by simply iterating through the visualization steps but also talks of a dynamic algorithm visualization method that allows the students to do experimentation with the pre-designed algorithms as well as the algorithms designed by their own and then talks about the effective of learning using such environments.[2]

C. *Supli AA, Shiratuddin N, Zaibon SB. Critical Analysis on Algorithm Visualization Study. International Journal of Computer Applications. 2016 Jan 1;150(11).*

(PDF) Critical Analysis on Algorithm Visualization Study it talks about the principle of interactive algorithm visualization on hybrid mobile application with the aim to inculcate effective learning among students on data structures and algorithms. It proposes several design guidelines and describes the gaps present in the hybrid model described in the study. It also compares the present desktop and mobile algorithm visualization software's present today and compares the drawbacks and also suggests that there is only sorting algorithm visualizer present for mobile platforms and that there is a need for an effective mobile algorithm visualization application.[3]

D. *Korhonen A, Malmi L. Algorithm simulation with automatic assessment. InProceedings of the 5th annual SIGCSE/SIGCUE ITiCSEconference on Innovation and technology in computer science education 2000 Jul 13 (pp. 160-163).*

(PDF) Algorithm simulation with automatic assessment ,the correctness of any algorithm visualizer is discussed in this work. There are many software available for the visualization but students many a times do not get the appropriateness of the results made available by the software and thus take the concept in the wrong way. Therefore, this paper talks about the need of a software that provides confirmation for the correctness of the interpretation made by the visualizer by providing feedback. This paper presents a learning environment with the combination of simulation, visualization and automatic assessments which provides accurate results with the help of feedback mechanism.[4]

Table -1: Review of Existing System

S. no	Existing System/Application	Advantages	Disadvantages	Gaps Identified
1.	OS Algorithm Simulator App	Several preemptive and on-preemptive algorithms	Only a few algorithms are available. Most of the people prefer online tools for this type of facility instead of Android application.	Some bugs , Problem with Storage
2.	AnimOS CPU-Scheduling	Visualize almost all CPU Scheduling algo. visualization is easy to understand.	Complicated to understand errors Takes more time to generate output.	Output table for some algorithms is wrong.
3.	CPU Simulator	CPU scheduling algorithms on a Gantt chart	Only a few algorithms are available. Most of the people prefer online tools for this type of facility instead of Android Application.	Problem occurred with preemptive (Priority Scheduling)
4.	CPU Simulator (CPU Scheduling)	CPU simulator resolves and graphs different CPU algorithms.	Only a few algorithms are available. Most of the people prefer online tools for this type of facility instead of Android Application.	Does not accept burst time more than 9. Generate some incorrect output. Sometimes Ads hide the content.
5.	CPU Scheduling algorithm visualization	Website for checking cpu algorithms.	Only a few algorithms are available. Most of the people prefer online tools for this type of facility instead of android Application	Generate incorrect output. Slow processing takes more time to generate charts.

III.METHODOLOGY

In our work we have implemented an algorithm simulator/visualizer that will help engineering students to learn various operating systems algorithms in a better and effective way. It will make their learning very easy and also accurate.
We have implemented it using Vanilla Js, HTML,CSS, Python 3 and Django. It has several modules each with the number of algorithms present in it. Some algorithms can be visualized step by step while some give direct detailed results on execution.

Some of the Algorithms that it supports currently are as follows:

1) Process Scheduling (SJF,FCFS,RR,SRTF)
2) Process Synchronization (Producer Consumer Problem,Reader Writer Problem,Dining Philosopher Problem,Cigarette smokers Problem,Smoking barber Problem).
3) Banker's Algorithm
4) Memory Management.
5) Page Replacement (Least Recently Used,Most Recently Used,Optimal Page Replacement,etc.)
6) Disk Scheduling Algorithms
7) File Allocation Technique (Contiguous File Allocation,Linked File Allocation)
8) File Management (Single level Structured directory, Two level Structured directory, Tree Structured directory)

In our model the main logic of the simulation and calculation is written in vanilla js and python 3 and then the application different modules are linked using the Django framework and also easily managed using it.
The exact workflow of the application can be understood from the fig A.

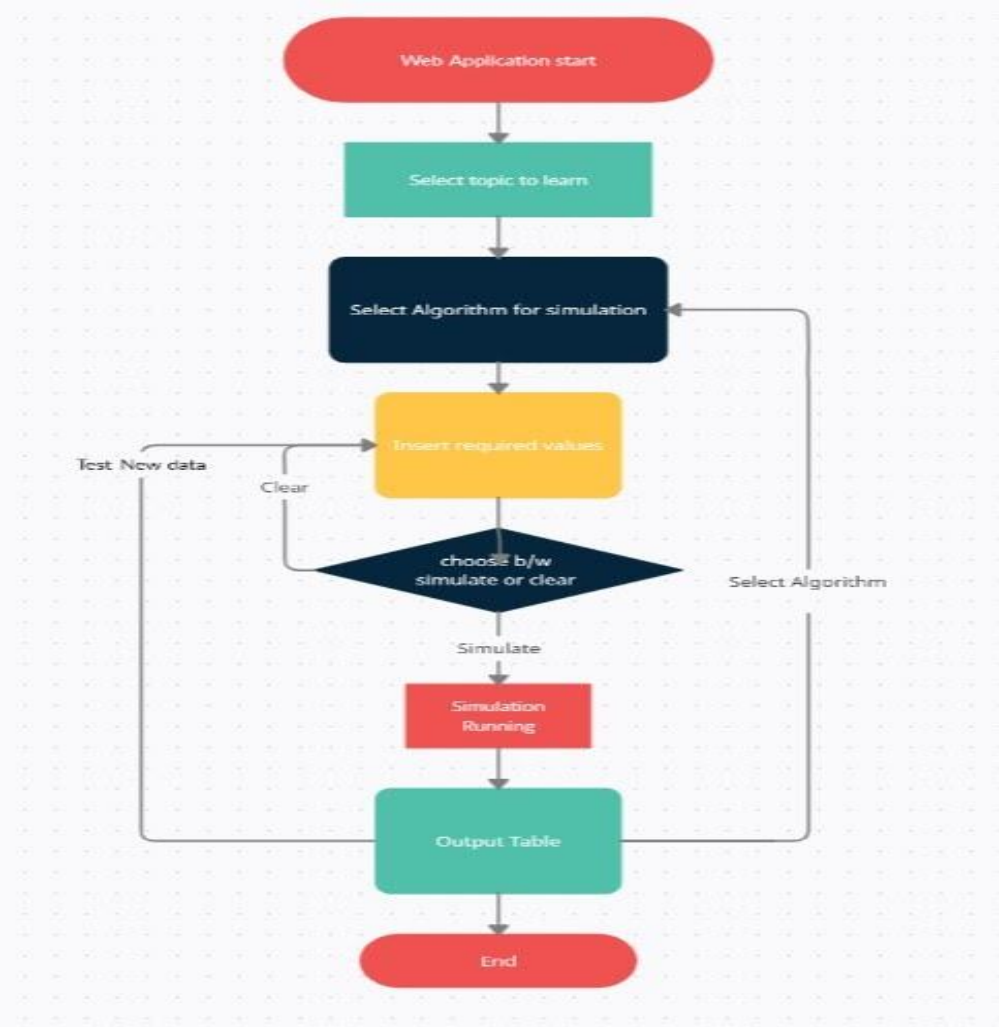


Fig 1 Flow diagram of Algorithm Sim.

Now for an illustration of the clear working of one of the Process Synchronization algorithms named as Dining Philosopher problem is shown in below images.

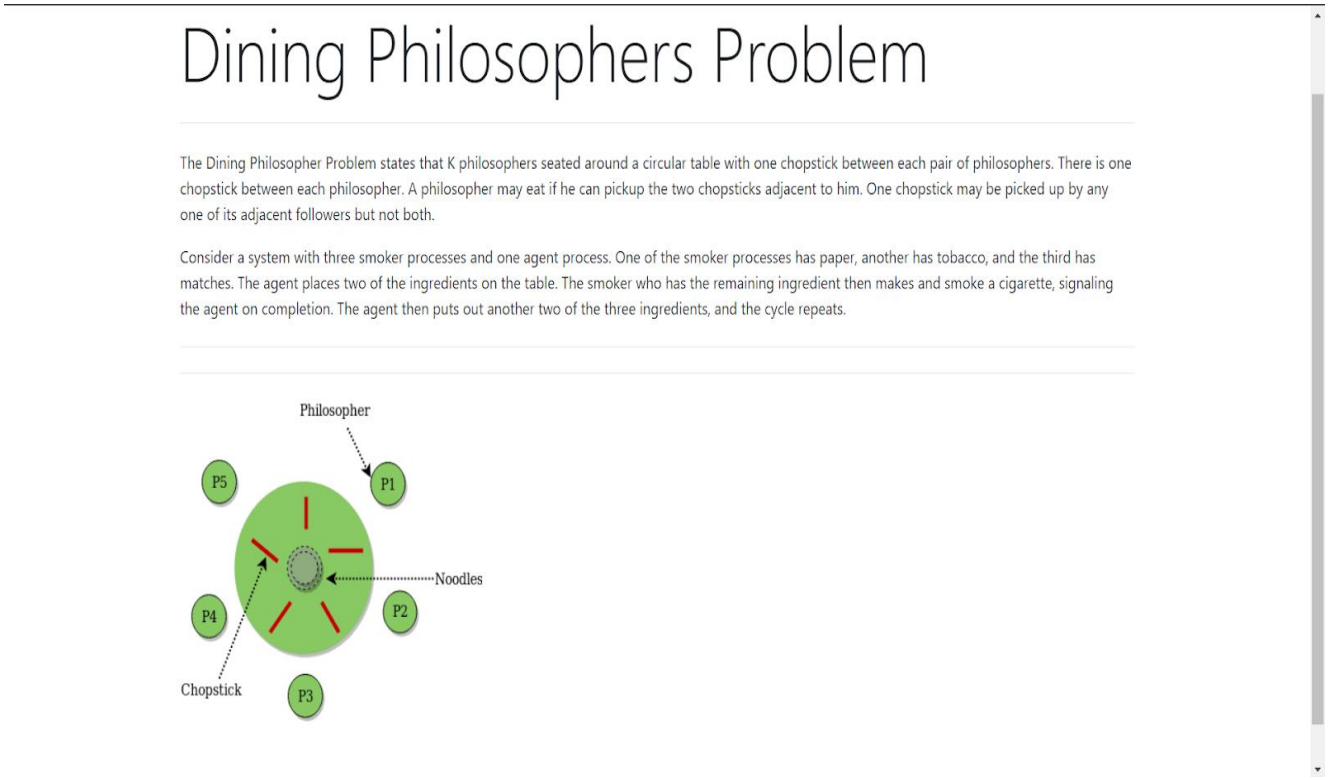


Fig 2.1

First the user is given a brief information about the particular algorithm as shown in the above fig. After that the user moves on the simulation process.

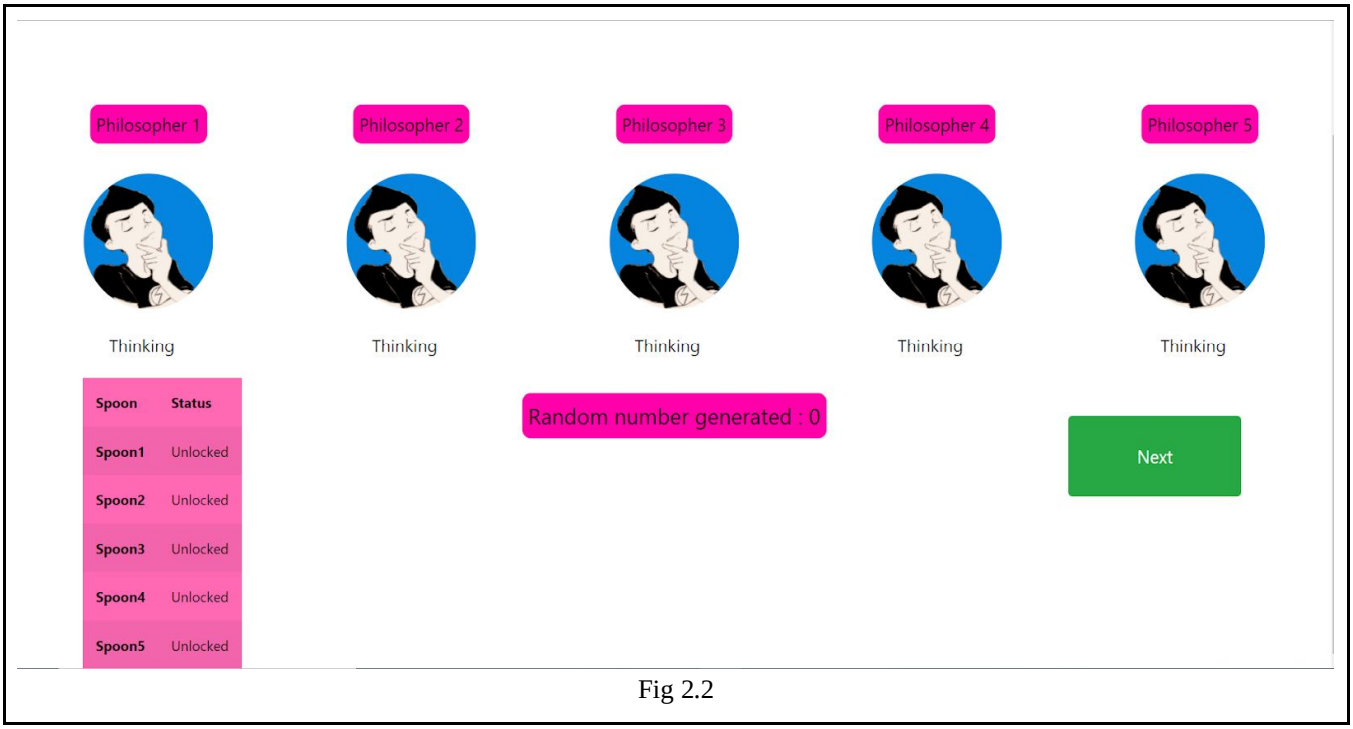


Fig 2.2

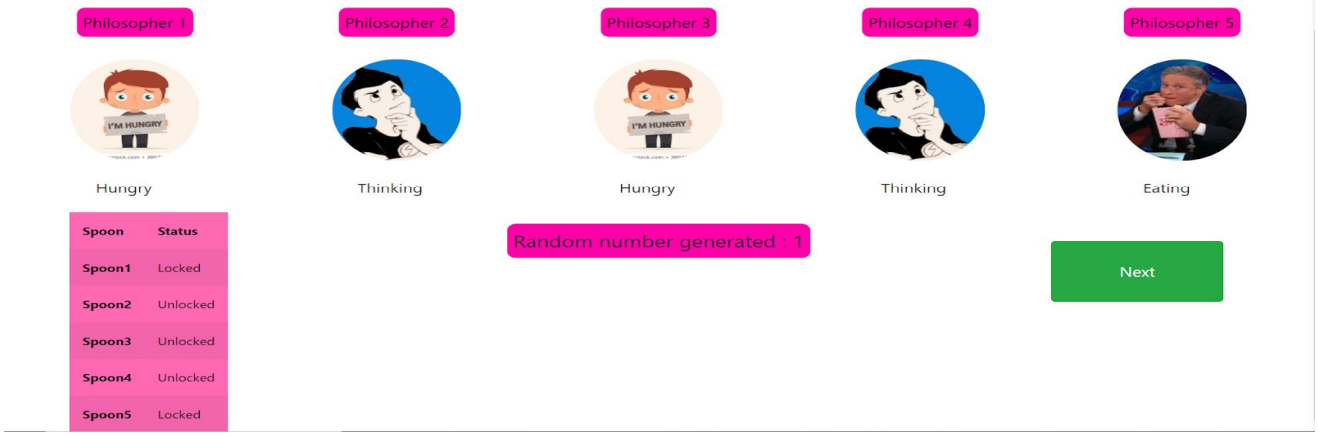


Fig 2.3

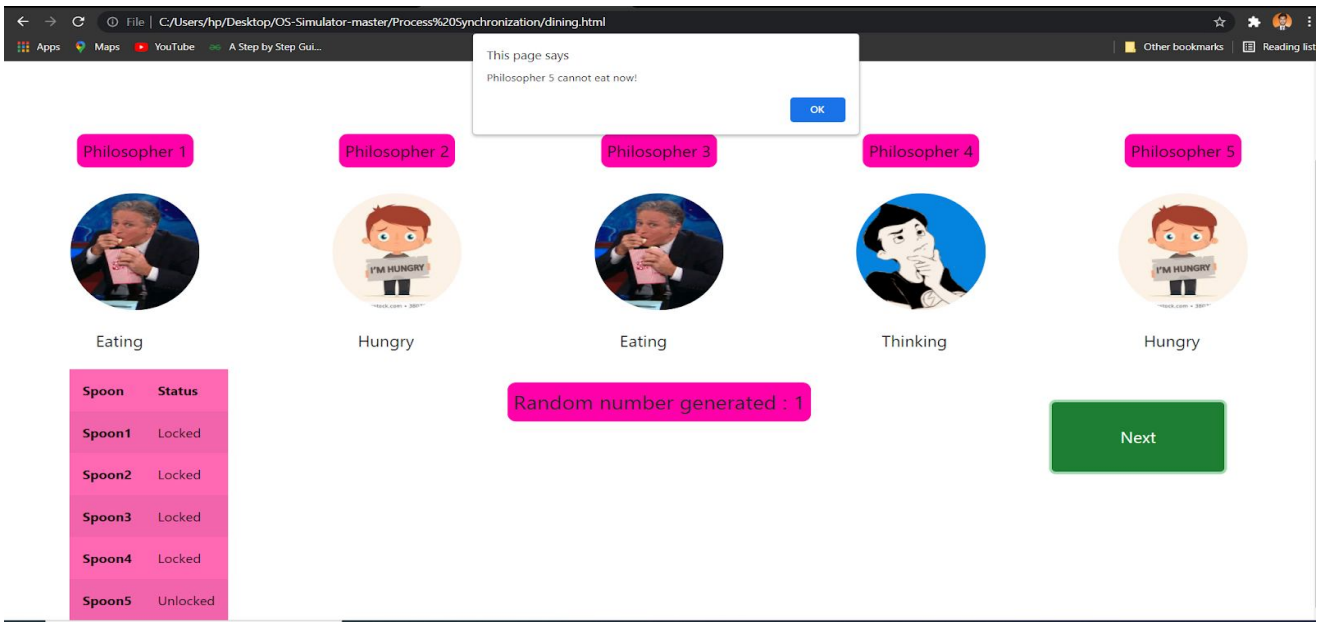


Fig 2.4

In the above images the stepwise visualization of algorithm is shown as initially all the people are thinking and all the spoons are free but as we move forward philosopher 1&3 becomes hungry whereas 2&4 are still thinking and 5 is in eating mode which results in spoon 1 and 5 as locked and others as free. Further, 1&3 starts eating and 2&5 becomes hungry but now no more philosophers can eat as all the 4 spoons are locked and only one is free.

In this way the algorithm is visualized very easily and effectively through pictorial representation and also it automatically generates all the output with full accuracy which gives the user to learn and try the algorithm on different cases.

.

IV. IMPLEMENTATION AND APPLICATIONS

The algorithm Simulator/visualizer can be developed in various forms and can be present to the user.

1) It can be built in the form of a web application which we will be doing in our work.

2) It can also be delivered as a mobile application.

3) It can also be presented in hybrid format by building REST API and then calling the REST API from the mobile application by sending the required data.[5]

V. RESULTS AND DISCUSSIONS

There are various Operating System Algorithm Simulator/Visualizers present. The simulation of various algorithms helps us to evaluate them more efficiently and precisely. The arrival and burst time of different processes is taken as an input from the user. Before that a brief detailing of the algorithm is shown to the user which gives a clearer idea about that algorithm to the user. After getting the required process times the average waiting time is calculated at each step as the user moves the algorithm forward. The user can see the simulation of different algorithms like FCFS, SJF, Priority Scheduling and many other algorithms present.[6]

Some Screenshots of our work are shown below. In the fig 2, the simulation of the FCFS algorithm is shown as in this algorithm step wise data cannot be shown so direct results have been shown.

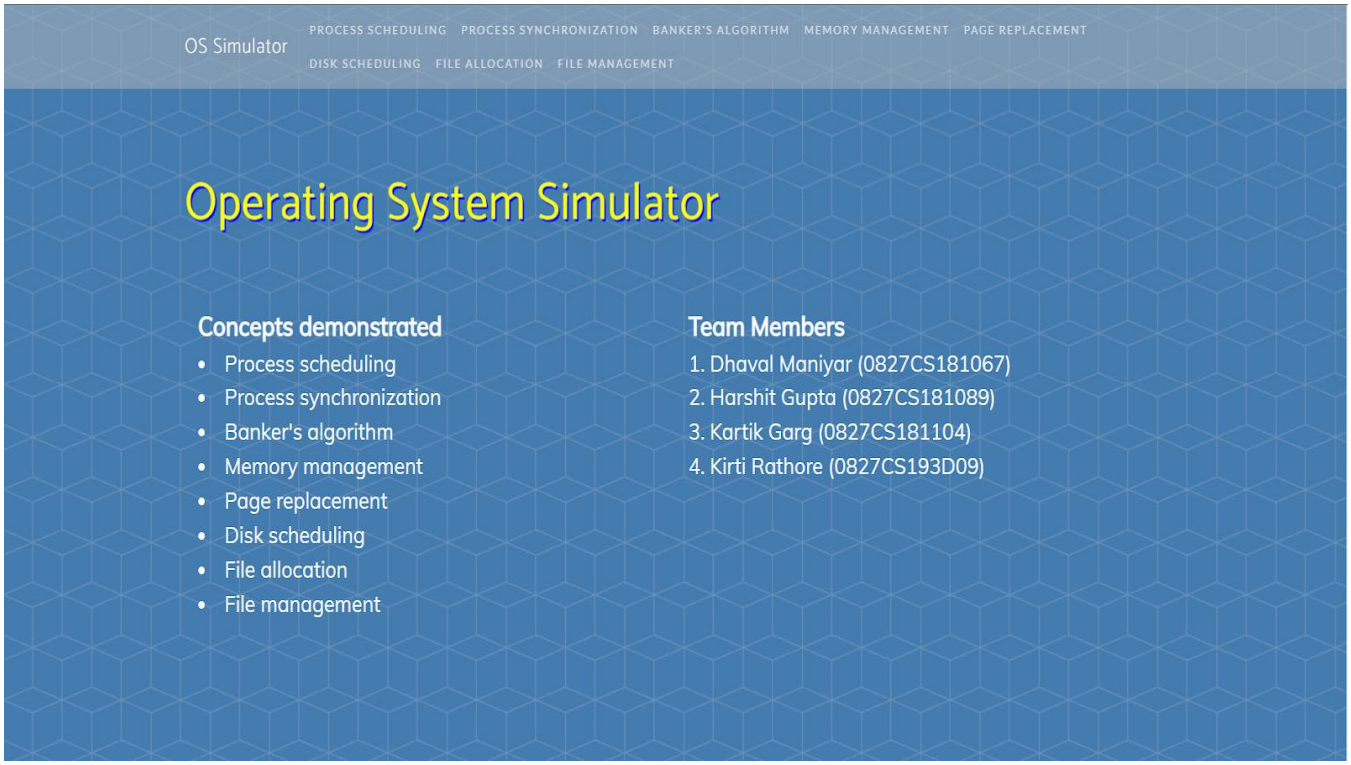


Fig 3 List of Algorithms supported by the application

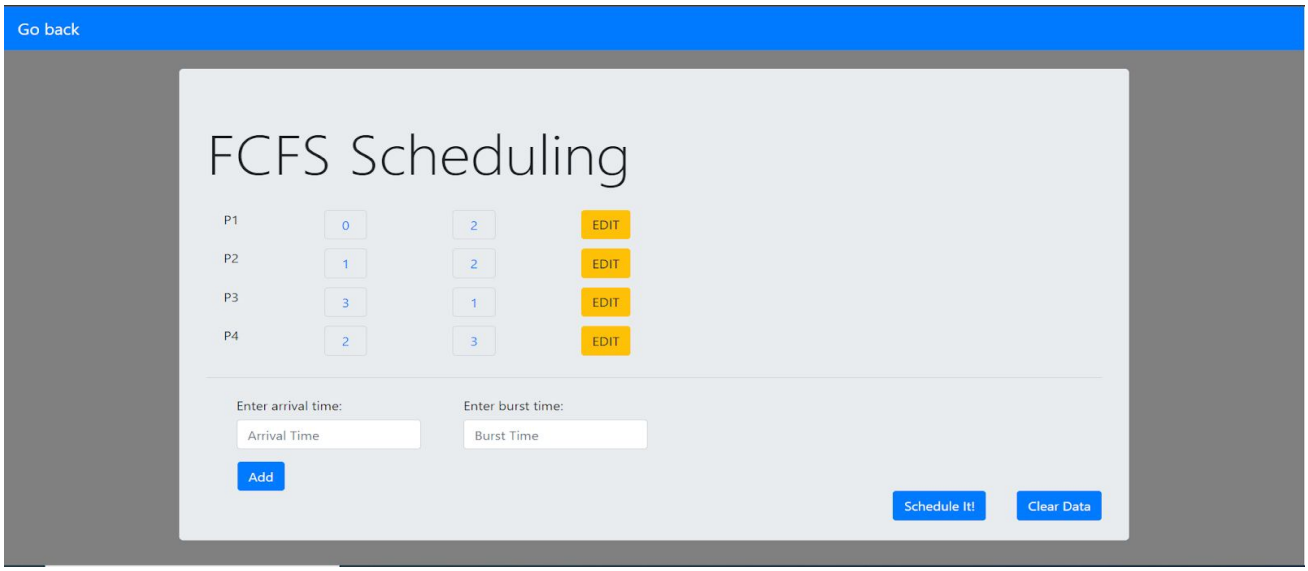


Fig 4.1 Input by user for different process

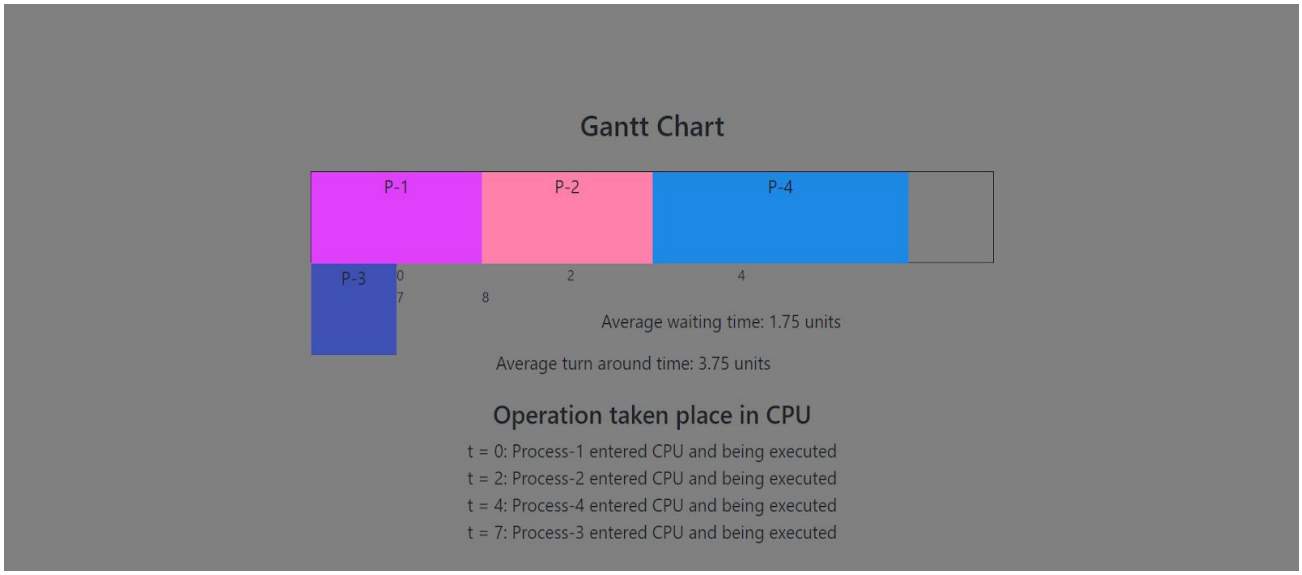


Fig 4.2 Output of FCFS in form of Gantt chart and each step

Output Table

Process ID	Arrival Time	Burst Time	Completion Time	Waiting Time	Turnaround Time
1	0	2	2	0	2
2	1	2	4	1	3
4	2	3	7	2	5
3	3	1	8	4	5

Fig 4.3 Output table containing all the process time for FCFS

Fig 4 FCFS Scheduling

VI. CONCLUSIONS

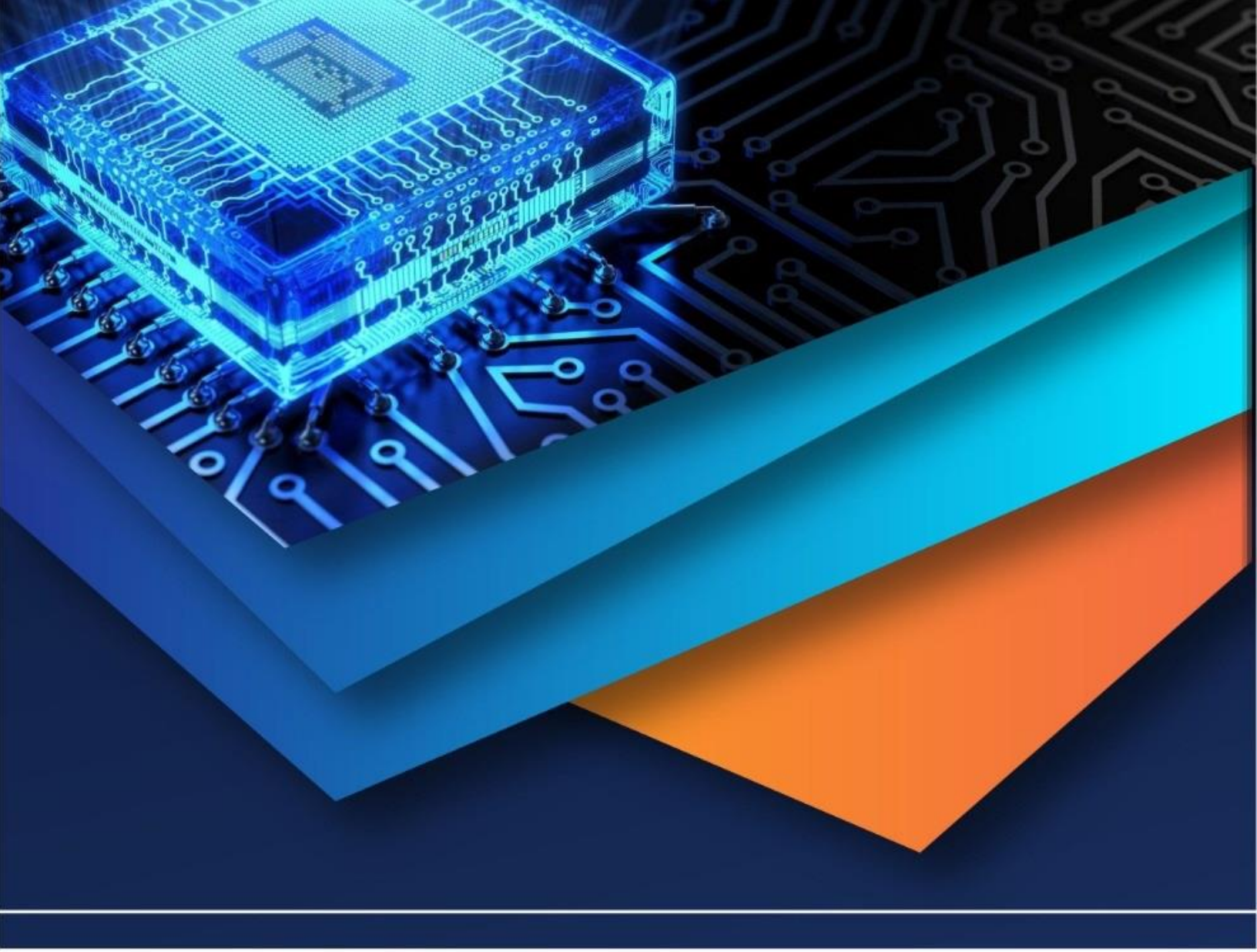
In our work We have developed a web application algorithm Simulator Cum visualizer Which will help the students of the engineering background to easily learn all the algorithms and complex concepts present in the Operating System subject. The application Simply allows the user to navigate between the different algorithms present in os and then learn it using the information present there. Not only limited to this the user can then give the inputs and run the algorithms according to its choice and see the stepwise simulation of it with the necessary outputs shown on the screen at each step. We have carefully worked on all the gaps present in already developed simulators and our application is much better as compared to the present works. In future further addition of algorithms of different subjects can also be done.[7]

VII. ACKNOWLEDGEMENT

We as the authors would like to extend a special thanks of vote to the reviewers of this paper for their valuable suggestions to improve this paper.

REFERENCES

[1] Hansen S., Narayanan H. N., Hegarty M., Designing educationally effective algorithm visualizations, Journal of Visual Languages and Computing, 13, 291-317, 2002.
[2] Vrachnos E, Jimoyiannis A. Dave: A dynamic algorithm visualization environment for novice learners. In2008 Eighth IEEE International Conference on Advanced Learning Technologies 2008 Jul 1 (pp. 319-323). IEEE.
[3] Supli AA, Shiratuddin N, Zaibon SB. Critical Analysis on Algorithm Visualization Study. International Journal of Computer Applications. 2016 Jan 1;150(11).
[4] Korhonen A, Malmi L. Algorithm simulation with automatic assessment. InProceedings of the 5th annual SIGCSE/SIGCUE ITiCSEconference on Innovation and technology in computer science education 2000 Jul 13 (pp. 160-163).
[5] Hundhausen CD, Douglas SA, Stasko JT. A meta-study of algorithm visualization effectiveness. Journal of Visual Languages & Computing. 2002 Jun 1;13(3):259-90.
[6] Carson E., Parberry I., and Jensen B., "Algorithm explorer: visualizing algorithms in a 3D multimedia environment", 38th SIGCSE Technical Symposium on Computer Science Education, pp. 155-159, 2007.
[7] Osman, Waleed Ibrahim, & Elmusharaf, Mudawi M. (2014). Effectiveness of Combining Algorithm and Program Animation: A Case Study with Data Structure Course. Issues in Informing Science and Information Technology, 11.
[8] Halim, Steven, Koh, Zi Chun, Loh, V, & Halim, Felix. (2012). Learning Algorithms with Unified and Interactive Web-Based Visualization. Olympiads in Informatics, 6, 53-68.
[9] Urquiza-Fuentes, Jaime, & Velázquez-Iturbide, J Ángel. (2009). A survey of successful evaluations of program visualization and algorithm animation systems. ACM Transactions on Computing Education (TOCE), 9(2), 9.
[10] Simonák, Slavomír. (2013). Algorithm Visualization Using the VizAlgo Platform. Acta Electrotechnica et Informatica, 13(2), 54.
[11] Lazaridis, Vassilios, Samaras, Nikolaos, & Sifaleras, Angelo. (2013). An empirical study on factors influencing the effectiveness of algorithm visualization. Computer Applications in Engineering Education, 21(3), 410-420.
[12] Haajanen J, Pesonius M, Sutinen E, Tarhio J, Terasvirta T, Vanninen P. Animation of user algorithms on the Web. InProceedings. 1997 IEEE Symposium on Visual Languages (Cat. No. 97TB100180) 1997 Sep 23 (pp. 356-363). IEEE.
[13] Singh V, Gabba T. Comparative study of processes scheduling algorithms using simulator. Int'l Journal of Computing and Business. 2013 May.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY