ReadMe for OS Project

❑ Introduction:

Through this website, we aim that the people who visit it learn about page replacement algorithms, an integral part of the Operating Systems subject. We have taken care of both the theoretical and practical aspects of the topic. In theory, we have explained the basic idea and concept of the algorithms in a very understandable and lucid language. The theory pages are static pages mainly implemented using HTML5 and CSS3 for styling. The layout and design are kept very elegant and simple for better viewing and easy navigation. The pages are responsive and can be opened on mobile phones, tablets, laptops, and computers without any problem. The home page of the website is the index.HTML5 file in our code folder. We have implemented 6 algorithms, namely First In First Out, Belady's Anomaly, Optimal Page Replacement, Least Recently Used, Most Recently Used, and finally Random Page Replacement. It is further linked to the pages of the different algorithms that we have implemented. The pages for the algorithm too are mainly static pages implemented in HTML5 and CSS3. Each page contains the theory of the algorithm along with an example. Further, for the practical aspect of the concepts, we have provided a live simulation for the algorithms about which the document states in the later part. We have tried to simulate the environment of a lab for page replacement algorithms where you can put in different values and find out the answer for yourself. The algorithms are mainly implemented using HTML5 CSS3 as well as JavaScript for reactivity. All the images that we have used are kept in the images folder. All the HTML5 files for the algorithms implemented can be found by opening the Page Replacement Algorithms folder. The CSS styling files for each page are in CSS and CSS_1 folder. JavaScript files for different pages can be found in JS folders. We mainly have coded on VS Code and Sublime Text.

❑ Technologies Used:

- HTML5
- CSS3
- JavaScript
- Bootstrap

❑ **Algorithms Implemented**: All the below mentioned algorithm pages are are static pages mainly implemented using HTML5 and CSS3 for styling:

➢ On clicking "First in First Out" algorithm from the home page you will be redirected to the First in First out algorithm page where you will be able to see the basic introduction and example of the algorithm.

- ➢ On clicking “Least Recently Used” algorithm from the home page you will be redirected to the Least Recently Used algorithm page where you will be able to see the basic introduction and example of the algorithm.
- ➢ On clicking “Most Recently Used” algorithm from the home page you will be redirected to the Most Recently Used algorithm page where you will be able to see the basic introduction and example of the algorithm.
- ➢ On clicking “Optimal Page Replacement” algorithm from the home page you will be redirected to the Optimal Page Replacement algorithm page where you will be able to see the basic introduction and example of this algorithm.
- ➢ On clicking “Random Page Replacement” algorithm from the home page you will be redirected to the Random Page Replacement algorithm page where you will be able to see the basic introduction and example of the algorithm.

On clicking the Live Simulation from the home page you will be redirected to the live simulation page. In the live simulation,the algorithms have been implemented using HTML5 CSS3 and JavaScript.Once you open the live simulation page,you will be able to implement all the page replacement algorithms one at a time or all at once too as per your wish. You will have to follow the below mentioned steps:

- Firstly,you will have to enter the page data.We have also kept an option to generate random page data.
- You can change the minimum and maximum page length and the buffer size.
- By clicking on the algorithm you want to execute,you will be able to see the step by step implementation of the algorithm,number of page faults and the execution time.
- We have also provided the graphical representation of the algorithm where you will be able to see the graph between the number of page faults and the particular algorithm. You will also be able to compare two or more than two algorithms through this graphical representation.