

A portrait of a young man with dark, curly hair, smiling at the camera. He is wearing a bright yellow short-sleeved shirt. He is holding a white tablet in his hands. In the background, there is a closet with several shirts hanging on hangers.

Rank Distribution and Seat Allotement

Vandan Patel (202001041)
Shrey Pobaru (202001021)

Problem Statement

- A single platform which can effortlessly distribute ranks and allocate seats without any man power would be highly useful to educational institutes.
- The algorithm behind the platform will make sure that most of the resulting ties are resolved. Hence providing no unfair advantages to any candidate.
- So that no deserving candidate will be at loss.
- This system can be readily applied at school or college level institutes.



Approach

We developed an algorithm similar to that of JoSAA

We needed a system which can give priority to Total, Mathematics, Physics and Chemistry marks respectively. And if further needed, the priority would be given to older candidate.

We sorted the candidates according to the above mentioned priority order

Then we allocated the seats as per the preference provided by the candidates



Things we learned

To store such a large amount of students' data we needed a primary data structure which is versatile.

So, we did a little research and that's when we came across **vectors**. And throughout the project we learned much more about **STL in C++**.

We used **file handling** in order to store the large amount of data in a way that we can manipulate it as per our needs. And we stored the data in .dat files to observe it at different stages.
In this process, we learned a lot about file handling.

We felt as if showing the result in a console was doing injustice to the whole project. So we decided to go for a webpage. But we couldn't link C++ and HTML. So, we used file handling to generate a **HTML + CSS** code for the webpage in a separate file.



LIMITATION 1

The execution time is a little more than what one would expect, when dealing with larger data (Ex. 50000 candidates)

LIMITATION 2

We used a lot of dynamic storage in the process

LIMITATION 3

We showed the output on a HTML + CSS webpage, but we were unable to take input from the page.

**Thank you
We enjoyed the project :)**