**AIM:**

To create an online voting System using Linear Data Structures.

**OBJECTIVES:**

* Implementation of the Linear Data Structure-LINKED LISTS.
* Multiple methods creation for various tasks under the project.
* In this system, authorized voters can give his\her votes online without going to any physical polling station.

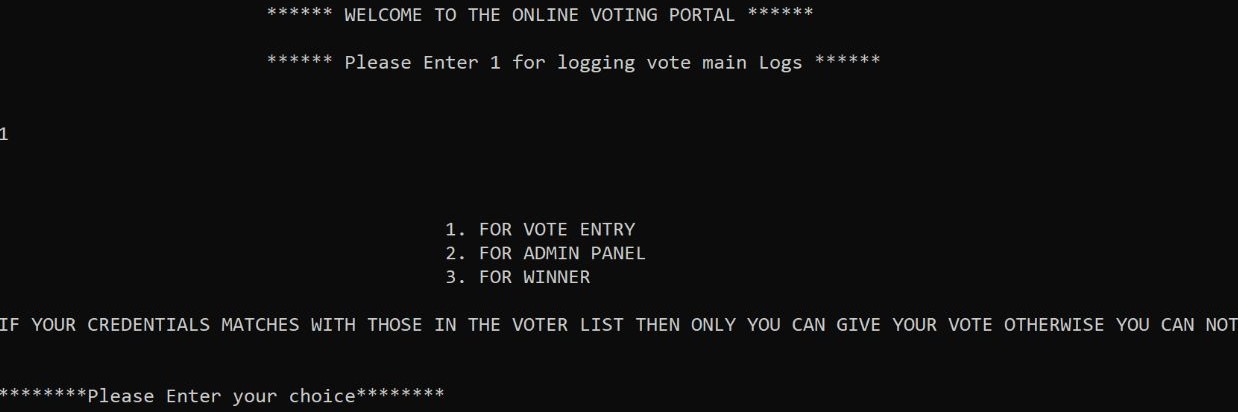
**METHODOLOGY:**

* Our application utilises the linear data structure ,Singly Linked Lists, to implement an voting system
* Linked list is a linear data structure ,in which elements are not sorted over contiguous memory locations.
* The elements in linked list are linked using pointers .
* A linked list is preferred over other structures is the that they are dynamic in nature and have faster deletion/insertion time.
* Singly linked list is preferred when we need to save memory and searching is not required as pointer of single index is stored.
* Linked lists are always better than Arrays as they do not lead to Memory Fragmentation.
* In our application ,we first the take input of credentials from the users to check they belong in the voter list based on their DSE registration no. (210968XXX)
* If they are eligible to vote then they are redirected to a voter list containing the names of the candidates or else they can retry for a maximum of 3 times failing to the portal will be terminated .
* After selecting the candidate you want , their name will be displayed and then you’ll return back to the main page (main logs)
* The admin panel will help you check the no. of votes each candidate has received and the check winner option displays the winner

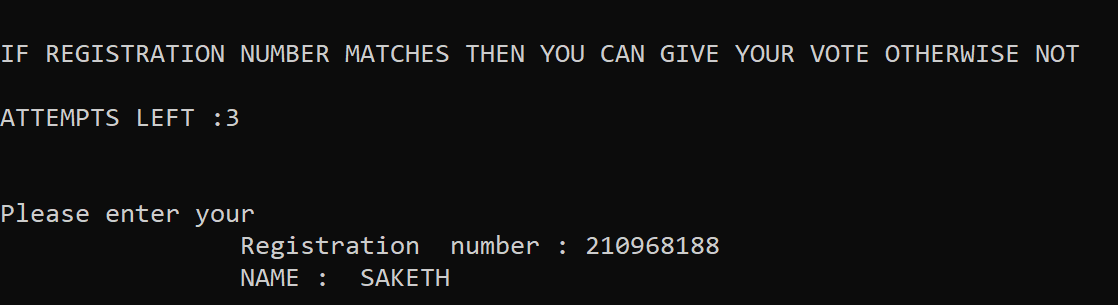
**Work Flow**

**Results Obtained**

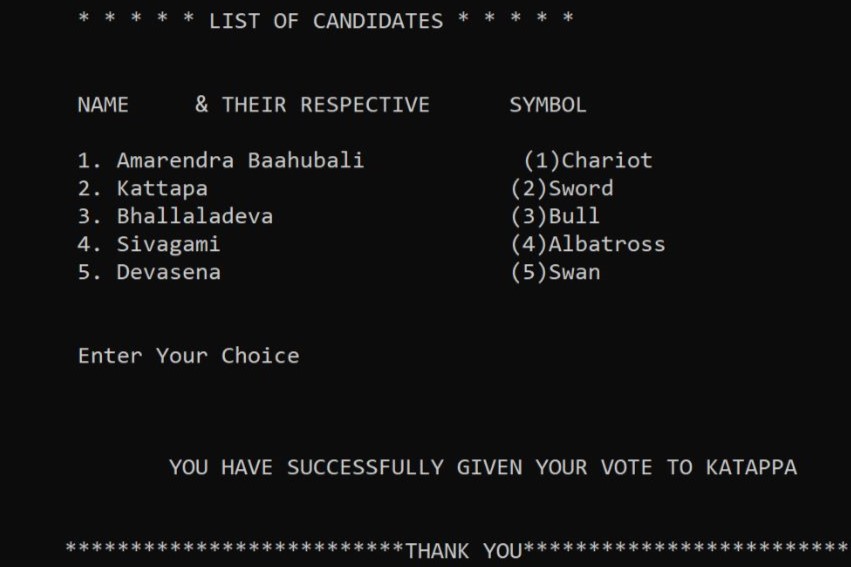
* As soon as you enter the portal by entering you’ll be shown the main logs as shown below



* Once you select vote entry, you’ll have to enter your credentials



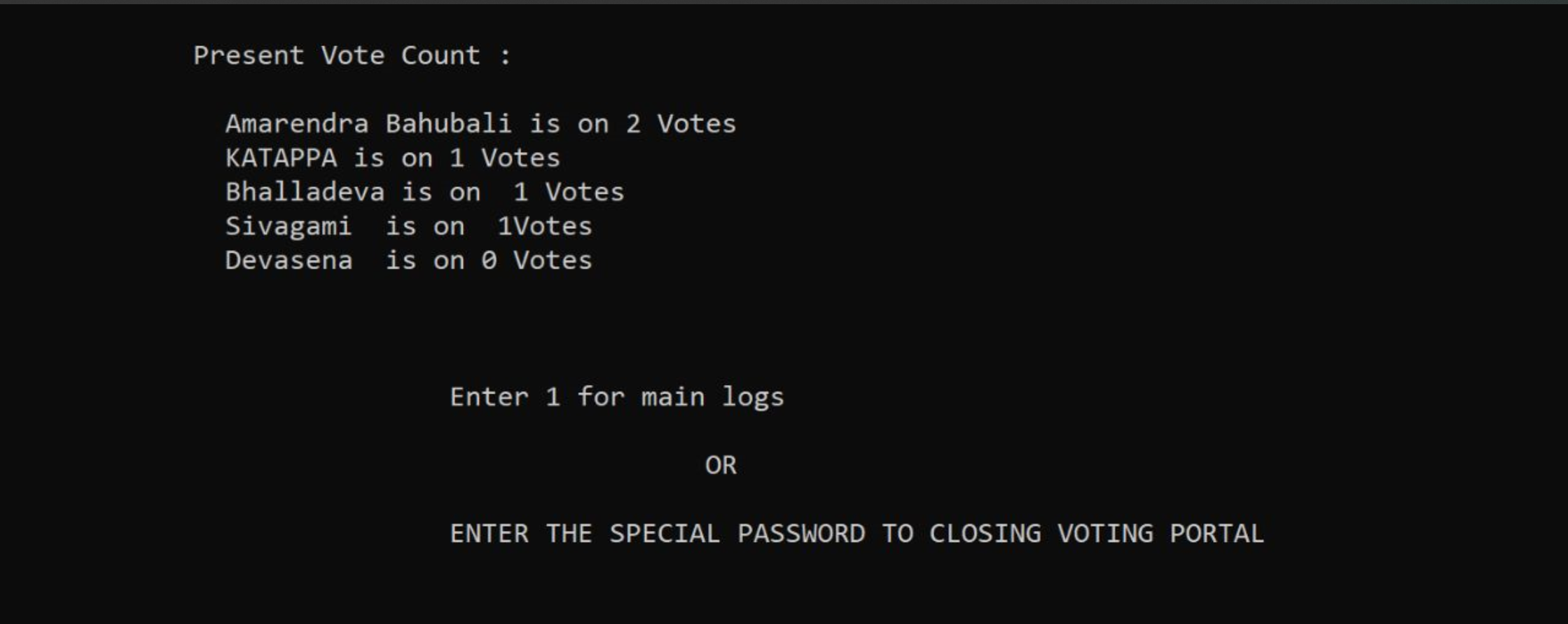
* You can choose the candidate of your choice over here



* Program Displays the current winner every time the winner function is called.

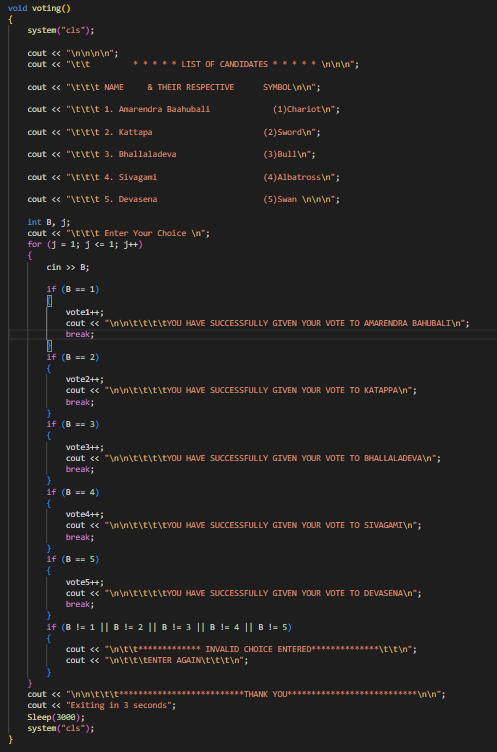
Text

Description automatically generated

* You can also access the voting count of all the candidates using the admin panel. 

**CONTRIBUTIONS:**

* I have helped in the process of developing the code for the **voting()** function which Shows the list of candidates, symbols, and their serial numbers that are participating in that particular election and takes the Serial number assigned to the candidate you want to vote as input and checks if the serial number entered matches with the candidate's pre-defined number. If yes then the vote count of that candidate increases by 1.
* If the entered input does not match the serial number pre-defined for the candidate. Then it will automatically redirect you to enter the input again.
* I also used the system("cls") function to clear the Output Screen Before entering the function and after exiting the function. So that the output looks more readable.



**CONCLUSIONS:**

* This project has been really helpful for me as I have learned the importance and implementation of linked lists in real-life applications and I would like to thank our teachers for the same.
* I’ve learned the following concepts by doing this program
* Implementation of data structures
* Singly-linked lists are useful data structures, especially if you need to automatically allocate and de-allocate space in a list.
* Structures for data encapsulation.
* Dynamic memory allocation using a new operator.

**REFERENCES:**

* <https://www.geeksforgeeks.org/category/linked-list/>
* https://stackoverflow.com/questions/1127396/struct-constructor-in-c

**CODE REPOSITORY:**

GitHub: https://github.com/Shrock-221B/DS\_PROJECT\_FISAC.git