**Term Paper Title**

**Bachelor of Technology**

**Computer Science and Engineering**

Submitted By

SHILPA AGARWAL(13000118052)

RUPNARAYAN KUMAR(13000118061)

SUMIT ANAND(13000118029)

SONAM RAJA(13000118043)

JUNE 2020



**Techno Main**

**EM-4/1, Sector-V, Salt Lake**

**Kolkata- 700091**

**West Bengal**

**India**

**TABLE OF CONTENTS**

1. Abstract
2. Introduction
3. Body

I. DEVELOPMENT TOOLS

II. SYSTEM AND HARWARE REQUIREMENTS

III. FEATURES

IV. SNAPSHOT

IV. SOURCE CODE

V. FUTURE SCOPE

1. Conclusion
2. References

**1. ABSTRACT:-**

**The issue of introducing a new facility is very paramount in any company. Therefore we intend to start voting from employee to introduce the winning facility by bringing a new software which fullfill the company policy which says the facility which will be introduced must have more than half of the total votes otherwise no facility will introduce.**

**For that we have to use any of the programming language and here we use the C language along with Data Structure. Now again the company wants that the technique must be optimal so that it works on every plateform so use Hashing technique here instead of counting each time votes individually for each facility. And also have to use dynamic memory allocation to allocate the memory of complex data structure(i.e.. for array and structure). It can be used as remotely on server if the company wants to it in future. From this project we hope to build an voting machine software as per company requirement.**

**2. INTRODUCTION :-**

**The main object of this project is design a technique for a company who wants to introduce a new facility through voting and the company policy says that the facility who get more than half of the total no of votes will be introduced here otherwise no facility will be introduced using trending technology in market with maximum efficiency and optimal speed.**

**I. DEVELOPMENT TOOLS :-**

**1. We make our module with C , Data Structure and Hashing.**

**2. Language Used : C Langauge.**

**3. We use Dev C as our editor.**

**4. We used hashing for optimized the time complexity.**

**II. SYSTEM AND HARDWARE REQUIREMENTS :-**

**1. P IV or above processor.**

**2. 1 GB RAM.**

**3. 120 GB HDD**

**4. Operating System(any)**

**III. FEATURES :-**

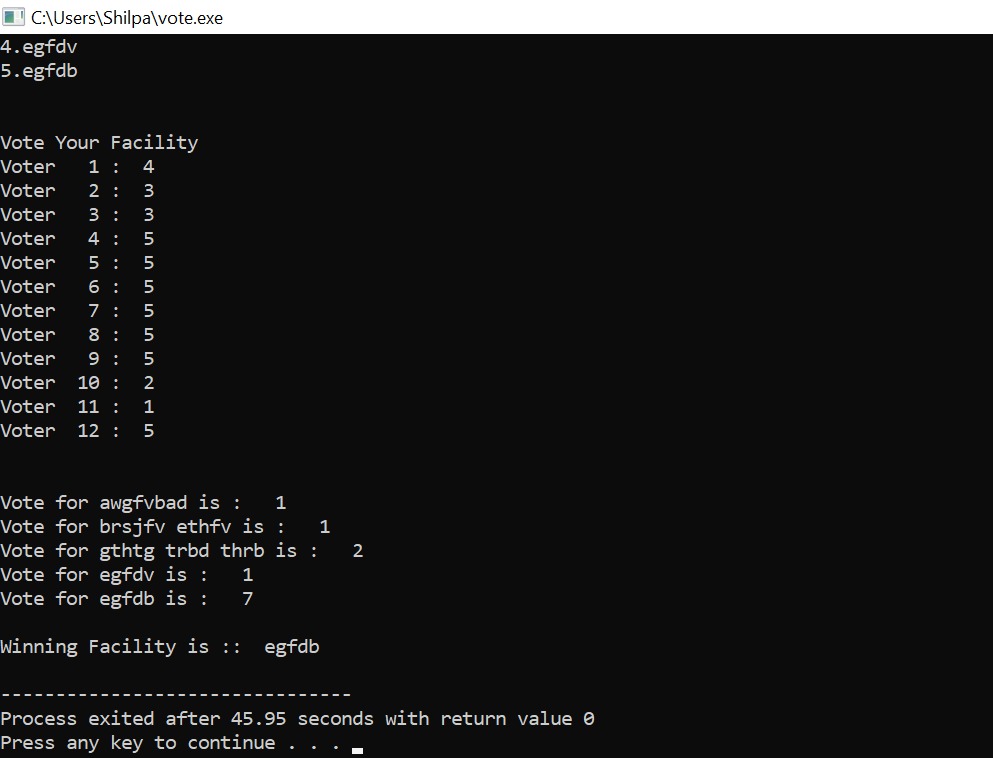
**1. Inexpensive**

**2. User Friendly**

**3. Optimized**

**4. Support All OS**

**IV.SNAPSHOT :-**

****

**V. SOURCE CODE :-**

**#include<stdio.h>**

**#include<stdlib.h>**

**#include<string.h>**

**struct facility**

**{**

**char f[50];**

**};**

**void takeInput(int \*facilityNumber, int \*votersNumber)**

**{**

**int i;**

**printf("Enter Number of Facility: ");**

**scanf("%d", facilityNumber);**

**printf("\nEnter Number of Voters: ");**

**scanf("%d", votersNumber);**

**}**

**void printingFacilities(struct facility \*fac, int facilityNumber)**

**{**

**int i;**

**printf("\nFacilities Listed for Voting: \n");**

**for(i=0;i<facilityNumber;i++)**

**{**

**printf("%d.",(i+1));**

**puts(fac[i].f);**

**}**

**printf("\n");**

**}**

**void takingVotes(int \*vote, int facilityNumber, int votersNumber)**

**{**

**int num, n=9, currVote;**

**printf("\nVote Your Facility\n");**

**num = 1;**

**n = votersNumber;**

**while(n--)**

**{**

**printf("Voter %3d : ", num++);**

**scanf("%d", &currVote);**

**if(currVote < votersNumber)**

**vote[currVote-1]++;**

**}**

**printf("\n");**

**}**

**void calculatingResult(struct facility \*fac, int \*vote, int votersNumber, int facilityNumber)**

**{**

**int index=-1,i;**

**for(i=0; i<facilityNumber; i++)**

**{**

**printf("\nVote for ");**

**printf("%s", fac[i].f);**

**printf(" is : %3d", vote[i]);**

**if(vote[i]>(votersNumber/2))**

**index=i;**

**}**

**printf("\n");**

**if(index==-1)**

**printf("\nNo Winning facility \n");**

**else**

**printf("\nWinning Facility is :: %s \n",fac[index].f);**

**}**

**int main()**

**{**

**int votersNumber, facilityNumber, n, buffer, i;**

**struct facility \*fac;**

**int \*vote;**

**takeInput(&facilityNumber, &votersNumber);**

**fac = (struct facility \*)malloc(facilityNumber \* sizeof(struct facility));**

**vote = (int \*)malloc(facilityNumber \* sizeof(int));**

**scanf("%c", &buffer);**

**printf("\nEnter Facilities: \n");**

**for(i=0; i<facilityNumber; i++)**

**{**

**gets(fac[i].f);**

**vote[i] = 0;**

**}**

**printingFacilities(fac, facilityNumber);**

**takingVotes(vote, facilityNumber, votersNumber);**

**calculatingResult(fac, vote, votersNumber, facilityNumber);**

**return 0;**

**}**

**VI.FUTURE SCOPE :-**

* **More Optimized**
* **Voter Details Verification**
* **Improving Security**
* **More User Friendly**
* **Remotely work**

**CONCLUSION :-**

* **This type of software is very common and useful**
* **This can be used not only in company to install a new facility,but also worldwide to elect a party for nation,elect class representative for school and colleges etc.**
* **This can be work remotely also so it is very much user friendly because user can vote from their respective places.**
* **We use Hashing Technique to minimize the time complexity while counting votes for each facilitywhich provides it to become more optimal than as usual software.**

**REFRENCES :-**

* [Kernighan, Brian W.](https://en.wikipedia.org/wiki/Brian_Kernighan);[Ritchie, Dennis M.](https://en.wikipedia.org/wiki/Dennis_Ritchie)(1996). *The C Programming Language*(2nd ed.).[Prentice Hall](https://en.wikipedia.org/wiki/Prentice_Hall). pp.192, 259.[ISBN](https://en.wikipedia.org/wiki/ISBN_(identifier)).
* King, K.N. (2008).*C Programming: A Modern Approach*(2 ed.). W. W. Norton.[ISBN](https://en.wikipedia.org/wiki/ISBN_(identifier)).
* Citavi Refrence tool.