**MINI VOTING BOOTH**

**21CSS101J – PROGRAMMING FOR PROBLEM SOLVING**

**Mini Project Report**

*Submitted by*

**Rishiram.B [Reg. No.: RA2211026010553]**

**B.Tech. CSE - <<AI&ML>>**

**Arjun Prathick.V [Reg. No.: RA2212701010054]**

**M.Tech. CSE - <<Integrated AI>>**



**SCHOOL OF COMPUTING**

**COLLEGE OF ENGINEERING AND TECHNOLOGY**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(Under Section 3 of UGC Act, 1956)**

S.R.M. NAGAR, KATTANKULATHUR – 603 203

KANCHEEPURAM DISTRICT

**December 2022**

***TABLE OF CONTENTS***

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Title** | **Page No.** |
| 1 | Problem Statement | 1 |
| 2 | Methodology / Procedure | 2 |
| 3 | Coding (C or Python) | 3 |
| 4 | Results | 5 |
| 5 | Conclusion | 5 |

***Problem Statement***

-> Allows us to set up a flexible and trustworthy voting system

-> Applicable for large as well small group of people

E.g A batch, A class.

-> Keeps a record of every voting

This voting management system is useful for storing as many votes as possible. The is a model of how the original voting system works in a voting booth.

***Methodology***

1. Provide the following options to the person who is accessing as shown below:
   * Vote for your favorite Candidate.
   * Check the number of votes of each Candidate.
   * Check the candidate who is leading and then [Exit](https://www.geeksforgeeks.org/understanding-exit-abort-and-assert/).
2. The user chooses one of the options.
3. If the user chooses 1, then the list of candidates is displayed and the user can now choose from this list of candidates.
4. If the user chooses 2, then the list of candidates along with their current number of votes is displayed.
5. If the user chooses 3, the name of the candidate with the maximum number of votes is displayed. If there is more than one candidate with maximum votes, display an [error message](https://www.geeksforgeeks.org/error-handling-c-programs/) stating “No winner”.
6. This program continues until the user chooses **0** to **exit()**.

***Coding (C-program)***

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<stdlib.h>

Struct currentValidID{

Int year;

Char branch[6];

Int totalVoters;

};

Typedef struct candidate{

Int cid;

Char cname[20];

Int votes;

}CANDIDATE;

//GLOBALS --------------------------------------------------------

Struct currentValidID currentValidID; //stores current Valid user ID parameters

CANDIDATE candidateArray[20]; //to store information all candidates

Int numberOfCandidates; //Total number of candidates standing for election

Char studentVotes[200]; //to store information of votes given by each student

//----------------------------------------------------------------

//To extract year from userID – For example, userID:2018btecs00064 year:2018

Int extractYear(char userID[15])

{

Int year=0;

Char tmp;

For(int i=0;i<4;i++){

Tmp=userID[i];

Year=(year\*10)+(tmp-48);

}

Return year;

}

Int extractRollNo(char userID[15])

{

Int rollno=0;

Char tmp;

For(int i=9;i<14;i++){

Tmp=userID[i];

Rollno=(rollno\*10)+(tmp-48);

}

Return rollno;

}

//Will check whether the global branch code and inputed branch code is matching or not

Int checkBranchCode(char userID[15])

{

Char branchCode[6];

For(int i=4;i<9;i++){

branchCode[i-4]=userID[i];

}

branchCode[5]=’\0’;

if(strcmp(branchCode,currentValidID.branch)==0)

return 1;

else

return 0;

}

Int authenticateAdmin(){

Char username[15], password[6];

Printf(“\nEnter username: “);

Scanf(“%s”,username);

If((strcmp(username,”Admin”))!=0)

Return 0;

Else

{

Printf(“Enter Password: “);

Int i=0;

For(i=0;i<5;i++)

{

Password[i]=getch();

Printf(“%c”,’\*’);

}

Password[i]=’\0’;

If((strcmp(password,”admiN”))!=0)

Return 0;

}

Return 1;

}

Void banID(){

Printf(“\nCreating Banned.txt…\n”);

FILE \*fp=fopen(“Banned.txt”, “w”);

If(fp==NULL){

Printf(“Error: Banned.txt not created.\n”);

Fclose(fp);

Return;

}

Printf(“Just Enter last roll no to ban\nPress 0 to exit… “);

Int input;

While(1){

Printf(“\nEnter Number: “);

Scanf(“%d”,&input);

If(input==0)

Break;

studentVotes[input-1]=’$’;

fprintf(fp,”%d\n”,input);

}

Fclose(fp);

Printf(“Created Successfully\n”);

}

Void createCandidateFiles(){

Printf(“\nCreating candidate files…\n”);

FILE \*fp;

Char filename[20];

For(int I = 1;I <= numberOfCandidates; i++){

Sprintf(filename,”candidate%d.txt”,i);

Fp=fopen(filename,”w”);

Fprintf(

Fp,”0\n%s”,

candidateArray[i-1].cname

);

Fclose(fp);

}

Printf(“Created Files successfully\n”);

}

Void deleteIllegalVote(char userID[15])

{

FILE \*fp,\*fcp;

Char filename[20];

Char line[20];

Int location = extractRollNo(userID);

Sprintf(filename,”candidate%d.txt”,candidateArray[studentVotes[location-1]-49].cid);

candidateArray[studentVotes[location-1]-49].votes--;

studentVotes[location-1]=’0’;

if ((fp = fopen(filename,”r”)) == NULL)

{

Printf(“\nFile cannot be opened…Operation Failed”);

Return;

}

Printf(“\nDeleting in process…\n “);

If ((fcp = fopen(“tmp.txt”,”w”)) == NULL)

{

Printf(“\nFile cannot be opened…Operation Failed”);

Return;

}

While (!feof(fp))

{

Fscanf(fp,”%s”,line);

Fprintf(fcp,”%s\n”,line);

}

Fclose(fp);

Fclose(fcp);

If ((fp = fopen(filename,”w”)) == NULL)

{

Printf(“\nFile cannot be opened…Operation Failed”);

Return;

}

Int numFromFile;

Char cnameFromFile[20];

Fcp = fopen(“tmp.txt”,”r”);

Fscanf(fcp,”%d”,&numFromFile);

Fprintf(fp,”%d”,numFromFile-1);

Fscanf(fcp,”%s”,cnameFromFile);

Fprintf(fp,”\n%s”,cnameFromFile);

While(!feof(fcp)){

Fscanf(fcp,”%d”,&numFromFile);

If(numFromFile!=location)

Fprintf(fp,”\n%d”,numFromFile);

}

Fclose(fp);

Fclose(fcp);

Remove(“tmp.txt”);

Printf(“\nVote deleted successfully\nPress any key to continue…”);

Getch();

}

Int getWinner(){

Int maxV = -1;

Int winnerCid;

For(int I = 0;I < numberOfCandidates; i++){

If(candidateArray[i].votes > maxV) {

winnerCid = candidateArray[i].cid;

maxV = candidateArray[i].votes;

}

Else if(candidateArray[i].votes == maxV) {

Return -1;

}

}

Return winnerCid;

}

Void initiateNewElection()

{

Printf(“\nNew Election Initiation:\n”);

Printf(“\nElections for which Year: “);

Scanf(“%d”,&currentValidID.year);

Printf(“Enter branch code:”);

Scanf(“%s”,currentValidID.branch);

Printf(“Enter max roll no.:”);

Scanf(“%d”,&currentValidID.totalVoters);

Printf(“Enter the no. of candidates:”);

Scanf(“%d”,&numberOfCandidates);

For (int I = 0; I < currentValidID.totalVoters; i++)

{

studentVotes[i] = ‘0’;

}

For (int I = 0;I < numberOfCandidates; i++)

{

candidateArray[i].cid=i+1;

printf(“Enter name of candidate %d: “,i+1);

scanf(“ %s”,candidateArray[i].cname);

candidateArray[i].votes=0;

}

Return;

}

Void saveElectionInfoInFile(){

Printf(“Saving Election Info in File…\n”);

FILE \*fp = fopen(“ElectionInfo.txt”, “w”);

If(fp==NULL)

{

Printf(“\nError in file creation\n”);

Fclose(fp);

Return;

}

Fprintf(

Fp,”%d\n%s\n%d\n%d”,

currentValidID.year,

currentValidID.branch,

currentValidID.totalVoters,

numberOfCandidates

);

Fclose(fp);

Printf(“Saved Successfully : )”);

}

Void loadElectionInfoFromFile()

{

FILE \*f1,\*f2,\*f3;

F1=fopen(“ElectionInfo.txt”,”r”);

If(f1==NULL)

Printf(“Not Exist”);

Fscanf(f1,”%d”,&currentValidID.year);

Fseek(f1,2,SEEK\_CUR);

Fscanf(f1,”%s”,currentValidID.branch);

Fseek(f1,2,SEEK\_CUR);

Fscanf(f1,”%d”,&currentValidID.totalVoters);

Fseek(f1,2,SEEK\_CUR);

Fscanf(f1,”%d”,&numberOfCandidates);

Fclose(f1);

//load candidates info and student votes

For (int I = 0; I < currentValidID.totalVoters; i++)

{

studentVotes[i] = ‘0’;

}

For(int i=1;i<=numberOfCandidates;i++)

{

Int location;

Char filename[20];

Sprintf(filename,”candidate%d.txt”,i);

F2=fopen(filename,”r+”);

candidateArray[i-1].cid=I;

fscanf(f2,”%d”,&candidateArray[i-1].votes);

fscanf(f2,”%s”,candidateArray[i-1].cname);

while(!feof(f2)){

fscanf(f2,”%d”,&location);

studentVotes[location-1] = i+48;

}

Fclose(f2);

}

//load banned votes

Int location;

F3=fopen(“banned.txt”,”r+”);

While(!feof(f3)){

Fscanf(f3,”%d”,&location);

studentVotes[location-1] = ‘$’;

}

Fclose(f3);

}

Void adminPanel()

{

While(1){

If(authenticateAdmin()!=1){

Printf(“\n Wrong Username or Password \n”);

Break;

}

Printf(“\n\nLOGGED IN SUCCESSFULLY (Press Enter)”);

Getch();

While(1)

{

Char inputID[15];

Char input;char banInp;

Int WinnerCid, totalVotedNow=0;

Printf(“\n1.New Election\n2.Continue Previous Election\n3.Delete Illegal Vote\n4.Ban User IDs\n5.Result\n6.Logout\nOption:”);

Scanf(“ %c”,&input);

Switch(input)

{

Case ‘1’:

initiateNewElection();

saveElectionInfoInFile();

createCandidateFiles();

break;

case ‘2’:

loadElectionInfoFromFile();

break;

case ‘3’:

printf(“\nEnter user ID to delete its vote: “);

scanf(“%s”,inputID);

deleteIllegalVote(inputID);

break;

case ‘4’:

printf(“Do you want to ban particular ID’s?\nPress 1 if yes or any other key to continue…”);

scanf(“ %c”,&banInp);

if(banInp==’1’){

banID();

}

Break;

Case ‘5’:

WinnerCid = getWinner();

If(WinnerCid != -1){

Printf(“\nWinner is %s with %d votes\n”,candidateArray[WinnerCid-1].cname,candidateArray[WinnerCid-1].votes);

}

Else{

Printf(“\nIts A TIE”);

}

Printf(“\nFull Result\n”);

For(int i=0;i<numberOfCandidates;i++){

totalVotedNow+=candidateArray[i].votes;

printf(“%d. %s -> %d votes\n”,candidateArray[i].cid,candidateArray[i].cname,candidateArray[i].votes);

}

Printf(“\nVoting Percentage: %d %%\n\n”,(totalVotedNow\*100)/currentValidID.totalVoters);

Break;

Case ‘6’:

Return;

Default:

Printf(“Invalid Option”);

Getch();

}

}

}

};

Int isValid(char userID[15])

{

If(strlen(userID)!=14)

Return 0;

Int inputedYear=extractYear(userID);

Int inputedRollNo = extractRollNo(userID);

If(inputedYear!=currentValidID.year || checkBranchCode(userID)!=1 || inputedRollNo>currentValidID.totalVoters)

Return 0;

Return 1;

}

Int isVoted(char userID[15])

{

Int location=extractRollNo(userID);

If(studentVotes[location-1]==’0’)

Return 0;

Else

Return 1;

}

Int isBanned(char userID[15]){

Int location=extractRollNo(userID);

If(studentVotes[location-1]==’$’)

Return 1;

Else

Return 0;

}

Void saveVote(char userID[15],char voteInput)

{

Char filename[20];

Sprintf(filename,”candidate%d.txt”,voteInput-48);

FILE \*fp = fopen(filename,”r+”);

Int location=extractRollNo(userID);

studentVotes[location-1]=voteInput;

candidateArray[voteInput-49].votes++;

fseek(fp, 0, SEEK\_SET);

fprintf(fp,”%d\n”,candidateArray[voteInput-49].votes);

fseek(fp, 0, SEEK\_END);

fprintf(fp,”\n%d”,location);

fclose(fp);

}

Void studentPanel()

{

Char userID[15];

Char voteInput;

While(1)

{

Printf(“\n\n To exit press 0”);

Printf(“\n Enter user ID:”);

Scanf(“%s”,userID);

If(strcmp(userID, “0”)==0)

Return;

If(isValid(userID)!=1)

{

Printf(“\n Invalid User ID(Press Enter)”);

Getch();

Continue;

}

If(isBanned(userID)!=0)

{

Printf(“\nThis User ID is currently banned…\nContact Admin for the reason…(Press Enter to continue)”);

Getch();

Continue;

}

If(isVoted(userID)!=0)

{

Printf(“\n Your PRN entered is already voted\n Contact Admin for furthur query”);

Getch();

Continue;

}

Printf(“\n\n Candidates for election:”);

For (int I = 0; I < numberOfCandidates; i++)

{

Printf(“\n %d. %s”,i+1,candidateArray[i].cname);

}

Printf(“\n\n Your Vote(Enter Number):”);

voteInput=getch();

printf(“\*”);

if(voteInput-48 < 1 || voteInput-48 > numberOfCandidates)

{

Printf(“\nInvalid Vote\nTry Again…”);

Getch();

Continue;

}

saveVote(userID,voteInput);

printf(“\n\nThanks for your precious vote(Press Enter)”);

getch();

}

};

#include"election.h"

int main(){

while(1){

printf("\n\t\t\t 1.Student panel \n\t\t\t 2.Admin panel \n\t\t\t 3.Exit \n\t\t\t Option:");

char input;

scanf(" %c",&input);

switch(input){

case '1':

studentPanel();

break;

case '2':

adminPanel();

break;

case '3':

return 0;

default:

printf("\nInvalid option");

getch();

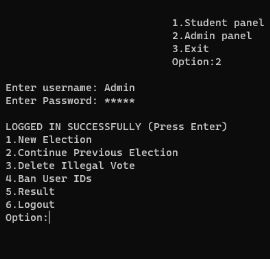
}

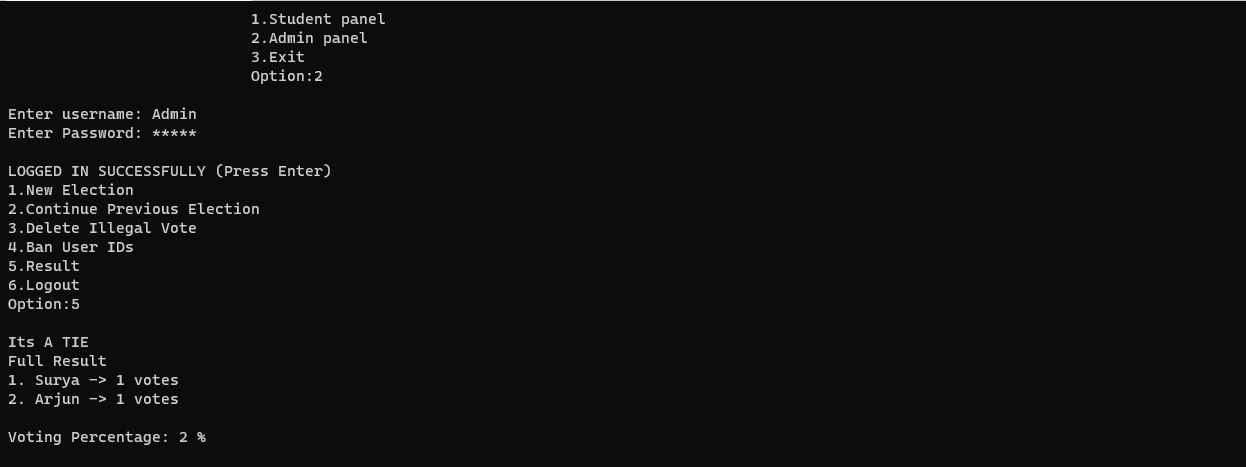
}

return 0;

}

***Result:***









***Conclusion:***

Election system using c-program studied and compile properly