VOTING SYSTEM

A Project Report

Submitted by

Group No : **12**

Group Members :

SAI NIVEDH V (CB.SC.U4AIE23062)

MATHIVANAN S (CB.SC.U4AIE23042)

SYED ANEES ASHRAF (CB.SC.U4AIE23068)

C V SHREYAAS ADITYA (CB.SC.U4AIE23018)

As a part of the subject

22AIE101- PROBLEM SOLVING & C PROGRAMMING

Department of Artificial Intelligence

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE – 641 112 (INDIA)

December - 2023

AMRITA SCHOOL OF ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE – 641 112

DECLARATION

We hereby declare that the project work entitled ***"Voting System"*** *submitted to*

Mrs. Aswathy P

Assistant Professor

Centre of Excellence and Networking (CEN)

Amrita School of Engineering

Place : Coimbatore

Date: 22-12-2023

ACKNOWLEDGEMENT

We would like to express my sincere gratitude to everyone who has contributed to the successful completion of this project. First and foremost, we would like to thank my project guide Mrs. Aswathy P for providing us with valuable guidance and support throughout the project.

We would also like to thank the Election Commission for providing me with the necessary resources and information to complete this project. Their support was invaluable in ensuring the success of this project.

And we would like to express our gratitude to our family and friends for their unwavering support and encouragement throughout the project. Their support gave me the motivation to complete this project to the best of my abilities.

Finally, we would like to thank all our team mates who have spent their valuable time to research about this project and made this project a great success.

CONTENTS

ABSTRACT

In our project, we have implemented an online voting system using the C programming language. The system allows candidates to register, verify their documents, and generate unique user IDs and passwords for themselves and voters. The Election Commission is responsible for the Admin Login, while the Candidate Login is taken care of by the candidates themselves. Voters are given a unique ID and password by each candidate, which they can use to vote for that candidate only once per election.

The system enables candidates to access their profiles and submit all of their information, including prior milestones. The administrator may review each candidate's information and papers; only after that, the candidate's ID and password will be produced, and incorrect accounts can be removed. Voters may access a list of candidates in their region via the software system. The administrator has full control over the system and may regulate and remove any information that isn't related to the election rules.

INTRODUCTION

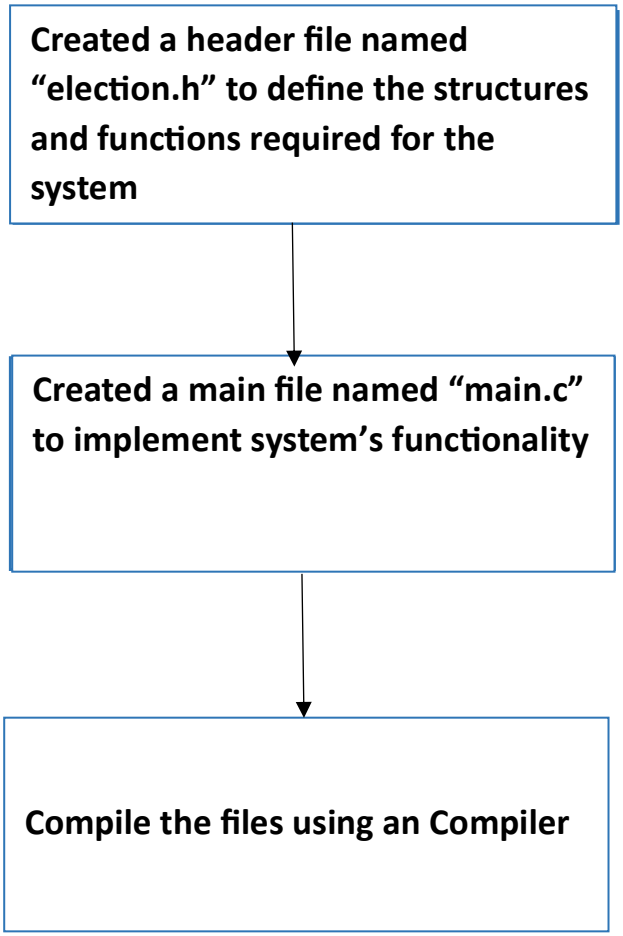
Our project aims to provide a secure and efficient way of conducting elections. The system allows candidates to register, verify their documents, and generate unique user IDs and passwords for themselves and voters. The Election Commission is responsible for the Admin Login, while the Candidate Login is taken care of by the candidates themselves. Voters are given a unique ID and password by each candidate, which they can use to vote for that candidate only once per election.

The system enables candidates to access their profiles and submit all of their information, including prior milestones. The administrator may review each candidate's information and papers; only after that, the candidate's ID and password will be produced, and incorrect accounts can be removed. Voters may access a list of candidates in their region via the software system. The administrator has full control over the system and may regulate and remove any information that isn't related to the election rules.

In this report, we will discuss the implementation of an online voting system using the C programming language. We will begin by providing an overview of the project and its objectives. Next, we will discuss the features of the system, including candidate registration, document verification, and user authentication. We will also describe the roles of the Election Commission, candidates, and voters in the system. Finally, we will provide a detailed explanation of the system's architecture and implementation.

METHODOLOGY

We implemented an online voting system using the C programming language. The system allows candidates to register, verify their documents, and generate unique user IDs and passwords for themselves and voters. The Election Commission is responsible for the Admin Login, while the Candidate Login is taken care of by the candidates themselves. Voters are given a unique ID and password by each candidate, which they can use to vote for that candidate only once per election.

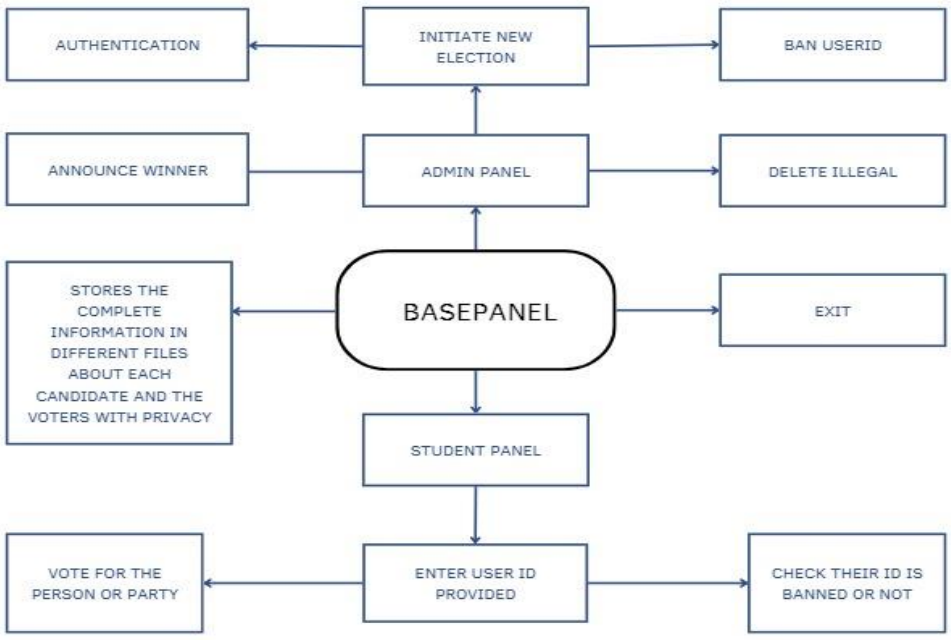


Ran the compiled program to test system's functionality and we have made sure that all our features are working as expected

↓

We have tested our program with edge cases to ensure that it can handle unexpected inputs and situation

Now we will see the detailed overview of our Voting system :



ALGORITHM

1. Main Function:

- The program starts by calling the main function.
- The main function initiates the program by calling either the adminPanel or studentPanel function based on user input.

2. Admin Panel:

- The adminPanel function begins by authenticating the administrator using the authenticateAdmin function.
- If authentication is successful, the admin is presented with a menu to perform various actions related to election management.
- The actions include:
  1. Initiating a new election (initiateNewElection).
  2. Continuing a previous election (loadElectionInfoFromFile).
  3. Deleting an illegal vote (deleteIllegalVote).
  4. Banning user IDs (banID).
  5. Displaying election results (getWinner).
  6. Logging out.

3. Initiating a New Election:

- The initiateNewElection function prompts the admin to input details for a new election, including the year, branch code, total voters, and number of candidates.

- The function initializes relevant global arrays to prepare for the new election.

- It then calls saveElectionInfoInFile to save the election information to a file and createCandidateFiles to create files for each candidate.

4. Saving Election Info in File:

- The saveElectionInfoInFile function opens a file ("ElectionInfo.txt") and writes the current election details (year, branch, total voters, and number of candidates) to the file.

5. Creating Candidate Files:

- The createCandidateFiles function creates a file for each candidate with the candidate's initial vote count and name.

6. Loading Election Info from File:

- The loadElectionInfoFromFile function reads election information from the "ElectionInfo.txt" file, including the year, branch, total voters, and number of candidates.

- It then loads candidate information and votes from individual candidate files and banned votes from the "Banned.txt" file.

7. Deleting an Illegal Vote:

- The deleteIllegalVote function deletes an illegal vote for a specific user ID.
- It reduces the vote count for the corresponding candidate and updates the student votes array.
- The function reads from and writes to temporary files to update the candidate file.

8. Banning User IDs:

- The banID function creates a file ("Banned.txt") to store banned user IDs.
- The admin enters roll numbers to be banned, and the function updates the student votes array accordingly.

9. Getting Election Winner:

- The getWinner function determines the candidate with the highest votes.
- If there is a clear winner, the candidate ID is returned; otherwise, -1 is returned for a tie.

- It will also display The percentage of people, voted for an election .

10. Student Panel:

- The studentPanel function allows students to cast their votes.
- Students enter their user ID and choose a candidate to vote for.
- The function checks the validity of the user ID and whether the user has already voted or is banned.
- If conditions are met, the vote is saved using the saveVote function.

11. Checking Validity of User ID:

- The isValid function checks the length, year, branch code, and roll number of a user ID for validity.

12. Checking if User ID has Voted:

- The isVoted function checks if a user ID has already cast a vote based on the student votes array.

13. Checking if User ID is Banned:

- The isBanned function checks if a user ID is banned based on the student votes array.

14. Saving a Vote:

- The saveVote function updates the student votes array, increments the candidate's vote count, and updates the corresponding candidate file.

15. Authentication for Admin:

- The authenticateAdmin function prompts the admin to enter a username and password.
- If the input matches the predefined admin credentials, authentication is successful.

16. Extraction Functions for User ID:

- extractYear extracts the year from a user ID.
- extractRollNo extracts the roll number from a user ID.
- checkBranchCode checks if the branch code in a user ID matches the global branch code.

SOURCE CODE

1.election.h <header file>

```
#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:2023btech00064
year:2023
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].vo
tes);
                    }
                    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);
```

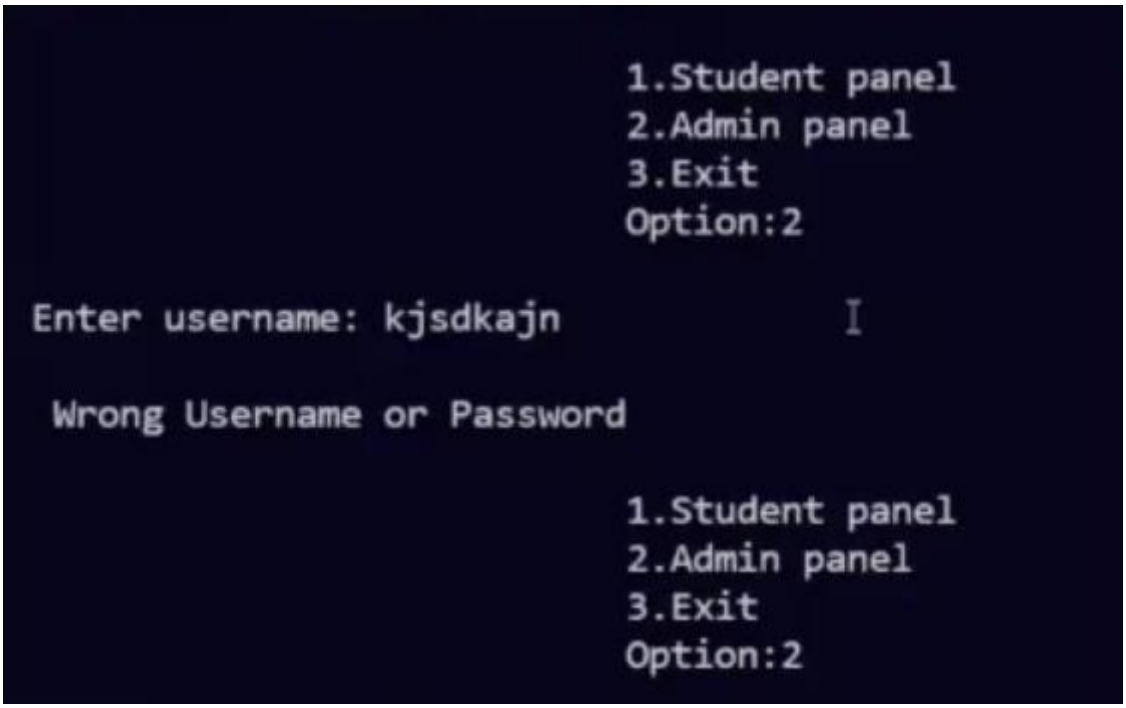
```
    fprintf(fp,"%d\n",candidateArray[voteInput-49].votes);
    fseek(fp, 0, SEEK_END);
    fprintf(fp,"\n%d",location);
    fclose(fp);
}


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");
```

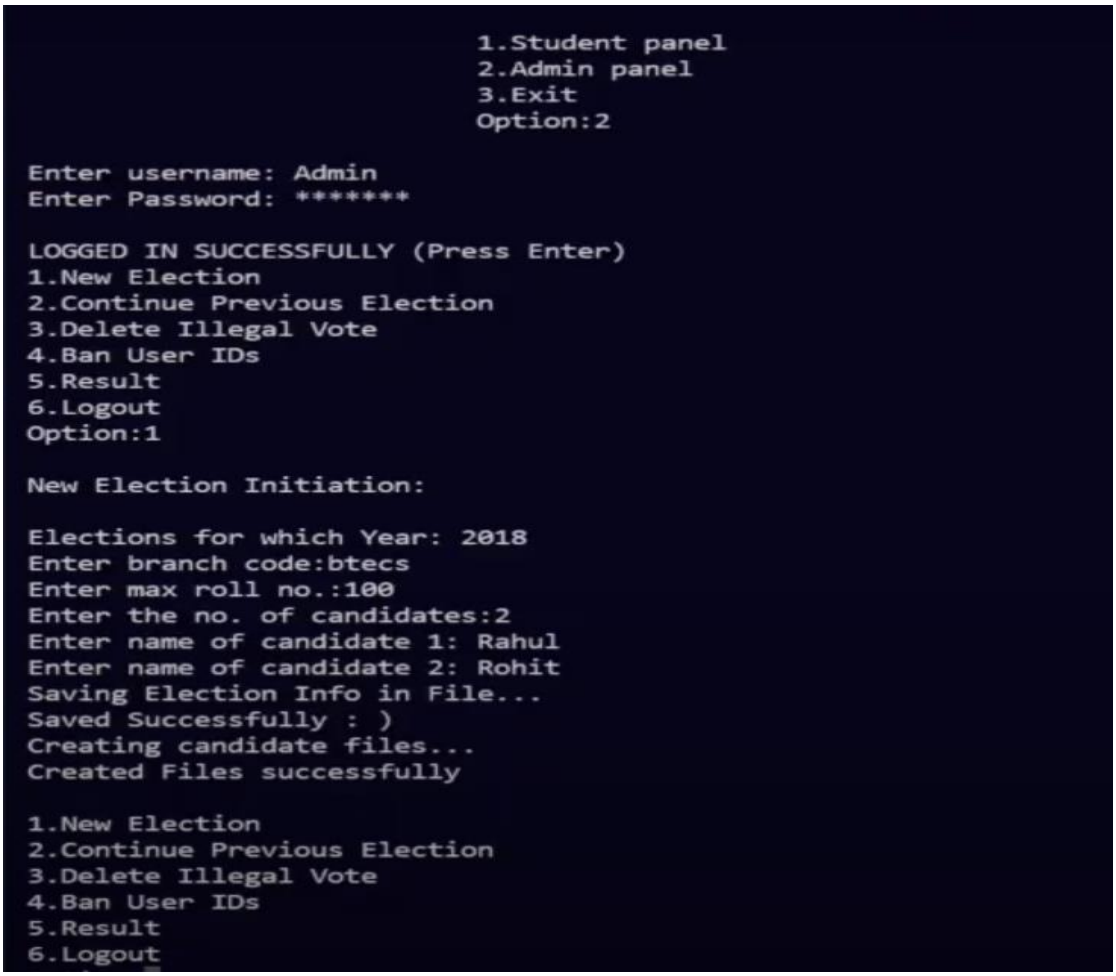
```
            getch();
            continue;
        }
        printf("\n\n  Candidates for election:");
            for (int i = 0; i < numberOfCandidates; i++)
        {
            printf("\n  %d. %s",i+1,candidateArray[i].cname);
        }
        printf("\n\n  Your Vote(Enter Number):");
        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();
    }
};
```

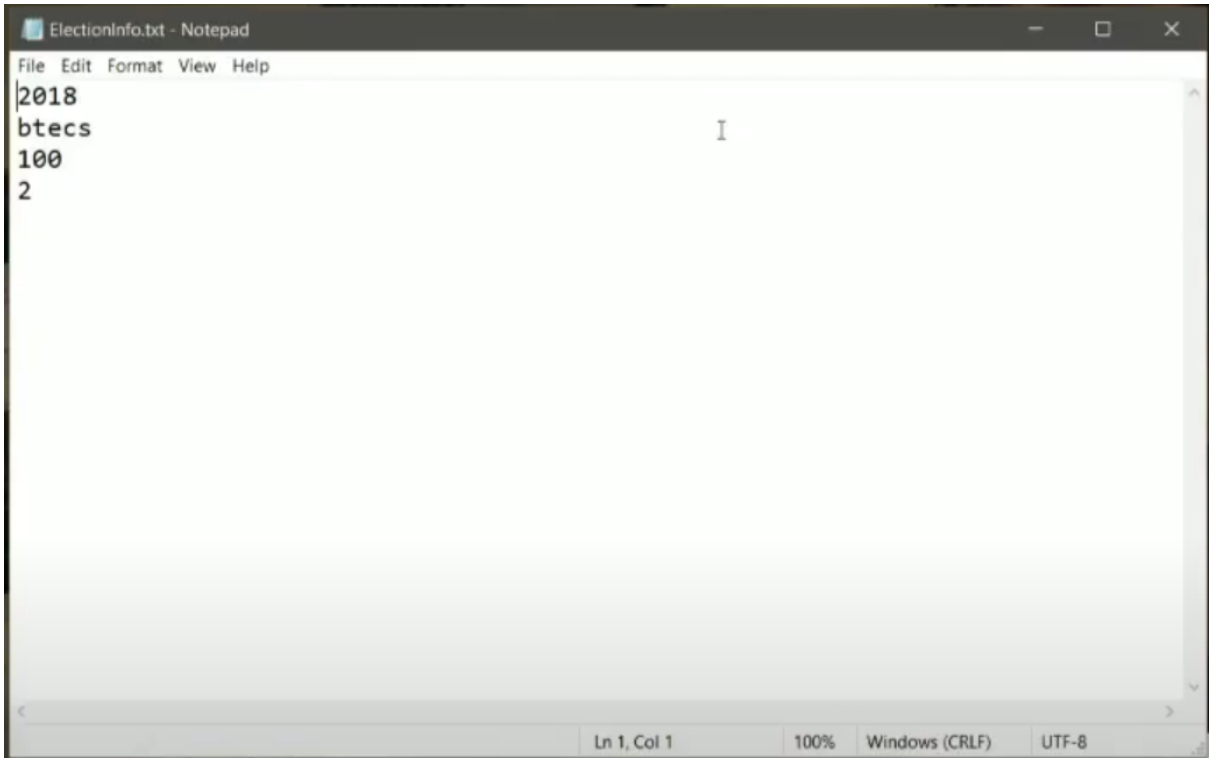
2.main.c <main file>

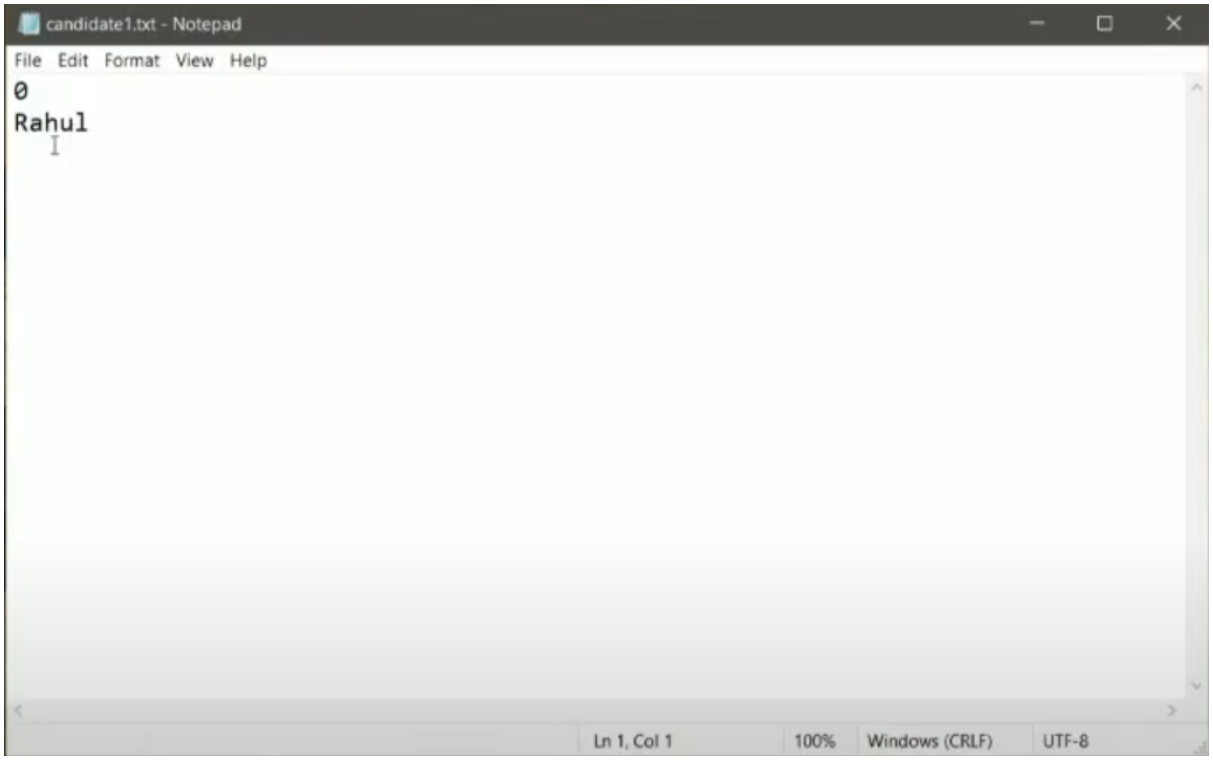
```
#include"election.h"


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


1.Student panel
2.Admin panel
3.Exit
Option:2
Enter username: kjsdkajn
Wrong Username or Password
1.Student panel
2.Admin panel
3.Exit
Option:2


1.Student panel
2.Admin panel
3.Exit
Option:2
Enter username: Admin
Enter Password: *******
LOGGED IN SUCCESSFULLY (Press Enter)
1.New Election
2.Continue Previous Election
3.Delete Illegal Vote
4.Ban User IDs
5.Result
6.Logout
Option:1
New Election Initiation:
Elections for which Year: 2018
Enter branch code:btecs
Enter max roll no.:100
Enter the no. of candidates:2
Enter name of candidate 1: Rahul
Enter name of candidate 2: Rohit
Saving Election Info in File...
Saved Successfully :)
Creating candidate files...
Created Files successfully
1.New Election
2.Continue Previous Election
3.Delete Illegal Vote
4.Ban User IDs
5.Result
6.Logout


ElectionInfo.txt - Notepad
File Edit Format View Help
2018
btecs
100
2
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8

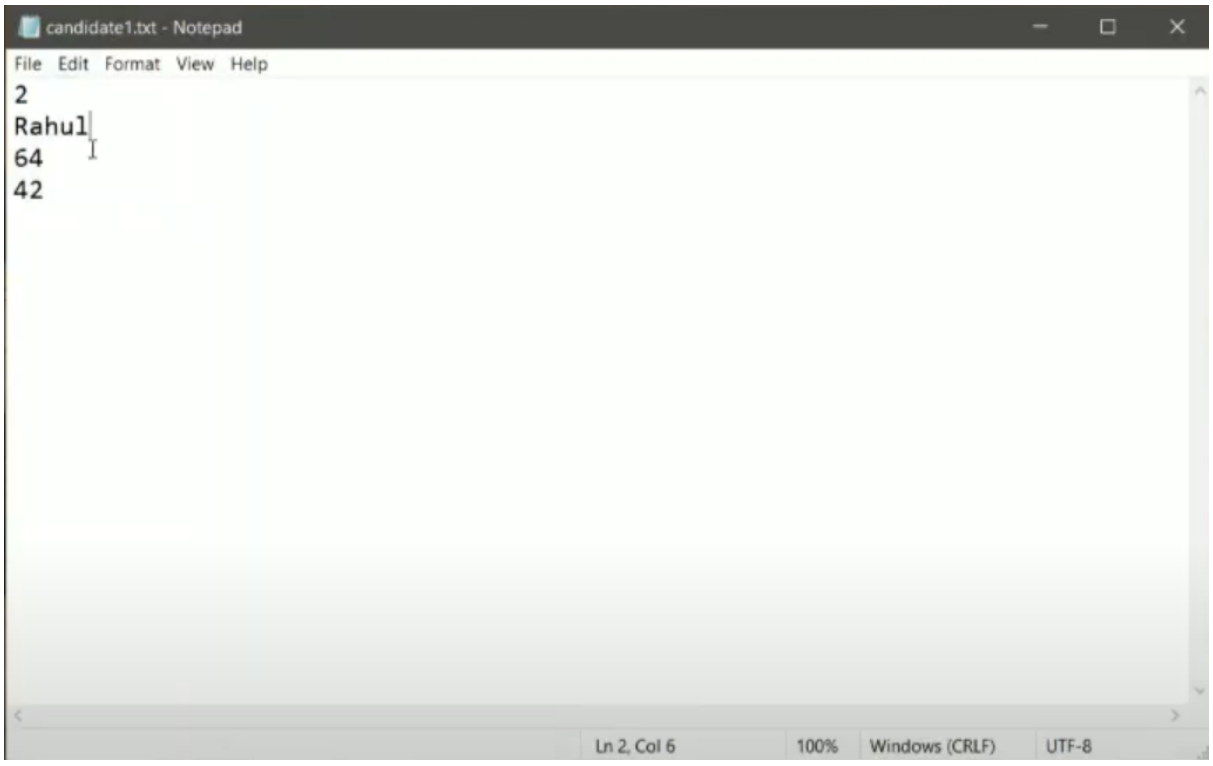

candidate1.txt - Notepad
File Edit Format View Help
0
Rahul
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8

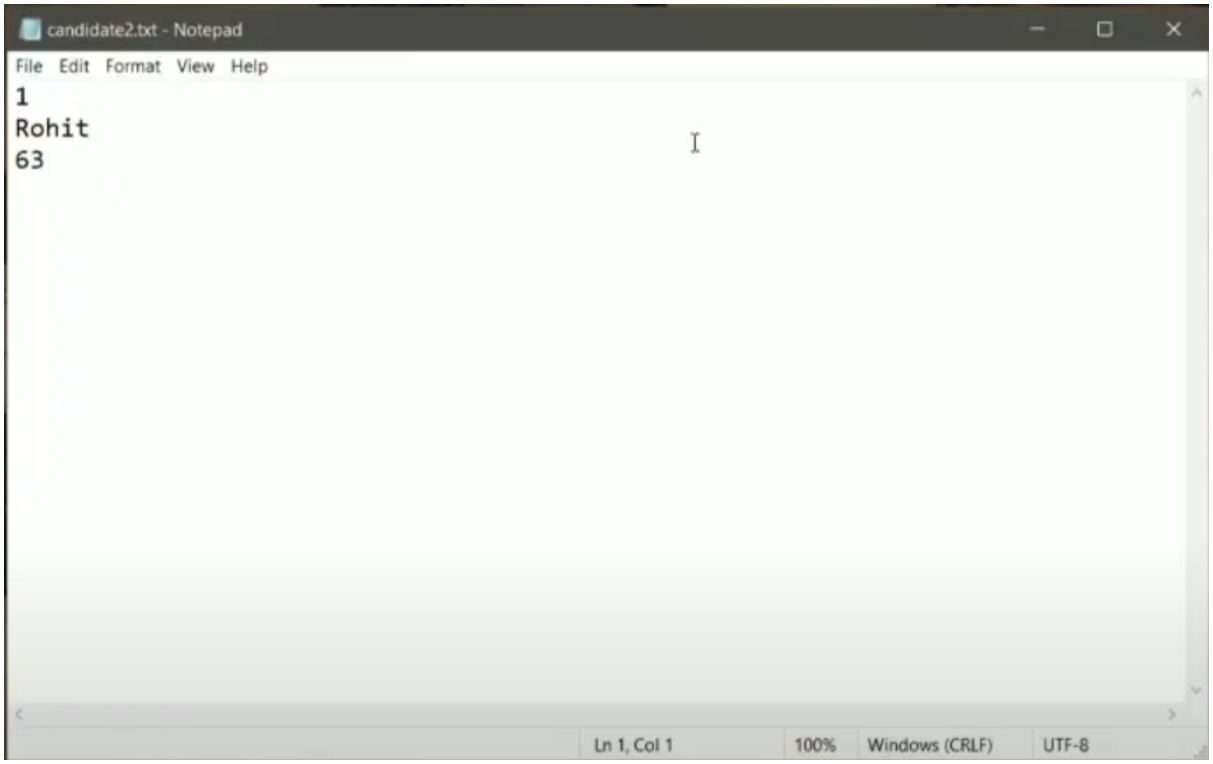
```
                        1.Student panel
                        2.Admin panel
                        3.Exit
                        Option:1


  To exit press 0
  Enter user ID:2019btecs00064

  Invalid User ID(Press Enter)

  To exit press 0
  Enter user ID:2018btecs00064


  Candidates for election:
  1. Rahul
  2. Rohit

  Your Vote(Enter Number):*

Thanks for your precious vote(Press Enter)

  To exit press 0
  Enter user ID:2018btecs00063


  Candidates for election:
  1. Rahul
  2. Rohit

  Your Vote(Enter Number):*

Thanks for your precious vote(Press Enter)

  To exit press 0
  Enter user ID:2018btecs00042
```

```
  Candidates for election:
  1. Rahul
  2. Rohit

  Your Vote(Enter Number):*

Thanks for your precious vote(Press Enter)

  To exit press 0
  Enter user ID:2018btecs00064

  Your PRN entered is already voted
  Contact Admin for furthur query

  To exit press 0
  Enter user ID:0
```

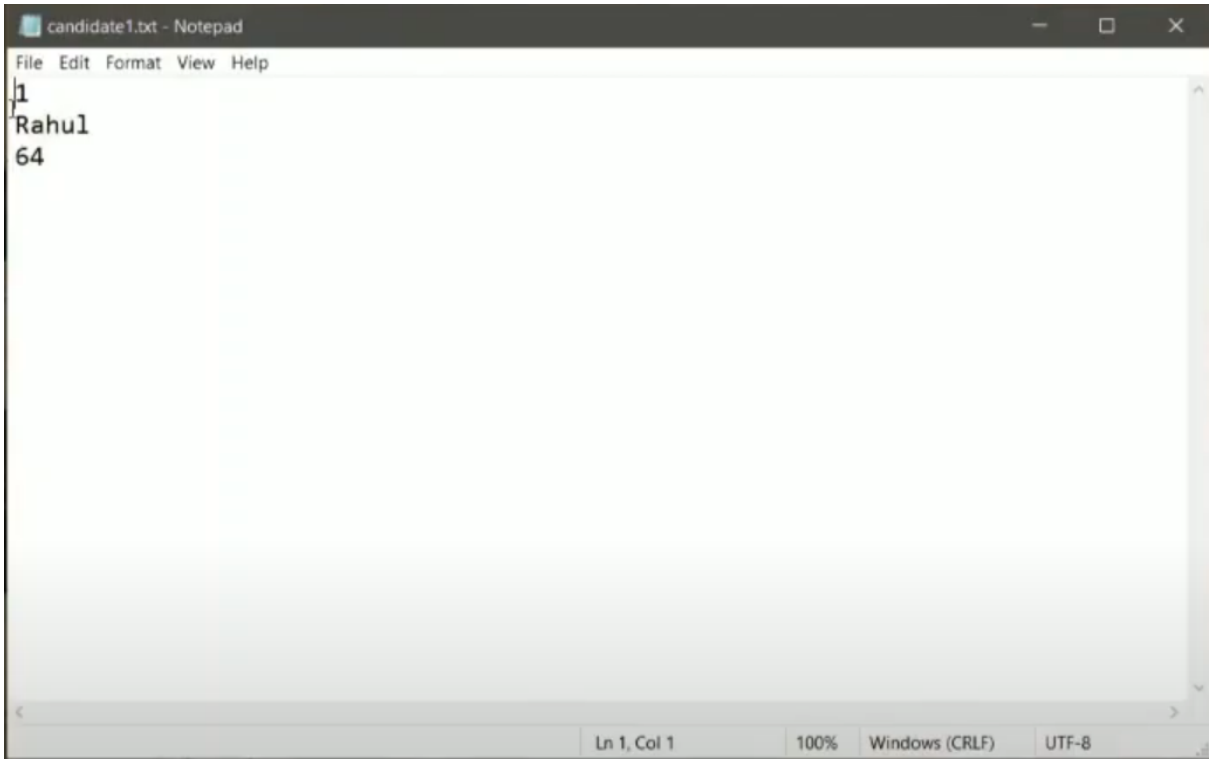
```
                          1.Student panel
                          2.Admin panel
                          3.Exit
                          Option:2

Enter username: Admin
Enter Password: *******

LOGGED IN SUCCESSFULLY (Press Enter)
1.New Election
2.Continue Previous Election
3.Delete Illegal Vote
4.Ban User IDs
5.Result
6.Logout
Option:4
Do you want to ban particular ID's?
Press 1 if yes or any other key to continue...1

Creating Banned.txt...
Just Enter last roll no to ban
Press 0 to exit...
Enter Number: 10

Enter Number: 50

Enter Number: 0
Created Successfully
```

```
1.New Election
2.Continue Previous Election
3.Delete Illegal Vote
4.Ban User IDs
5.Result
6.Logout
Option:6

                        1.Student panel
                        2.Admin panel
                        3.Exit
                        Option:1


  To exit press 0
  Enter user ID:2018btecs00010

This User ID is currently banned...
Contact Admin for the reason...(Press Enter to continue)
```

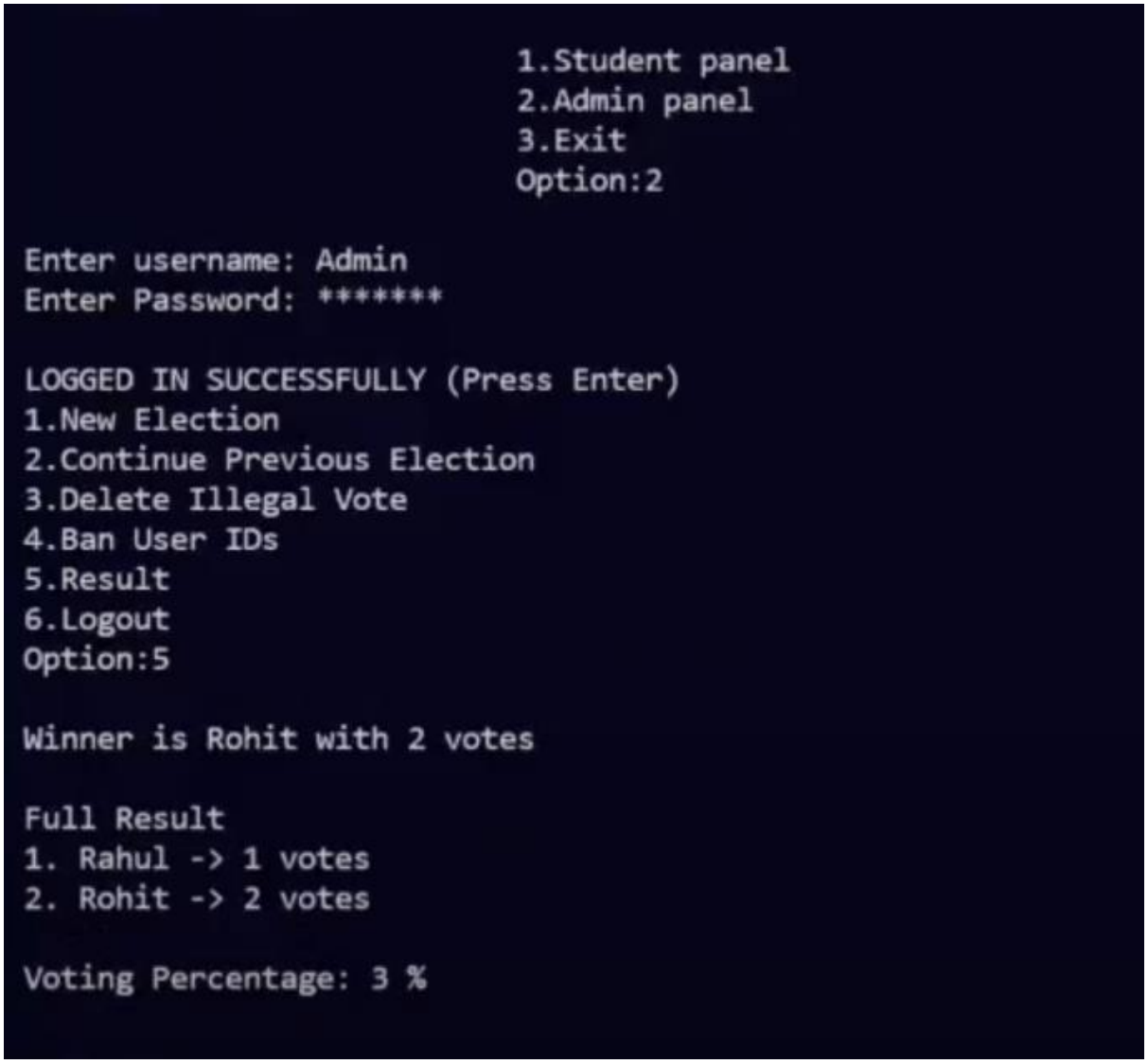





```
                        1.Student panel
                        2.Admin panel
                        3.Exit
                        Option:2

Enter username: Admin
Enter Password: *******

LOGGED IN SUCCESSFULLY (Press Enter)
1.New Election
2.Continue Previous Election
3.Delete Illegal Vote
4.Ban User IDs
5.Result
6.Logout
Option:3

Enter user ID to delete its vote: 2018btecs00042

Deleting in process...

Vote deleted successfully
```

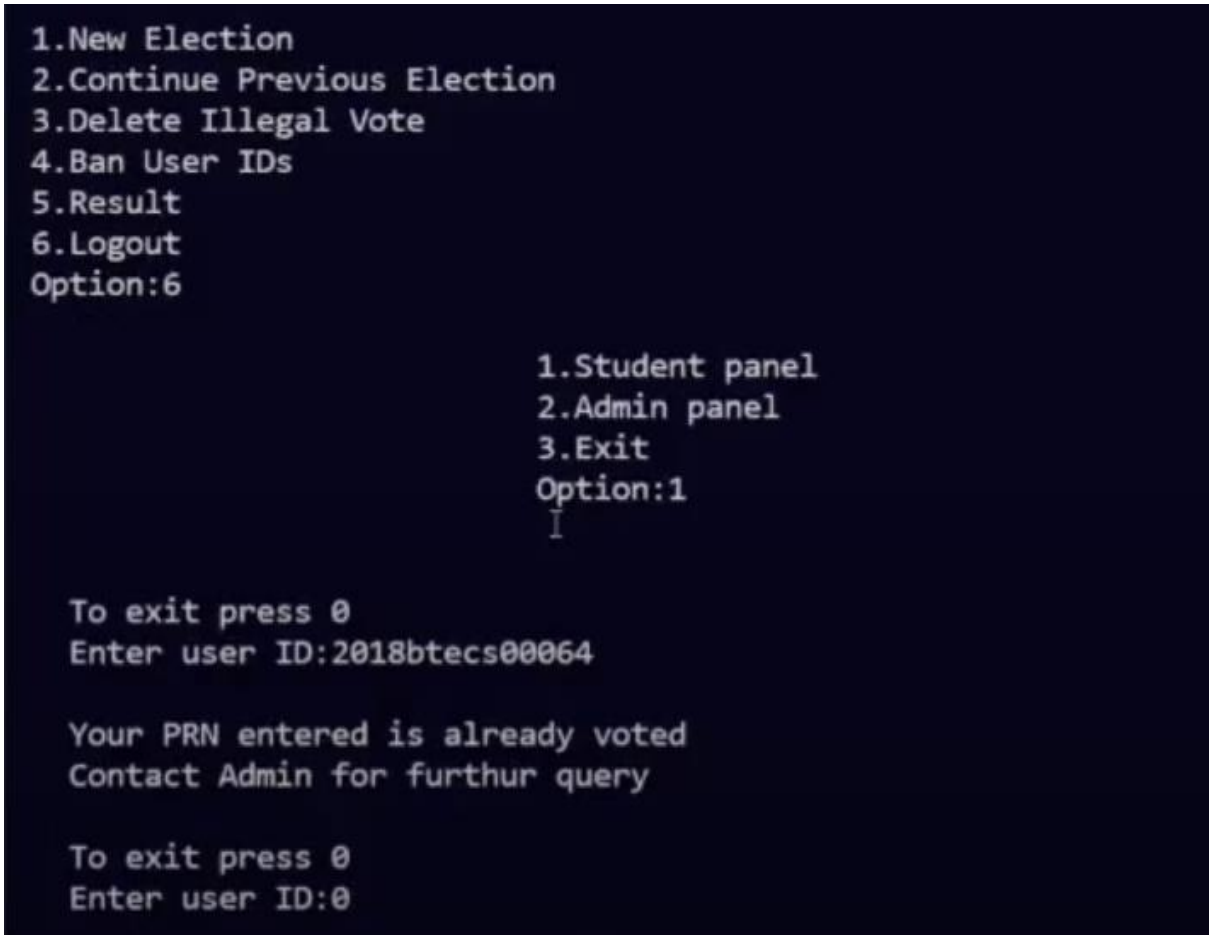



```
                        1.Student panel
                        2.Admin panel
                        3.Exit
                        Option:1


  To exit press 0
  Enter user ID:2018btecs00042


  Candidates for election:
  1. Rahul
  2. Rohit

  Your Vote(Enter Number):*

Thanks for your precious vote(Press Enter)

  To exit press 0
  Enter user ID:0
```


1.Student panel
2.Admin panel
3.Exit
Option:2
Enter username: Admin
Enter Password: *******
LOGGED IN SUCCESSFULLY (Press Enter)
1.New Election
2.Continue Previous Election
3.Delete Illegal Vote
4.Ban User IDs
5.Result
6.Logout
Option:5
Winner is Rohit with 2 votes
Full Result
1. Rahul -> 1 votes
2. Rohit -> 2 votes
Voting Percentage: 3 %

```
                          1.Student panel
                          2.Admin panel
                          3.Exit
                          Option:2

Enter username: Admin
Enter Password: *******

LOGGED IN SUCCESSFULLY (Press Enter)
1.New Election
2.Continue Previous Election
3.Delete Illegal Vote
4.Ban User IDs
5.Result
6.Logout
Option:2

1.New Election
2.Continue Previous Election
3.Delete Illegal Vote
4.Ban User IDs
5.Result
6.Logout
Option:5
```

```
Winner is Rohit with 2 votes

Full Result
1. Rahul -> 1 votes
2. Rohit -> 2 votes

Voting Percentage: 3 %
```


1.New Election
2.Continue Previous Election
3.Delete Illegal Vote
4.Ban User IDs
5.Result
6.Logout
Option:6
1.Student panel
2.Admin panel
3.Exit
Option:1
To exit press 0
Enter user ID:2018btecs00064
Your PRN entered is already voted
Contact Admin for furthur query
To exit press 0
Enter user ID:0

CONCLUSION

ACHIEVEMENTS

1. Election Management:

- Successful implementation of a comprehensive election management system allowing the initiation of new elections and continuation of previous ones.

- Efficient storage and retrieval of election-related data using file handling.

2. User Authentication and Security:

- Implementation of a secure user authentication system for administrators, ensuring system integrity.

- Successful banning of user IDs to maintain the fairness of the election process.

3. Voting System:

- Implementation of a user-friendly interface for students to cast their votes.

- Proper validation mechanisms to ensure the validity of user IDs and prevent duplicate voting.

4. Candidate Management:

- Effective storage and management of candidate information in separate files.

- Accurate tracking of the number of votes each candidate receives.

5. Result Reporting:

- Successful determination of election results, including the identification of the winner and overall voting percentages.

- Informative reporting of election outcomes for both administrators and students.

CHALLENGES FACED

1. File Handling and Data Integrity:

- Ensuring data integrity and proper error handling during file operations posed challenges, particularly when reading and writing candidate and election information.

2. User Interface Design:

- Designing an intuitive and user-friendly interface, especially for student voters, required careful consideration and testing to ensure a positive user experience.

3. Security Concerns:

- Addressing potential security vulnerabilities and ensuring that the system is robust against unauthorized access or manipulation was a continuous challenge.

4. Testing and Validation:

- Rigorous testing was essential to identify and resolve potential issues related to user authentication, vote validation, and overall system reliability.

5. Incomplete Features:

- The project is currently incomplete, lacking detailed implementation in certain functions such as loading election information from files and determining the winner.

FUTURE CONSIDERATIONS

To enhance the project, future considerations should include:

- Completing the implementation of missing features.

- Conducting more extensive testing, especially for security vulnerabilities.

- Implementing a graphical user interface (GUI) for a more user-friendly experience.

- Addressing potential legal and ethical considerations for a real-world deployment.

REFERENCES

- File Handling in C Language
- https://www.geeksforgeeks.org/basics-file-handling-c/
- Pointers Application – detailed overview