CHAPTER - 1

1. INTRODUCTION

Online Voting System is developed with the aim of creating a platform which will make voting process easier, allowing citizens above 18 years of age and any sex can give his\her vote without going to any physical polling station. There is a database consisting of records and details of the citizens and candidates who are eligible to vote are maintained by the system. This mini project is implemented in C++

This program can perform following function:

1. A new user can create account if and only if he satisfied following requirements:
   a) User minimum age should be 18yrs
   b) 10 Digit UID (Unique Identity) –Aadhar No.
   c) 10 Digit valid Phone no
2. If user already has an account they can login by entering their UID and password.
3. One User is allowed to vote only once.
4. Every user can change their password or phone no at any time. Other than that user will not be allowed to change details.
5. Any user can become candidate. (Admin access)
6. Admin can see the result Statistics like
   a) Total vote given.
   b) No of vote obtained by each candidate.
   c) Vote obtained in percentage.
   d) Winner (who have the max no of votes).
7. Data are stored in binary form so very hard to hack.

CHAPTER – 2

2.1 Objectives of the project

- Making voting procedure easy
- Saving time
- Avoids errors such as invalid votes and miscalculation of votes
- Fast voting result
- counting the total number of votes cast
- calculating the percentage of total votes
- calculating the total votes cast for each candidate
- calculating the percentage of votes for each candidate
- finding the wining candidates in the election based on the total votes they received

2.2 REQUIREMENTS

2.2.1 HARDWARES

- Processor: Intel Premium IV or V/AMD Athlon or Higher
- RAM: 512 MB
- Hard disk-100 GB

2.2.2 SOFTWARES

- Operating System: Windows 7 or any updated version of windows
- OOPs Concept used: Classes, Object, Polymorphism, Inheritance, Friend Function.
- Language: C++
- Compiler: DEV C++

2.3 Algorithm/Pseudo code

2.3.1 For Login

```
int user::login(char id[20])
{
        user u;
        char p[20];
        int found=0;
        ifstream fdata;
        fdata.open("Data\\user.txt",ios::in|ios::binary);

        while(fdata.read((char*)&u,sizeof(u)))
        {
                if(stricmp(u.uid,id)==0)
                {
                        found=1;
                        cout<<"Enter Password : ";
                        strcpy(p,password());
                        if(strcmp(p,u.pwd)==0)
                                return(1);
                        else
                        {
                                cout<<"Incorrect Password";
                                return(0);
                        }
                }
        }
        fdata.close();
        if(found==0)
        {
                cout<<"\nUID not found";
                return(0);
        }
}
```

2.3.2 For Registration

```
void user::input_data()
{
        user u;
        char co_pwd[20];
        int ch;
        ofstream fdata;
        fdata.open("Data\\user.txt",ios::app|ios::binary);

        system("cls");
        dispchar();
        cout<<"\t\t\t\t NEW USER ";
        dispchar();

        cout<<"Enter Unique ID : ";
        cin>>u.uid;
        if(check_uid(u.uid)==0 && check_pno_uid(u.uid)==1)
        {
                cout<<"Enter name : ";
                cin>>u.name;
                cout<<"Gender M/F :";
                cin>>u.gender;
        cout<<"Enter Age : ";
        cin>>u.age;
        if(u.age>=18)
        {
            cout<<"Father's Name : ";
            cin>>u.fathers_name;
            cout<<"Enter pno : ";
            cin>>u.pno;
if(check_pno_uid(u.pno) && (u.pno[0]=='9' || u.pno[0]=='8' || u.pno[0]=='7'))
            {
                cout<<"Enter password : ";
                strcpy(u.pwd,password());
                cout<<"\nConfirm password : ";
                strcpy(co_pwd,password());

                if(strcmp(u.pwd,co_pwd)==0)
                {
            cout<<"\n\nDo you want to save :\n";
            cout<<"1: Save \n2: Exit Without Saving\nEnter Choice : ";
                    cin>>ch;
                    if(ch==1)
                    {
                        fdata.write((char*)&u,sizeof(u));
                        cout<<"\nData Saved";
                    }
                }
                else
                {
                    cout<<"\nBoth Passowrds are diffrent";
```

```
                        }
                    }
                    else
                    {
                        cout<<"\n\nInvalid pno";
                    }
                }
                else
                {
                    cout<<"\nUnderAge";
                }
            }
            else
            {
                if(check_uid(u.uid)!=0)
                    cout<<"UID already present";
                else
                    cout<<"Invalid UID";
            }

            fdata.close();
        }
```

2.3.3 for voting:

```
  void vote::voting(char id[20])
  {
      vote v;
      candidate c;
      int found=0;
      ofstream fvote;
```

```
        ifstream fcandidate;
        fvote.open("Data\\vote.txt",ios::app|ios::binary);
        fcandidate.open("Data\\candidate.txt",ios::in|ios::binary);
        system("cls");
        dispchar();
        cout<<"\t\t\t\t VOTING PAGE";
        dispchar();
        cout<<"\n\n\tCandidate Info";
        c.display_data();
        cout<<"\n\n\n\n\n\tEnter Party name for vote : ";
        cin>>v.choice;

        while(fcandidate.read((char*)&c,sizeof(c)))
        {
                if(stricmp(c.party_name,v.choice)==0)
                {
                        strcpy(v.uid,id);
                        fvote.write((char*)&v,sizeof(v));
                        cout<<"\n\nThank For the Voting\n";
                        found=1;
                }
        }
        fvote.close();
        fcandidate.close();
        if(found==0)
        {
                cout<<"Party not found\n";
        }
    }
```

2.4 Class Diagram

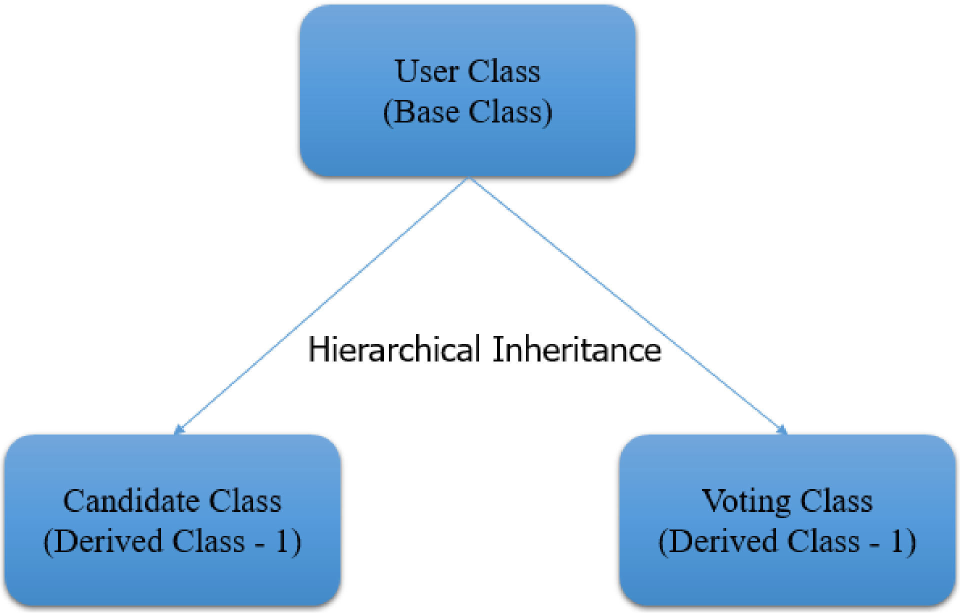


Fig : 4.1

CHAPTER 3

3. IMPLEMENTATION

3.1 Data Abstraction:

Data abstraction is the method of providing only essential details by hiding certain implementation details to user is known as data abstraction. This can be implemented using “private” modes or “protected” mode.

```
class user
{
protected:
    char uid[20],name[20],gender[6],fathers_name[20],pno[11],pwd[20];
    int age;
};

class candidate : public user
{
 protected:
   char party_name[20];
};

class vote
{
protected:
  char uid[20];
  char choice[20];
};
```

3.2 Encapsulation, Classes:

Data encapsulation is defined as wrapping of data members and member function into a single unit is known as encapsulation.

All the member functions in the project uses encapsulation concept as all of them uses or candidate class data member.

```
class user
{
    protected:
    char uid[20],name[20],gender[6],fathers_name[20],pno[11],pwd[20];
    int age;
    public:
        friend class candidate;
        void input_data();
        void display_data();
        void display_one_user(char []);
        int check_uid(char []);
        int login(char []);
        void change_pno(char []);
        void change_pwd(char []);
};

class candidate: public user
{
    protected:
        char party_name[20];
        public:
        void input_data();
```

```
                void display_data();
                friend class vote;
};

class vote
{
        protected:
                char uid[20];
                char choice[20];
        public:
                void voting(char []);
                void vote_stat();
                int count_vote();
                int count_vote(char []);
                int check_vote(char []);
};
```

3.3 Inheritance:

Inheritance is the process by which object of one class acquire the properties of object of another class. In this project all the data member of user class and candidate call are common so data member of user class will get inherited to candidate class, i.e single level inheritance.

3.4 Polymorphism:

Polymorphism is the act of representing the data in more than one form.

This project contain two class with the same name but there work are difference. Function name : count_vote() will count total no of vote, and sunction with count_vote(candidate_party_name) will count total no of vote given to that party.

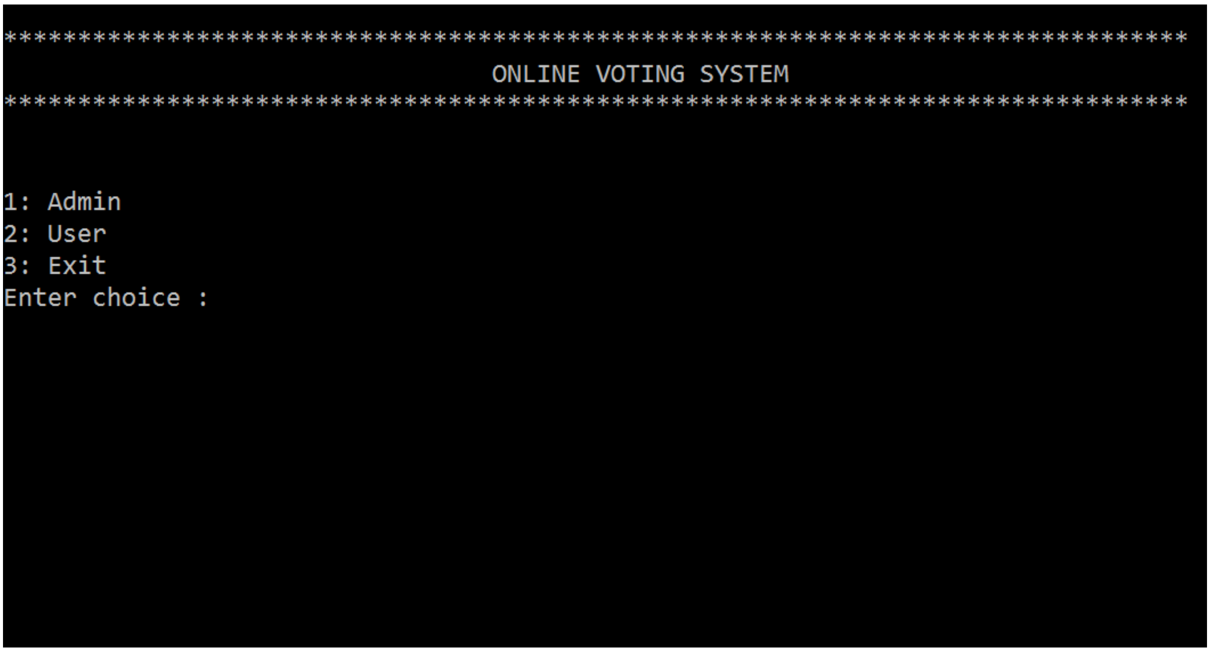
```
int vote::count_vote(char pname[20])
{
        int count=0;
        vote v;
        ifstream fvote;
        fvote.open("Data\\vote.txt",ios::in|ios::binary);
        while(fvote.read(((char*)&v),sizeof(v)))
        {
                if(stricmp(v.choice,pname)==0)
                count++;
        }
        fvote.close();
        return(count);
}
```

```
int vote::count_vote()
{
        int count=0;
        vote v;
        ifstream fvote;
        fvote.open("Data\\vote.txt",ios::in|ios::binary);
        while(fvote.read(((char*)&v),sizeof(v)))
        {
                count++;
        }
        fvote.close();
        return(count);
}
```

3.5 Friend functions:

Friend functions are the functions which are used to access the data members of a class and are not the member functions. Friend functions are only declared inside a class and is defined outside a class.

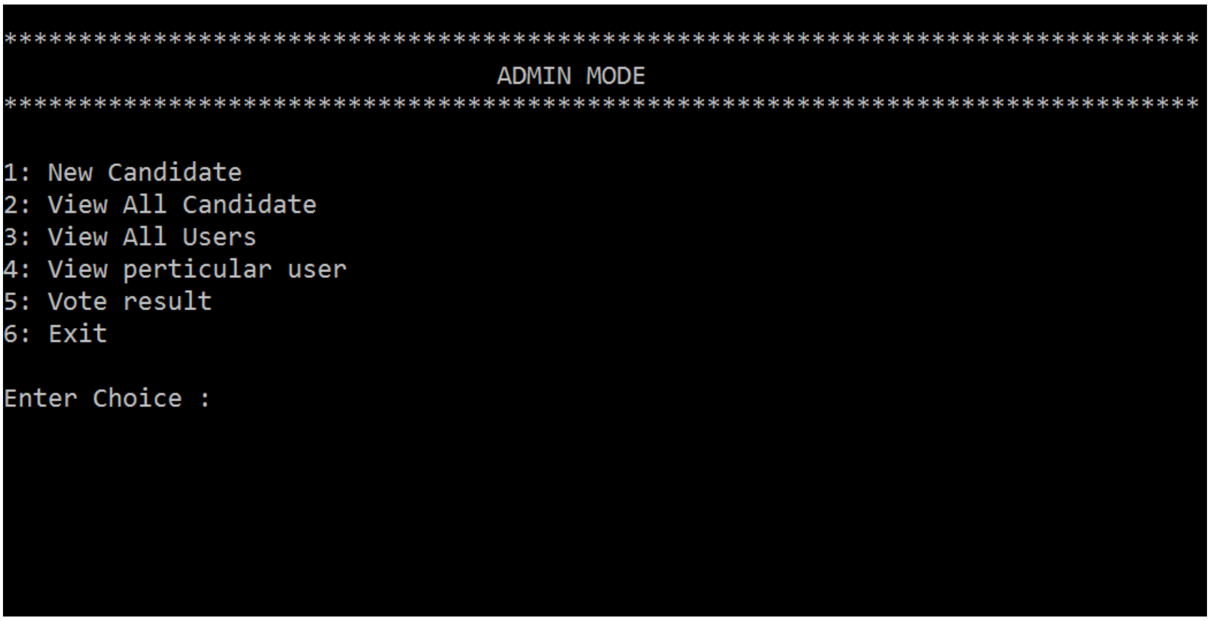
4. SAMPLE OUTPUT



```
********************************************************************************
                                 ONLINE VOTING SYSTEM
********************************************************************************


1: Admin
2: User
3: Exit
Enter choice :
```

Fig 4.1: Login Page

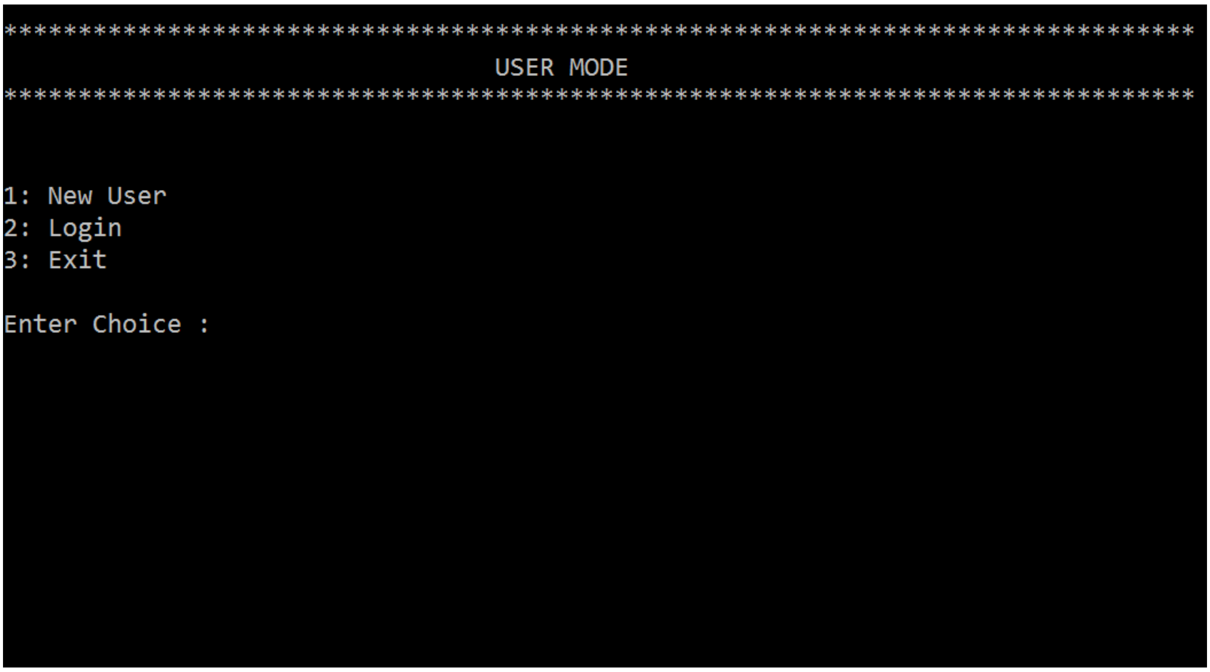


```
*********************************************************************************
                                 ADMIN MODE
*********************************************************************************

1: New Candidate
2: View All Candidate
3: View All Users
4: View perticular user
5: Vote result
6: Exit

Enter Choice :
```

Fig 4.2: Admin Mode



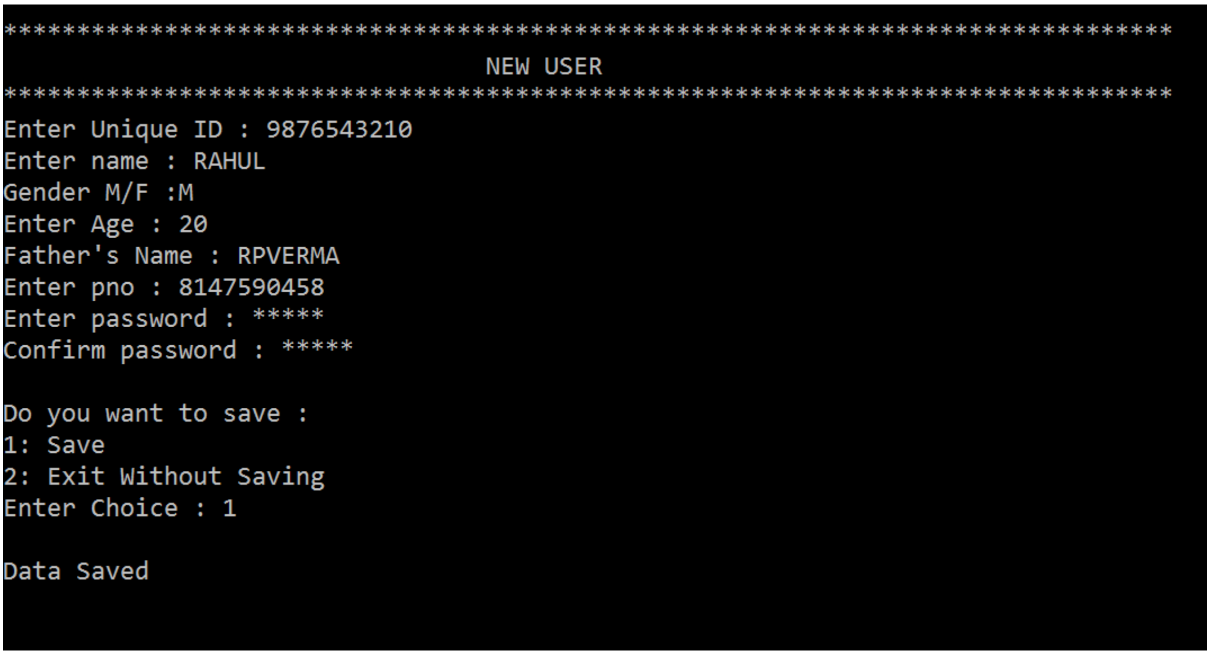
```
********************************************************************************
                                 USER MODE
********************************************************************************



1: New User
2: Login
3: Exit

Enter Choice :
```

Fig 4.3: User Mode

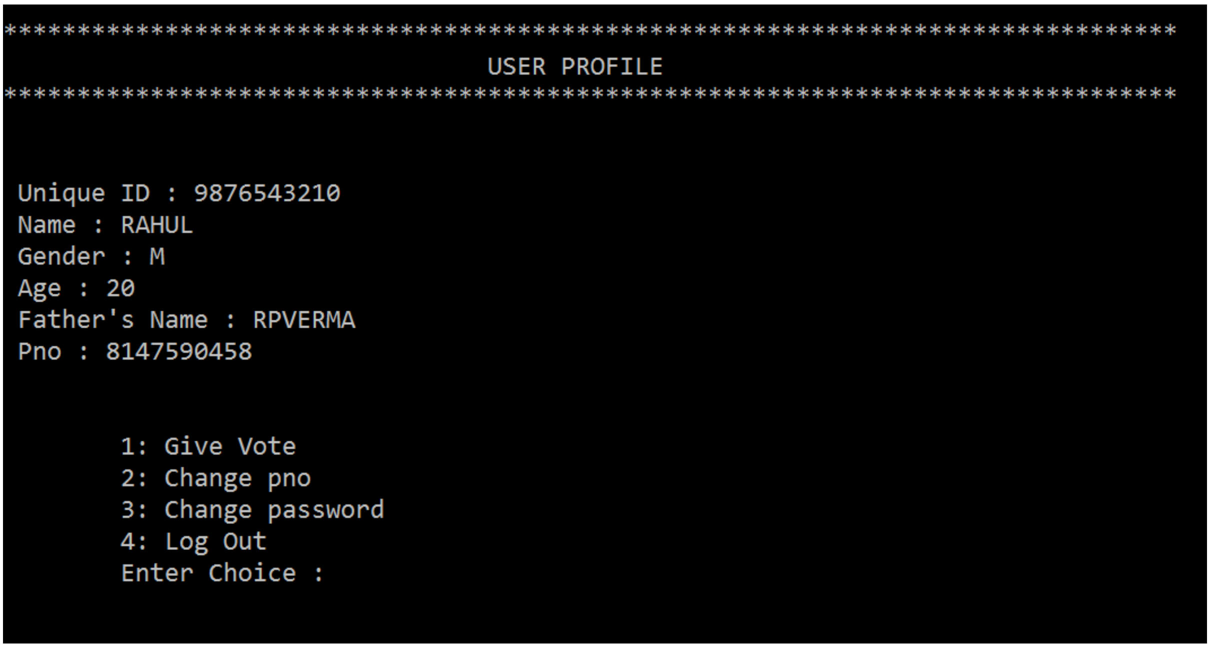


```
********************************************************************************
                                 NEW USER
********************************************************************************
Enter Unique ID : 9876543210
Enter name : RAHUL
Gender M/F :M
Enter Age : 20
Father's Name : RPVERMA
Enter pno : 8147590458
Enter password : *****
Confirm password : *****

Do you want to save :
1: Save
2: Exit Without Saving
Enter Choice : 1

Data Saved
```

Fig 4.4: New User

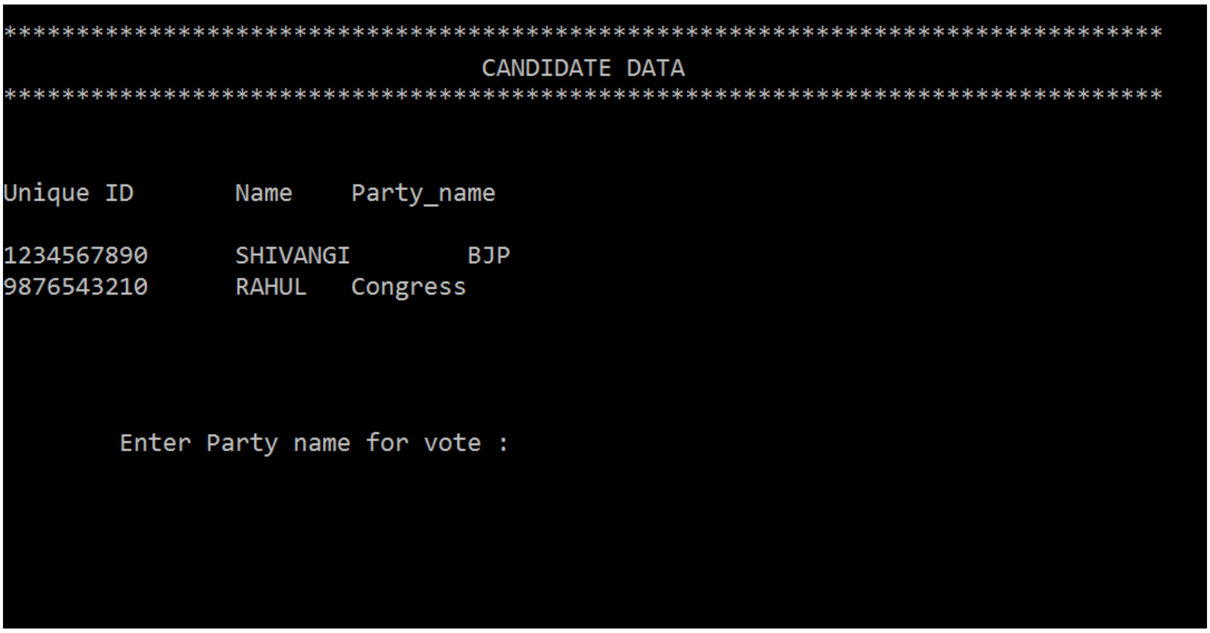


```
********************************************************************************
                                 USER PROFILE
********************************************************************************


 Unique ID : 9876543210
 Name : RAHUL
 Gender : M
 Age : 20
 Father's Name : RPVERMA
 Pno : 8147590458


        1: Give Vote
        2: Change pno
        3: Change password
        4: Log Out
        Enter Choice :
```

Fig 4.5: User Profile



```
*******************************************************************************
                                 CANDIDATE DATA
*******************************************************************************


Unique ID       Name    Party_name

1234567890      SHIVANGI        BJP
9876543210      RAHUL   Congress




        Enter Party name for vote :
```

Fig 4.6: Voting Screen

5. Conclusion

This Online Voting system will manage the Voter's information by which voter can login and use his voting rights. The system will incorporate all features of Voting system. It provides the tools for maintaining voter's vote to every party and it count total no. of votes of every party. There is a DATABASE which is maintained by the admin in which all the names of voter with complete information is stored.

In this user who is above 18 year's register his/her information on the database and when he/she want to vote he/she has to login by his id and password and can vote to any party only single time. Voting detail store in database and the result is displayed by calculation. By online voting system percentage of voting is increases. It decreases the cost and time of voting process. It is very easy to use and It is very less time consuming. It is very easy to debug.

6. BIBLIOGRAPHY

1. https://www.wikipedia.org/
2. Object oriented programming By Sumita Arora
3. https://www.geeksforgeeks.org/