ELECTION SYSTEM

SEM III C++ PROJECT

CONTENTS

What we are going to discuss

TEAM

Group 13

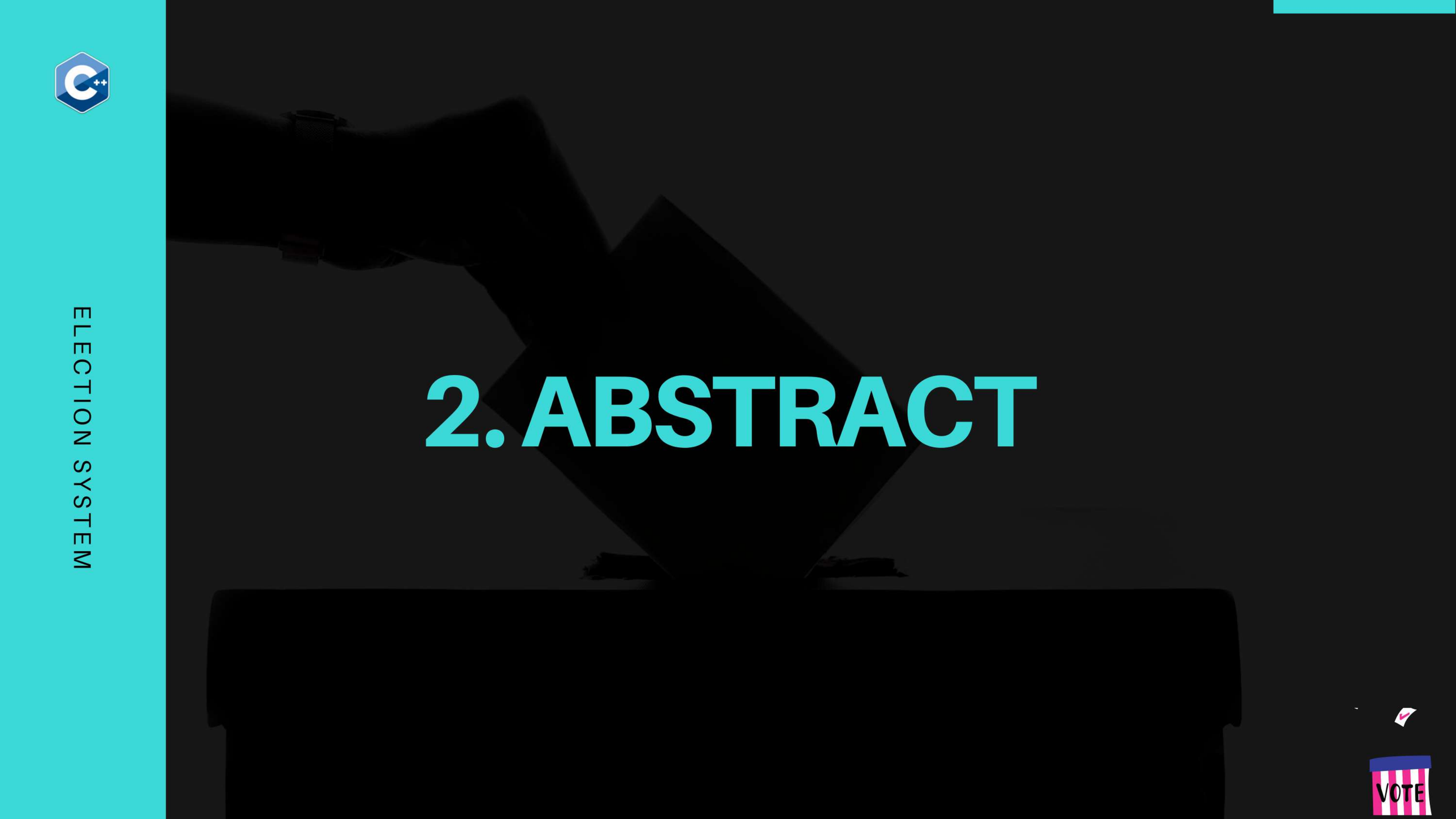
Guided by: Dr. Rajiv Senapati

1 Abdul Jawad Khan | AP21110010131

2 Subodh Amru Kiliveti | AP21110010133

3 Mahendra Kumar | AP21110010193

4 Suprabath Gangiredla | AP21110011185



2. ABSTRACT

ELECTION SYSTEM

ABSTRACT

General information about the project

1

"vote" means to choose from a list, to elect, or to determine

2

Goal of voting is to come up with leaders of the people's choice

3

This program allows eligible voters to cast their electronically.

4

The program gives faster, accurate results

5

However, the users/citizens shall be introduced to the concept of this type of election beforehand.

ELECTION SYSTEM

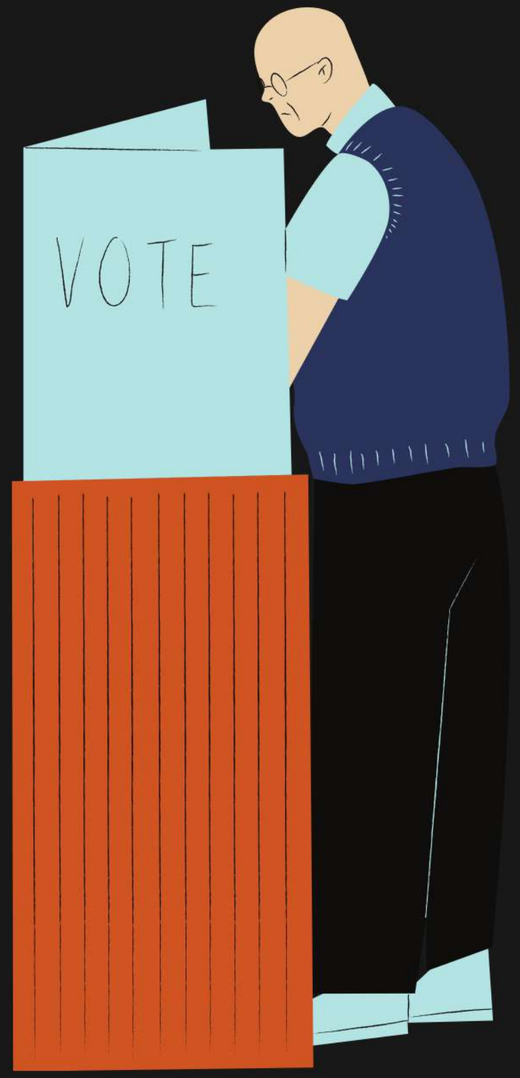
3. INTRODUCTION

INTRODUCTION

WHAT WE INTEND TO DO THROUGH THIS PROJECT

OBJECTIVES OF THIS PROJECT:

- Get rid of traditional ballot boxes or EVMs and create a faster voting system.
- To eliminate the worry for EVM's malfunctioning.
- Can work on any system including the low end.



INTRODUCTION

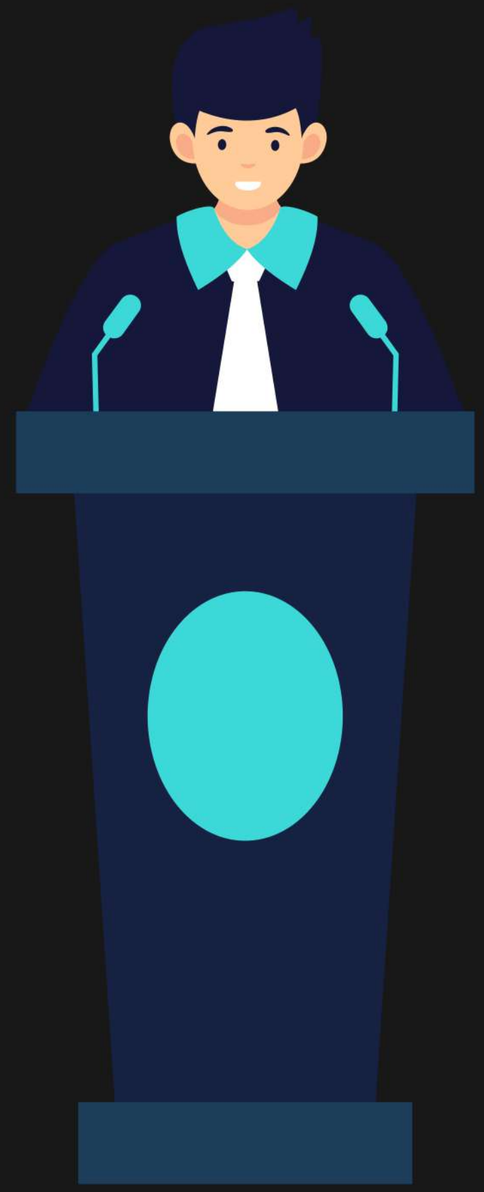
FUNCTIONALITIES OF THE ELECTRONIC VOTING SYSTEM

ALLOWS THE VOTER TO

- Register themselves
- Cast to vote for any candidate of their choice

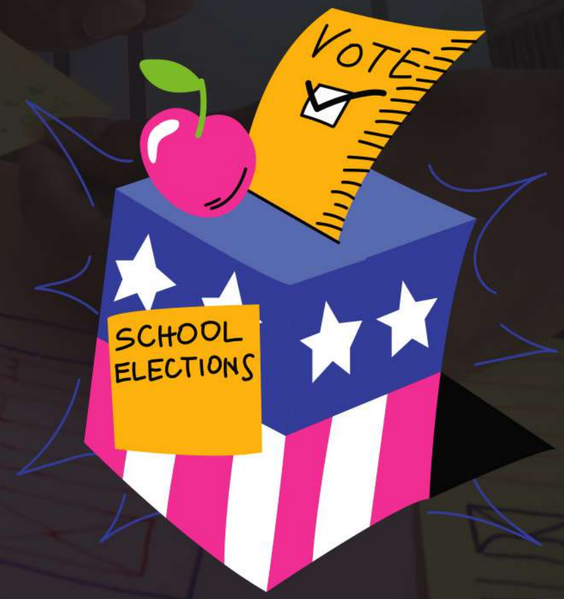
ALLOWS THE OFFICER TO

- Update the candidates contesting in the elections before the voters can start voting.
- See the number of votes that were cast for each contesting candidate, after the voters have given their votes.



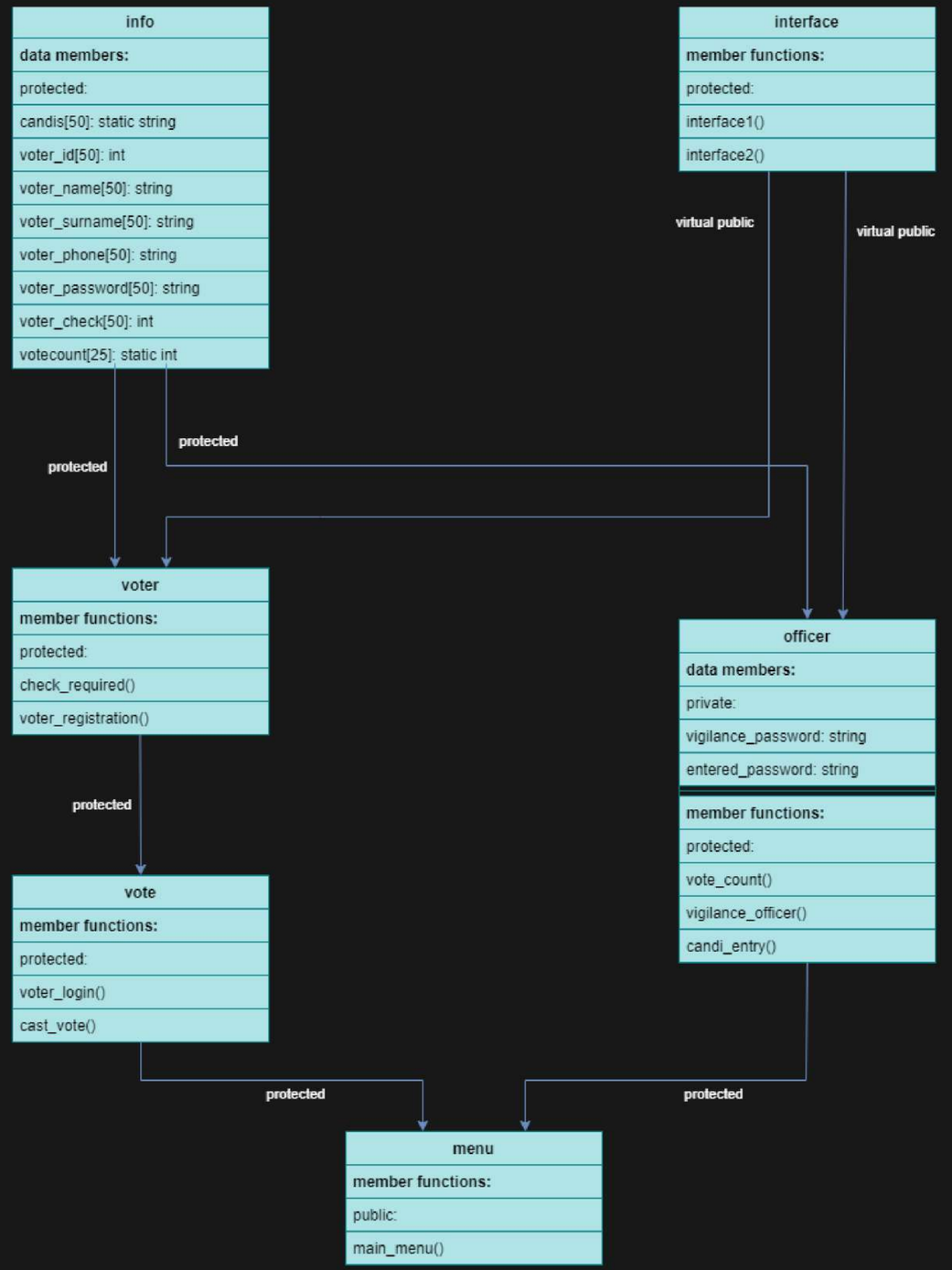
ELECTION SYSTEM

4. PROJECT DESIGN



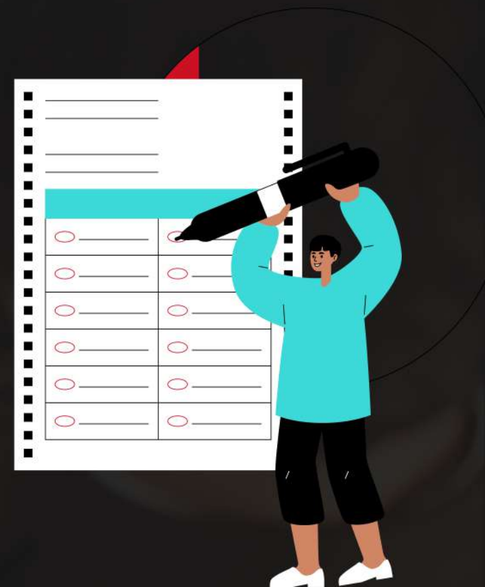
PROJECT DESIGN

- We designed this project using the core concepts of object-oriented programming (OOP).
- Classes and objects, data encapsulation and abstraction, array data members, static data members, and inheritance are the Object-Oriented concepts that we used to design this project.
- While designing the project, we tried to keep data members as protected as possible because the data stored in this project is very sensitive.
- The concepts of protected and virtual public inheritance were used. The UML diagram for this project is as follows:



ELECTION SYSTEM

5. PROJECT IMPLEMENTATION



PROJECT IMPLEMENTATION

class info:

- This is one of the base classes used in our program.
- The purpose of this class is to store all the data that will be used throughout the execution of the program.
- This class stores the voter ID, name, surname, phone number, and password of all the voters.
- It also has a variable voter_check which checks if the voter has previously voted.
- Apart from this, this class also has two static variables namely static string candis and static int votecount which store the names of candidates that are contesting in the elections and also the number of votes that were cast for them.

```
7   class info
8   {
9       protected:
10      static string candis[25];
11      int voter_id[50];
12      string voter_name[50];
13      string voter_surname[50];
14      string voter_phone[50];
15      string voter_password[50];
16      int voter_check[50];
17      static int votecount[25];
18  };
19
20  string info::candis[25];
21  int info::votecount[25];
22
```

PROJECT IMPLEMENTATION

class interface:

This is another base class that is used in our program. It has two member functions namely interface1() and interface2() both of which are protected.

void interface1() and void interface2():

- These two functions are used to give good readability and look to the output.

```
23   class interface
24   {
25       protected:
26       void interface1();
27       void interface2();
28   };
```

```
62   void interface::interface1()
63   {
64       cout<<"\n\t\t\t|=============================================|";
65       cout<<"\n\t\t\t|=====       ONLINE ELECTION SYSTEM       ======|";
66       cout<<"\n\t\t\t|=====            C++ PROJECT             ======|";
67       cout<<"\n\t\t\t|=============================================|";
68       cout<<"\n";
69   }
70
71   void interface::interface2()
72   {
73       cout<<"\n\t\t\t|=============================================|";
74       cout<<"\n\t\t\t|=====    WELCOME TO ELECTIONS - 2022     ======|";
75       cout<<"\n\t\t\t|=============================================|";
76       cout<<"\n";
77   }
```

PROJECT IMPLEMENTATION

class voter:

This class is protectively inherited from the base class info and virtually inherited from the base class interface. This class has two member functions namely check_required() and voter_registration() both of which are declared under the access specifier ‘protected’.

void check_required():

- This function belongs to the 'voter' class. This function is basically to check if the voter has voted before or not.

void voter_registration():

- This function is used to register the voter. It is called by void_voterlogin(). It gives a unique voter Id to each voter. Then it takes the inputs from the user such as name, surname, and phone number, and finally allows the user to set a password. Following all the steps the message ‘You have successfully completed the registration is displayed.

```
30   class voter:protected info,virtual public interface
31   {
32       protected:
33       void check_required();
34       void voter_registration();
35   };
79   void voter::check_required()
80   {
81       int noo=0;
82       for(noo=0;noo<n;noo++)
83       {
84           voter_check[noo]=0;
85       }
86       getc;
87   }
88
89   void voter::voter_registration()
90   {
91       system("cls");
92       interface1();
93       interface2();
94       static int x=0;
95       static int id=3458889;
96
97       check_required();
98       cout<<"\n\t\t Your voter id is "<<id;
99       cout<<"\n\t\t Please enter the voter id alloted to you: ";
100      cin>>voter_id[i];
101      cout<<"\n\t\t Enter your first name: ";
102      cin>>voter_name[i];
103      cout<<"\n\t\t Enter your last name: ";
104      cin>>voter_surname[i];
105      cout<<"\n\t\t Enter your phone number: ";
106      cin>>voter_phone[i];
107      cout<<"\n\t\t Enter your password: ";
108      cin>>voter_password[i];
109      cout<<"\n\t\t You have successfully completed the registration!!!";
110      voter_check[i] = 1;
111      i++;
112      id++;
113
114      fflush(stdin);
115      getchar();
116  }
```

PROJECT IMPLEMENTATION

class vote:

This class is protectively inherited from the class voter. It has two member functions namely voter_login() and cast_vote() both of which are protected.

```
37  class vote:protected voter
38  {
39      protected:
40      void voter_login();
41      void cast_vote();
42  };
```

void voter_login():

- This function is used to log the voter in. The sub-function used here is cast_vote(), voter_registration(). Depending on the input from the user the control flows into any one of these subfunctions or the user is directed back to the home page.

```
117 void vote::voter_login()
118 {
119     system("cls");
120     int choice_voter;
121     do
122     {
123         system("cls");
124         interface1();
125         interface2();
126         cout<<"\n\t\t\t|=============================================|";
127         cout<<"\n\t\t\t|=====           |VOTER MENU|            ======|";
128         cout<<"\n\t\t\t|=============================================|";
129         cout<<"\n";
130         cout<<"\n\t\t If you are a new voter please register yourself first by selecting choice 2";
131         cout<<"\n\t\t 1. Cast your vote";
132         cout<<"\n\t\t 2. Voter registration";
133         cout<<"\n\t\t 3. Back";
134         cout<<"\n\t\t Please enter your choice:";
135         cin>>choice_voter;
136
137         switch(choice_voter)
138         {
139             case 1: cast_vote();break;
140             case 2: voter_registration();break;
141             case 3: cout<<"\n Home page";system("cls");break;
142             default: cout<<"\n Kindly give a valid Input!";
143         }
144     }while(choice_voter!=3);
145     fflush(stdin);
146 }
```

PROJECT IMPLEMENTATION

void cast_vote():

- This is a function through which we allow the user to cast their vote. It is in the class 'vote'.

- Requirements For voting:
  1) Firstly, the contesting candidates must be updated by the officer.
  2) Secondly, the voter ID must be registered and the user must enter the password of that voter ID correctly, only then the list of candidates will be displayed to them. The voter will be given 3 tries to enter the password correctly.

- After voting, the voter_check of that voter ID is updated which prevents the same voter from voting again. (This is the crucial part which is shown in the snippet beside).

```
if(voter_check[hi]==1)
{
    cout<<"\n\t\t Voter first name : "<<voter_name[hi];
    cout<<"\n\t\t Voter last name : "<<voter_surname[hi];
    cout<<"\n\t\t Voter phone number : "<<voter_phone[hi];
    cout<<"\n\n\t\t ### Please choose your Candidate ####";
    for(int i=0;i<n;i++)
    {
        cout<<"\n\t\t"<<i+1<<". "<<candis[i];
    }
    cout<<"\n\n\t\t Input your choice (1 - "<<n<<" ) : ";
    cin>>choice;
    votecount[choice-1]++;
    voter_check[hi]++;
}
else
{
    cout<<"\n\t\t Sorry you have already voted!!!";
}
```

PROJECT IMPLEMENTATION

class officer:

This class is protectively inherited from the class info and virtually inherited from the class interface. It has two private data members namely, vigilance_password and entered_password. This class also has three member functions namely, void vote_count(), void vigilance_officer(), void candi_entry() which are all protected.

void vote_count():

- This function is used to know the votes obtained by each candidate. It is called by the function 'void vigilanceofficer( )'.

void vote_count():

- This function is used by the officer to add the number of candidates contesting in the elections. Once the number of candidates are entered only then the voters are allowed to cast their votes.

```
44  class officer:protected info,virtual public interface
45  {
46      private:
47      string vigilance_password="#Good@Officer";
48      string entered_password="";
49      protected:
50      void vote_count();
51      void vigilance_officer();
52      void candi_entry();
53  };
219  void officer::vote_count()
220  {
221      system("cls");
222      interface1();
223      interface2();
224      cout<<"\n\t\t ##### Voting Statics ####";
225      for(int i=0;i<n;i++)
226      {
227          cout<<"\n\t\t"<< candis[i]<<  "                "<<votecount[i];
228      }
229      fflush(stdin);
230      getchar();
231  }
232
278  void officer::candi_entry()
279  {
280      system("cls");
281      interface1();
282      interface2();
283      int n1,p;
284      cout<<"\n\t\tEnter Number of Parties:" ;
285      cin>>n1;
286      n=n+n1;
287      for(p=i;p<n;p++)
288      {
289          cout<<"\n\t\tParty "<<p+1<<" name :";
290          cin>>candis[p];
291      }
292      cout<<"\n\t\tParties Updated Successfully!";
293
294      i=p;
295      fflush(stdin);
296      getchar();
297  }
```

PROJECT IMPLEMENTATION

class menu:

This class is used to display the main menu of the program. It is protectively inherited from the class vote and officer. This class only has one member function main_menu() which is public.

void main_menu():

- This is the only public member function in our entire program. It was declared publicly so that it can be called by using the object of the menu class in the main() function.

- The purpose of this function is to display the main menu of our program.

```
55  class menu:protected vote,protected officer
56  {
57      public:
58      void main_menu();
59  };
299 void menu::main_menu()
300 {
301     system("cls");
302 
303     int i;
304     int choice;
305 
306     while(1)
307     {
308         interface1();
309         interface2();
310         cout<<"\n\t\t\t-------------------------------------------------";
311         cout<<"\n\t\t\t--------              LOGIN              --------";
312         cout<<"\n\t\t\t-------------------------------------------------";
313         cout<<"\n\t\t\t-------        1. Vigilance officer        -------";
314         cout<<"\n\t\t\t-------             2. Voter               -------";
315         cout<<"\n\t\t\t-------             3. Exit                -------";
316         cout<<"\n\t\t\t-------------------------------------------------";
317         cout<<"\n\t\t\tPlease enter your choice : ";
318         cin>>choice;
319         switch(choice)
320         {
321             case 1: vigilance_officer();break;
322             case 2: voter_login();break;
323             case 3: cout<<"\n Session ended";exit(0);break;
324             default: system("cls");cout<<" Kindly give a valid Input!\n ";break;
325         }
326     }
327     fflush(stdin);
328     getchar();
329 }
```

PROJECT IMPLEMENTATION

int main():

- This is the main function of the program. From here the execution of the program begins.

- Here the object of the class menu is declared as m. Using this object, we access the main_menu() member function of the menu class.

```
331    int main()
332    {
333        menu m;
334        m.main_menu();
335        return 0;
336    }
```

ELECTION SYSTEM

6. RESULTS

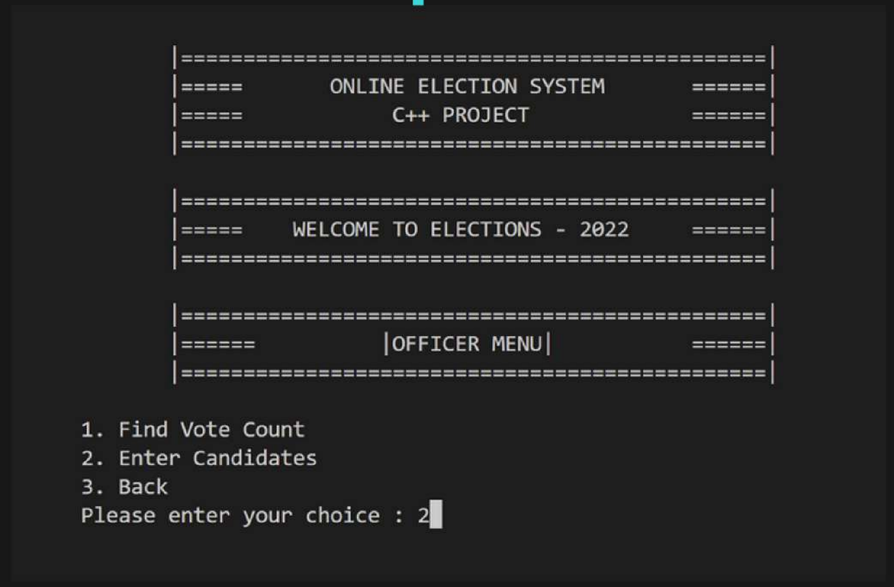
Results

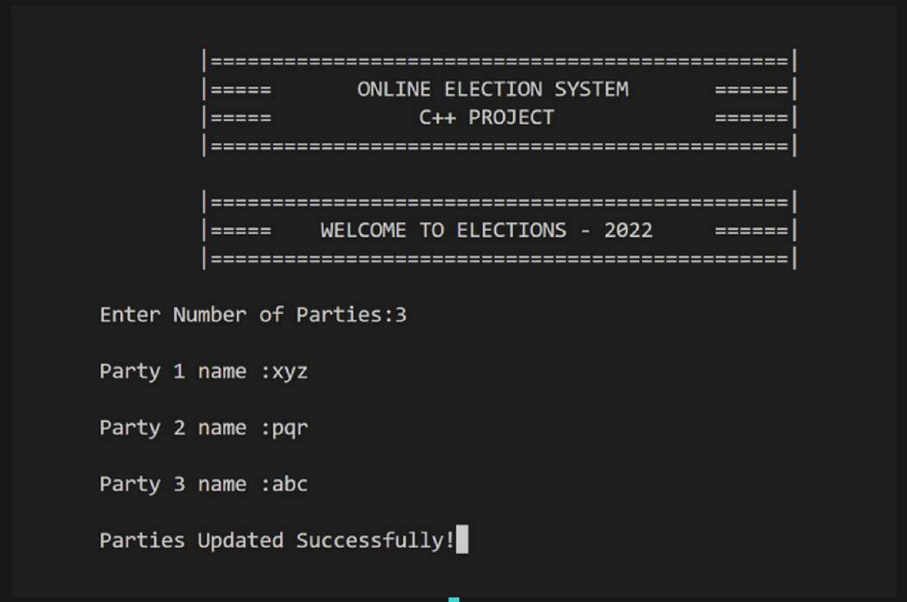
```
|================================================|
|=====       ONLINE ELECTION SYSTEM       ======|
|=====            C++ PROJECT             ======|
|================================================|

|================================================|
|=====    WELCOME TO ELECTIONS - 2022     ======|
|================================================|

------------------------------------------------
--------              LOGIN              --------
------------------------------------------------
-------        1. Vigilance officer        -------
-------             2. Voter               -------
-------             3. Exit                -------
------------------------------------------------
Please enter your choice :
```

Program starts with displaying the main menu of the program.
Here the user gets an option to login as a officer or a voter

Results

On selecting option 1 on the main menu we get the officer menu as shown below





By selecting option number 2 from officer menu the officer will be able to update the contesting candidates as shown above

Results

On selecting option 2 on the main menu we get the voter menu as shown below

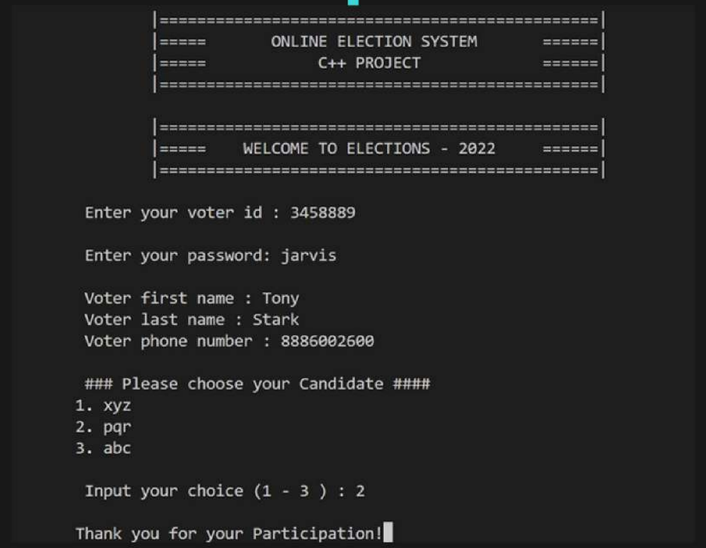
```
|=============================================|
|=====       ONLINE ELECTION SYSTEM       ======|
|=====            C++ PROJECT             ======|
|=============================================|

|=============================================|
|=====    WELCOME TO ELECTIONS - 2022     ======|
|=============================================|

|=============================================|
|=====            |VOTER MENU|             ======|
|=============================================|

If you are a new voter please register yourself first by selecting choice 2
1. Cast your vote
2. Voter registration
3. Back
Please enter your choice:2
```

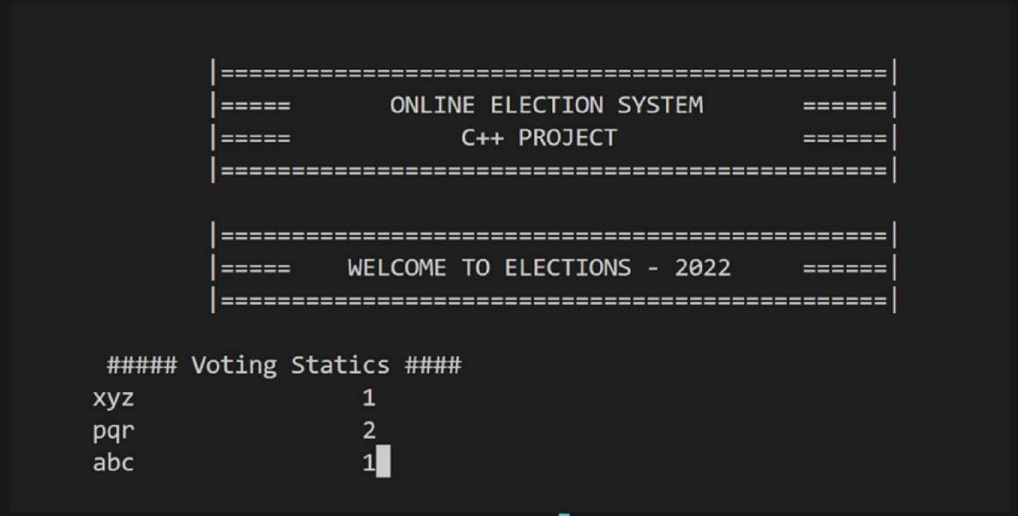
```
|=============================================|
|=====       ONLINE ELECTION SYSTEM       ======|
|=====            C++ PROJECT             ======|
|=============================================|

|=============================================|
|=====    WELCOME TO ELECTIONS - 2022     ======|
|=============================================|

Your voter id is 3458889
Please enter the voter id alloted to you: 3458889

Enter your first name: Tony

Enter your last name: Stark

Enter your phone number: 8886002600

Enter your password: jarvis

You have successfully completed the registration!!!
```

By selecting option number 2 from voter menu the voters will be able to register themselves by entering the required details one by one as shown above

Results

On selecting option 1 on the voter menu, the voters will be able to cast their votes for any candidate of their choice. The voters are required to enter the registered voter ID and password .





By selecting option number 1 from officer menu the officer will be able to see the vote count of each candidate and announce the results of elections as shown above

THANK YOU