COMPUTER SCIENCE INVESTIGATORY PROJECT

AIM:
DEVELOPMENT OF A VOTING PROGRAM FOR SCHOOLS

RAUNAK SENGUPTA
GRADE XII B
14

Presidency School Bangalore South, Bilekahalli, off Bannerghatta Road, Bangalore- 560076

CERTIFICATE

Name: Raunak Sengupta

Class: XII B

Roll no: 14

Exam No: _____

Institution: Presidency School Bangalore South

This is certify that the project work named ‘STUDENT VOTING SYSTEM’ has been carried out and completed sincerely and satisfactorily.

_____

Teacher In-charge

_____

Examiner

_____

Principal

Date: _____

Institution Rubber Stamp

ACKNOWLEDGEMENT

I warmly acknowledge the continuous encouragement and timely suggestions offered by our Principal Ms. J Bhuvaneswari and I am thankful for the opportunity to make use of the facilities available in the campus to carry out the project successfully.

I am highly indebted to Ms. Tamil Selvi for her constant supervision and guidance during the course of the project. Conclusively, I would like to thank all those involved directly or indirectly in the successful completion of my project once again.

Sincerely,

Signature of candidate

CONTENTS

INTRODUCTION

The aim of this program is to modernize and computerize the system of voting done in a school. This program will ensure that that the elections are conducted in a quick and efficient way.

Since the admin menu of this program is password protected it will ensure that the elections are not rigged.

The main menu offers the following options to the user:

Are you:-

1. Administrator

2. Voter

3. Exit

Further,

The Administrator menu offers the following options which can be used to display/enter the posts and candidates according to users' choice

1. Enter new posts

2. Enter new candidate

3. List all posts

4. List all candidates for a particular post

5. Show the winner of every post

6. Delete post

7. Delete candidate

8. Return to welcome screen

9. Save and exit

The Voter menu on the other hand is for the students who are voting . The following options implement our prospect

1. List all posts
2. List all candidates for a post
3. Vote for a candidate for each post
4. Return to welcome screen
5. Save and exit

Modules:-

Input Module:

Input module enables user to input data of their choice. It's implemented through functions and includes the following functionalities

1. Get the new post-name and the candidates standing for that particular post (pname, cname, etc)

2. Input the vote of a student to the particular candidate.

Some functions involved in the Input Module are:

FUNCTION	INPUT REQUIRED
void newpost()	Post name
void newcand()	Candidate name
void vote()	Counts the vote by students by choosing candidate

Display Module:

All outputs in the program are made in this module. This module consists of the following functionality:

1. Display all the post in the elections.

2. Display the candidates for each post.

3. Display the winner of every post.

Some Functions involved:

FUNCTION	INPUT REQUIRED
void cdisp()	Post Name
void pdisp()	-
void winner()	-

Update Module:

Used to maintain and update information in the system. Functionalities include

1. Getting input of the new votes (data through input module and insertion via a separate function) and writing the votes into a file.

2. Deleting a particular post.

3. Deleting a particular candidate.

The Function involved:

FUNCTION	INPUT REQUIRED
void store()	-
void pdel()	Post name
void cdel()	Candidate name

Files used:

'Votes.DAT' which contains the data in the form of records implemented through structure variables.

SYSTEM SPECIFICATIONS AND DATA DESIGN

This is an introduction of the structures ‘cand’ and ‘post’ used:

structure ‘cand’ and ‘post’

DATA MEMBERS(Public)

Data Name	Data type
cand:-	
Cname	Character array(40)
Votes	Integer
cls	Character array(5)
post:-	
Pname	Character array(30)
*clist	Pointer array(20)
*link	Pointer
Cno	Integer

FUNCTION/PROCEDURE DESCRIPTION

Functions used in this program have been listed here along with their uses:

newpost() //creates a new post

newcand() //adds a candidate to a particular post

pdisp() //displays all the posts

cdisp() //displays all the candidate's name under a particular post

vote() //to take vote of voter

winner() //displays the candidate with the highest no of votes for each post

pdel() //to delete a post

cdel() //to delete a candidate

store() // to store the details of posts and candidates in a file

REQUIREMENT ANALYSIS

Hardware Requirements

•Processor: Intel® Pentium® 4 CPU

•CPU speed: 2.67 GHz

•RAM: 512 MB

•Hard disk memory: 80GB

•Cache: 512 KB

Software Components

•Operating System: Microsoft Windows XP

•Software: Turbo C++

C++ OVERVIEW

C++ is one of the most widely used development tools in the market today.
A language can be classified into low-level and high-level languages. The low level languages are machine dependent. Machine language and Assembly language are the low-level languages.

High-level language uses statements in English to give an instruction to the machine. These instructions are machine independent. It uses the translators to translate the source program in to machine executable form because the computer can understand only binary language.
A high-level language program can be translated using interpreter or compiler. Interpreter takes the source program line by line as input, translates one line at a time into an executable form if it has no errors otherwise displays an error message.

Compiler first translates the source program into object program if there are no errors and if there are any errors in the program it displays the list of the errors. After translating the object program can be executed. C++ is a compiler language.

Features of C++:-

1. Operators and operator overloading

C++ provides more than 35 operators, covering basic arithmetic, bit manipulation, indirection, comparisons, logical operations and others.

2. Objects

C++ introduces object-oriented programming (OOP) features to C. It offers classes, which provide the four features commonly present in OOP (and some non-OOP) languages: abstraction, encapsulation, inheritance, and polymorphism.

a. Abstraction

Abstraction is the process by which data and programs are defined with a representation similar in form to its meaning (semantics), while hiding away the implementation details.

b. Encapsulation

Encapsulation is the process of hiding information in order to ensure that data structures and operators are used as intended and to make the usage model more obvious to the developer.

c. Inheritance

Inheritance allows one data type to acquire properties of other data types. Inheritance from a base class may be declared as public, protected, or private.

d. Polymorphism

Polymorphism enables one common interface for many implementations, and enables objects to act differently under different circumstances.

The structure of a C++ program:-

Comments:-

A comment is text that is ignored by the compiler but which nonetheless conveys information to programmers. They are important and must be used to make code clear easier to understand. (// for lines, /* . . . */ for blocks of code)

Library inclusions:-

The inclusion of a header file indicates that the program uses methods from a library, which is a collection of stored functions.

Function prototypes:-

Computation in a C++ program is carried out in the context of functions.

A function is a unit of code that:

(1) Performs a specific operation and

(2) Is identified by name.

The main program:-

Every C++ program must contain a function with the name main. This function specifies the starting point for the computation and is called when the program starts up. When main has finished its work and returns, execution of the program ends.

Function definitions:-

Because large programs are difficult to understand in their entirety, most programs are broken down into several smaller functions, each of which is easier to understand. This implements modularity.

Variables:-

Data values in a program are usually stored in variables. You must declare that variable before you use it. Declaring a variable establishes the following properties:

Name:

Every variable has a name, which is formed according to a certain set of rules.

Type:

Each variable in a C++ program is constrained to hold values of a particular data type.

Lifetime:

Depending on how they are declared, some variables persist throughout the entire program, while others are created and destroyed dynamically as the program moves through various levels of function call.

Scope:

The declaration of a variable also controls what parts of the program have access to the variable, which is called its scope.

Local and global variables:-

Variables declared with the body of a function are called local variables. The scope of a local variable extends to the end of the block in which it is declared. The lifetime of a local variable is the time during which the function is active. When the function is called, space for each local variable is allocated for the duration of that function call. When the function returns, all its local variables disappear. Variable declared outside any function definition are called global variables. The scope of a global variable is the remainder of the file in which it is declared. Its lifetime continues throughout the entire execution of a program.

The concept of a data type:-

One of the reasons C++ requires all variables to be declared is that doing so constrains their contents to values of a particular data type.

Integer types:

Type int corresponds to the standard representation of an integer on the computer system. Values of type int are stored internally in storage units that have a limited capacity.

To overcome this, C++ defines three integer types—short, int, and long—distinguished from each other by the size of their domains.

Floating-point types:

Numbers that include a decimal fraction are called floating-point numbers, which are used to approximate real numbers in mathematics. C++ defines three different floating-point types: float, double, and long double.

Text types:

In C++, the coding system used to represent characters is called ASCII. Characters are collected together into sequences called strings. Strings make it possible to display messages on the screen. Strings can be manipulated and stored in much the same way as other types.

Header Files

Header File	Associated Functions
iostream.h	cout cin endl
conio,h	clrscr() getch()
string.h	strcmpi()
stdio.h	gets()
dos.h	delay()
fstream.h	open() write() close()

SOURCE CODE

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
#include<stdio.h>
#include<dos.h>
#include<fstream.h>
struct cand
{  char cname[40];
   int votes;
   char cls[5];
};

struct post
{  char pname[30];
   post *link;
   cand *clist[20];
   int cno;
};
post *top=NULL;
```

```
void newpost()
{  post *temp=new post;
   post *save;
   cout<<"Enter name of post:"<<endl;
   gets(temp->pname);
   (temp->link)=NULL;
   (temp->cno)=0;
   for(int i=0;i<20;i++)
     (temp->clist[i])=NULL;
   if(top==NULL)
     top=temp;
   else
   {  save=top;
      top=temp;
      (temp->link)=save;
   }
   cout<<"Post added\n\n";
}
```

```
void newcand()
{  cand *newc=new cand;
   char p[30];
   post *temp;
   temp=top;
   cout<<"Enter the name of the candidate:"<<endl;
   gets(newc->cname);
   cout<<"Enter the class of the candidate (eg:10c):"<<endl;
   gets(newc->cls);
   (newc->votes)=0;
   cout<<"Enter the post that the candidate is competing for:"<<endl;
   gets(p);
   while((temp!=NULL)&&(strcmpi(p,(temp->pname))!=0))
   {  temp=(temp->link);
   }
   if(temp==NULL)
     cout<<"Post not found\n";
   else
   {  (temp->clist[(temp->cno)])=newc;
      (temp->cno)++;
      cout<<"Candidate added\n";
   }
   cout<<endl;
}
```

```
void pdisp()
{  post *temp;
   temp=top;
   int i=1,flag=0;
   while(temp!=NULL)
   {  cout<<"Post "<<i<<':'<<(temp->pname)<<endl;
      temp=(temp->link);
      i++;
      flag=1;
   }
   if(flag==0)
     cout<<"No posts found\n";
   cout<<endl;
}

void cdisp()
{  char p[30];
   post *temp;
   temp=top;
   cout << "Please enter the post of the candidate\n";
   gets(p);
   while((temp!=NULL)&&(strcmpi(p,(temp->pname))!=0))
   {  temp=(temp->link);
```

```
  }
  if(temp==NULL)
    cout<<"Post not found\n";
  else
  {  for(int i=0;i<(temp->cno);i++)
     {  cout<<"Candidate "<<(i+1)<<":\nName:"<<(temp->clist[i]->cname)<<"\nClass:"<<(temp->clist[i]->cls)<<"\nVotes:"<<(temp->clist[i]->votes)<<"\n\n";
     }
  }
  cout<<endl;
}

void vote()
{  post *temp;
  temp=top;
  int ch,flag=0;
  while(temp!=NULL)
  {  flag=1;
     cout<<"Post:\n"<<(temp->pname)<<endl;
     for(int i=0;i<(temp->cno);i++)
     {  cout<<"Candidate "<<(i+1)<<":\nName:"<<(temp->clist[i]->cname)<<"\tClass:"<<(temp->clist[i]->cls)<<endl;
     }
     cout<<"Enter your choice:\n";
```

```
    cin>>ch;
    for(i=0;i<(temp->cno);i++)
    { if((i+1)==ch)
        (temp->clist[i]->votes)++;
    }
    temp=(temp->link);
    cout<<"Vote casted\n\n";
  }
  if(flag==0)
   cout<<"No posts found\n\n";
}

void winner()
{  post *temp;
  temp=top;
  int highest=0,flag=0;
  while(temp!=NULL)
  {  flag=1;
    cout<<"Post:\n"<<(temp->pname)<<endl;
    for(int i=0;i<(temp->cno);i++)
    {  if(highest<(temp->clist[i]->votes))
        highest=(temp->clist[i]->votes);
    }
    for(i=0;i<(temp->cno);i++)
```

```
    {  if(highest==(temp->clist[i]->votes))
          cout<<"Winner is:"<<(temp->clist[i]->cname)<<"\tFrom class:"<<(temp->clist[i]->cls)<<endl;
    }
    temp=(temp->link);
    cout<<endl;
  }
  if(flag==0)
   cout<<"No posts found\n\n";
}

void pdel()
{  char p[30];
  int flag=0;
  cout<<"Enter name of post to be deleted\n";
  gets(p);
  post *temp,*save;
  temp=top;
  while(temp!=NULL)
  {  flag=1;
    if(strcmpi(p,(temp->link->pname))==0)
    {  save=(temp->link);
       (temp->link)=(temp->link->link);
       for(int i=0;i<(save->cno);i++)
```

```
            delete (save->clist[i]);
          delete save;
          cout<<"The post and all its candidates were deleted\n\n";
          goto lab3;
      }
   }
   if(flag==0)
     cout<<"No posts found\n\n";
   lab3:
}

void cdel()
{  post *temp;
   temp=top;
   int i,flag=0,flag1=0;
   char p[40];
   cout<<"Enter candidate to be removed\n";
   gets(p);
   while(temp!=NULL)
   {  flag=1;
      for(i=0;i<(temp->cno);i++)
      {  if(strcmpi(p,(temp->clist[i]->cname))==0)
          {  delete (temp->clist[i]);
             (temp->cno)--;
```

```
            cout<<"Candidate deleted\n\n";
            flag1=1;
            goto lab1;
          }
      }
   }
   if(flag==0)
     cout<<"No posts found\n";
   if(flag1==0)
     cout<<"No candidates found\n\n";
   lab1:
}

void store()
{  fstream fout;
   post *temp=top;
   fout.open("Votes.DAT",ios::out|ios::binary);
   while(temp!=NULL)
   {  fout.write((char*)temp,sizeof(post));
      for (int i=0;i<(temp->cno);i++)
      {  fout.write((char*)temp->clist[i],sizeof(cand));
      }
      temp=temp->link;
   }  fout.close();  }
```

```
void main()
{  clrscr();
int ch1,ch2,ch6;
char ch3='y',ch4='y',ch5='y',ch7='y',ch8='y';
char pw[20],pc[20];
cout<<"WELCOME TO THE PSBS VOTING PROGRAM\nBy Raunak, Sharon and Parth\n(Loading";
delay(1000); cout<<'.'; delay(1000); cout<<'.'; delay(1000); cout<<".)"; delay(1000);
cout<<"\nSet new admin password:\n";
gets(pw);
while(ch7=='y'||ch7=='Y')
{  clrscr();
  set:
  cout<<"Are you an Administrator(1) or a Voter(2)? (1/2)\nPress 3 to exit\n";
  cin>>ch1;
  switch(ch1)
  {  case 1: cout<<"Enter password (case sensitive):\n";
             gets(pc);
             if(strcmp(pc,pw)!=0)
             {  cout<<"Wrong password\n";
                goto lab2;
             }
```

```
        else
        {  ch4='y';
           while(ch4=='y'||ch4=='Y')
           {  clrscr();
              cout<<"Would you like to:\n"<<"1:Enter new posts\n";
              cout<<"2:Enter new candidate (Warning: Need to enter post of candidate beforehand, Max 20 per post)\n";
              cout<<"3:List all posts\n"<<"4:List all candidates for a particular post\n";
              cout<<"5:Show the winner of every post\n"<<"6:Delete a post\n"<<"7:Delete a candidate\n";
              cout<<"8:Return to welcome screen\n";
              cout<<"9:Save and exit\n";
              cin>>ch2;
              switch(ch2)
              {  case 1: clrscr();
                         newpost();
                         break;
                 case 2: clrscr();
                         newcand();
                         break;
                 case 3: clrscr();
                         pdisp();
                         break;
                 case 4: clrscr();
                         ch3='y';
```

```
                    while(ch3=='y'||ch3=='Y')
                    {  clrscr();
                       cdisp();
                       cout<<"Would you like to check candidate list for a different
post? (y/n)\n";
                       cin>>ch3;
                    }
                    cout<<endl;
                    break;
               case 5: clrscr();
                    winner();
                    break;
               case 6: clrscr();
                    pdel();
                    break;
               case 7: clrscr();
                    cdel();
                    break;
               case 8: clrscr();
                    goto set;
                    break;
               case 9: goto lab;
                    break;
```

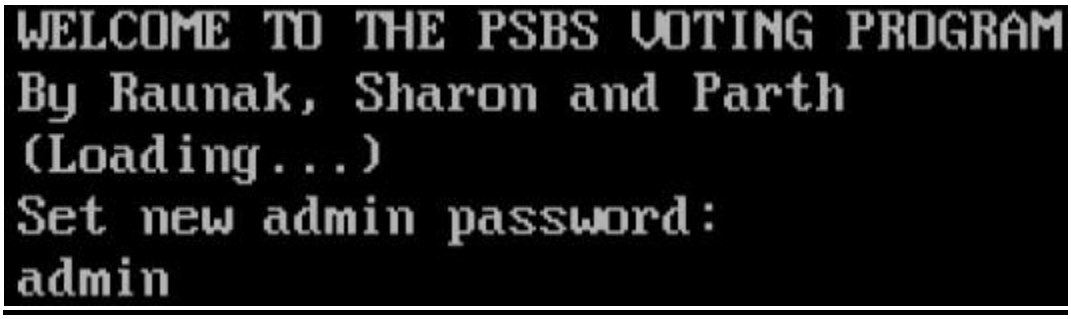
```
                default: cout<<"Invalid choice\n\n";
                         break;
                }
                cout<<"Would you like to return to Administrator Main Menu? (y/n)\n";
                cin>>ch4;
              }
              cout<<endl;
            }
            break;
case 2: ch5='y';
            while(ch5=='y'||ch5=='Y')
            {  clrscr();
               cout<<"Would you like to:\n"<<"1:List all posts\n";
               cout<<"2:List all candidates for a post\n";
               cout<<"3:Vote for a candidate for each post\n";
               cout<<"4:Return to welcome screen\n"<<"5:Save and exit\n";
               cin>>ch6;
               switch(ch6)
               {  case 1: clrscr();
                          pdisp();
                          break;
                  case 2: clrscr();
                          ch8='y';
```

```
                    while(ch8=='y'||ch8=='Y')
                    {  clrscr();
                       cdisp();
                       cout<<"Would you like to check candidate list for a different post?
(y/n)\n";
                       cin>>ch8;
                    }
                    cout<<endl;
                    break;
               case 3: clrscr();
                    vote();
                    break;
               case 4: clrscr();
                    goto set;
                    break;
               case 5: goto lab;
                    break;
               default: cout<<"Invalid choice\n\n";
                     break;
             }
             cout<<"Would you like to return to Voter main menu? (y/n)\n";
             cin>>ch5;
          }
```

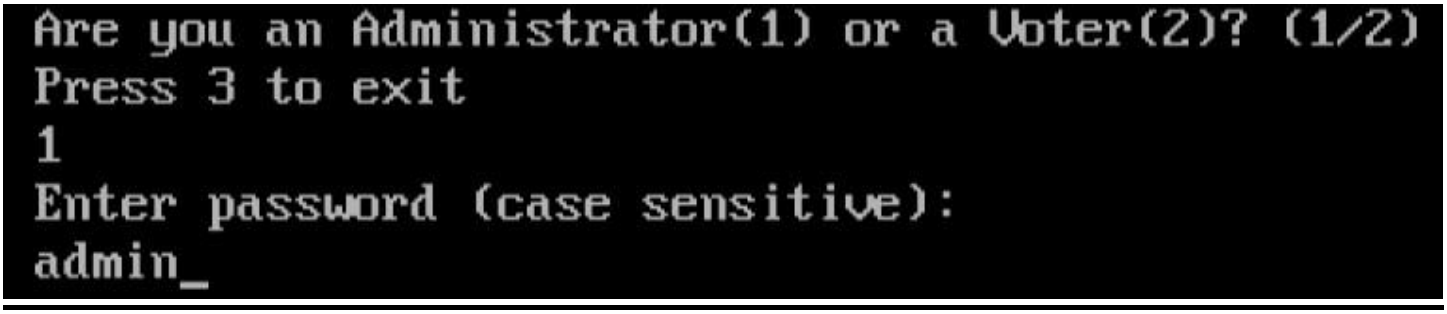
```
            cout<<endl;
            break;
    case 3: goto lab;
            break;
    default: cout<<"Invalid choice\n\n";
  }
  lab2:
  cout<<"Would you like to return to the main menu? (y/n)\n";
  cin>>ch7;
  cout<<endl;
}
  lab:
  clrscr();
  store();
  getch();
}
```

OUTPUT SCREENS

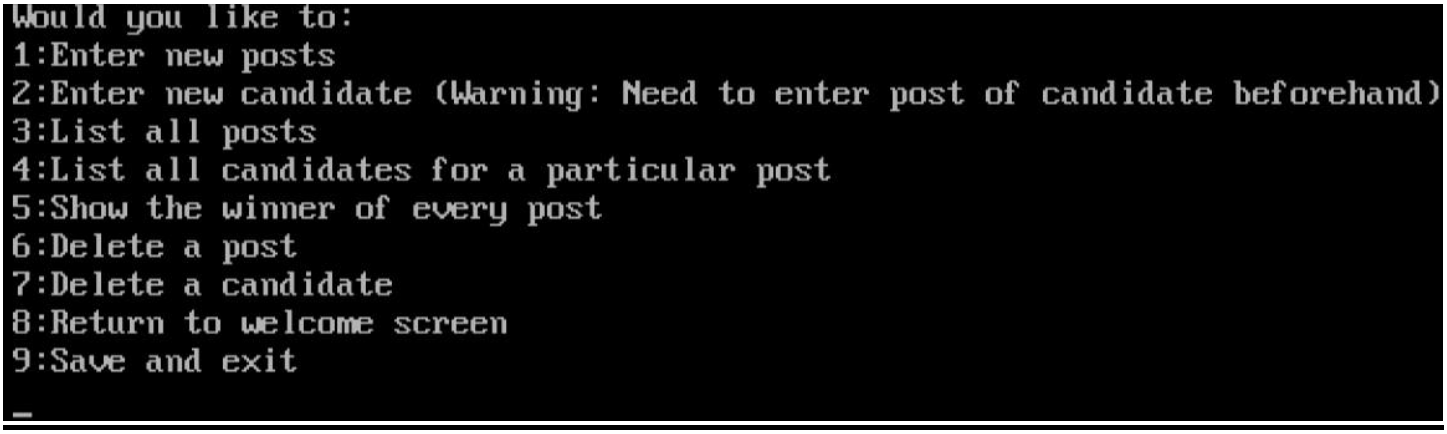
Welcome screen



Main menu



Administrator menu



New post

```
Enter name of post:
headboy
Post added

Would you like to return to Administrator Main Menu? (y/n)
```

Add candidate

```
Enter the name of the candidate:
student1
Enter the class of the candidate (eg:10c):
12b
Enter the post that the candidate is competing for:
headboy
Candidate added

Would you like to return to Administrator Main Menu? (y/n)
_
```

Post display

```
Post 1:sports captain
Post 2:headgirl
Post 3:headboy

Would you like to return to Administrator Main Menu? (y/n)
```

Candidate display for a particular post

```
Please enter the post of the candidate
headboy
Candidate 1:
Name:student1
Class:12b
Votes:0

Candidate 2:
Name:student2
Class:12a
Votes:0

Candidate 3:
Name:student3
Class:12c
Votes:0
```

Voter menu

```
Would you like to:
1:List all posts
2:List all candidates for a post
3:Vote for a candidate for each post
4:Return to welcome screen
5:Save and exit
```

Vote

```
Post:
headgirl
Candidate 1:
Name:stud3      Class:12a
Candidate 2:
Name:stud4      Class:12c
Enter your choice:
2
Vote casted

Post:
headboy
Candidate 1:
Name:stud1      Class:11b
Candidate 2:
Name:stud2      Class:11d
Enter your choice:
1
Vote casted

Would you like to return to Voter main menu? (y/n)
```

Winner declaration for all posts

```
Post:
headgirl
Winner is:stud3 From class:12a

Post:
headboy
Winner is:stud1 From class:11b

Would you like to return to Administrator Main Menu? (y/n)
```

Candidate deletion

```
Enter candidate to be removed
stud3
Candidate deleted

Would you like to return to Administrator Main Menu? (y/n)
_
```

Post deletion

```
Enter name of post to be deleted
headgirl
The post and all its candidates were deleted
```

FUTURE UPGRADES

Because the program has been created keeping in mind the concepts of good programming practices like comments, modularity and extensive use of functions, upgrades can be easily made without much hassle.

Some feasible upgrades are:

1. Graphics can be used in the intro function without affecting the functioning in any way. It will make the program more attractive and unique.

2. Since there is extensive use of classes, we can add new details of candidates etc. easily and the input and output can be facilitated by adding merely a few lines of code in the respective functions.

3. Advanced security features can be added to prevent tampering of votes.

BIBLIOGRAPHY

1. Computer science-C++(Part 1) by Sumita Arora

2. www.google.co.in

3. en.wikipedia.org

4. www.cpp-project.blogspot.in