Muhammad Fahad Shabbir

BS\_CS\_2B

11-02-2024

OOP LAB assignment # 1

Voting Management System

My voting management system consists of the following entities which include: -

1. National Assembly Candidate Management System
2. Provincial Assembly Candidate Management
3. Voter Details Management

The following entities give user option to fully handle easy and efficient voting through the system. It has a proper menu driven interface provided at all the stages of the program. The NA Candidate Management system as well as PA Candidate Management allows to enter and view (using functions) the data (name, constituency, party, and votes of the candidate). Specifically, have restricted the user to enter only 10 candidates in each assembly standing for elections through diverse constituencies. The third entity which is voter details allows the user to enter, display or search (using functions) the data of voter such as (name, house number, CNIC, PA constituency, NA constituency, serial number of the voter).

* MODULAR PROGRAMMING

1. Main.cpp file
2. NA\_Candidate1.h
3. PA\_Candidate1.h
4. Voters\_Details1.h

implementations include structures, arrays, file handling, loops, functions are used exceptionally.

Code Snippets

Main.cpp

#include <iostream>

#include <fstream>

#include <iomanip>

#include "NA\_candidate1.h"

#include "PA\_candidate1.h"

#include"Voters\_Details1.h"

using namespace std;

void addNA\_Candidate();

void viewNA\_Candidate();

void addPA\_Candidate();

void viewPA\_Candidate();

void addVoter\_List();

void viewVoter\_List();

void searchVoter\_List();

void menu();

int main() {

while (true) {

menu();

}

return 0;

}

void menu() {

cout << "~~~~~~Voting-Management-System~~~~~" << endl;

cout << endl;

int menu;

int choice;

cout << "~~Choose one out of following options~~" << endl;

cout << "1:Management of NA Candidate\n2:Management of PA Candidate\n3:Check Voter details\n0:Exit" << endl;

cin >> menu;

switch (menu) {

case 1:

cout << "Choose one out of following options:" << endl;

cout << "1:Add-NA-Candidate\n2:View-NA-Candidate\n0:Exit" << endl;

cin >> choice;

switch (choice) {

case 1:

addNA\_Candidate();

break;

case 2:

viewNA\_Candidate();

break;

case 0:

exit(0);

}

break;

case 2:

cout << "Choose one out of following options:" << endl;

cout << "1:Add-PA-Candidate\n2:View-PA-Candidate\n0:Exit" << endl;

cin >> choice;

switch (choice) {

case 1:

addPA\_Candidate();

break;

case 2:

viewPA\_Candidate();

break;

case 0:

exit(0);

}

break;

case 3:

cout << "Choose one out of following options:" << endl;

cout << "1:Add-Voter-Info\n2:View-Voter-Info\n3:Search-Voter-Info\n0:Exit" << endl;

cin >> choice;

switch (choice) {

case 1:

addVoter\_List();

break;

case 2:

viewVoter\_List();

break;

case 3:

searchVoter\_List();

break;

case 0:

exit(0);

}

break;

case 0:

exit(0);

}

}

* HEADER FILE: NA\_Candidate1.h

#pragma once

#include <iostream>

#include <fstream>

#include <iomanip>

using namespace std;

const int MAX\_NA\_CANDIDATES = 10;

struct PollingDataNA {

string name;

string constituency;

string party;

int votes;

};

bool addNA\_Candidate(PollingDataNA candidates[], int& count) {

if (count >= MAX\_NA\_CANDIDATES) {

cout << "Maximum candidates reached." << endl;

return false;

}

ofstream fout("NA.txt", ios::app);

if (!fout.is\_open()) {

cout << "Error opening file." << endl;

return false;

}

PollingDataNA newCandidate;

cout << "Enter the Name of the Candidate: ";

cin >> newCandidate.name;

for (int i = 0; i < count; ++i) {

if (candidates[i].name == newCandidate.name) {

cout << "The candidate is already in the list! Please enter another." << endl;

return false;

}

}

cout << "Enter the Constituency of the Candidate: ";

cin >> newCandidate.constituency;

cout << "Enter the Party of the Candidate: ";

cin >> newCandidate.party;

cout << "Enter the Votes of the Candidate: ";

cin >> newCandidate.votes;

fout << newCandidate.name << " " << newCandidate.constituency << " " << newCandidate.party << " " << newCandidate.votes << endl;

candidates[count++] = newCandidate;

fout.close();

return true;

}

void viewNA\_Candidate(const PollingDataNA candidates[], int count) {

ifstream fin("NA.txt");

if (!fin.is\_open()) {

cout << "File is not available" << endl;

return;

}

cout << left << setw(25) << "Name" << setw(15) << "Constituency" << setw(20) << "Party" << setw(10) << "Votes" << endl;

for (int i = 0; i < count; ++i) {

cout << left << setw(25) << candidates[i].name << setw(15) << candidates[i].constituency << setw(20) << candidates[i].party << setw(10) << candidates[i].votes << endl;

}

fin.close();

}

* HEADER FILE: PA\_Candidate1.h

#pragma once

#include <iostream>

#include <fstream>

#include <iomanip>

#include <string>

using namespace std;

const int MAX\_PA\_CANDIDATES = 10;

struct PollingDataPA {

string name;

string constituency;

string party;

int votes;

};

bool addPA\_Candidate(PollingDataPA candidates[], int& count) {

if (count >= MAX\_PA\_CANDIDATES) {

cout << "Maximum candidates reached." << endl;

return false;

}

ofstream fout("PA.txt", ios::app);

if (!fout.is\_open()) {

cout << "Error opening file." << endl;

return false;

}

PollingDataPA newCandidate;

cout << "Enter the Name of the Candidate: ";

cin >> newCandidate.name;

for (int i = 0; i < count; ++i) {

if (candidates[i].name == newCandidate.name) {

cout << "The candidate is already in the list! Please enter another." << endl;

return false;

}

}

cout << "Enter the Constituency of the Candidate: ";

cin >> newCandidate.constituency;

cout << "Enter the Party of the Candidate: ";

cin >> newCandidate.party;

cout << "Enter the Votes of the Candidate: ";

cin >> newCandidate.votes;

fout << newCandidate.name << " " << newCandidate.constituency << " " << newCandidate.party << " " << newCandidate.votes << endl;

candidates[count++] = newCandidate;

fout.close();

return true;

}

void viewPA\_Candidate(const PollingDataPA candidates[], int count) {

ifstream fin("PA.txt");

if (!fin.is\_open()) {

cout << "File is not available" << endl;

return;

}

cout << left << setw(25) << "Name" << setw(15) << "Constituency" << setw(20) << "Party" << setw(10) << "Votes" << endl;

for (int i = 0; i < count; ++i) {

cout << left << setw(25) << candidates[i].name << setw(15) << candidates[i].constituency << setw(20) << candidates[i].party << setw(10) << candidates[i].votes << endl;

}

fin.close();

}

=>HEADER FILE: Voters\_Details1.h

#pragma once

#include<iostream>

#include<fstream>

#include<iomanip>

using namespace std;

const int MAX\_VOTERS = 10000;

struct voter {

string name;

int House\_No;

string CNIC;

string PA\_Constituency;

string NA\_Constituency;

int Serial\_num;

} info[MAX\_VOTERS];

void addVoter\_List() {

ofstream fout("Voter.txt", ios::app);

if (!fout.is\_open()) {

cout << "Error opening file." << endl;

return;

}

char choice;

for (int i = 0; i < MAX\_VOTERS; i++) {

cout << "Enter the CNIC of the Voter" << endl;

cin >> info[i].CNIC;

cout << "Enter the Name of the Voter" << endl;

cin >> info[i].name;

cout << "Enter the House Number of the Voter" << endl;

cin >> info[i].House\_No;

cout << "Enter the Serial Number of the Voter" << endl;

cin >> info[i].Serial\_num;

cout << "Enter the PA\_Constituency of the Voter" << endl;

cin >> info[i].PA\_Constituency;

cout << "Enter the NA\_Constituency of the Voter" << endl;

cin >> info[i].NA\_Constituency;

fout << info[i].CNIC << " " << info[i].name << " " << info[i].House\_No << " " << info[i].Serial\_num << " " << info[i].PA\_Constituency << " " << info[i].NA\_Constituency << endl;

cout << "Enter 'Y' if you want to add the data of another Voter and press any other button to exit" << endl;

cin >> choice;

if (choice != 'y' && choice != 'Y') {

break;

}

}

fout.close();

}

void viewVoter\_List() {

ifstream fin("Voter.txt");

if (!fin.is\_open()) {

cout << "File is not available" << endl;

return;

}

cout << left << setw(15) << "CNIC" << setw(15) << "Name" << setw(15) << "House\_Number" << setw(15) << "Serial\_Number" << setw(20) << "PA\_Constituency" << setw(20) << "NA\_Constituency" << endl;

int count = 0;

while (fin >> info[count].CNIC >> info[count].name >> info[count].House\_No >> info[count].Serial\_num >> info[count].PA\_Constituency >> info[count].NA\_Constituency) {

cout << left << setw(15) << info[count].CNIC << setw(15) << info[count].name << setw(15) << info[count].House\_No << setw(15) << info[count].Serial\_num << setw(20) << info[count].PA\_Constituency << setw(20) << info[count].NA\_Constituency << endl;

count++;

if (count >= MAX\_VOTERS) {

break;

}

}

fin.close();

}

void searchVoter\_List() {

ifstream fin("Voter.txt");

if (!fin.is\_open()) {

cout << "File is not Available" << endl;

return;

}

string Search\_CNIC;

cout << "Enter the CNIC of the person for searching Voter Details:" << endl;

cin >> Search\_CNIC;

cout << left << setw(15) << "CNIC" << setw(15) << "Name" << setw(15) << "House\_Number" << setw(15) << "Serial\_Number" << setw(20) << "PA\_Constituency" << setw(20) << "NA\_Constituency" << endl;

voter temp;

while (fin >> temp.CNIC >> temp.name >> temp.House\_No >> temp.Serial\_num >> temp.PA\_Constituency >> temp.NA\_Constituency) {

if (Search\_CNIC == temp.CNIC) {

cout << left << setw(15) << temp.CNIC << setw(15) << temp.name << setw(15) << temp.House\_No << setw(15) << temp.Serial\_num << setw(20) << temp.PA\_Constituency << setw(20) << temp.NA\_Constituency << endl;

break;

}

}

fin.close();

}

**THE END!**

**MUHAMMAD FAHAD SHABBIR**

**233017**

**BS-CS-2B**