

Electronic Voting System

M. Ahabb Sheraz, Reg. no. 2021327, CS103 B Semester Project

Problem Statement:

A valid voter will be casting their ballot for the candidates in provincial and federal general elections. After ending the election, the results of the election will be announced.

Introduction:

Firstly, Users will register as a voter using their IDs (CNICs). This will be done using voter_registration.cpp program. Voters will be able to vote for their parties in the election.cpp program but only if their IDs are registered.

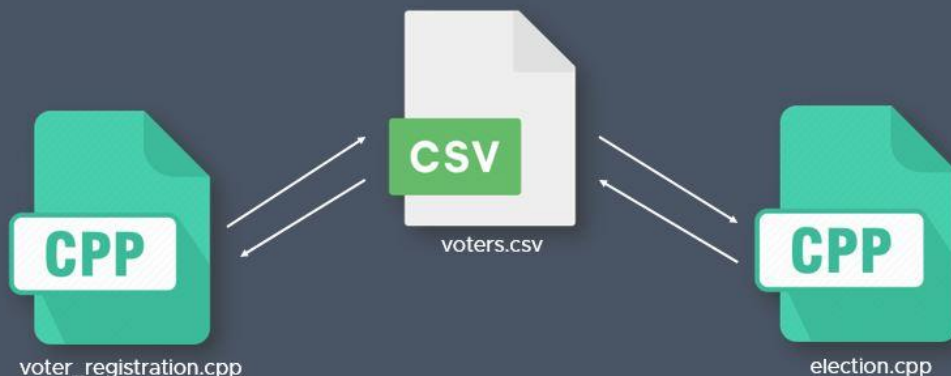
Algorithm Design and Code:

This section has been divided into two main parts:

- Algorithm design and code of voter_registration.cpp
- Algorithm design and code of election.cpp

How are 2 programs linked?

Users will register their IDs in the voter_registration.cpp program. Then these IDs get stored into voters.csv file. The elections.cpp program will then fetch these IDs from the voters.csv file.



voter_registration.cpp

This code consists of the following functions:

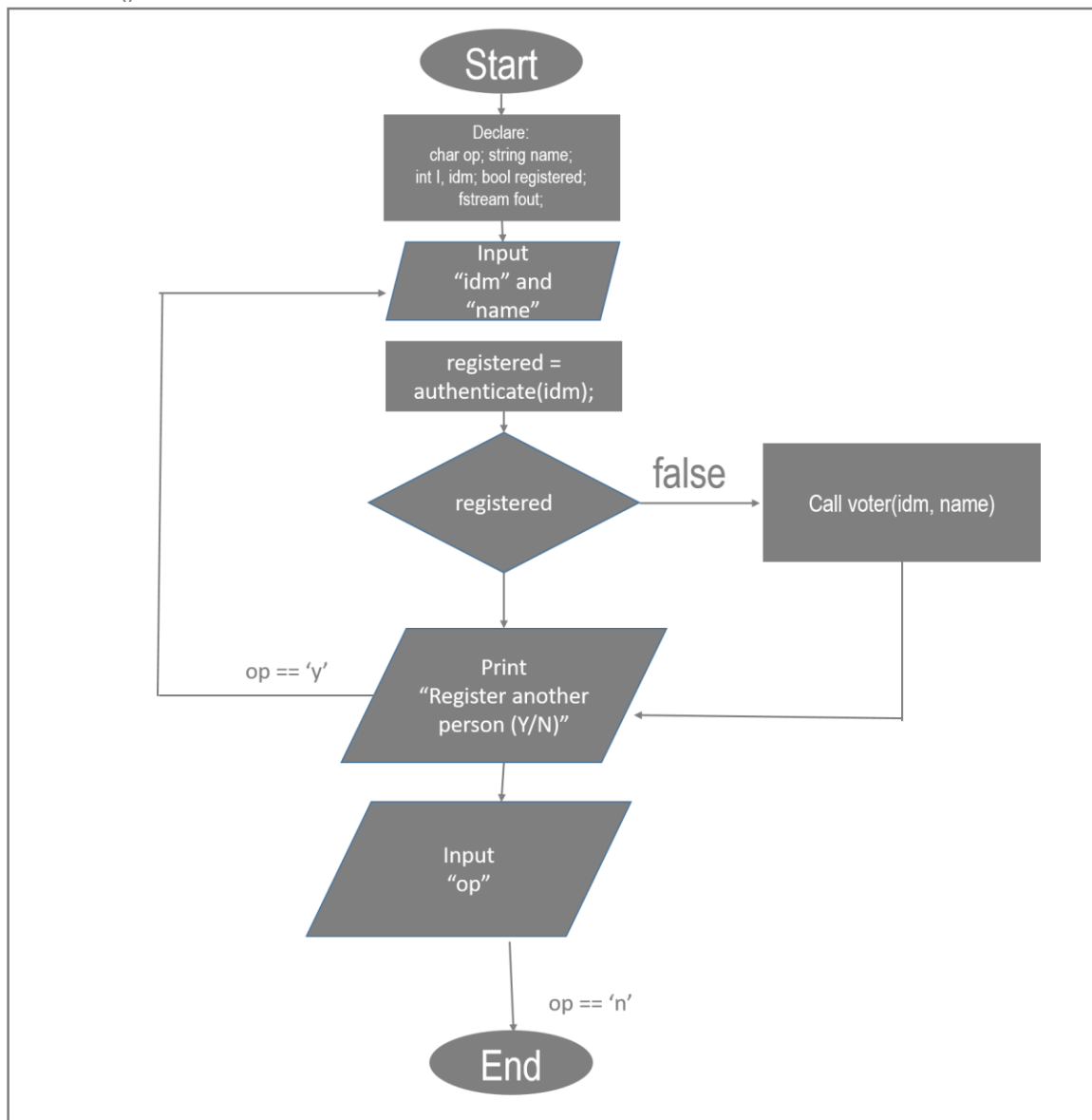
- `authenticate()`:

This function verifies if the typed ID has already been registered or not. It returns the Boolean value of 0 if ID has not been registered before.

- `voter()`:

Unregisters IDs get stored into the voters.csv using this function. No return value.

`int main()`



Flowchart of voter_registration

■ authenticate()

```
1 #include <iostream>
2 #include <fstream>
3 #include <vector>
4 #include <string>
5 #include <bits/stdc++.h>
6
7 using namespace std;
8
9 bool authenticate(int& x)
10 {
11     bool registered = false;
12
13     // File pointer
14     fstream fin;
15
16     // Open an existing file
17     fin.open("voters.csv", ios::in);
18
19     // Get the roll number of which the data is required
20     // Read the Data from the file as String Vector
21     vector<string> row;
22     string line, word;
23     int id, count = 0;
24
25     while (!fin.eof()) {
26         row.clear();
27
28         // read an entire row and store it in a string variable 'line'
29         getline(fin, line);
30
31     }
```

```
32 // used for breaking words
33 stringstream s(line);
34
35 // read every column data of a row and store it in a string variable, 'word'
36 while (getline(s, word, ',')) {
37     // add all the column data
38     // of a row to a vector
39     row.push_back(word);
40 }
41
42 // convert string to integer for comparison
43 while (!fin.eof()) {
44     id = stoi(row[0]);
45     // Compare the roll number
46     if (id == x) {
47         count += 1;
48         registered = true;
49         cout << "Error: This CNIC is already registered" << endl;
50         cout << "Please enter your CNIC again:\n";
51         cin >> x;
52     }
53     if (id != x) {
54         registered = false;
55         break;
56     }
57 }
58
59 }
60
61 fin.close();
62 return registered;
63 }
```

■ voter()

```
65 void voter(int idm, string name)
66 {
67     fstream fout;
68     fout.open("voters.csv", ios::out|ios::app);
69     fout << idm << "," << name << endl;
70     fout.close();
71 }
```

■ main()

```

77 int main()
78 {
79     // file pointer
80     fstream fout;
81     char op;
82
83     string name;
84     int i = 0, id;
85     bool registered = false;
86
87     // exception handling to check if voters.csv file exists or not
88     try{
89         if(!fout){
90             throw "404! File voters.csv not found";
91         }
92     }
93     catch(const char* er){
94         cout << er << endl;
95     }
96
97     while(true){
98         cout << "\t ***** \t\n"
99         << "\t  Voter Registration System \t\n"
100         << "\t ***** \t\n";
101         cout << "Please enter your details"
102         << " (CNIC Name):" << endl;
103         cin >> id >> name;
104
105         registered = authenticate(id); //returns boolean value on registration status
106
107         if (registered == false){
108             // Insert the data to file and add newly registered id to voters list
109             voter(id, name);
110         }
111         cout << "Register another person(y/n)\n";
112         cin >> op;
113         if (op == 'n' || op == 'N'){
114             break;
115         }
116     }
117 }
118
119
120

```

Program in action

```

C:\Users\Lenovo\Documents\e-voting\ voter_registration.exe
*****
Voter Registration System
*****
Please enter your details (CNIC Name):
1234 ali
Register another person(y/n)
n
-----
Process exited after 12.87 seconds with return value 0
Press any key to continue . . .

```

```

voters - Notepad
File Edit Format View Help
1000,srd
1234,ali
2345,ahabb

```

```

C:\Users\Lenovo\Documents\e-voting\ voter_registration.exe
*****
Voter Registration System
*****
Please enter your details (CNIC Name):
1234 ahabb
Error: This CNIC is already registered
Please enter your CNIC again:
2345
Register another person(y/n)
n
-----
Process exited after 17.73 seconds with return value 0
Press any key to continue . . .

```

election.cpp

This code consists of the following functions:

- voter():

This function verifies if the typed ID has already been registered or not. In this function, I defined a fstream class named fin which has been linked to voters.csv file to compare the content in the file to the function parameters, given in the form of an address, given by the user.

- voter()

```
1  #include <iostream>
2  #include <fstream>
3  #include <vector>
4  #include <string>
5  #include <bits/stdc++.h>
6
7  using namespace std;
8
9  bool authenticate(int& x)
10 {
11     bool registered = false;
12
13     // File pointer
14     fstream fin;
15
16     // Open an existing file
17     fin.open("voters.csv", ios::in);
18
19     // Get the roll number of which the data is required
20     // Read the Data from the file as String Vector
21     vector<string> row;
22     string line, word;
23     int id, count = 0;
24
25     while (!fin.eof()) {
26         row.clear();
27
28         // read an entire row and store it in a string variable 'line'
29         getline(fin, line);
30
31         // used for breaking words
32         stringstream s(line);
33
34         // read every column data of a row and store it in a string variable, 'word'
35         while (getline(s, word, ',')) {
36             // add all the column data
37             // of a row to a vector
38             row.push_back(word);
39         }
40
41         // convert string to integer for comparison
42         while (!fin.eof()) {
43             id = stoi(row[0]);
44             // Compare the roll number
45             if (id == x) {
46                 count++;
47                 registered = true;
48                 cout << "Error: This CNIC is already registered" << endl;
49                 cout << "Please enter your CNIC again:\n";
50                 cin >> x;
51             }
52             if (id != x) {
53                 registered = false;
54                 break;
55             }
56         }
57     }
58
59     fin.close();
60     return registered;
61 }
```

```
32 // used for breaking words
33 stringstream s(line);
34
35 // read every column data of a row and store it in a string variable, 'word'
36 while (getline(s, word, ',')) {
37     // add all the column data
38     // of a row to a vector
39     row.push_back(word);
40 }
41
42 // convert string to integer for comparison
43 while (!fin.eof()) {
44     id = stoi(row[0]);
45     // Compare the roll number
46     if (id == x) {
47         count++;
48         registered = true;
49         cout << "Error: This CNIC is already registered" << endl;
50         cout << "Please enter your CNIC again:\n";
51         cin >> x;
52     }
53     if (id != x) {
54         registered = false;
55         break;
56     }
57 }
58
59 fin.close();
60 return registered;
61 }
```

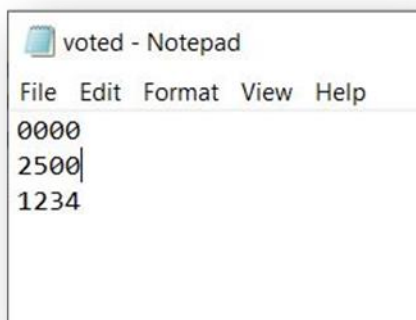
- `vote_status()`:

This function will not allow users who have already casted their vote to vote again as records of those who have already voted are kept in another file named `voted.csv` file. It also returns a Boolean value.

- `vote_status()`

```
66 bool vote_status(int& x)
67 {
68     bool voted = false;
69     ifstream fin;
70     int count = 0, id;
71     vector<string> row;
72     string line, word;
73     fin.open("voted.csv", ios::in);
74     while (!fin.eof()) {
75         row.clear();
76
77         // read an entire row and
78         // store it in a string variable 'line'
79         getline(fin, line);
80
81         // used for breaking words
82         stringstream s(line);
83
84         // read every column data of a row and
85         // store it in a string variable, 'word'
86         while (getline(s, word, ',')) {
87             // add all the column data
88             // of a row to a vector
89             row.push_back(word);
90         }
91
92         // convert string to integer for comparison
93         //while (!fin.eof()) {
94             id = stoi(row[0]);
95             // Compare the roll number
```

```
94         id = stoi(row[0]);
95         // Compare the roll number
96         if (id == x) {
97             voted = true;
98             count = 1;
99             //count = 1;
100             cout << "You have already casted a vote" << endl;
101             break;
102         }
103         if (count == 0) {
104             voted = false;
105         }
106     }
107     fin.close();
108     return voted;
109 }
110
111
```



IDs of those who have already voted are kept in `voted.csv`.

- record_vote_n():

This function updates voting results for National Assembly candidates by updating file na_results.csv. At first, it adds existing data of na_results.csv with new changes to the newly created file na_resultsnew.csv and then deletes the original na_results.csv and renames na_resultsnew.csv to it.

- record_vote_p():

This function updates voting results for Provincial Assembly candidates. Working is very similar to function record_vote_n().

- record_vote_n()
similar for record_vote_p()

```

112 void record_vote_n(string selectparty, int tvotes)
113 {
114     // File pointer
115     fstream fin, fout;
116
117     // Open an existing record
118     fin.open("results.csv", ios::in);
119
120     // Create a new file to store updated data
121     fout.open("resultsnew.csv", ios::out);
122
123     int rollnum, rolli, marks, found = 0, i, votes=0;
124     char sub;
125     int index, new_marks;
126     string line, word, party;
127     vector<string> row;
128
129     // Traverse the file
130     while (!fin.eof()) {
131         row.clear();
132         getline(fin, line);
133         stringstream s(line);
134
135         while (getline(s, word, ',')) {
136             row.push_back(word);
137         }
138
139         party = row[0];
140         int row_size = row.size();
141
142
143

```

```

144         if (party == selectparty) {
145             found = 1;
146             stringstream convert;
147
148             // sending a number as a stream into output string
149             votes += tvotes + stoi(row[1]);
150             convert << votes;
151
152             // the str() converts number into string
153             row[1] = convert.str();
154
155             if (!fin.eof()) {
156                 for (i = 0; i < row_size - 1; i++) {
157                     // write the updated data
158                     // into a new file 'resultsnew.csv'
159                     // using fout
160                     fout << row[i] << ",";
161                 }
162                 fout << row[row_size - 1] << endl;
163             }
164             else {
165                 if (!fin.eof()) {
166                     for (i = 0; i < row_size - 1; i++) {
167                         // writing other existing records
168                         // into the new file using fout.
169                         fout << row[i] << ",";
170                     }
171                     // the last column data ends with a '\n'
172                     fout << row[row_size - 1] << endl;
173                 }
174             }
175             if (fin.eof())
176                 break;
177             if (found == 0)
178                 cout << "error" << endl;
179             fin.close();
180             fout.close();
181
182
183
184
185
186
187
188
189
190
191
192
193
194

```

```

185     fin.close();
186     fout.close();
187
188     // removing the existing file
189     remove("results.csv");
190     // renaming the updated file with the existing file name
191     rename("resultsnew.csv", "results.csv");
192
193
194

```

- na_displayresults():

Function displays National Assembly election results.

- pa_displayresults():

Function displays Provincial Assembly election results.

- na_displayresults()
similar for pa_displayresults()

```
278 void na_displayresults()
279 {
280     // File pointer
281     fstream fin;
282
283     // Open an existing file
284     fin.open("na_results.csv", ios::in);
285
286     int count = 0;
287
288     // Read the data from the file
289     // as string vector
290     vector<string> row;
291     string line, word;
292
293     cout << "NA Election results:\n";
294
295     while (!fin.eof()) {
296         row.clear();
297
298         // read an entire row and
299         // store it in a string variable 'line'
300         getline(fin, line);
301
302         // used for breaking words
303         stringstream s(line);
304
305         // read every column data of a row and
306         // store it in a string variable, 'word'
307         while (getline(s, word, ',')) {
308             // add all the column data
309             // of a row to a vector
310             row.push_back(word);
311         }
312         cout << row[0] << " : ";
313         cout << row[1] << endl;
314     }
315     fin.close();
316 }
317
```


- main():

It is in main function where the user will vote for their party and get verified for voting eligibility.

```

375 int main()
376 {
377     int id;
378     bool voted, registered;
379     char op, partyselect, admin;
380     string party, adminpw;
381     int partyvote[4] = {0};
382     int pvote[4] = {0};
383
384     // file pointer
385     fstream foutn;
386     fstream foutp;
387
388     // opens an existing csv file or creates a new file.
389     foutn.open("voted.csv", ios::out|ios::app);
390
391     cout << "\t ***** \t\n"
392     << "\t E-Voting System \t\n"
393     << "\t ***** \t\n";
394
395     while(true){
396         cout << "Enter your CNIC: ";
397         cin >> id;
398         cout << endl;
399
400         registered = voter(id);
401         voted = vote_status(id);
402
403         if(voted == false && registered == true){
404             cout << "\t ***** \t\n"
405             << "\t Please cast your vote for NA \t\n"
406             << "\t ***** \t\n";
407             cout << "\na.PTI\n" << "b.PML(N)\n" << "c.PPP\n" << "d.MMA\n" << "e.TLP\n" << "f.Independent" << endl;
408             cin >> partyselect;
409             if(partyselect == 'a'){
410                 party = "PTI";
411                 partyvote[0]++;
412                 record_vote_n(party, partyvote[0]);
413             }
414             else if (partyselect == 'b')
415             {
416                 party = "PMLN";
417                 partyvote[1]++;
418                 record_vote_n(party, partyvote[1]);
419             }
420             else if (partyselect == 'c')
421             {
422                 party = "PPP";
423                 partyvote[2]++;
424                 record_vote_n(party, partyvote[2]);
425             }
426             else if (partyselect == 'd'){
427                 party = "MMA";
428                 partyvote[3]++;
429                 record_vote_n(party, partyvote[3]);
430             }
431             else if (partyselect == 'e')
432             {
433                 party = "TLP";
434                 partyvote[4]++;
435                 record_vote_n(party, partyvote[4]);
436             }
437             else if (partyselect == 'f'){
438                 party = "Indie";
439                 partyvote[5]++;
440                 record_vote_n(party, partyvote[5]);
441             }
442             else{
443                 cout << "invalid input\n";
444             }
445             foutn << id << endl;
446             cout << "Voted successfully.\n";
447
448             cout << "\t ***** \t\n"
449             << "\t Please cast your vote for PA \t\n"
450             << "\t ***** \t\n";
451             cout << "\na.PTI\n" << "b.PML(N)\n" << "c.PPP\n" << "d.MMA\n" << "e.TLP\n" << "f.Independent" << endl;
452             cin >> partyselect;
453             if(partyselect == 'a'){
454                 party = "PTI";
455                 pvote[0]++;
456                 record_vote_p(party, pvote[0]);
457             }

```

```

459 {
460     party = "PMLN";
461     pvote[1]++;
462     record_vote_p(party, pvote[1]);
463 }
464
465 else if (partyselect == 'c')
466 {
467     party = "PPP";
468     pvote[2]++;
469     record_vote_p(party, pvote[2]);
470 }
471
472 else if (partyselect == 'd'){
473     party = "MMA";
474     pvote[3]++;
475     record_vote_p(party, pvote[3]);
476 }
477
478 else if (partyselect == 'e')
479 {
480     party = "TLP";
481     pvote[4]++;
482     record_vote_p(party, pvote[4]);
483 }
484
485 else if (partyselect == 'f'){
486     party = "Indie";
487     pvote[5]++;
488     record_vote_p(party, pvote[5]);
489 }
490
491 else{
492     cout << "invalid input\n";
493 }
494
495 foutp << id << endl;
496 cout << "Voted successfully.\n";
497 }
498
499 cout << "\nProceed to other voter.(Y/N)\n";
500 cin >> op;
501 if(op == 'n' || op == 'N'){
502     cout << "Do you want to end the election and display election results.(Y/N)\n";
503     cin >> admin;
504     if(admin == 'Y' || admin == 'y'){
505         cout << "Please enter the admin. password to gain administrative access.\n";
506         cin >> adminpw;
507         if(adminpw == "1234e")
508         {
509             na_displayresults();
510             pa_displayresults();
511         }
512         else{
513             cout << "Wrong Password\n";
514         }
515     }
516     break;
517 }
518
519 }
520
521 foutn.close();
522 foutp.close();
523
524 return 0;
525 }

```

Conclusion

This project was definitely interesting but also challenging. It tested my knowledge of CS103 to its full extend. By making File-based databases using File handling, I was able to complete this task. It was fun creating this project.