S.No	Platforms	Page.No
01	Introduction	01
02	Online voting system	01
03	Module	01
04	Module	02
05	Conclusion	02

Index

INTRODUCTION:

The project is aimed at developing an online voting system using the C language. The online voting system will allow users to register, log in, choose positions for voting, give their vote, and update their profile. This pre-submission report provides an overview of each module of the project and the online voting system as a whole.

ONLINE VOTING SYSTEM

An online voting system is a platform that allows voters to cast their vote electronically over the Internet_ The system can be used for both public and private elections, and it can provide several advantages such as increased accessibility, security, and efficiency. However, it also comes with some potential drawbacks like possible security breaches and technical issues.

Module 1: Registration

The registration module will allow new users to create an account on the online voting system. The users will be required to provide some basic personal information, such as their name, address, and email address, and create a username and password. The registration module should also verify the user's email address to ensure the validity of the registration.

Module 2: Login/Logout

The login module will enable registered users to access the online voting system by providing their usernames and pages,' ord. The login module will also ensure that the user's credentials are authenticated before they are granted access to the system. The logout module will allow the user to log out of the system after they have completed their tasks.

Module 3: Update Profile

The update profile module will allow users to modify their personal information and update their profile. The module should enable the user to change their email address, address, or password. This module is necessary to ensure that the user's information is up-to-date and accurate.

Module 4: Choose Positions for Voting

The choose positions for the voting module will enable the user to select the positions they want to vote for. The positions can be grouped according to different categories such as political parties, school offices, or sports teams. The module should provide the user with an overview Of the available positions, including the candidates running for each position.

Module 5: Give a Vote

The give vote module will enable the user to cast their vote in the chosen position. The module should ensure that the User can vote only once for each position and that the vote is recorded accurately, The module should also confirm to the user that their vote has been cast successfully.

Conclusion: -

In conclusion, the online voting system project consists of five modules: registration, login/logout, updating profile, choosing positions for voting, and giving votes. Each module plays an essential role in the overall functionality Of the system. The project aims to develop a secure and efficient online voting system that can be used for public or private elections.

PROGRAM:

voting.c

```c
#include<stdio.h>
#include<string.h>
#include "color.h"

#define MAX_VOTERS 100
#define MAX_CANDIDATES 5

struct Date{
    unsigned int day;
    unsigned int month;
    unsigned int year;
};

typedef struct voter{ //Voters who will vote
    char uname[50];
    struct Date dob;
    int id;
    char password[50];
    int area;
    int has_voted; //voted or not?
}voting;

typedef struct candidate { // Candidates who need votes
    char fname[25];
    char lname[25];
    char vote_sign[1];
    int vote_count;
}candid;
```

Compiler Resources Compile Log Debug Find Results

Line: 1 Col: 1 Sel: 0 Lines: 443 Length: 12108 Insert Done parsing in 0.093 seconds

voting.c

```c
void login(int *Verify);
void register_voter(voting voter_arr,int num_voters,int log);
void update_profile(voting *voter_arr,int num_voters,int log,int index);
void display_voting_date();
void insert_candid(candid cand_arr,int num_candid);
void display_results(candid *cand_arr, int num_candidates);
void vote(candid *cand_arr, int num_candidates,int log,int index,voting *voter_arr);
void login_logout(voting voter_arr,int num_voters,int log,int* index);
void display_candidates(voting *voter_arr,int log,int index);

int main() {
    int choice = 0,num_voters,num_candid; //Variable to perform Operation
    int Verify=0; // variable to check if the password entered is correct or not
    login(&Verify); // Login/Authentication
    voting voter_arr[MAX_VOTERS]; //Creating array to store details of 100 Voters
    candid cand_arr[MAX_CANDIDATES]; //Creating array to store details of 5 Candidates
    num_voters=0; //Number of voters in the database
    num_candid=0; //Number of candidates standing for election in the database
    int log=0;//Variable to check if the user is logged in or not
    int index=-1; //index where the details of the voters is stored in the database
    if (Verify==1) {
    do {
    cyan();
    printf("\n\n");
    printf("-----------------------------------\n");
    printf("\tONLINE VOTING SYSTEM\n");
    printf("-----------------------------------\n");
    printf("\n1. Insert Candidate\n");
    printf("2. Register Voter\n");
```

Compiler Resources Compile Log Debug Find Results

Line: 1 Col: 1 Sel: 0 Lines: 443 Length: 12108 Insert Done parsing in 0.093 seconds

voting.c

```c
            case 7:
                vote(cand_arr,num_candid,log,index,voter_arr);
                break;
            case 8:
                display_results(cand_arr,num_candid);
                break;
            case 9:
                system("cls");
                green();
                printf("Exiting Program. Thank You!\n");
                reset();
                break;
            default:
                printf("Invalid choice.\n");
                break;
        }
    } while(choice != 9);
    }
        red();
        printf("You have successfully exited the program...\n\n");
        reset();
        return 0;
}

void login(int *Verify) {
    char password[15];
    blue();
    printf("\nEnter Password: ");
    scanf("%s",&password);
```

Compiler Resources Compile Log Debug Find Results

Line: 443 Col: 2 Sel: 0 Lines: 443 Length: 12108 Insert Done parsing in 0.093 seconds

voting.c

```c
    FILE *pass;
    pass=fopen("Password.txt","r");
    char myString[15];
    fgets(myString,15,pass);
    fclose(pass);
    if (strcmp(password, myString) == 0) {
        green();
        printf("Login Successful!\n\n");
        *Verify=1; //Value changed as pass is true
        reset();
    }
    else {
        red();
        printf("Wrong Password, Login Failed. Try Again!.\n");
        reset(); //else program is closed
    }
}

void register_voter(voting voter_arr,int num_voters,int log) {
    if (log==2){
        red();
        printf("You are already logged in!\n,Try Again...\n");
        reset();
        return;
    }
    if(*num_voters >= MAX_VOTERS) {
        red();
        printf("Sorry, Maximum number of voters reached.\n");
        reset();
```

Compiler Resources Compile Log Debug Find Results

Line: 443 Col: 2 Sel: 0 Lines: 443 Length: 12108 Insert Done parsing in 0.093 seconds

voting.c

```c
    }
    yellow();
    voting new_voter; // Creating new structure and assign the details in the existing structure in the end
    printf("Enter User name: ");
    scanf("%s",&new_voter.uname);
    printf("Enter Day of Date Of Birth (DD): ");
    scanf("%d",&new_voter.dob.day);
    printf("Enter Month of Date Of Birth (MM): ");
    scanf("%d",&new_voter.dob.month);
    printf("Enter Year of Date Of Birth (YYYY): ");
    scanf("%d",&new_voter.dob.year);
    reset();
    if (new_voter.dob.year>2005)
    {
        system("cls");
        red();
        printf("Sorry, you must be 18 years or above to be eligible for voting.\n");
        reset();
        return;
    }
    yellow();
    printf("Enter Area (1, 2, 3): ");
    scanf("%d",&new_voter.area);
    if (new_voter.area!=1&&new_voter.area!=2&&new_voter.area!=3)
    {
        system("cls");
        red();
        printf("Sorry,Invalid Address\nPlease Choose address b/w 1 and 3\n");
        reset();
```

Compiler Resources Compile Log Debug Find Results

Line: 443 Col: 2 Sel: 0 Lines: 443 Length: 12108 Insert Done parsing in 0.093 seconds

voting.c

```c
    }
    printf("Enter Your password: ");
    scanf("%s",&new_voter.password);
    new_voter.has_voted = 0; //Initially the user hasn't voted
    reset();

    // Generate unique voter ID
    new_voter.id=*num_voters+1;

    voter_arr[*num_voters] = new_voter;
    *num_voters=*num_voters+1; //Moving the pointer to the next memory address
    green();
    system("cls");
    printf("Registration Successful. Your voter ID is %d\n", new_voter.id);
    printf("Please remember your id and password to perform other operations in Future ...\n");
    reset();
}

void update_profile(voting *voter_arr,int num_voters,int log,int index) {
    if (log==0){
        red();
        printf("Log in to your ID First!\n");
        reset();
        return;
    }
        yellow();
        voting new_voter;
        printf("Enter New User name: ");
        scanf("%s",&new_voter.uname);
```

Compiler Resources Compile Log Debug Find Results

Line: 443 Col: 2 Sel: 0 Lines: 443 Length: 12108 Insert Done parsing in 0.093 seconds

voting.c

```
213         printf("Enter Year of Date Of Birth (YYYY): ");
214         scanf("%d",&new_voter.dob.year);
215         reset();
216         if (new_voter.dob.year>2005)
217         {
218         system("cls");
219         red();
220         printf("Sorry, you must be 18 years or above to be eligible for voting.\n");
221         reset();
222         return;
223         }
224         yellow();
225         printf("Enter New Area (1, 2, 3): ");
226         scanf("%d",&new_voter.area);
227         if (new_voter.area!=1&&new_voter.area!=2&&new_voter.area!=3)
228     {
229         system("cls");
230         red();
231         printf("Sorry,Invalid Address\nPlease Choose address b/w 1 and 3\n");
232         reset();
233         return;
234     }
235         reset();
236         voter_arr[index] = new_voter;
237         system("cls");
238         green();
239         printf("\n\nProfile updated successfully.\n");
240         reset();
241         }
```

Compiler Resources Compile Log Debug Find Results

Line: 443 Col: 2 Sel: 0 Lines: 443 Length: 12108 Insert Done parsing in 0.093 seconds

voting.c

```
243 void display_results(candid *cand_arr, int num_candidates) {
244     if (num_candidates==0){
245         red();
246         printf("Candidate List is Empty!\n");
247         reset();
248         return;
249     }
250 green();
251 printf("Results:-\n\n");
252 reset();
253 int i;
254 white();
255 for(i = 0; i < num_candidates; i++) {
256 printf("Candidate %d: %s %s [%s]-> %d\n",i+1, cand_arr[i].fname,cand_arr[i].lname,cand_arr[i].vote_sign, cand_arr[i].vote_count);
257 }
258 reset();
259 }
260
261 void display_voting_date() {
262     green();
263 printf("The Voting Date is 10th April 2023.\n");
264 reset();
265 }
266
267 void insert_candid(candid cand_arr,int num_candid){
268     if(*num_candid >= MAX_CANDIDATES) {
269         red();
270         printf("Sorry, Maximum number of Candidates has been reached.\n");
271         reset();
```

Compiler Resources Compile Log Debug Find Results

Line: 443 Col: 2 Sel: 0 Lines: 443 Length: 12108 Insert Done parsing in 0.093 seconds

voting.c

```
273     }
274     yellow();
275     candid new_candid;
276     int z;
277     printf("Enter First name: ");
278     scanf("%s",&new_candid.fname);
279     printf("Enter Last name: ");
280     scanf("%s",&new_candid.lname);
281     printf("Enter Your Vote sign (*,^,$,#,@): ");
282     scanf("%s",&new_candid.vote_sign);
283     reset();
284     for(z=0;z<*num_candid;z++){
285         if(strcmp(new_candid.vote_sign,cand_arr[z].vote_sign)==0){
286             system("cls");
287             red();
288             printf("Sorry, Selected Voting Symbol is already chosen by other candidate!\n");
289             reset();
290             return;
291         }
292     }
293     new_candid.vote_count = 0;
294     reset();
295
296     cand_arr[*num_candid] = new_candid;
297     *num_candid=*num_candid+1; //Moving the pointer to the next memory address
298     green();
299     system("cls");
300     printf("\n\nCandidate values inserted Successfully.\n");
301     reset();
```

Compiler Resources Compile Log Debug Find Results

Line: 443 Col: 2 Sel: 0 Lines: 443 Length: 12108 Insert Done parsing in 0.093 seconds

voting.c

```
306         red();
307         printf("Log in to your ID First!\n");
308         reset();
309         return;
310     }
311         if (num_candidates==0){
312         system("cls");
313         red();
314         printf("Candidate List is Empty!\n");
315         reset();
316         return;
317     }
318     if (voter_arr[index].has_voted==1){
319         system("cls");
320         red();
321         printf("Sorry, You have already Voted!\n");
322         reset();
323         return;
324     }
325     system("cls");
326     green();
327     printf("Please Choose the Party from the below to vote:- \nParty Name\t\t Party Symbol\n");
328     reset();
329     white();
330     int z;
331     for(z=0;z<num_candidates;z++){
332         printf("%d: %s %s \t %s\n",z+1,cand_arr[z].fname,cand_arr[z].lname,cand_arr[z].vote_sign);
333     }
334     reset();
```

Compiler Resources Compile Log Debug Find Results

Line: 443 Col: 2 Sel: 0 Lines: 443 Length: 12108 Insert Done parsing in 0.093 seconds

voting.c

```
339     scanf("%s",&sign);
340     reset();
341     int j;
342     for(j=0;j<num_candidates;j++){
343         if(strcmp(cand_arr[j].vote_sign,sign)==0){
344             found2++;
345             cand_arr[j].vote_count+=1;
346             voter_arr[index].has_voted=1;
347             system("cls");
348             green();
349             printf("Your Vote has been taken successfully!\n");
350             reset();
351         }
352     }
353
354     if(found2==0){
355         system("cls");
356         red();
357         printf("Invalid Choice, Please Try Again!\n");
358         reset();
359         return;
360     }
361 }
362
363 void login_logout(voting voter_arr,int num_voters,int log,int* index){
364     int id;
365     char pass[50];
366     if (num_voters==0){
367         red();
```

Compiler Resources Compile Log Debug Find Results

ine: 443 Col: 2 Sel: 0 Lines: 443 Length: 12108 Insert Done parsing in 0.093 seconds

voting.c

```
369             reset();
370             return;
371         }
372         else{
373         if(*log==2){
374             system("cls");
375             green();
376             printf("ID %d have been successfully logged out!\n",*index+1);
377             *log=0;
378             *index=-1;
379             reset();
380             return;
381         }
382         if(*log==0){
383             purple();
384             printf("Please enter your id: ");
385             reset();
386             scanf("%d",&id);
387             int i;
388             int found=0;
389             for(i=0;i<num_voters;i++){
390                 if(voter_arr[i].id==id){
391                     purple();
392                     printf("Please enter your password to continue: ");
393                     scanf("%s",&pass);
394                     reset();
395                     if(strcmp(voter_arr[i].password,pass)!=0){
396                         red();
397                         system("cls");
```

Compiler | Resources | Compile Log | Debug | Find Results

Line: 443 Col: 2 Sel: 0 Lines: 443 Length: 12108 Insert Done parsing in 0.093 seconds

```
402             }
403             system("cls");
404             green();
405             printf("User Verified :-\n\n");
406             found++;
407             printf("ID %d have been successfully logged in!\n\n",i+1);
408             *log=2;
409             *index=i;
410             reset();
411             return;
412         }
413         }
414         if(found==0){
415             red();
416             system("cls");
417             printf("Record of the user not found\nPlease Try Again!\n\n");
418             reset();
419             return;
420         }
421         }
422     }
423 }
424
425 void display_candidates(voting *voter_arr,int log,int index){
426     if (log==0){
427         red();
428         printf("Log in to your ID First!\n");
429         reset();
430         return;
```

Compiler | Resources | Compile Log | Debug | Find Results

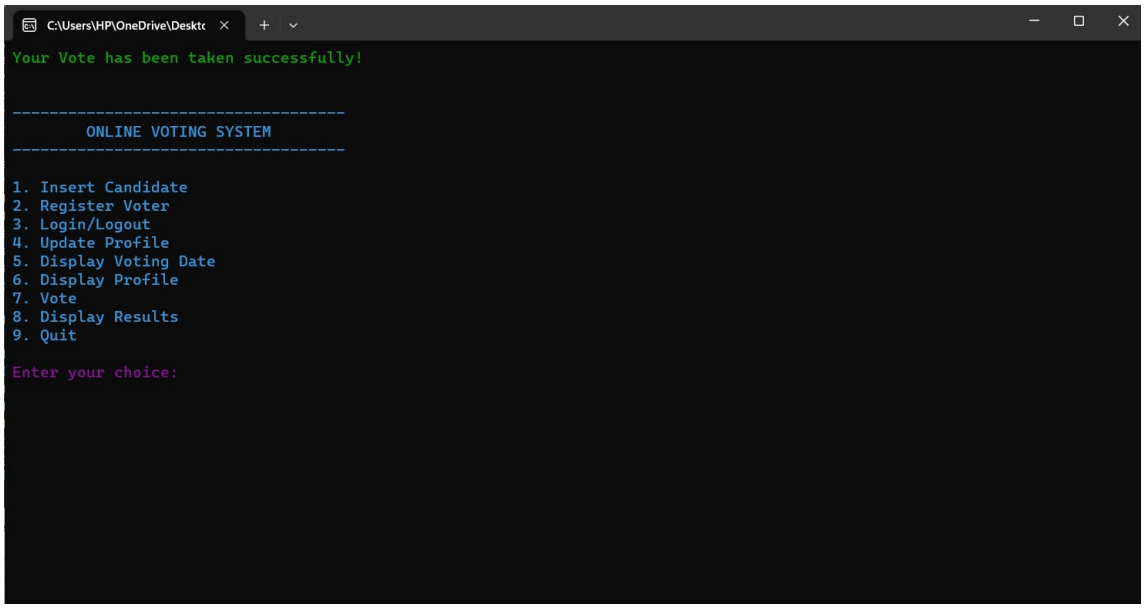
ine: 443 Col: 2 Sel: 0 Lines: 443 Length: 12108 Insert Done parsing in 0.093 seconds

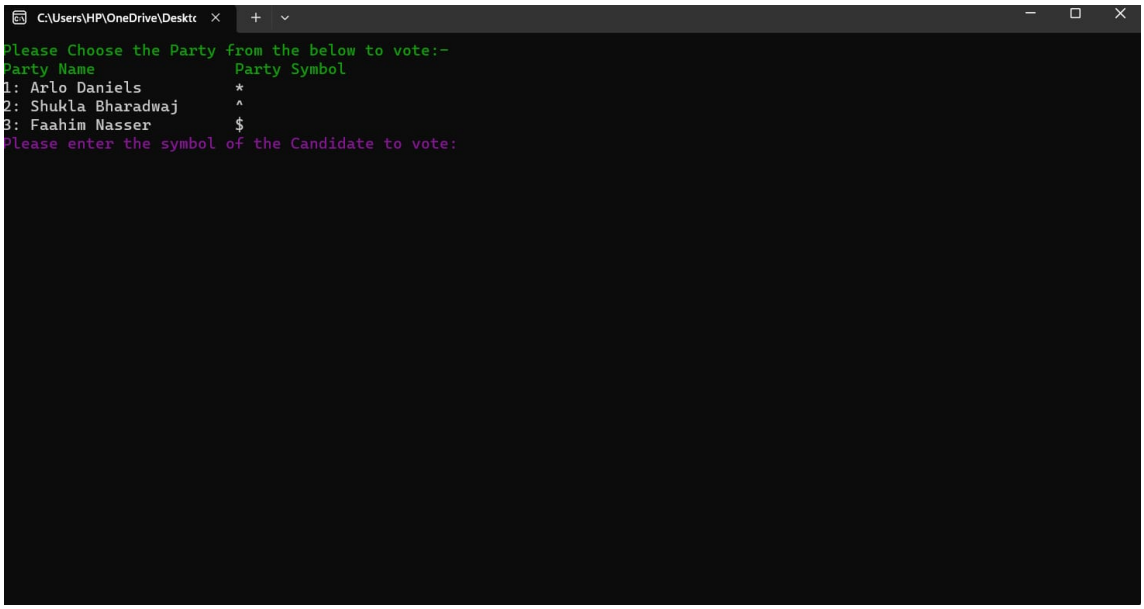
```
425 void display_candidates(voting *voter_arr,int log,int index){
426     if (log==0){
427         red();
428         printf("Log in to your ID First!\n");
429         reset();
430         return;
431     }
432     green();
433     printf("Voter Details are :-\n");
434     reset();
435     white();
436     printf("\nID:- %d",voter_arr[index].id);
437     printf("\nCurrent Password :- %s",voter_arr[index].password);
438     printf("\nUsername:- %s",voter_arr[index].uname);
439     printf("\nD.O.B:- %d/%d/%d",voter_arr[index].dob.day,voter_arr[index].dob.month,voter_arr[index].dob.year);
440     printf("\nArea:- %d",voter_arr[index].area);
441     printf("\nHas voted: %s\n\n", voter_arr[index].has_voted==1?"Yes":"No");
442     reset();
443 }
```

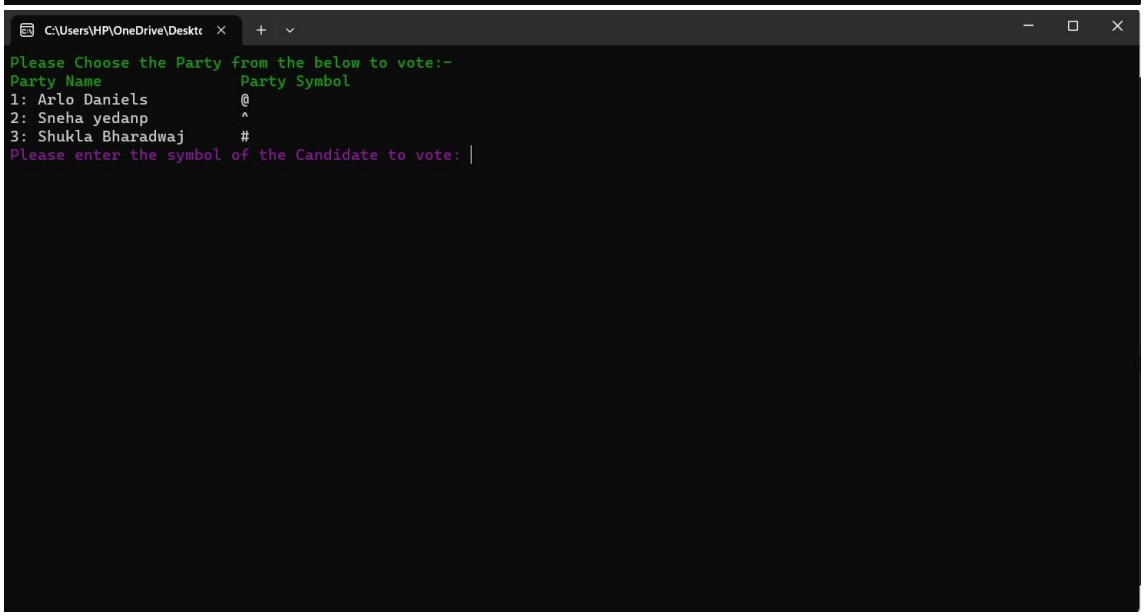
Compiler | Resources | Compile Log | Debug | Find Results

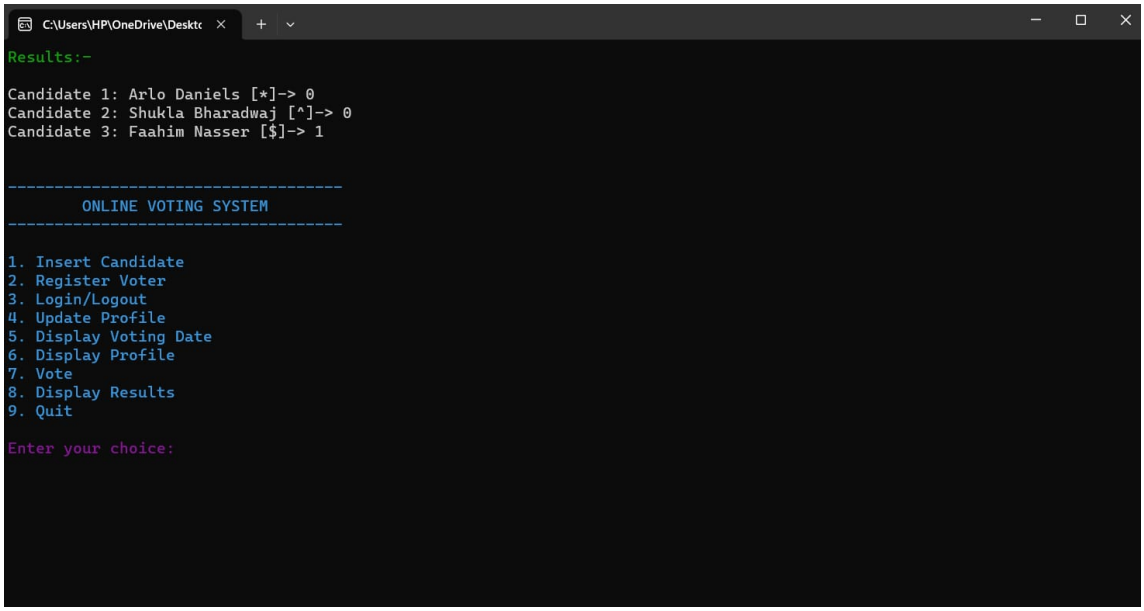
Line: 443 Col: 2 Sel: 0 Lines: 443 Length: 12108 Insert Done parsing in 0.093 seconds

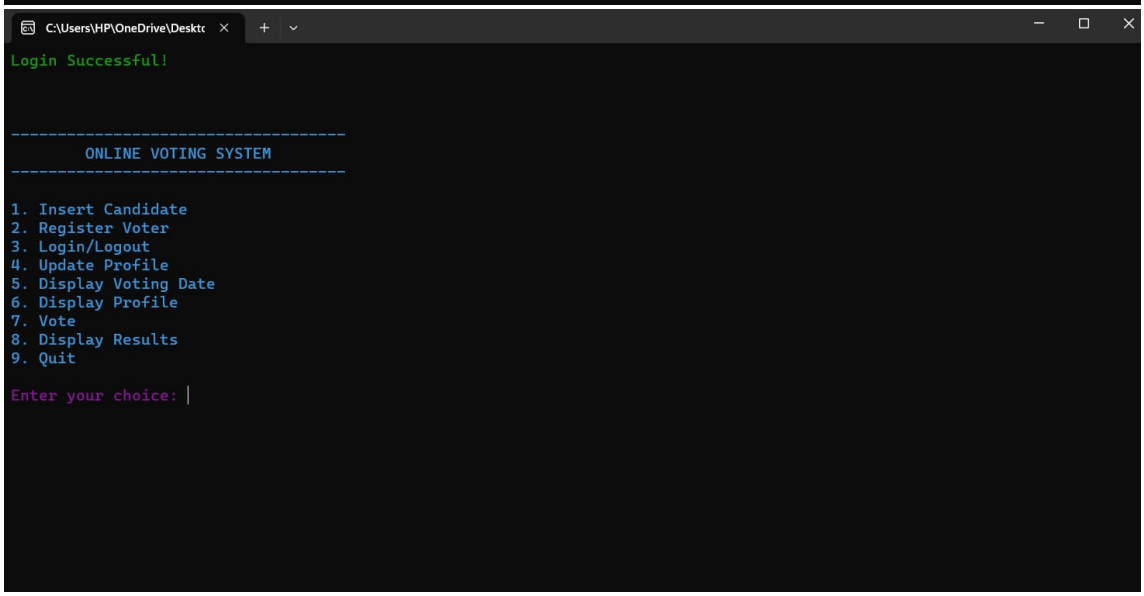
OUTPUT:


C:\Users\HP\OneDrive\Desktc
Your Vote has been taken successfully!
ONLINE VOTING SYSTEM
1. Insert Candidate
2. Register Voter
3. Login/Logout
4. Update Profile
5. Display Voting Date
6. Display Profile
7. Vote
8. Display Results
9. Quit
Enter your choice:


C:\Users\HP\OneDrive\Desktc
Please Choose the Party from the below to vote:-
Party Name
Party Symbol
1: Arlo Daniels
*
2: Shukla Bharadwaj
^
3: Faahim Nasser
$
Please enter the symbol of the Candidate to vote:


C:\Users\HP\OneDrive\Desktc
Please Choose the Party from the below to vote:-
Party Name
Party Symbol
1: Arlo Daniels
@
2: Sneha yedanp
^
3: Shukla Bharadwaj
#
Please enter the symbol of the Candidate to vote:


C:\Users\HP\OneDrive\Desktc
Results:-
Candidate 1: Arlo Daniels [*]-> 0
Candidate 2: Shukla Bharadwaj [^]-> 0
Candidate 3: Faahim Nasser [$]-> 1
ONLINE VOTING SYSTEM
1. Insert Candidate
2. Register Voter
3. Login/Logout
4. Update Profile
5. Display Voting Date
6. Display Profile
7. Vote
8. Display Results
9. Quit
Enter your choice:


C:\Users\HP\OneDrive\Desktc
Login Successful!
ONLINE VOTING SYSTEM
1. Insert Candidate
2. Register Voter
3. Login/Logout
4. Update Profile
5. Display Voting Date
6. Display Profile
7. Vote
8. Display Results
9. Quit
Enter your choice: