Object Oriented Programming

Mini-Project Report

Topic: Futuristic Voting System

Group Details:

1.Kushagra Goswami (23) (A 1)

2.Atharva Toshniwal (08) (A1)

3.Dhruv Gadge (14) (A1)

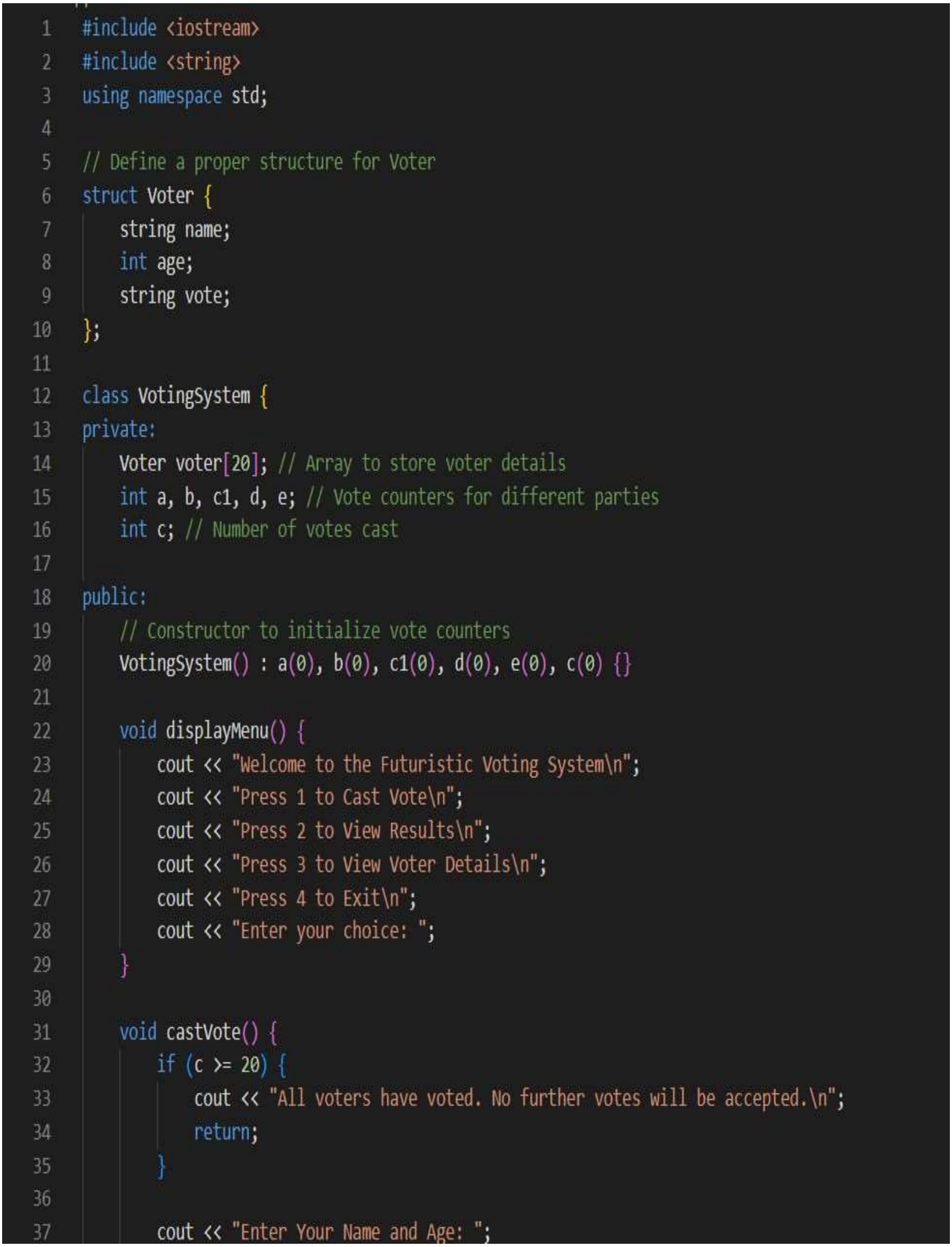
Aim of the Experiment:

The aim of this experiment is to implement a voting system using object-oriented programming (OOP) concepts in C++. The project simulates a real-world voting scenario where voters can cast their votes for different political parties.

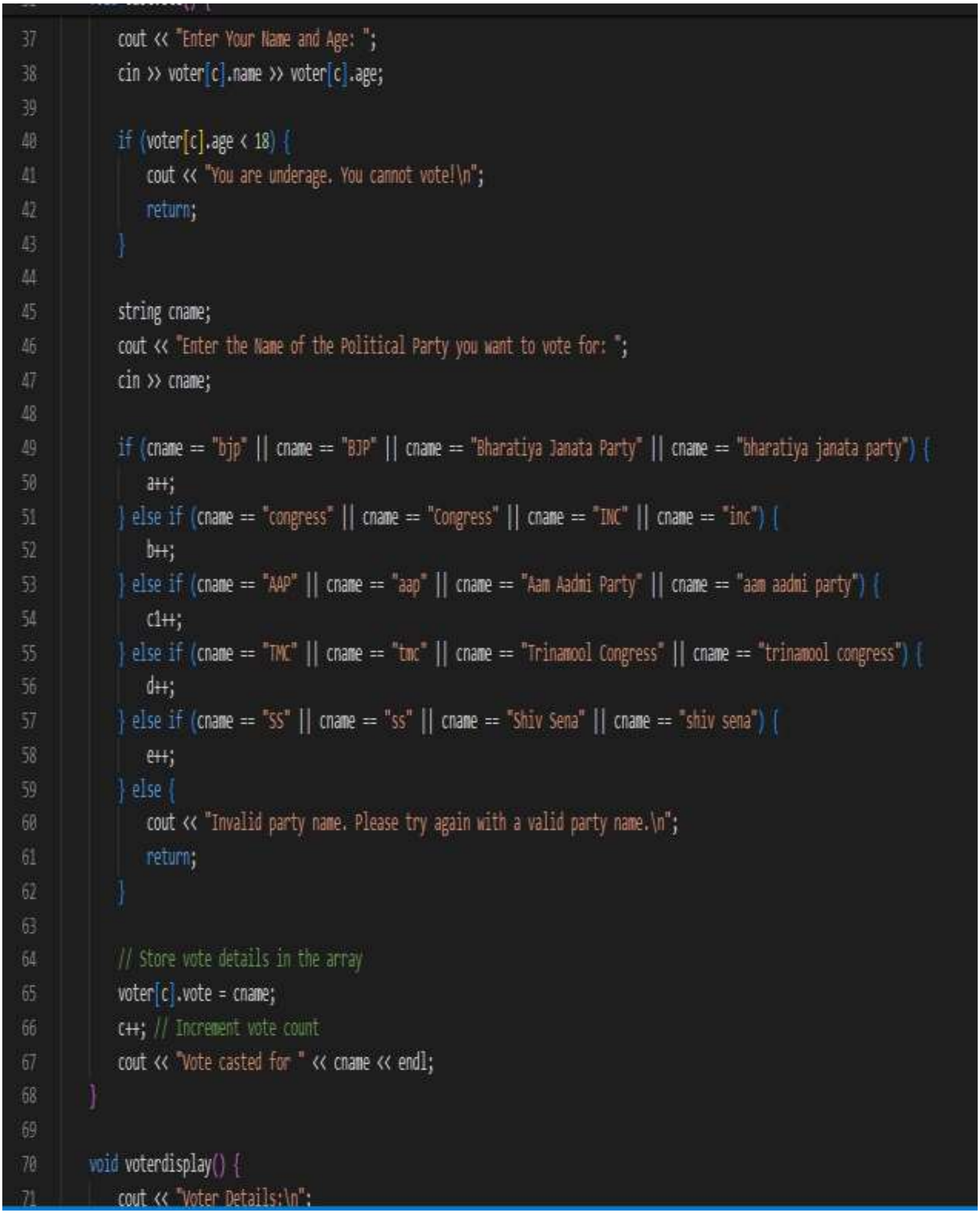
Objective:

The objective of this C++ project is to develop a comprehensive and efficient voting system utilizing object-oriented programming principles. The project aims to create a structured representation of voter details through a Voter structure, encompassing attributes such as name, age, and voting choice. The core focus is on implementing the Voting System class to oversee the entire voting process, including registering votes, displaying results, and managing user interactions via a console-based interface. The project aims to deliver a functional and professional voting system that highlights best practices in software design and object-oriented programming, demonstrating the effectiveness of OOP concepts in real-world application development.

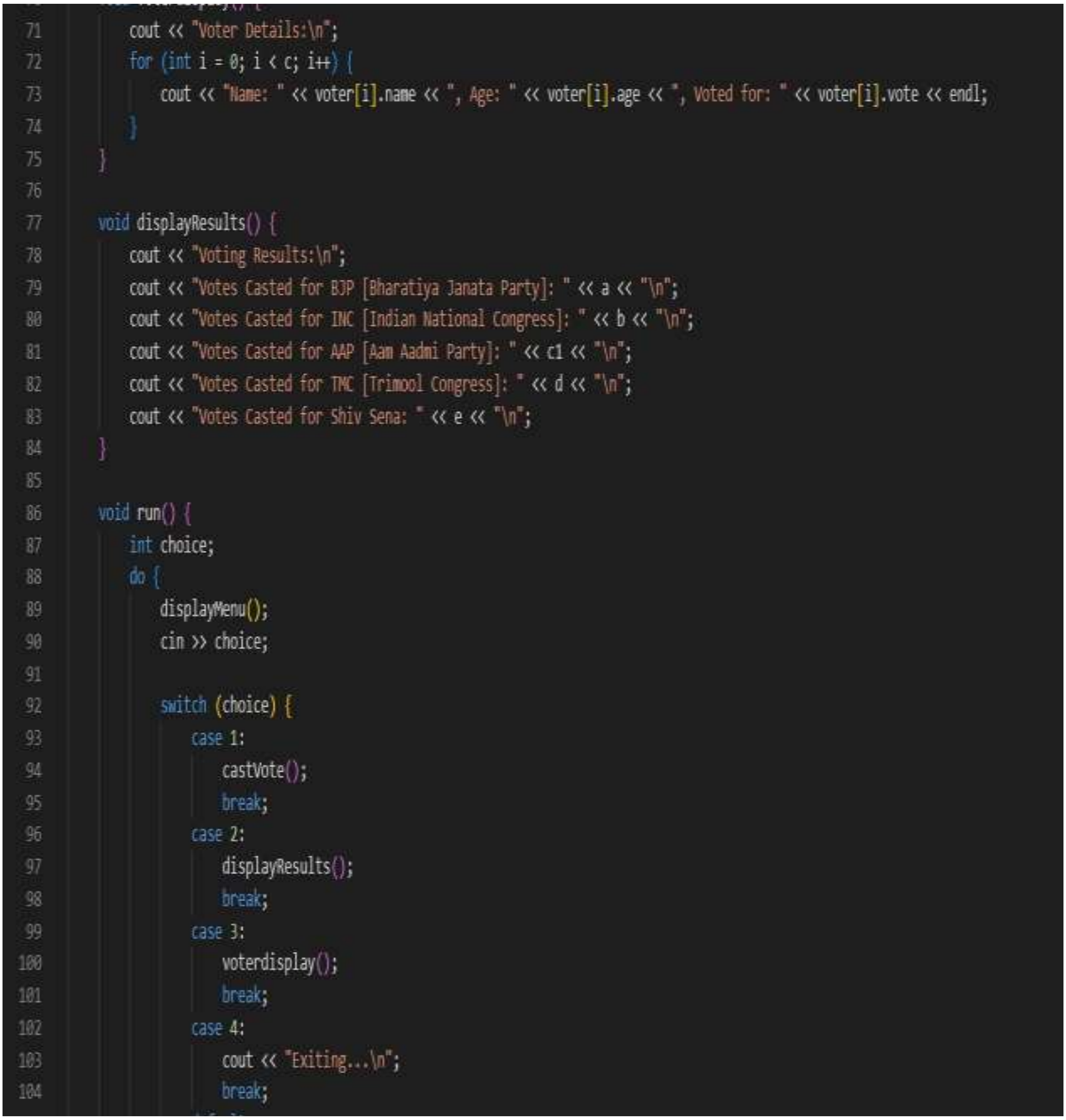
Code:



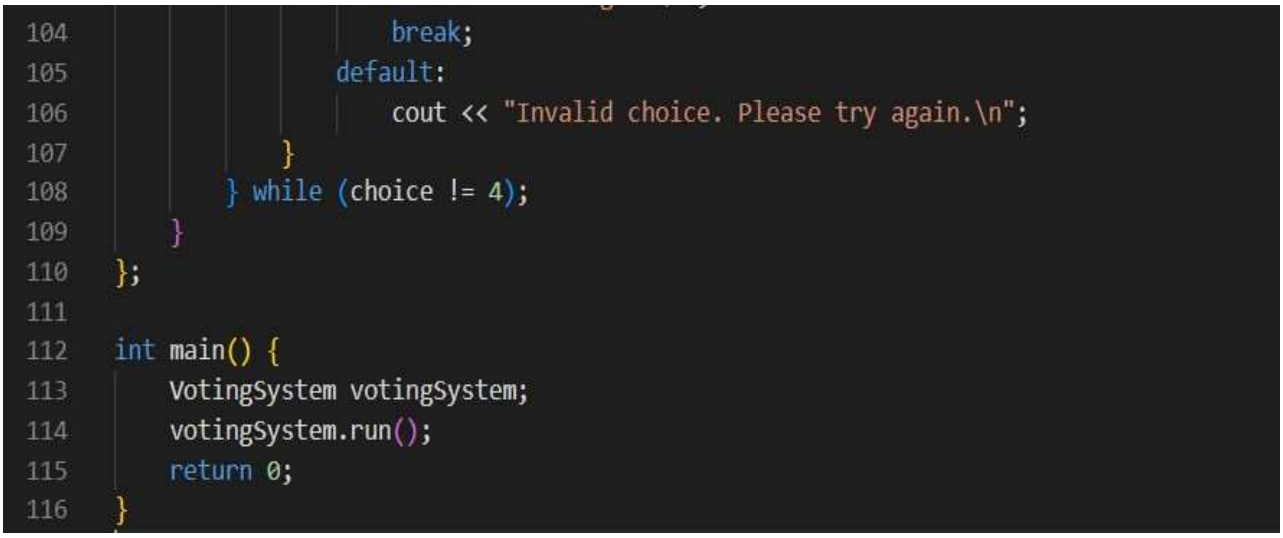
```
#include <iostream>
#include <string>
using namespace std;

// Define a proper structure for Voter
struct Voter {
    string name;
    int age;
    string vote;
};

class VotingSystem {
private:
    Voter voter[20]; // Array to store voter details
    int a, b, c1, d, e; // Vote counters for different parties
    int c; // Number of votes cast

public:
    // Constructor to initialize vote counters
    VotingSystem() : a(0), b(0), c1(0), d(0), e(0), c(0) {}

    void displayMenu() {
        cout << "Welcome to the Futuristic Voting System\n";
        cout << "Press 1 to Cast Vote\n";
        cout << "Press 2 to View Results\n";
        cout << "Press 3 to View Voter Details\n";
        cout << "Press 4 to Exit\n";
        cout << "Enter your choice: ";
    }

    void castVote() {
        if (c >= 20) {
            cout << "All voters have voted. No further votes will be accepted.\n";
            return;
        }

        cout << "Enter Your Name and Age: ";
```



```
    cout << "Enter Your Name and Age: ";
    cin >> voter[c].name >> voter[c].age;

    if (voter[c].age < 18) {
        cout << "You are underage. You cannot vote!\n";
        return;
    }

    string cname;
    cout << "Enter the Name of the Political Party you want to vote for: ";
    cin >> cname;

    if (cname == "bjp" || cname == "BJP" || cname == "Bharatiya Janata Party" || cname == "bharatiya janata party") {
        a++;
    } else if (cname == "congress" || cname == "Congress" || cname == "INC" || cname == "inc") {
        b++;
    } else if (cname == "AAP" || cname == "aap" || cname == "Aam Aadmi Party" || cname == "aam aadmi party") {
        c1++;
    } else if (cname == "TMC" || cname == "tmc" || cname == "Trinamool Congress" || cname == "trinamool congress") {
        d++;
    } else if (cname == "SS" || cname == "ss" || cname == "Shiv Sena" || cname == "shiv sena") {
        e++;
    } else {
        cout << "Invalid party name. Please try again with a valid party name.\n";
        return;
    }

    // Store vote details in the array
    voter[c].vote = cname;
    c++; // Increment vote count
    cout << "Vote casted for " << cname << endl;
}

void voterdisplay() {
    cout << "Voter Details:\n";
```

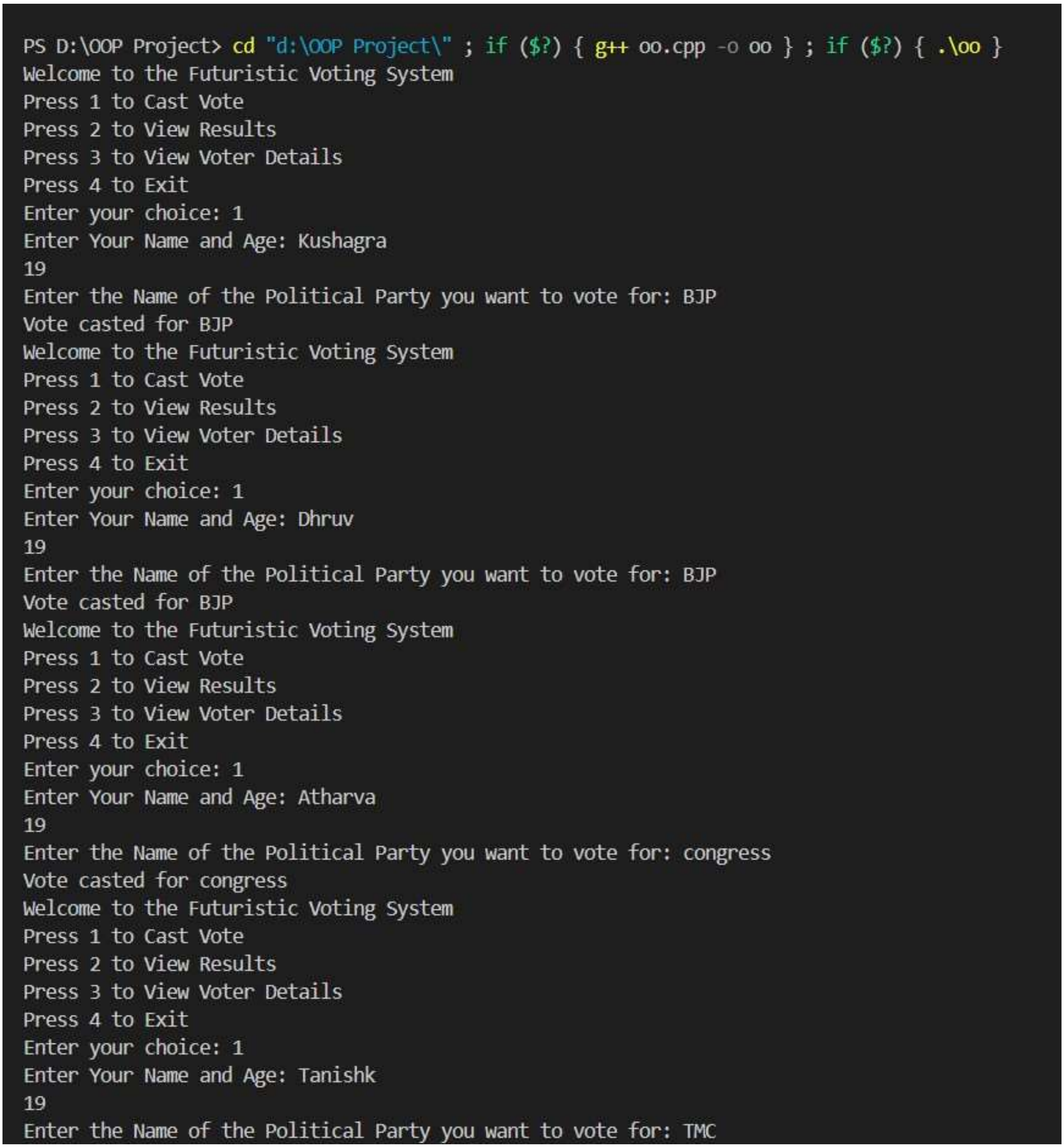


```
        cout << "Voter Details:\n";
        for (int i = 0; i < c; i++) {
            cout << "Name: " << voter[i].name << ", Age: " << voter[i].age << ", Voted for: " << voter[i].vote << endl;
        }
    }

    void displayResults() {
        cout << "Voting Results:\n";
        cout << "Votes Casted for BJP [Bharatiya Janata Party]: " << a << "\n";
        cout << "Votes Casted for INC [Indian National Congress]: " << b << "\n";
        cout << "Votes Casted for AAP [Aam Aadmi Party]: " << c1 << "\n";
        cout << "Votes Casted for TMC [Trimool Congress]: " << d << "\n";
        cout << "Votes Casted for Shiv Sena: " << e << "\n";
    }

    void run() {
        int choice;
        do {
            displayMenu();
            cin >> choice;

            switch (choice) {
                case 1:
                    castVote();
                    break;
                case 2:
                    displayResults();
                    break;
                case 3:
                    voterdisplay();
                    break;
                case 4:
                    cout << "Exiting...\n";
                    break;
```

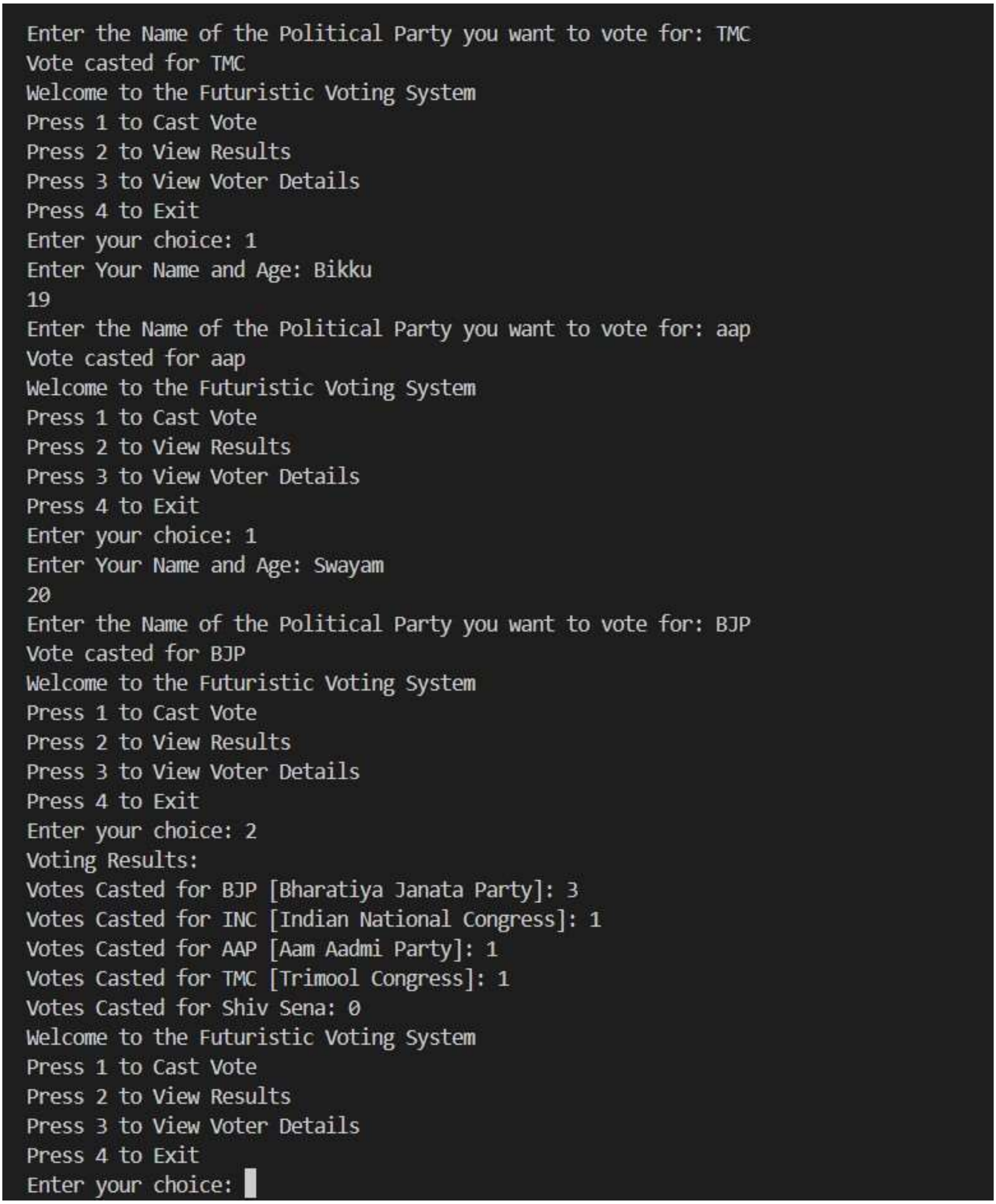


```
                    break;
                default:
                    cout << "Invalid choice. Please try again.\n";
            }
        } while (choice != 4);
    }
};

int main() {
    VotingSystem votingSystem;
    votingSystem.run();
    return 0;
}
```

Output:



```
PS D:\OOP Project> cd "d:\OOP Project\" ; if ($?) { g++ oo.cpp -o oo } ; if ($?) { .\oo }
Welcome to the Futuristic Voting System
Press 1 to Cast Vote
Press 2 to View Results
Press 3 to View Voter Details
Press 4 to Exit
Enter your choice: 1
Enter Your Name and Age: Kushagra
19
Enter the Name of the Political Party you want to vote for: BJP
Vote casted for BJP
Welcome to the Futuristic Voting System
Press 1 to Cast Vote
Press 2 to View Results
Press 3 to View Voter Details
Press 4 to Exit
Enter your choice: 1
Enter Your Name and Age: Dhruv
19
Enter the Name of the Political Party you want to vote for: BJP
Vote casted for BJP
Welcome to the Futuristic Voting System
Press 1 to Cast Vote
Press 2 to View Results
Press 3 to View Voter Details
Press 4 to Exit
Enter your choice: 1
Enter Your Name and Age: Atharva
19
Enter the Name of the Political Party you want to vote for: congress
Vote casted for congress
Welcome to the Futuristic Voting System
Press 1 to Cast Vote
Press 2 to View Results
Press 3 to View Voter Details
Press 4 to Exit
Enter your choice: 1
Enter Your Name and Age: Tanishk
19
Enter the Name of the Political Party you want to vote for: TMC
```

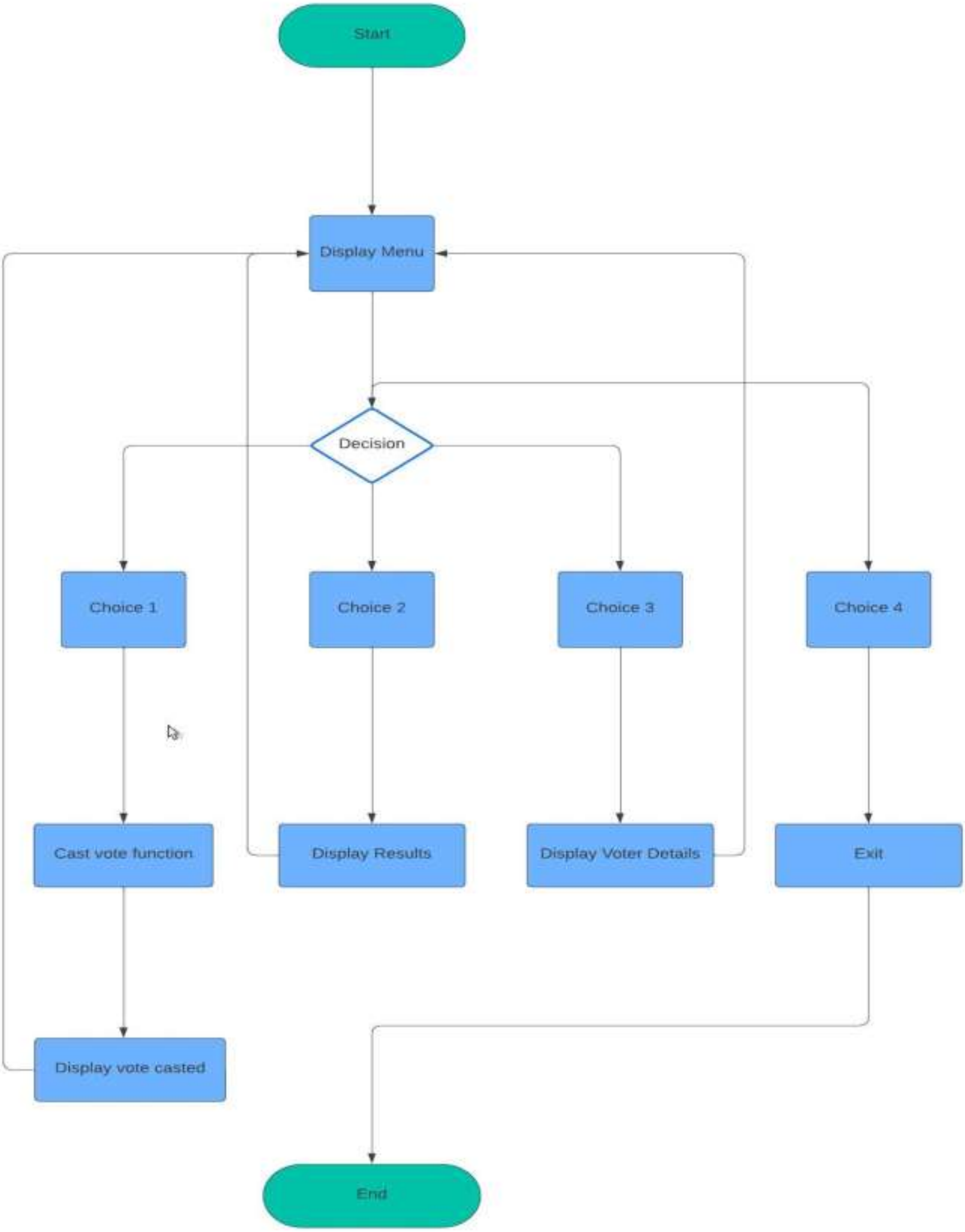


```
Enter the Name of the Political Party you want to vote for: TMC
Vote casted for TMC
Welcome to the Futuristic Voting System
Press 1 to Cast Vote
Press 2 to View Results
Press 3 to View Voter Details
Press 4 to Exit
Enter your choice: 1
Enter Your Name and Age: Bikku
19
Enter the Name of the Political Party you want to vote for: aap
Vote casted for aap
Welcome to the Futuristic Voting System
Press 1 to Cast Vote
Press 2 to View Results
Press 3 to View Voter Details
Press 4 to Exit
Enter your choice: 1
Enter Your Name and Age: Swayam
20
Enter the Name of the Political Party you want to vote for: BJP
Vote casted for BJP
Welcome to the Futuristic Voting System
Press 1 to Cast Vote
Press 2 to View Results
Press 3 to View Voter Details
Press 4 to Exit
Enter your choice: 2
Voting Results:
Votes Casted for BJP [Bharatiya Janata Party]: 3
Votes Casted for INC [Indian National Congress]: 1
Votes Casted for AAP [Aam Aadmi Party]: 1
Votes Casted for TMC [Trimool Congress]: 1
Votes Casted for Shiv Sena: 0
Welcome to the Futuristic Voting System
Press 1 to Cast Vote
Press 2 to View Results
Press 3 to View Voter Details
Press 4 to Exit
Enter your choice: 
```

Methodology:

1. **Define Voter Structure**:
   Start by defining a **Voter** structure with attributes (**name**, **age**, **vote**) to represent individual voters.
2. **Implement Voting System Class**:
   Create a **Voting System** class with member variables (**voter[]**, **a**, **b**, **c1**, **d**, **e**, **c**) to manage the voting process and store voting-related data.
3. **Encapsulate Data and Methods**:
   Use access specifiers (**private**, **public**) to encapsulate voter data and voting operations within the class. Provide public methods (**castVote()**, **displayResults()**, **voterdisplay()**) to interact with the voting system.
4. **Implement Voting Operations**:
   Write methods (**castVote()**, **displayResults()**, **voterdisplay()**) to handle specific voting operations, such as registering votes based on user input and displaying voting results and voter details.
5. **Develop User Interface**:
   Design a console-based user interface (**displayMenu()**) to interact with the voting system. Present a menu with options for casting votes, viewing results, displaying voter details, and exiting the program.
6. **Execute the Program**:
   Instantiate the **VotingSystem** class in the **main()** function and run the program (**run()**) to start the voting system. Allow users to interact with the voting system by choosing different options from the menu.
7. **Test and Validate**:
   Test the program by casting votes, viewing results, and displaying voter details to ensure the voting system functions correctly and handles various scenarios (e.g., underage voters, invalid party names).

Flowchart:


Start
Display Menu
Decision
Choice 1
Choice 2
Choice 3
Choice 4
Cast vote function
Display Results
Display Voter Details
Exit
Display vote casted
End

Conclusion:

The C++ voting system project showcases the effective application of object-oriented programming (OOP) principles to create a comprehensive and user-friendly software application. By utilizing a structured Voter data representation and implementing the Voting System class to manage voting processes, the project embodies core OOP concepts such as encapsulation, abstraction, and modularity

Thank You