A

PROJECT REPORT

ON

“ONLINE VOTING MANAGEMENT SYSTEM”

SUBMITTED BY:

Miss. BHAKTI AJAY KADAM

PRN:2124UCEF1086

Class: F.Y.B-Tech

SUBJECT:

C++ PROGRAMMING

UNDER THE GUIDANCE OF:

PROFESSOR. ISHWARI TIRSE

Department of Computer Science and Engineering

Sanjivani Rural Education Society’s

SANJIVANI UNIVERSITY

KOPARGAON – 423603, DISTRICT: AHMEDNAGAR

2024-2025

INDEX

INTRODUCTION

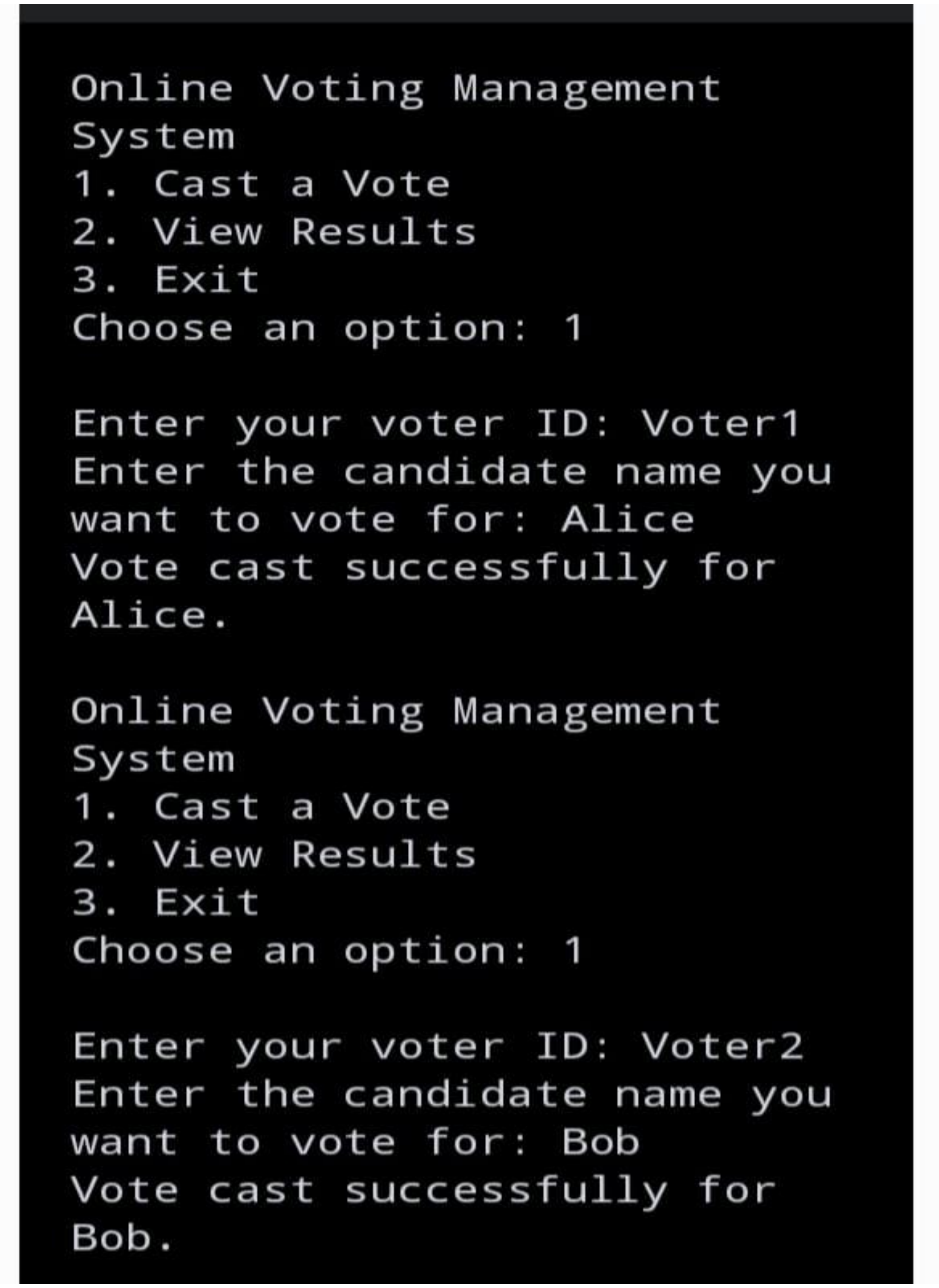
An Online Voting Management System is a software application designed to facilitate secure, efficient, and accessible voting processes for elections. This system automates the entire voting lifecycle, from voter registration to ballot casting and results tallying. It aims to enhance transparency, reduce administrative overhead, and increase voter participation by providing a user-friendly interface for both voters and election officials. Key features of an Online Voting Management System include Voter Authentication, Ballot Management, Vote Casting, Results Calculation, and Audit Trails. These features ensure that the voting process is secure, anonymous, and verifiable, thereby maintaining the integrity of the electoral system.

C++ is a powerful programming language well-suited for developing an Online Voting Management System due to its performance efficiency, strong type checking, and rich library support. C++ allows for fine-grained control over system resources, making it ideal for building secure and responsive applications. The benefits of a C++-based Online Voting Management System include High Performance, Enhanced Security, Flexibility in System Design, and the ability to handle complex data structures efficiently. By leveraging C++, the system can ensure reliable operation even under high load during elections, making it a compelling choice for modern electoral processes.

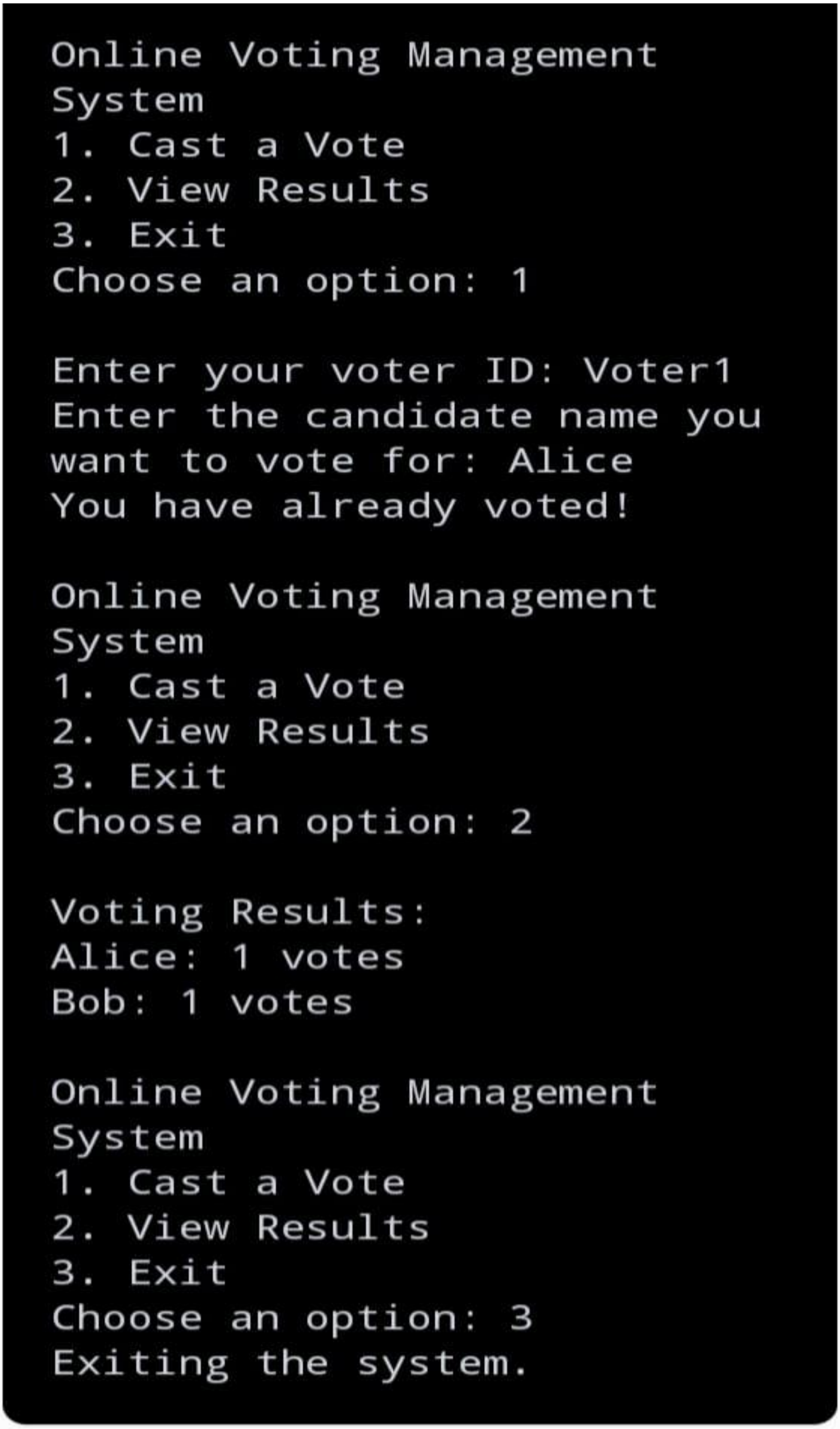
CODE

```
#include <iostream>
#include <vector>
#include <string>
#include <algorithm>
using namespace std;
class Candidate {
private:
    string name;
    int votes;
public:
    Candidate(string name) : name(name), votes(0) {}
    string getName() const { // Mark as const since it doesn't modify the object
        return name;
    }
    void addVote() {
        votes++;
    }
    int getVotes() const { // Mark as const since it doesn't modify the object
        return votes;
    }
    void display() const { // Mark as const since it doesn't modify the object
        cout << name << ": " << votes << " votes" << endl;
    }
};

class VotingSystem {
private:
    vector<Candidate> candidates;
    vector<string> voters;

public:
    void addCandidate(const string& name) { // Pass by const reference for efficiency
        candidates.push_back(Candidate(name));
    }

    bool vote(const string& voterId, const string& candidateName) { // Pass by const reference
        if (find(voters.begin(), voters.end(), voterId) != voters.end()) {
            cout << "You have already voted!" << endl;
            return false;
        }

        // Find the candidate by reference to modify the votes
        for (auto& candidate : candidates) {
            if (candidate.getName() == candidateName) {
                candidate.addVote();
                voters.push_back(voterId);
                cout << "Vote cast successfully for " << candidateName << "." << endl;
                return true;
            }
        }
        cout << "Candidate not found!" << endl;
        return false;
    }

    void viewResults() const { // Mark as const since it doesn't modify the object
        cout << "Voting Results:" << endl;
        for (const auto& candidate : candidates) {
            candidate.display();
        }
    }
};

int main() {
    VotingSystem votingSystem;

    votingSystem.addCandidate("Alice");
    votingSystem.addCandidate("Bob");

    while (true) {
        cout << "\nOnline Voting Management System" << endl;
        cout << "1. Cast a Vote" << endl;
        cout << "2. View Results" << endl;
        cout << "3. Exit" << endl;
        cout << "Choose an option: ";

        int choice;
        cin >> choice;
        cin.ignore(); // Consume newline

        switch (choice) {
            case 1: {
                string voterId, candidateName;
                cout << "Enter your voter ID: ";
                getline(cin, voterId);
                cout << "Enter candidate name to vote for: ";
                getline(cin, candidateName);
                votingSystem.vote(voterId, candidateName);
                break;
            }
            case 2:
                votingSystem.viewResults();
                break;
            case 3:
                cout << "Exiting the system." << endl;
                return 0;
            default:
                cout << "Invalid choice. Please try again." << endl;
        }
    }
    return 0;}
```

OUTPUT



```
Online Voting Management
System
1. Cast a Vote
2. View Results
3. Exit
Choose an option: 1

Enter your voter ID: Voter1
Enter the candidate name you
want to vote for: Alice
Vote cast successfully for
Alice.

Online Voting Management
System
1. Cast a Vote
2. View Results
3. Exit
Choose an option: 1

Enter your voter ID: Voter2
Enter the candidate name you
want to vote for: Bob
Vote cast successfully for
Bob.
```



```
Online Voting Management
System
1. Cast a Vote
2. View Results
3. Exit
Choose an option: 1

Enter your voter ID: Voter1
Enter the candidate name you
want to vote for: Alice
You have already voted!

Online Voting Management
System
1. Cast a Vote
2. View Results
3. Exit
Choose an option: 2

Voting Results:
Alice: 1 votes
Bob: 1 votes

Online Voting Management
System
1. Cast a Vote
2. View Results
3. Exit
Choose an option: 3
Exiting the system.
```

CONCLUSION

In conclusion, an Online Voting Management System developed in C++ is an effective solution for facilitating secure and efficient electoral processes. With its ability to manage candidate information, ensure voter authentication, and maintain the integrity of votes, this system enhances transparency and accessibility in voting. The performance efficiency and strong type-checking capabilities of C++ make it a suitable choice for handling the complexities involved in online voting. Overall, this system is a valuable tool for modern elections, promoting civic engagement and trust in democratic processes.