Microsoft account

**election MANAGEMENT SYSTEM**

**PREPARED BY:**

**PRATEEK SHARMA (1/23/SET/BCS/579)**

**SHIV SHANKAR (1/23/SET/BCS/ )**

**AKSHAT SHARMA (1/23/SET/BCS/ )**

**SAMRIDH SINGH (1/23/SET/BCS/ )**

**ABSTRACT**

This project presents a **C++-based Election Management System** that facilitates the key operations required for conducting a small-scale election. Designed to ensure security, accuracy, and ease of use, this system provides a comprehensive solution for managing election processes, from voter and candidate registration to voting and results calculation. With a focus on a user-friendly experience, the system enables both administrators and voters to participate in and manage elections efficiently.

The system is structured around several core functionalities:

1. **Voter Registration**: This module securely registers voters by collecting essential information such as voter ID, name, and age, with validation checks to prevent duplicate registrations. It ensures that only registered voters can participate in the election, enhancing security and authenticity.
2. **Candidate Registration**: In the candidate management module, administrators can add and remove candidates while storing details like the candidate’s ID, name, political party, and manifesto. This feature organizes the candidate information in a structured format, allowing easy access during the voting process.
3. **Voting System**: A secure voting system allows registered voters to cast their votes for candidates. The system ensures that each voter can vote only once, with a real-time tally of votes for each candidate. During voting, a list of candidates is displayed to the voter, making it easy to select a preferred candidate. This feature reinforces the integrity of the election by preventing unauthorized votes and duplicate voting.
4. **Election Management**: The system allows administrators to manage the election lifecycle by opening and closing the voting phase. This feature controls when voters are allowed to cast their votes, ensuring orderly and timely election processes.
5. **Result Calculation**: After voting ends, the system automatically calculates the results by counting the votes received by each candidate. It then displays the vote count and announces the winner. This process is designed to be efficient and accurate, minimizing the need for manual tallying and reducing the risk of errors.

This Election Management System leverages basic data structures in C++, such as lists and maps, to store and retrieve information dynamically, ensuring smooth interaction with users. The system employs basic control flow constructs, such as conditionals and loops, to handle user inputs and enforce rules, making it a practical demonstration of fundamental programming concepts in C++.

In summary, this Election Management System offers a reliable, secure, and user-friendly solution for conducting elections. It is ideal for educational purposes or small organizations, demonstrating how C++ can be used to build effective management systems that emphasize security, accessibility, and simplicity. This project also highlights the potential for expanding the system to accommodate larger elections or integrate more sophisticated features like a graphical interface or networked voting.

**SYSTEM CONFIGURATION**

**HARDWARE CONFIGURATION:**

RAM: 2 GB

Processor: Dual-core

Disk Space: 500 MB

Monitor: 800x600

Keyboard: Standard

Mouse: 2-button

**SOFTWARE CONFIGURATION:**

The software used for the development of the project is:

* OPERATING SYSTEM : WINDOWS 10
* ENVIRONMENT : VISUAL STUDIO CODE or code blocks
* LANGUAGE : C++

**OBJECTIVE**

The objective of this Election Management System project is to provide a secure, efficient, and user-friendly platform for managing and conducting elections digitally. This system is designed to automate the core functions involved in an election process, such as:

Voter Registration: Allowing only eligible and verified individuals to register as voters.

Candidate Registration: Enabling administrators to register candidates, providing essential details like name and party.

Voting: Enabling voters to securely cast their vote for registered candidates, with mechanisms to prevent multiple voting by the same person.

Result Calculation: Automatically calculating and displaying results based on the votes each candidate receives.

By integrating these functions into a single system, the project aims to eliminate manual errors, ensure fair and transparent voting, and simplify the election process for both administrators and voters. This project demonstrates the use of C++ programming concepts, such as object-oriented principles, to develop a reliable and easily manageable election solution.

**MODULE DESCRIPTION**

#### **1. Administrator Modules**

The Administrator manages and oversees the election process, ensuring smooth operation and secure data management.

* **Voter Registration Management**
  + Registers eligible voters, collecting and verifying necessary details such as voter ID, name, and age.
  + Prevents duplicate entries and maintains a secure voter database.
* **Candidate Registration Management**
  + Adds and organizes candidate information, including name, ID, and party affiliation.
  + Allows administrators to update or remove candidates as needed.
* **Election Control**
  + Manages the election status, including starting and ending the voting phase.
  + Restricts voting to an active election period to ensure accuracy and fairness.
* **Result Calculation and Display**
  + Counts votes for each candidate once voting closes.
  + Displays total votes for each candidate and announces the winner automatically.

#### **2. Voter Modules**

The Voter participates in the election by registering, viewing candidates, and casting their vote.

* **Voter Registration**
  + Allows eligible individuals to register as voters by entering and verifying their details.
  + Ensures only registered and verified voters can participate in the election.
* **Candidate Viewing and Voting**
  + Displays a list of registered candidates to help the voter make an informed decision.
  + Allows each voter to cast a single vote, ensuring one-person, one-vote integrity