🗳️Voting System Project Proposal

Programming Fundamentals - CCP

Project Title: Voting System

Language Used: C

Submitted By:

1. Mariya Vayani (CT-061)
2. Rubaisha Arif (CT-067)
3. Ujala Usman (CT-073)

Submitted To:

Instructor: Mr. Abdullah

Submission Date: 10th November 2025

---

1. Introduction

The **Voting System Project** is a console-based program developed in the **C programming language** to simulate a basic and interactive voting process. It allows users to cast votes for predefined candidates, view live results, detect ties, and exit the system safely. The project showcases fundamental programming concepts such as loops, conditionals, arrays, and functions, providing hands-on experience in modular and structured programming.

---

2. Problem Statement

Manual vote counting can be time-consuming and prone to human error. This project automates the voting process by implementing a digital system that:

- Allows users to cast votes conveniently.
- Automatically calculates and displays results.
- Detects ties or declares the winner immediately.

The project is designed for simplicity, making it ideal for classroom use, group activities, or demonstration purposes.

3. Objectives

1. To design a simple, interactive voting system.
2. To use arrays and loops for managing votes efficiently.
3. To separate logic into functions for clarity and modularity.
4. To implement real-time result computation.
5. To provide clean and formatted output for users.

4. Tools and Technologies

Tool / Component	Purpose
Language: C	Program development and logic implementation
Compiler: GCC / Code::Blocks / Dev-C++	Code execution and testing
Operating System: Windows / Linux	Execution environment
Header Files: stdio.h, string.h	Input/output and string manipulation

5. Methodology

Step 1: Initialization

• Define arrays for storing candidate names and vote counts.

Step 2: Display Menu

• Create a `Menu()` function to show voting options and guide user input.

Step 3: Input Processing

• Accept user choice and process it using conditional statements.
• If 1–3 → increment vote count for the respective candidate.
• If 4 → display results via `Results()` function.
• If 5 → exit the program.

Step 4: Display Results

• The `Results()` function displays total votes, identifies the winner, and handles tie conditions.

Step 5: Looping

• A while loop keeps the system active until the user chooses to exit.

6. Expected Output

```
        VOTING SYSTEM
            MENU
[1]: Vote for Candidate 1
[2]: Vote for Candidate 2
[3]: Vote for Candidate 3
[4]: Display Result
[5]: Exit
enter choice : 1
Candidate 1 : Vote has been cast

enter choice : 4
        VOTING RESULTS
Candidate 1--Votes: 1
Candidate 2--Votes: 0
Candidate 3--Votes: 0
WINNER: Candidate 1 with 1 votes
```

---

7. Advantages

- Provides accurate and instant vote counting.
- Simple and user-friendly interface.
- Demonstrates modular programming with clear function division.
- Easily extendable for additional candidates or options.

---

8. Future Enhancements

- Add **file handling** to store votes permanently.
- Include **voter authentication** to prevent duplicate voting.
- Add **dynamic candidate input** and variable number of voters.
- Create a **graphical interface** for enhanced presentation.

---

9. Conclusion

The **Voting System in C** project successfully demonstrates how programming fundamentals can be applied to simulate a real-world process. It utilizes **loops**, **arrays**, **conditional statements**, and **functions** to create a logical, structured, and modular program. The system performs accurate vote counting, tie detection, and result announcement in a clear and interactive manner.

This project not only meets all its functional objectives but also serves as an excellent base for further enhancements, such as secure voting systems with persistent data storage and graphical representation of results.