

Project Report

Project title: Electronic Voting system

Department: Department of Software Engineering

Student: Sarim Sadiq

Student ID: 25K-3081

Student: Zayan Kamran

Student ID: 25K-3043

Submitted To: Miss Izzah Salam

Semester: Fall 2025

Submission Date: 25/11/25

Abstract

Our project implements a simple command-line Electronic Voting Recorder written in C. It demonstrates basic file I/O, string processing, structured data (via struct), input validation (Pakistan CNIC format), duplicate-voter prevention, age check (18+), and vote tallying. The program logs each valid vote to a persistent file (voters.txt) and shows final tallies for three candidates. This report explains the problem, system design, implementation details, testing, results, limitations, and future improvements.

1. Introduction

Electronic voting systems are an important topic in computer science and civic technology. This small-scale project simulates the essential aspects of an election frontend: collecting voter information, validating identity (CNIC), ensuring eligibility (age), preventing duplicate votes, recording votes persistently, and tallying results.

2. Objectives

- Build a console-based C application to capture votes and voter data.
 - Implement robust CNIC validation function.
 - Provide duplicate CNIC detection in-memory per run.
 - Store voter records in a persistent file (voters.txt) with clear formatting.
 - Produce readable documentation and an appendix containing the full source code.
-

3. System Design

System Overview

This program collects voter information, verifies eligibility, prevents duplicate voting, and records each valid vote in a file. After all entries, it displays the final voting results.

Flow of the Program:

Start → Input Voter Details → Validate CNIC → Check Age → Check Duplicate CNIC → Record Vote → Ask to Continue → Show Final Results → Exit

Algorithm

1. Start the program
2. Open the file voters.txt in append mode
3. Repeat the following steps until the user chooses to stop:
 - a. Ask the voter for name, CNIC, and age
 - b. Validate CNIC format (xxxxx-xxxxxxx-x)
 - c. Check if CNIC is already used in the current session
 - d. Verify age is 18 or above
 - e. Ask the voter to choose a candidate
 - f. Record the vote and save details in the file
 - g. Ask if another vote should be entered
4. Display the final vote count for all candidates
5. End the program

Input & Output

Input:

- Voter name
- CNIC number
- Age
- Candidate choice

Output:

- Messages for validation (invalid CNIC, duplicate CNIC, underage, invalid choice)
- Confirmation of recorded vote
- Final vote tally for each candidate

4. Implementation

The application is implemented in ANSI C with standard libraries:

- <stdio.h> for I/O.

- `<string.h>` for string operations.

Key implementation notes:

- Input for names uses `scanf("%[^\\n]s", v.name);` so names can contain spaces.
- CNIC input is read as `%s` (no spaces expected).
- Duplicate detection uses `strcmp` between usedCNICs entries and the newly-entered CNIC.
- Votes are logged to `voters.txt` with readable lines, e.g.:
- Name: John Doe
- CNIC: 12345-1234567-1
- Age: 25
- Voted For: Candidate A
- -----

File handling: `fopen("voters.txt", "a")` opens file in append mode to preserve previous runs.

User prompts: The program uses a loop controlled by `do { ... } while (again == 'Y' || again == 'y');` to accept multiple voters.

5. Testing & Results

5.1 Test cases executed

1. Valid voter:

- Name: Ali
 - CNIC: 12345-1234567-1
 - Age: 20
 - Choice: 1
- Expected:* Vote recorded, file appended, votesA incremented.

2. Invalid CNIC format:

- Example: 1234-1234567-1 (wrong length)
- Expected:* Validation fails, prompt error and continue loop.

3. Underage voter:

- Age: 16

Expected: Message "You are under 18. Cannot vote." and no record saved.

4. Duplicate CNIC:

- Enter same CNIC as first valid voter in a later attempt.

Expected: Message "This CNIC has already voted!" and no vote recorded.

5. Invalid candidate choice:

- Choice: 5

Expected: "Invalid choice. Try again." and no record saved.

5.2 Results

The program correctly validated CNIC formats, blocked underage and duplicate CNIC submissions, logged valid votes to voters.txt, and printed the final tallies. Example final output:

--- Final Voting Results ---

Candidate A: 2

Candidate B: 1

Candidate C: 0