N.B.K.R INSTITUTE OF SCIENCE AND TECHNOLOGY

## Vidyanagar, Tirupathi Dist

### Voting System [Team – 13]

Team Members:

1. M. Ganesh [24KB1A05BE]

2. P. Srinadh [24KB1A05EQ]

3. M. Vasu [24KB1A05AR]

## 

## Course: Data Structures

## Department: CSE

## Date: 29 - 04 – 25

Guide: Ashok Sekhar Kumar E

Certificate

This is to certify that [ M. Ganesh (24KB1A05BE) , P. Srinadh (24KB1A05EQ) , M. Vasu (24KB1A05AR) ], students of [CSE], Section [c]. At [N.B.K.R INSTITUTE OF SCIENCE AND TECHNOLOGY], has successfully completed the project titled "Voting System in C" under my supervision. This project is a partial fulfillment of the curriculum of [ 1st B.Tech 2nd Semester].  
  
Guide Name: Ashok Sekhar Kumar E

(Signature)  
  
Date: \_29 – 04 - 25\_\_\_\_\_\_\_\_\_\_

Acknowledgment

I would like to express my deep gratitude to Ashok Sekhar Kumar E, my project guide, for their valuable support, guidance, and encouragement throughout this project. I am also thankful to my family and friends who have helped me throughout the development of this project.

Abstract

This project demonstrates the creation of a basic Voting System using C programming language. The system is designed to: secure access with a password, allow voting for 5 candidates, and display the results of the voting. The project focuses on fundamental concepts like arrays, loops, string comparison, and conditional statements.

Project Description

Problem Statement

In many basic elections or polls, a secure and straightforward system for collecting votes is required without complexity.

## Solution

A simple, password-protected voting system was developed using the C programming language to demonstrate secure and user-friendly voting.

## Key Features

- Secure access via password authentication  
- User-friendly interface for voting  
- Real-time vote counting and result display  
- Simple and lightweight application

Methodology

The system was developed using structured programming techniques in C. The program starts by securing access with a password. Once authenticated, users can cast their votes by selecting candidates from a menu-driven system. The votes are counted dynamically, and results can be displayed on user request. Extensive use of arrays, loops, conditional statements, and string handling functions ensure proper functionality.

Software & Hardware Requirements

Software:

- GCC Compiler (MinGW / CodeBlocks / Turbo C / Dev C++ / online Gbd)  
- Text Editor (Notepad / VS Code / Ms Word / Ms PowerPoint)  
  
Hardware:

- Any standard PC or Laptop  
- Windows/Linux Operating System

System Design

Flow:

1. Enter Password → If correct → Show Menu  
2. Choose Candidate (1-5) to vote  
3. Option to Show Results or Exit  
4. Invalid entries are handled properly.

Security:

- Simple password check using strcmp() function.

Code Explanation

- Password Check: Compares user input with stored password using strcmp().  
- Candidates: Names stored in a 2D array.  
- Votes: Votes are recorded in an integer array.  
- Menu Driven: Using loops and conditions to offer voting or showing results.

Testing and Validation

Several test cases were conducted to ensure:  
- Correct password authentication  
- Proper vote counting  
- Accurate result display  
- Handling of invalid inputs  
All functionalities were successfully validated with no major errors encountered during testing.

Source Code

#include <stdio.h>

#include <string.h>

#define PASSWORD "12345"

int main() {

int votes[5] = {0};

char candidates[5][20] = {"candidate 1", "candidate 2", "candidate 3", "candidate 4", "candidate 5"};

char password[20];

int choice, i, voting = 1;

printf("Enter password to access voting system: ");

scanf("%s", password);

if (strcmp(password, PASSWORD) != 0) {

printf("Incorrect password. Access denied.\n");

return 0;

}

while (voting) {

printf("\n---------- Voting System ----------\n");

printf("Please vote for your candidate:\n");

for (i = 0; i < 5; i++) {

printf("%d. %s\n", i + 1, candidates[i]);

}

printf("6. Show Results\n");

printf("7. Exit\n");

printf("Enter your choice (1-7): ");

scanf("%d", &choice);

if (choice >= 1 && choice <= 5) {

votes[choice - 1]++;

printf("Thank you for voting!\n");

} else if (choice == 6) {

printf("\n---------- Voting Results ----------\n");

for (i = 0; i < 5; i++) {

printf("%s: %d votes\n", candidates[i], votes[i]);

}

} else if (choice == 7) {

voting = 0;

printf("Exiting the voting system. Goodbye!\n");

} else {

printf("Invalid choice. Please try again.\n");

}

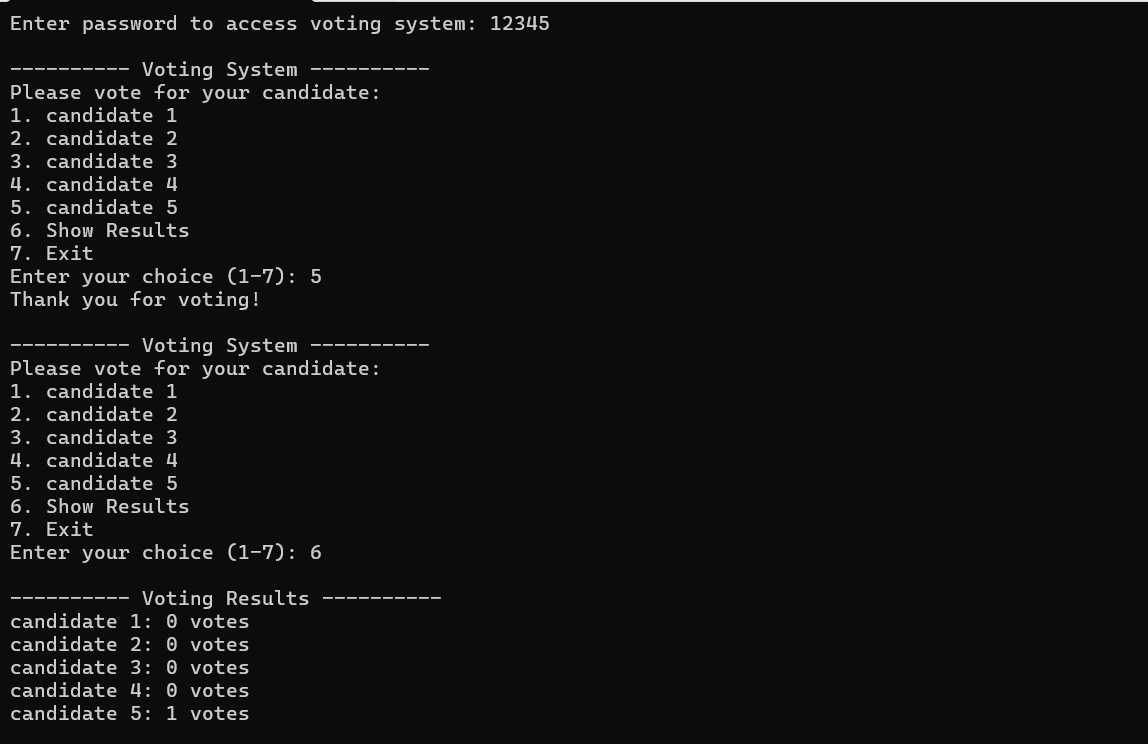
}

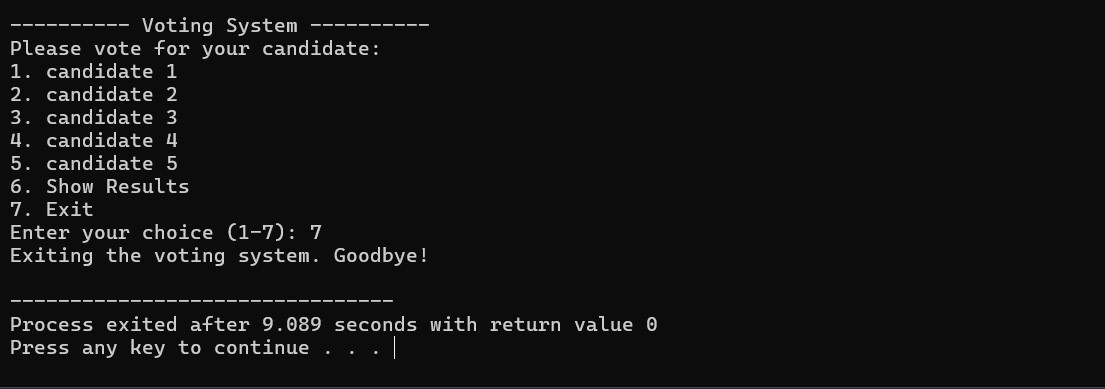
return 0;

}

Output

Login Page: Prompt for password  
Voting Menu: List of candidates  
Results: Display each candidate's votes  
  
Expected Output





Limitations

- No database connectivity; results are lost once program exits.  
- No prevention of multiple voting by same user.  
- Basic password protection only; no encryption.  
- No real user authentication.

Conclusion

This project successfully simulates a simple Voting System with basic security and functionality. It enhances understanding of real-world applications using C programming concepts.

Future Scope

- Adding authentication for multiple users.  
- Preventing double voting.  
- Implementing a graphical user interface (GUI).  
- Adding a database to store votes persistently.

References

- C Programming Language by E. Balagurusamy  
- Online resources and tutorials on C programming  
- MinGW GCC Compiler Documentation