**Simple Voting System**

**Project report submitted in partial fulfillment of the Requirements for the Award of the Degree of**

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

**By**

**Valmeti . Kamal Vaarshneya Reddy – 24kb1a05mk**

**Vannem . Aishwarya – 24kb1a05mr**

**Vuribindi . Lasya – 24kb1a05nu**

**Yelluru . Meghana – 24kb1a05ph**

**Under the Guidance of**

**Sivanraj Samy**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**N.B.K.R.I.S.T**

**Academic Year: 2024–2025**

**Simple Voting System**

**Project report submitted in partial fulfillment of the Requirements for the Award of the Degree of**

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

**By**

**Valmeti . Kamal Vaarshneya Reddy – 24kb1a05mk**

**Vannem . Aishwarya – 24kb1a05mr**

**Vuribindi . Lasya – 24kb1a05nu**

**Yelluru . Meghana – 24kb1a05ph**

**Under the Guidance of**

**Sivanraj Samy**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**N.B.K.R.I.S.T**

**Academic Year: 2024–2025**

**N.B.K.R.I.S.T**

**(AUTONOMOUS)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



###### CERTIFICATE

This is to certify that the project report entitled **Simple Voting System** being submitted by

**Valmeti . Kamal Vaarshneya Reddy – 24kb1a05mk**

**Vannem . Aishwarya – 24kb1a05mr**

**Vuribindi . Lasya – 24kb1a05nu**

**Yelluru . Meghana – 24kb1a05ph**

in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering to the Jawaharlal Nehru Technological University, Ananthapur is a record of bonafied work carried out under my guidance and supervision.

|  |  |
| --- | --- |
| **Sivanraj Samy**  **Designation** | **Dr. HOD Name**  **M.Tech, Ph.D**  **Head of the Department** |

**DECLARATION**

I hereby declare that the dissertation entitled Simple Voting System submitted for the B.Tech Degree is our original work and the dissertation has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles.

Place: vidyanagar Valmeti . Kamal Vaarshneya Reddy

24kb1a05mk

Vannem . Aishwarya

24kb1a05mr

Vuribindi . Lasya

24kb1a05nu

Yelluru . Meghana

24kb1a05ph

Date: 4-05-2025

**Acknowledgement**

The satisfaction that accompanies the successful completion of a project would be

incomplete without he people who made it possible for their constant guidance and

encouragement crowned our efforts with success.

We would like to express our profound sense of gratitude to our project guide Smt. S.

Anusha, Assistant Professor, Department of Computer Science & Engineering,

N.B.K.R.I.S.T affiliated to JNTUA, Ananthapur, Vidyanagar, for her masterful

guidance and the constant encouragement throughout the project. Our sincere appreciations for

her valuable and unmatched services without making this work would have been an unfulfilled dream.

We convey our special thanks to Dr. Y. VENKATA RAMI REDDY, respectable

chairman of N.B.K.R Institute of Science & Technology, for providing excellent infrastructure

in our campus for the completion of the project.

We convey our special thanks to Sri N. RAM KUMAR, respectable correspondent

of N.B.K.R Institute of Science & Technology, for providing excellent infrastructure in our

campus for the completion of the project.

We are grateful to Dr. V. VIJAYA KUMAR REDDY, Director of N.B.K.R Institute of

Science & Technology for allowing us to avail all the facilities in the college.

We express our s1cere gratitude to Dr. A. RAJASEKHAR REDDY, Professor & HOD,

Computer Science & Engineering, for providing exceptional facilities for successful completion

of our project work.

We would like to convey our heartful thanks to Staff Members, Lab Technicians, and

Friends who extended their cooperation in making this project a successful one.

We would like to thank one and all who have helped us, directly and indirectly, to complete

this project successfully.

Abstract

This project presents a voting system in C where candidates are stored using an array and each candidate’s voters are managed with a linked list. The system includes candidate registration, voting, result viewing, and voter detail tracking. This demonstrates data structure applications in real-world systems using C programming.

Table of contents

CHAPTER 1: INTRODUCTION

1.1 Problem Statement

1.2 Scope

1.3 Objective

CHAPTER 2: LITERATURE SURVEY / EXISTING SYSTEM

2.1 Existing System

2.2 Proposed System

CHAPTER 3: SOFTWARE REQUIREMENT ANALYSIS

3.1 Functional Requirements

3.2 Non-Functional Requirements

CHAPTER 4: SOFTWARE DESIGN

4.1 Control Flow

4.2 Modules

CHAPTER 5: CODING

CHAPTER 6: TESTING

CHAPTER 7: OUTPUT SCREENS

CHAPTER 8: CONCLUSION AND FURTHER WORK

CHAPTER 9: REFERENCES

**CHAPTER 1**

**INTRODUCTION**

1.1 **Problem Statement**  
To design a simple digital voting system using C that utilizes arrays and linked lists.

1.2 **Scope**

* CLI-based interface
* No GUI
* Local execution only

1.3 **Objective**

* Use arrays for candidate list
* Use linked lists for voter storage
* Provide complete basic voting functionality

**CHAPTER 2**

**LITERATURE SURVEY / EXISTING SYSTEM**

2.1 **Existing System**  
Most existing voting systems are paper-based or complex digital interfaces. Many do not store voter data or track it structurally for academic-level implementation.

2.2 **Proposed System**  
Our system uses an array for storing candidate info and a linked list for dynamically storing voter data. It is simple, interactive, and demonstrates fundamental C programming.

**CHAPTER 3**

**SOFTWARE REQUIREMENT ANALYSIS**

3.1 **Functional Requirements**

* Register candidates
* Vote
* Show results
* Show voter and candidate info

3.2 **Non-Functional Requirements**

* Easy to use
* Efficient memory handling
* Terminal-based

**CHAPTER 4**

**SOFTWARE DESIGN**

4.1 **Control Flow**

Main Menu  
├── Register Candidate  
├── Vote  
├── Show Results  
├── Show Voter Details  
└── Exit

4.2 **Modules**

* registerCandidate()
* vote()
* showResults()
* showVoters()
* showCandidates()

**CHAPTER 5**

**CODING**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

// Voter node (linked list)

struct Voter {

char name[50];

char phone[20];

char aadhar[20];

struct Voter\* next;

};

// Candidate node (array element)

struct Candidate {

char name[50];

int id;

int voteCount;

struct Voter\* voterHead; // Linked list of voters

};

struct Candidate candidates[100]; // Array of candidates

int candidateCount = 0;

//registration for new candidates

void registerCandidate() {

if (candidateCount >= 100) {

printf("Candidate limit reached.\n"); //max limit

return;

}

printf("Enter candidate name: ");

scanf("%s", candidates[candidateCount].name);

candidates[candidateCount].id = candidateCount + 1;

candidates[candidateCount].voteCount = 0;

candidates[candidateCount].voterHead = NULL;

printf("Candidate '%s' registered with ID %d\n\n", candidates[candidateCount].name, candidates[candidateCount].id);

candidateCount++;

}

//voting a candidate

void vote() {

if (candidateCount == 0) {

printf("No candidates registered yet.\n\n");

return;

}

printf("Enter your name: "); //name

char name[50];

scanf("%s", name);

printf("Enter your phone number: "); // number

char phone[20];

scanf("%s", phone);

printf("Enter your Aadhar number: "); // aaadhar

char aadhar[20];

scanf("%s", aadhar);

printf("\nCandidates:\n");

for (int i = 0; i < candidateCount; i++) {

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

}

printf("Enter candidate ID to vote for: ");

int voteID;

scanf("%d", &voteID);

if (voteID < 1 || voteID > candidateCount) {

printf("Invalid candidate ID.\n\n");

return;

}

// Create new voter node

struct Voter\* newVoter = (struct Voter\*)malloc(sizeof(struct Voter));

strcpy(newVoter->name, name);

strcpy(newVoter->phone, phone);

strcpy(newVoter->aadhar, aadhar);

newVoter->next = NULL;

// Add voter to the linked list of chosen candidate

struct Candidate\* chosen = &candidates[voteID - 1];

newVoter->next = chosen->voterHead;

chosen->voterHead = newVoter;

chosen->voteCount++;

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

}

// Function to show vote results

void showResults() {

if (candidateCount == 0) {

printf("No candidates registered.\n\n");

return;

}

printf("---- Vote Count ----\n");

for (int i = 0; i < candidateCount; i++) {

printf("%s - %d votes\n", candidates[i].name, candidates[i].voteCount);

}

printf("\n");

}

// Function to show voter details per candidate

void showVoters() {

if (candidateCount == 0) {

printf("No candidates registered.\n\n");

return;

}

for (int i = 0; i < candidateCount; i++) {

printf("\nVoters for %s:\n", candidates[i].name);

struct Voter\* current = candidates[i].voterHead;

if (current == NULL) {

printf(" No voters yet.\n");

}

while (current != NULL) {

printf(" Name: %s | Phone: %s | Aadhar: %s\n", current->name, current->phone, current->aadhar);

current = current->next;

}

}

printf("\n");

}

// Function to show all registered candidate details

void showCandidates() {

if (candidateCount == 0) {

printf("No candidates registered.\n\n");

return;

}

printf("---- Candidate List ----\n");

for (int i = 0; i < candidateCount; i++) {

printf("ID: %d | Name: %s\n", candidates[i].id, candidates[i].name);

}

printf("\n");

}

// Main driver function

int main() {

int choice;

while (1) {

printf("===== Simple Voting System =====\n");

printf("1. Register Candidate\n");

printf("2. Vote\n");

printf("3. Show Vote Results\n");

printf("4. Show Voter Details\n");

printf("5. Show Candidate List\n");

printf("6. Exit\n");

printf("Choose an option: ");

scanf("%d", &choice);

switch (choice) {

case 1:

registerCandidate();

break;

case 2:

vote();

break;

case 3:

showResults();

break;

case 4:

showVoters();

break;

case 5:

showCandidates();

break;

case 6:

printf("Exiting...\n");

return 0;

default:

printf("Invalid option. Try again.\n");

}

}

return 0;

}

**CHAPTER 6**

**TESTING**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Input** | **Expected Output** | **Result** |
| Valid Vote | Name + Vote ID | Vote added | Pass |
| Invalid Candidate | ID = 999 | Error message | Pass |

**CHAPTER 7**

**OUTPUT SCREENS**

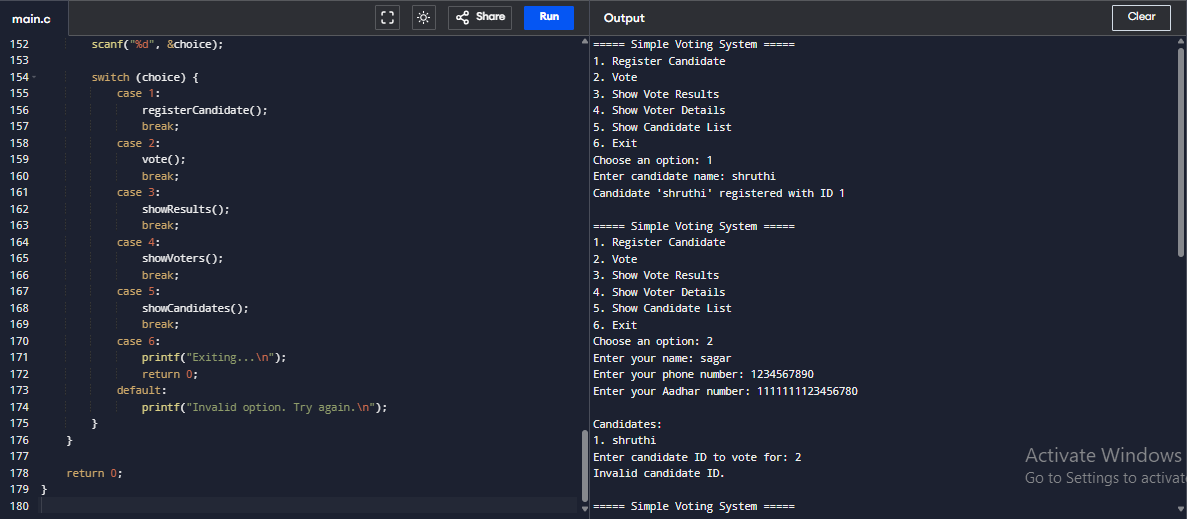


Fig 7.1

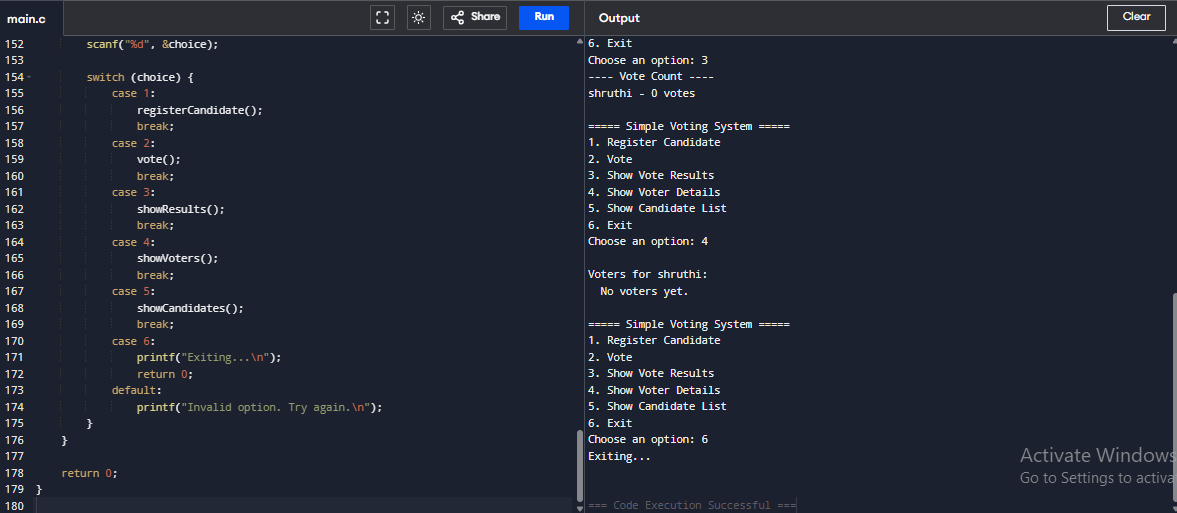


Fig 7.2

**CHAPTER 8**

**CONCLUSION AND FURTHER WORK**

We successfully built a basic voting system using C. This project reinforces the importance and application of arrays and linked lists. Future upgrades include GUI, duplicate vote checks, file storage, and result analytics.

**CHAPTER 9**

**REFERENCES**

1. The C Programming Language by Dennis Ritchie
2. GeeksforGeeks.org
3. Lecture notes & documentation