CODE:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

// Max number of voters and candidates

#define MAX\_VOTERS 100

#define MAX\_CANDIDATES 100

// Structure for candidate (Linked List)

typedef struct Candidate {

int id;

char name[50];

struct Candidate\* next;

} Candidate;

// Structure to track voters

typedef struct Voter {

int id;

int hasVoted;

} Voter;

Candidate\* head = NULL; // Head of candidate list

int voteCount[MAX\_CANDIDATES]; // Vote count per candidate

Voter voters[MAX\_VOTERS]; // Voter list

int totalVoters = 0;

// Function to add a candidate

void addCandidate(int id, const char\* name) {

Candidate\* newNode = (Candidate\*)malloc(sizeof(Candidate));

newNode->id = id;

strcpy(newNode->name, name);

newNode->next = head;

head = newNode;

}

// Function to register voters

void registerVoters(int count) {

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

voters[i].id = i;

voters[i].hasVoted = 0;

}

totalVoters = count;

}

// Function to check if voter is eligible

int isVoterEligible(int voterID) {

if (voterID < 0 || voterID >= totalVoters) {

return 0;

}

if (voters[voterID].hasVoted) {

return 0;

}

return 1;

}

// Display all candidates

void displayCandidates() {

Candidate\* temp = head;

printf("\nList of Candidates:\n");

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

while (temp != NULL) {

printf("ID: %d\tName: %s\n", temp->id, temp->name);

temp = temp->next;

}

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

}

// Cast a vote

void vote(int voterID, int candidateID) {

Candidate\* temp = head;

while (temp != NULL) {

if (temp->id == candidateID) {

if (!isVoterEligible(voterID)) {

printf("Error: Invalid or already used Voter ID.\n");

return;

}

voteCount[candidateID]++;

voters[voterID].hasVoted = 1;

printf("Vote cast successfully for %s.\n", temp->name);

return;

}

temp = temp->next;

}

printf("Error: Invalid Candidate ID.\n");

}

// Show vote count for each candidate

void displayResults() {

Candidate\* temp = head;

printf("\nVoting Results:\n");

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

while (temp != NULL) {

printf("Candidate: %s\tVotes: %d\n", temp->name, voteCount[temp->id]);

temp = temp->next;

}

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

}

// Display the winner

void displayWinner() {

Candidate\* temp = head;

int maxVotes = -1;

int winnerID = -1;

while (temp != NULL) {

if (voteCount[temp->id] > maxVotes) {

maxVotes = voteCount[temp->id];

winnerID = temp->id;

}

temp = temp->next;

}

temp = head;

while (temp != NULL) {

if (temp->id == winnerID) {

printf("\n🏆 Winner: %s with %d votes!\n", temp->name, maxVotes);

return;

}

temp = temp->next;

}

printf("No winner found.\n");

}

// Display voters and status

void displayVoterStatus() {

printf("\nVoter Participation:\n");

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

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

printf("Voter ID: %d\tStatus: %s\n", voters[i].id, voters[i].hasVoted ? "Voted" : "Not Voted");

}

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

}

// Clear memory

void freeCandidates() {

Candidate\* temp;

while (head != NULL) {

temp = head;

head = head->next;

free(temp);

}

}

// Main menu

void menu() {

int choice;

int voterID, candidateID;

do {

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

printf("1. Display Candidates\n");

printf("2. Vote\n");

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

printf("4. Show Winner\n");

printf("5. Voter Status\n");

printf("6. Exit\n");

printf("Enter choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

displayCandidates();

break;

case 2:

printf("Enter your Voter ID: ");

scanf("%d", &voterID);

printf("Enter Candidate ID to vote: ");

scanf("%d", &candidateID);

vote(voterID, candidateID);

break;

case 3:

displayResults();

break;

case 4:

displayWinner();

break;

case 5:

displayVoterStatus();

break;

case 6:

printf("Exiting... Thank you!\n");

break;

default:

printf("Invalid choice! Try again.\n");

}

} while (choice != 6);

}

// Main function

int main() {

int numCandidates, numVoters;

char name[50];

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

// Register candidates

printf("Enter number of candidates: ");

scanf("%d", &numCandidates);

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

printf("Enter name for Candidate %d: ", i);

scanf("%s", name);

addCandidate(i, name);

}

// Register voters

printf("Enter number of voters: ");

scanf("%d", &numVoters);

registerVoters(numVoters);

// Start menu loop

menu();

// Cleanup

freeCandidates();

return 0;

}