GOMENDRA MULTIPLE COLLEGE

Birtamode-04, Jhapa

Affiliated to Purbanchal University

Program: BCA-IT

Semester: First

Faculty of Science and Technology

Project report on

VOTING MANAGEMENT SYSTEM

Submitted by :	**Under the supervision of**
Kewal Dahal()	**Assistant lecturer:**
Prakhar Katwal()	**Mr. Nabin Prasain**
Bipin Giri()	

Acknowledgement

In the pursuit of our Bachelor in Computer Application degree, we embarked on an enriching journey during the first semester, tasked with the creation of a significant project - the "Voting Management System." This endeavor not only served as a pivotal component of our academic journey but also as a testament to our collaborative spirit and our introduction to the practical intricacies of software development.

This report serves as a comprehensive documentation of the system's requirements and design, meticulously articulated in natural language to provide a lucid understanding of each facet. We would like to extend our heartfelt gratitude to Gomendra Multiple College for providing us with the platform to undertake and complete this project successfully.

Our profound appreciation goes to our mentor and guide, Mr. Nabin Prasain, whose unwavering support and encouragement propelled us through the challenging phases of the project. His guidance, insightful suggestions, and willingness to navigate us through obstacles were instrumental in the project's triumph. We extend our thanks to the entire faculty of the Department of Computer Application for their continuous support and invaluable guidance, which nurtured our growth and learning throughout this endeavor.

Lastly, we want to express our gratitude to our friends, whose unwavering support, constant encouragement, and invaluable suggestions were a source of motivation and inspiration throughout the project's duration. This project has not only broadened our horizons but has also affirmed the importance of collaboration, dedication, and innovation in the field of computer science. We look forward to applying the knowledge and experiences gained here in our future endeavors.

Table of Contents

Introduction of Project

The "Voting Management System" is a sophisticated software solution developed as part of the BCA 1st-semester project to facilitate efficient and transparent election processes. This system is designed to handle various aspects of elections, from authenticating voters to recording and counting votes for multiple candidates. It addresses the core challenges of conducting elections by ensuring data integrity, preventing double voting, and protecting voter privacy.

Key Features:

This project offers a range of key features, including:

- Voter Authentication: Voters are required to enter their unique Voter ID for authentication, which is cross-referenced with a database to validate their eligibility.
- Candidate Selection: The system provides a user-friendly interface to display a list of candidates and their respective parties, allowing voters to make informed choices.
- Vote Recording: Voters can cast their votes for their preferred candidates or select a "None of These" option.
- Vote Count: The system maintains an up-to-date count of votes for each candidate and spoiled votes, ensuring transparency in the election process.
- Leading Candidate Determination: It can identify the leading candidate based on the highest number of votes received.
- Vote Details Logging: The system logs vote details, including the voter's choice and timestamp, for auditing and transparency purposes.

This project aims to streamline and modernize the election process, making it more accessible and efficient while maintaining the highest standards of fairness and security. It is a valuable tool for anyone involved in election management and provides an excellent foundation for future enhancements and real-world applications.

Objectives of the Project

Project Overview:

The "Voting Management System" is a dynamic software solution developed as part of the BCA 1st-semester project. It seeks to modernize and streamline the election process by providing secure voter authentication, efficient vote casting, and real-time result tracking. This system aims to uphold the principles of transparency, data integrity, and privacy protection, ensuring a fair and accurate electoral experience.

Objectives:

1. Authentication Assurance: Develop a secure authentication mechanism to validate voter eligibility.
2. Candidate Management: Enable administrators to manage candidate information seamlessly.
3. Efficient Voting: Implement an intuitive interface for hassle-free vote casting.
4. Vote Tallying: Create a robust vote counting system, including tracking spoiled votes
5. Leading Candidate Identification: Offer real-time updates on the leading candidate.
6. Data Logging: Maintain a transparent and auditable record of the election process.
7. Double Voting Prevention: Implement safeguards to prevent multiple votes from a single voter.
8. User-Friendly Interface: Ensure an accessible and user-friendly voting experience.
9. Data Integrity: Protect the integrity of election data through proper file handling
10. Privacy Assurance: Safeguard voter privacy by clearing screens and protecting sensitive information.

Importance of Voting Management System

The Voting Management System holds a pivotal role in the functioning of democratic society. It serves as the linchpin that upholds the principles of democracy, ensuring the integrity of elections and the fair representation of citizens' voices. By offering a secure and efficient means of authentication, it instills confidence in voters, assuring them that their votes will be counted accurately.

This system simplifies the voting process, making it accessible to a broader range of citizens. It also plays a critical role in maintaining transparency throughout the election cycle. By meticulously logging voter choices and outcomes, it creates an audit trail that can be reviewed for accountability, minimizing the potential for fraud or manipulation.

Furthermore, the Voting Management System contributes to the strengthening of democratic institutions, fostering public trust in the electoral process. It stands as a symbol of a just and inclusive society, where every individual's vote matters and the collective will of the people is reflected accurately. In essence, it is not merely a technological tool but a cornerstone of democracy itself, ensuring that the voice of the people remains at the heart of the nation's decision-making processes.

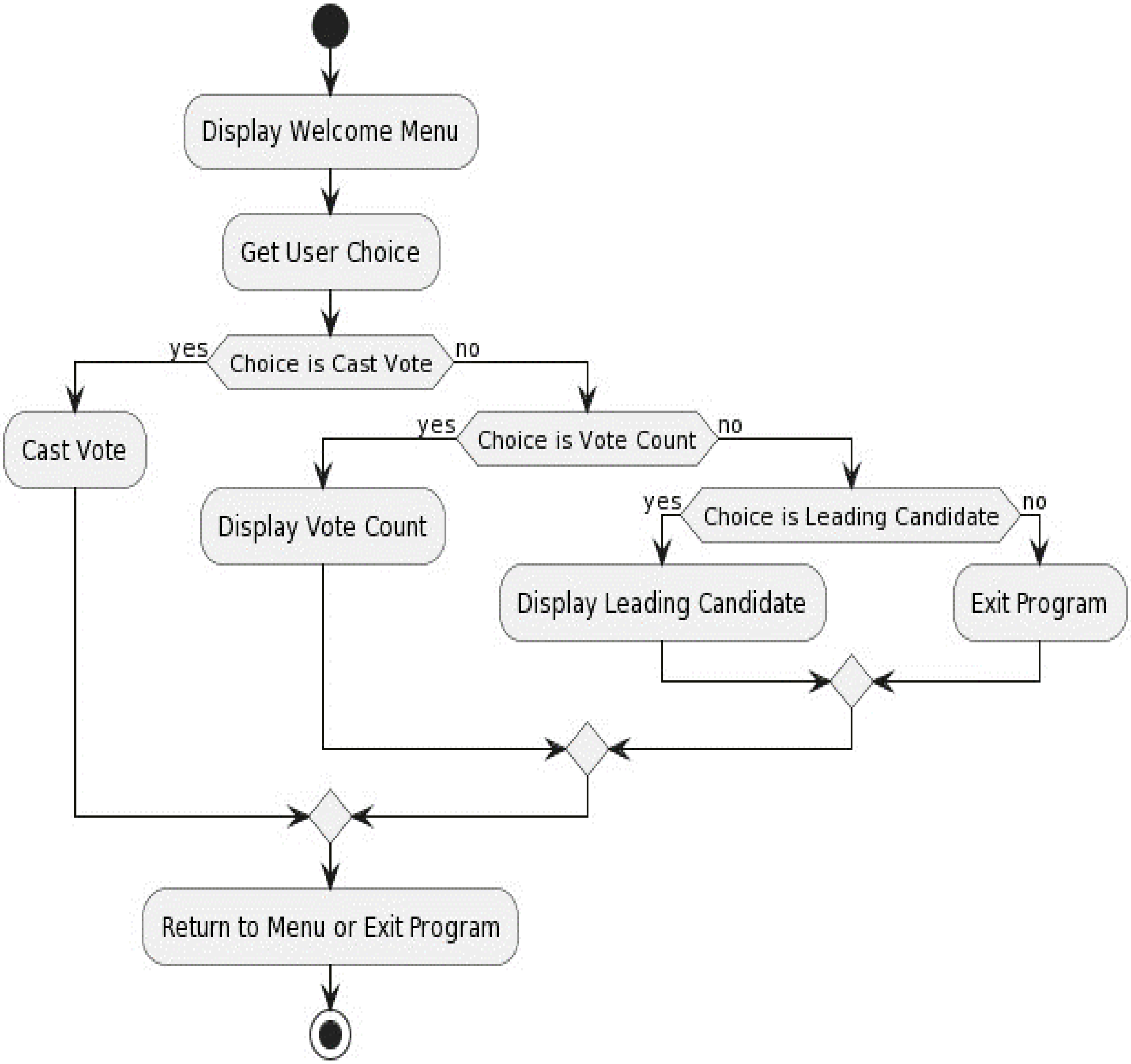
System Requirements

Minimum System Requirements:

1. Operating System: Windows, Linux, or macOS.
2. Processor: Dual-core processor (e.g., Intel Core i3 or equivalent).
3. Memory (RAM): 2GB of RAM or higher.
4. Storage: 50MB of free disk space for program installation and data storage.
5. Input Devices: Standard keyboard and mouse for user interactions.
6. Display: Monitor with a minimum resolution of 1024x768 pixels.
7. Software Dependencies: A C compiler (e.g., GCC for Linux or MinGW for Windows) and standard C libraries.

These minimum requirements should allow you to run the Voting Management System project smoothly without encountering any major issues. However, for optimal performance and to accommodate larger datasets, it's recommended to use a system that exceeds these minimum specifications, especially in terms of RAM and processor capabilities.

Flowchart


Display Welcome Menu
Get User Choice
yes
Choice is Cast Vote
no
Cast Vote
yes
Choice is Vote Count
no
Display Vote Count
yes
Choice is Leading Candidate
no
Display Leading Candidate
Exit Program
Return to Menu or Exit Program

Source Code

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <unistd.h>
#define CANDIDATE_COUNT 4
#define CANDIDATE1 "Kewal Dahal"
#define CANDIDATE2 "Bipin Giri"
#define CANDIDATE3 "Prakhar Katwal"
#define CANDIDATE4 "Sandesh Bharti"
struct Candidate {
char name[50];
char party[50];
};
typedef struct Candidate Candidate;
Candidate candidates[CANDIDATE_COUNT];
int votesCount[CANDIDATE_COUNT] = {0};
int spoiledVotes = 0;
// Function to load candidate names and parties into the candidates array
void loadCandidates() {
strcpy(candidates[0].name, CANDIDATE1);
strcpy(candidates[1].name, CANDIDATE2);
strcpy(candidates[2].name, CANDIDATE3);
strcpy(candidates[3].name, CANDIDATE4);
strcpy(candidates[0].party, "Party A");
strcpy(candidates[1].party, "Party B");
strcpy(candidates[2].party, "Party C");
strcpy(candidates[3].party, "Party D");
}
// Function to clear the console screen (system-dependent)
void clearScreen() {
system("cls"); // For Windows. Use "clear" for Linux/macOS.
}
// Function to display the list of candidates
void displayCandidates() {
printf("\n\n ### List of Candidates ###\n");
for (int i = 0; i < CANDIDATE_COUNT; i++) {
printf(" %d. %s (%s)\n", i + 1, candidates[i].name, candidates[i].party);
}
printf(" %d. None of These\n", CANDIDATE_COUNT + 1);
}
// Function to get the current time as a string
char *getCurrentTime() {
time_t now;
time(&now);
return ctime(&now);
}
// Function to check if a voter has already voted
int isVoterAlreadyVoted(const char *voterID) {
FILE *file = fopen("voted.txt", "r");
```

```
if (file == NULL) {
printf("Error opening voted ID file.\n");
return 0;
}

char line[100];
while (fgets(line, sizeof(line), file)) {
if (line[strlen(line) - 1] == '\n') {
line[strlen(line) - 1] = '\0';
}
if (strstr(line, voterID) != NULL) {
fclose(file);
return 1;
}
}
fclose(file);
return 0;
}
// Function to check if a voter is valid
int isValidVoter(const char *voterID) {
FILE *file = fopen("votingid.txt", "r");
if (file == NULL) {
printf("Error opening voter ID file.\n");
return 0;
}
char line[100];
while (fgets(line, sizeof(line), file)) {
```

```
if (line[strlen(line) - 1] == '\n') {
line[strlen(line) - 1] = '\0';
}
if (strcmp(line, voterID) == 0) {
fclose(file);
return 1;
}
}
fclose(file);
return 0;
}
// Function to save vote details to a file
void saveVoteDetailsToFile(const char *voterID, int candidateIndex) {
FILE *file = fopen("voting_details.txt", "a");
if (file == NULL) {
printf("Error opening file for writing.\n");
return;
}
fprintf(file, "%s - %s\n", candidates[candidateIndex].name, getCurrentTime());
fclose(file);
}
// Function to update vote count in the voting_details.txt file
void updateVoteCountToFile(const char *candidateName) {
FILE *file = fopen("voting_details.txt", "a");
if (file == NULL) {
printf("Error opening file for writing.\n");
return;
```

```
}

for (int i = 0; i < CANDIDATE_COUNT; i++) {
if (strcmp(candidates[i].name, candidateName) == 0) {
fprintf(file, "%s - %d votes\n", candidateName, votesCount[i]);
break;
}
}
fclose(file);
}
// Function for casting a vote
void castVote() {
clearScreen();
char voterID[20];
printf("\n\n ### Please enter your Voter ID to cast your vote: ");
scanf("%s", voterID);
if (isVoterAlreadyVoted(voterID)) {
printf("\n You have already cast your vote!\n");
sleep(2);
return;
}
if (!isValidVoter(voterID)) {
printf("\n Invalid Voter ID! Vote casting failed.\n");
sleep(2);
return;
}
displayCandidates();
```

```
int choice;
printf("\n Input your choice (1 - %d): ", CANDIDATE_COUNT + 1);
scanf("%d", &choice);
if (choice >= 1 && choice <= CANDIDATE_COUNT) {
votesCount[choice - 1]++;
printf("\n Thanks for voting!\n");
FILE *votedFile = fopen("voted.txt", "a");
if (votedFile == NULL) {
printf("Error opening voted ID file for writing.\n");
sleep(2);
return;
}
fprintf(votedFile, "%s - Voted for %s (%s)\n", voterID,
candidates[choice - 1].name, getCurrentTime());
fclose(votedFile);
saveVoteDetailsToFile(voterID, choice - 1);
updateVoteCountToFile(candidates[choice - 1].name);
sleep(2);
} else if (choice == CANDIDATE_COUNT + 1) {
spoiledVotes++;
printf("\n Your vote has been marked as spoiled.\n");
sleep(2);
} else {
printf("\n Error: Wrong Choice! Please retry.\n");
sleep(2);
}
```

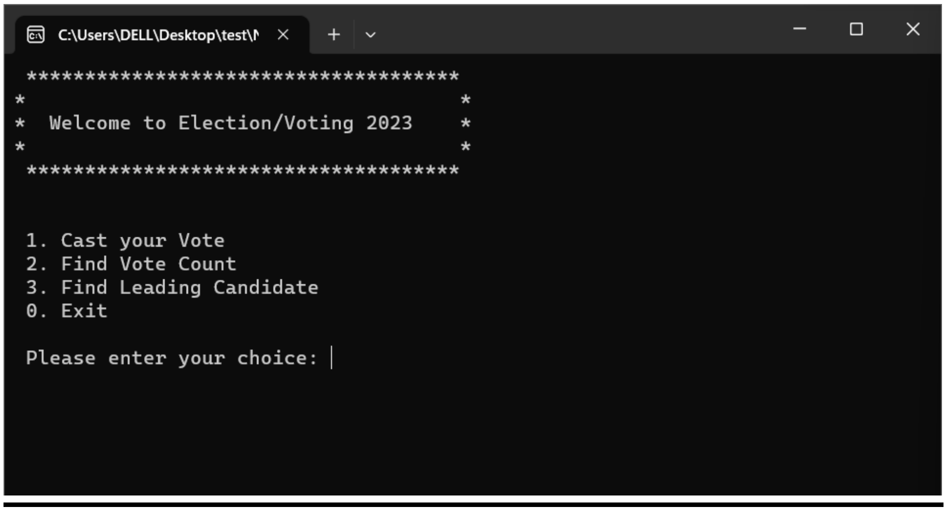
```
}
// Function to pause the screen until Enter key is pressed
void waitForEnter() {
printf("\nPress Enter to continue...");
while (getchar() != '\n');
getchar();  // Consume the Enter key
}
// Function to print voting statistics
void printVotesCount() {
clearScreen();
printf("\n\n ##### Voting Statistics ####\n");
for (int i = 0; i < CANDIDATE_COUNT; i++) {
printf(" %s - %d votes\n", candidates[i].name, votesCount[i]);
}
printf(" Spoiled Votes - %d\n", spoiledVotes);
// Display the contents of the "voting_details.txt" file
printf("\n\n ##### Voting Details ####\n");
FILE *votingDetailsFile = fopen("voting_details.txt", "r");
if (votingDetailsFile == NULL) {
printf("Error opening voting_details.txt file.\n");
} else {
char line[100];
while (fgets(line, sizeof(line), votingDetailsFile)) {
printf("%s", line);
}
fclose(votingDetailsFile);
}
```

```
waitForEnter();  // Wait for Enter key
}
// Function to find the leading candidate
void getLeadingCandidate() {
clearScreen();
printf("\n\n  #### Leading Candidate ####\n\n");
int maxVotes = votesCount[0];
int leadingCandidateIndex = 0;
for (int i = 1; i < CANDIDATE_COUNT; i++) {
if (votesCount[i] > maxVotes) {
maxVotes = votesCount[i];
leadingCandidateIndex = i;
}
}
if (maxVotes > 0) {
printf("[%s]\n", candidates[leadingCandidateIndex].name);
} else {
printf("----- Warning !!! No-win situation ----\n");
}
sleep(2);
}
// Function to display a loading animation
void displayLoading(int delay) {
printf("Loading");
fflush(stdout);
for (int i = 0; i < 3; i++) {
usleep(delay);
```

```
printf(".");
fflush(stdout);
}
usleep(delay);
printf("\n");
}
int main() {
int choice;
loadCandidates();
do {
clearScreen();
printf(" ****************************************\n");
printf("*                                        *\n");
printf("*          Welcome to Election/Voting 2023   *\n");
printf("*                                         *\n");
printf(" ****************************************\n");
printf("\n\n 1. Cast your Vote");
printf("\n 2. Find Vote Count");
printf("\n 3. Find Leading Candidate");
printf("\n 0. Exit");
printf("\n\n Please enter your choice: ");
scanf("%d", &choice);
switch (choice) {
case 1:
displayLoading(300000);
castVote();
break;
case 2:
displayLoading(200000);
printVotesCount();
break;
case 3:
displayLoading(200000);
getLeadingCandidate();
break;
case 0:
printf("\n Exiting the program. Goodbye!\n");
break;
default:
printf("\n Error: Invalid Choice\n");
}
} while (choice != 0);
return 0;
}
```

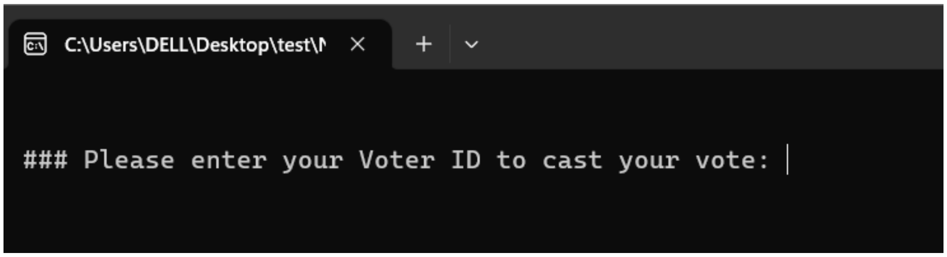
Output/ Application Overview Windows

Menu window



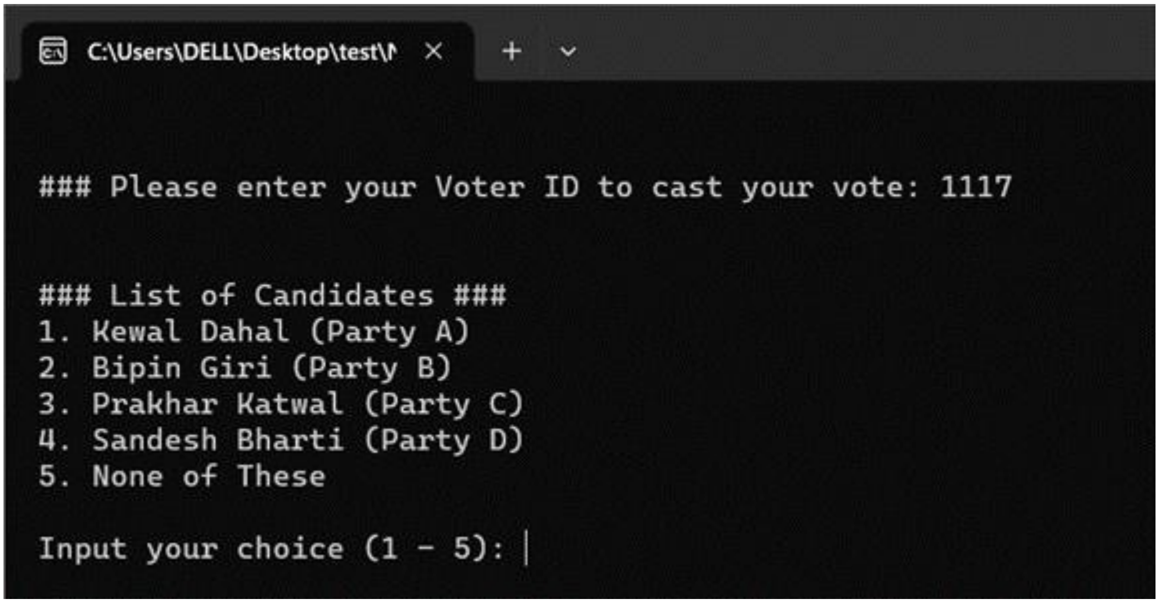
```
 **************************************
*                                      *
*  Welcome to Election/Voting 2023     *
*                                      *
 **************************************


 1. Cast your Vote
 2. Find Vote Count
 3. Find Leading Candidate
 0. Exit

 Please enter your choice: 
```

Authentication window to cast vote



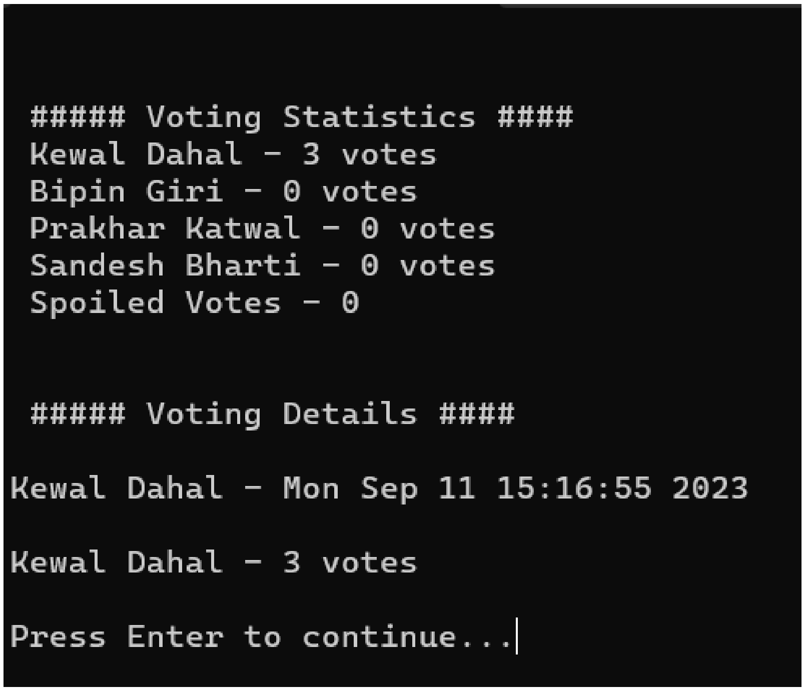
```
### Please enter your Voter ID to cast your vote: 
```

Voting window after authentication with candidates names and party



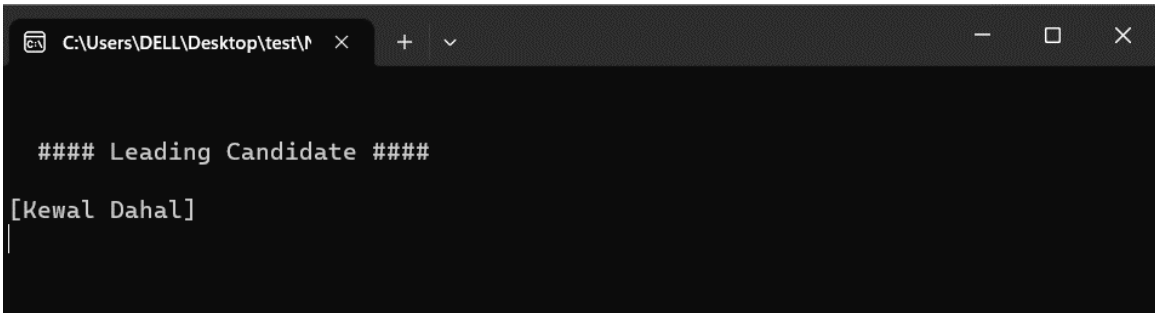
```
### Please enter your Voter ID to cast your vote: 1117


### List of Candidates ###
1. Kewal Dahal (Party A)
2. Bipin Giri (Party B)
3. Prakhar Katwal (Party C)
4. Sandesh Bharti (Party D)
5. None of These

Input your choice (1 - 5): 
```

Voting statistics window



```
 ##### Voting Statistics ####
 Kewal Dahal - 3 votes
 Bipin Giri - 0 votes
 Prakhar Katwal - 0 votes
 Sandesh Bharti - 0 votes
 Spoiled Votes - 0


 ##### Voting Details ####

Kewal Dahal - Mon Sep 11 15:16:55 2023

Kewal Dahal - 3 votes

Press Enter to continue...
```

Leading candidate window



```
  #### Leading Candidate ####

[Kewal Dahal]
```

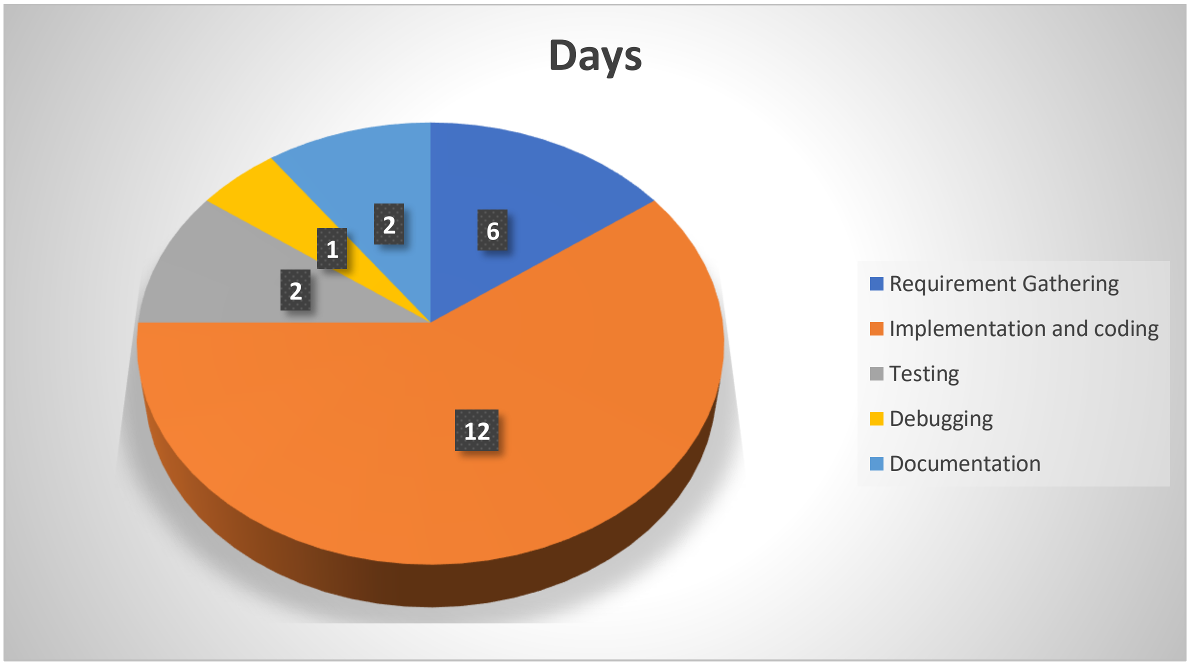
Future Use and Implementation

The Voting Management System project holds significant promise for future use and expansion. One potential avenue for its application is in the context of electronic voting systems, which are increasingly gaining traction globally. By enhancing the system to support online and remote voting, it can provide citizens with a secure and convenient means to cast their votes from the comfort of their homes or via mobile devices. This modernization can be particularly impactful during situations like pandemics, where ensuring safe voting practices is crucial. Additionally, the system could find application in proxy voting for overseas citizens, enabling them to participate in their home country's elections regardless of their geographical location.

Furthermore, the project can serve as a foundation for electoral reforms and enhancements. It can be adapted to implement ranked-choice voting systems, where voters prioritize multiple candidates in order of preference. This promotes a more nuanced and representative democracy by accommodating a wider spectrum of political ideologies. Additionally, by incorporating advanced data analytics, the system can provide valuable insights into voter demographics and preferences, aiding political campaigns and policymakers in making informed decisions. Moreover, the introduction of biometric authentication methods can further strengthen the security of the system, ensuring that only eligible voters can participate.

Looking beyond national elections, the Voting Management System can be tailored for use in various organizational and institutional voting scenarios. This includes student council elections in educational institutions, board member elections in corporations, and decision-making processes in non-profit organizations. Its adaptability and scalability make it a versatile tool that can be customized to suit the unique requirements of different voting scenarios. As technology continues to advance and societies evolve, the Voting Management System remains a dynamic platform ready to evolve and meet the ever-changing demands of modern democracy and organizational decision-making.

Feasibility Information / Group Information



Student Name: Kewal Dahal

S.N	Job description	Date from	Date to	Reference
1	Requirement Gathering	10th August	13th August	
2	Coding	14th August	24th August	
3	Testing	25th August	26th August	
4	Debugging	27th August	28th August	Gathered together and had discissions about project

Student Name: Bipin Giri

S.N	Job description	Date from	Date to	Reference
1	Requirement Gathering	10th August	13th August	
2	Testing	25th August	26th August	
3	Documentation	29th August	31st August	We both equally divided the documentation

Student Name: Prakhar Katwal

S.N	Job description	Date from	Date to	Reference
1	Coding	22nd August	24th August	
3	Testing	25th August	26th August	
4	Documentation	31th August	2nd September	We both equally divided the documentation

Conclusion

We hope that the experience we gained while making the “Voting Management System” will be beneficial for us in the coming days. As we are only beginners, there are a lot of areas where we need to improve and this project has helped us in a big way. We are thankful to the management of our college for providing the necessary resources. Without their cooperation, it would not have been possible to complete the project. We would like to express our gratitude to all friends and teachers who helped us by providing valuable suggestions to make the project better. The mistakes we made during the making of this project have taught us various new things about C and implement Logic. We hope we can create better programs in the Future or Upgrade this program to do Other Additional Works. Finally thank you to all individuals who have assisted and encouraged us in the development of this project

Bibliography

To complete this project we took help of following Sites , Books and Videos.

Websites:

- github.com/cognitive-ninja/Mini-Voting-System
- stackoverflow.com/questions/840501/how-do-function-pointers-in-c-work
- stackoverflow.com/questions/41857146/how-to-use-array-and-file-handling-in-c
- chat.openai.com
- www.plantuml.com/plantuml/uml

Books:

- C Programming A Modern Approach(Author: K.N. King)
- The C Programming Language(Author: Brian W. Kernighan)

Video Links

- youtube.com/watch?v=4dX_hQ7A_8M
- youtube.com/watch?v=irqbmMNs2Bo
- youtube.com/watch?v=RhLSYjDXB4Q
- youtube.com/watch?v=fltaqGek-oA