

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI – 590018**



PROJECT REPORT ON

“Voting Management System”

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE & ENGINEERING

FILE STRUCTURE LABORATORY WITH MINI PROJECT [18ISL67]

Submitted by

Hrithik N R – 4JK20IS020

Abhiram K R – 4JK20IS002

Under the guidance of

Prof Divya P

Assistant Professor

**Department of Information Science &
Engineering**

Prof Sivapuram Jayasri

Assistant Professor

**Department of Information Science &
Engineering**



DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

A.J. INSTITUTE OF ENGINEERING & TECHNOLOGY

NH-66, KOTTARA CHOWKI, MANGALURU – 575006

2022-2023

A. J. INSTITUTE OF ENGINEERING & TECHNOLOGY

NH – 66, Kottara Chowki, Mangaluru - 575006

A Unit of Laxmi Memorial Education Trust (R)

(Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING



CERTIFICATE

Certify that the Project entitled “**VOTING MANAGEMENT SYSTEM**” is carried out by Mr. **HRITHIK N R**, USN: **4JK20IS020**, and Mr. **ABHIRAM K R**, USN: **4JK20IS002**, Students of sixth semester B.E. Information Science & Engineering, and submitted as a part of the course **FILE STRUCTURE LABORATORY WITH MINI PROJECT [18ISL67]** during the academic year 2022-2023.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The report has been approved as it satisfies the academic requirements in respect of File Structure Project prescribed for the said Degree.

Prof Divya P
Project Guide

Prof Sivapuram Jayasri
Project Guide

Dr Suresha D
Head of the Department

Dr Shantharama Rai C
Principal

Examiners

Signature with Date

- 1.
- 2.

ACKNOWLEDGEMENT

First and foremost, we thank our parents for what we are and where we are today, without whose hard work and sacrifice we would not be here today.

We deem it a privilege to place on record the deep sense of gratitude to our Project Guide **Prof. Divya P, Assistant Professor**, Department of Information Science and Engineering, and **Prof. Sivapuram Jayasri, Assistant Professor**, Department of Information Science and Engineering, who always stood behind us and supported in each step of the project work.

We are grateful to **Dr. Suresha D**, Head of the Department, Information Science and Engineering for his support and encouragement.

We are indebted to our respected Principal **Dr. Shantharama Rai. C**, beloved Vice President **Mr. Prashanth Shetty** and the management of **A. J. Institute of Engineering and Technology, Mangaluru** for providing all the facilities that helped us in timely completion of this project report.

Finally, we would like to thank all the teaching and non-teaching staff of Department of Information Science and Engineering for their valuable help and support.

Hrithik N R - 4JK20IS020

Abhiram K R - 4JK20IS002

ABSTRACT

Voting Management System is an online voting system where a voter can use his/her voting right online without any difficulty. A unique user id is given to the voter, by using that id he/she can cast his/her vote. If conditions are wrong then that entry will be discarded. The voting management system is an essential tool that ensures the smooth running of elections. This system requires efficient file structuring to manage the vast amounts of data generated during the electoral process. The file structure consists of different files, such as the voter file, candidate file, ballot file, and election result file. These files store all the necessary information, such as voter details, candidate details, and voting results, in an organized manner. The voting management system's file structure also includes different access methods, such as sequential access and direct access, to retrieve information efficiently. The system's file structure is designed to ensure data integrity, security, and privacy. Effective file structuring is critical to the success of any voting management system, as it facilitates the retrieval of accurate and timely information during and after the electoral process.

TABLE OF CONTENTS

Chapter No.	Content	Page No.
	Acknowledgement	i
	Abstract	ii
	Table of Contents	iii
	List of Figures	iv
1	INTRODUCTION	01
	1.1 About the Project	01
	1.2 Scope of Project	02
	1.3 Purpose of Project	02
2	REQUIREMENT SPECIFICATION	03
	2.1 Software Requirements	03
	2.2 Hardware Requirements	03
	2.3 Functional Requirement	03
3	SYSTEM DESIGN	05
	3.1 Algorithm	05
	3.2 Flowchart	06
	3.2 Use Case Diagram	08
4	IMPLEMENTATION	10
	4.1 Details of the Language	10
	4.2 User defined functions	10
5	RESULTS	17
6	CONCLUSION AND FUTURE WORKS	25
	REFERENCES	

LISTS OF FIGURES

Figure No.	Figure Name	Page No.
3.2	Flowchart Diagram	6
3.3	Use Case Diagram	8
5.1.1	Home Page	17
5.1.2	Admin Login Page	17
5.1.3	Admin Main Page	18
5.1.4	View Result Page	18
5.1.5	Candidate Enrolment Page	19
5.1.6	Delete All Voters Page	19
5.1.7	Reset page	20
5.1.8	Add New Voters Page	20
5.1.9	Start Voting Page	21
5.2.0	Give Vote Page	21
5.2.1	Already Voted Page	22
5.2.2	Age Limit Page	22
5.2.3	Delete Voter Page	23
5.2.4	Voters Text File	23
5.2.5	Votes Text File	24
5.2.6	Candidates Text File	24

CHAPTER 1

INTRODUCTION

A voting management system is a comprehensive software solution designed to streamline and enhance the entire process of conducting elections. It serves as a centralized platform for managing voter registration, ballot creation, voter authentication, vote counting, and result tabulation. This system aims to ensure transparency, accuracy, and efficiency in electoral processes. The voting management system provides numerous benefits over traditional paper-based voting methods. It eliminates the need for manual data entry, reduces the chances of human errors, and enhances the security of the voting process. The system can authenticate voters through various means, such as biometric identification or unique voter IDs, to prevent fraud and ensure that only eligible individuals cast their votes. Moreover, the system allows for easy monitoring and auditing of the election process, as it maintains a digital record of all activities. It enables officials to track the progress of voting, manage polling stations, and generate real-time reports. This promotes accountability and facilitates prompt response to any issues or irregularities that may arise. In summary, a voting management system is a sophisticated software solution that revolutionizes the way elections are conducted, offering efficiency, security, and transparency in the electoral process.

1.1 About the Project

- A voting management system is a software solution designed to streamline and enhance the voting process in democratic societies. The system aims to efficiently manage voter registration, ensure secure and accurate vote casting, and facilitate result tabulation.
- The system should enable seamless registration of eligible voters, allowing them to provide accurate personal information and verify their identity securely.
- It should provide a user-friendly interface for creating and customizing electronic ballots, including options for multiple candidates or issues.
- The system must ensure the integrity and confidentiality of votes by employing robust encryption and authentication mechanisms. It should enable various methods of casting votes, such as in-person, mail-in, or online voting.
- The system should accurately and efficiently count the votes, aggregate the results, and generate comprehensive reports.
- The system should incorporate stringent security measures to prevent fraud, hacking, or unauthorized access.

1.2 Scope of Project

- Develop a comprehensive software solution for managing the voting process.
- Create a user-friendly interface for voter registration, identity verification, and secure voting.
- Implement a robust backend system to store and manage voter information securely.
- Enable ballot generation and distribution through the system.
- Ensure data integrity and security throughout the voting process.
- Provide administrative features for election officials, including candidate registration and ballot design.
- Enable real-time monitoring of polling stations and voting activities.
- Implement tools for accurate vote counting and result consolidation.

1.3 Purpose of Project

- The purpose of the "voting management system" project is to streamline and modernize the electoral process, ensuring efficient, transparent, and secure elections.
- The system aims to replace traditional paper-based voting methods with a digital platform that enables voters to cast their ballots electronically.
- The key objectives of the project include enhancing accessibility and convenience for voters by allowing them to vote from anywhere using their personal devices, reducing long queues at polling stations, and facilitating participation for individuals with disabilities or limited mobility.
- Moreover, the voting management system seeks to improve the accuracy and integrity of the electoral process by minimizing the potential for human error and fraud.
- It can incorporate advanced security measures such as encryption and authentication to safeguard the confidentiality and authenticity of each vote.
- The system provides efficient data management and analysis capabilities, allowing election officials to quickly compile and analyse the voting results. This speeds up declaring election outcomes and enhances public trust in the electoral system.
- Foster public confidence in the democratic process.

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

The purpose of this Software Requirement Specification (SRS) document is to outline the requirements for a note's application. The notes application aims to provide users with a convenient and efficient way to create, organize, and manage their personal notes. This document will cover the functional and non-functional requirements, as well as any constraints or assumptions associated with the development of the application.

2.1 Software Requirements

It is also important to consider the specific needs of your users when determining the hardware requirements for your Android notes application. Here are some of the software requirements for an Android notes application:

- Windows 10
- Code blocks, Visual studio
- Programming Languages: C++
- GCC compiler

2.2 Hardware Requirements

It is also important to consider the specific needs of your users when determining the hardware requirements for your Android notes application. Here are some of the hardware requirements for an Android notes application:

- PROCESSOR : Intel, Ryzen, Mac
- SPEED : 1 GHz
- RAM : 8 GB or more
- HARD DISK : 30 GB or more

2.3 Functional Requirements

A voting management system is a critical tool for ensuring the smooth and efficient conduct of elections. The functional requirements of such a system include:

- **User Registration:** The system should allow eligible voters to register and create their profiles, capturing essential information like name, address, and identification details.

- **Candidate Registration:** The system should provide a mechanism for candidates to register their candidature, including submission of required documents and information.
- **Voter Authentication:** The system should authenticate voters' identities to ensure only eligible individuals can participate in the voting process, using methods like biometric verification or unique identification codes.
- **Ballot Creation:** The system should facilitate the creation and management of electronic or physical ballots, including candidate lists and voting options.
- **Vote Casting:** The system should enable voters to cast their votes securely and anonymously, ensuring the integrity of the process.
- **Vote Counting:** The system should accurately count and tabulate the votes, providing real-time updates and generating final results.
- **Results Publication:** The system should publish the election results promptly and securely, making them accessible to the public.
- **Security Measures:** The system should incorporate robust security features to prevent tampering, hacking, or unauthorized access, safeguarding the integrity of the voting process.
- **Audit Trail:** The system should maintain an audit trail of all activities, ensuring transparency and accountability.
- **Accessibility:** The system should be accessible to voters with disabilities, providing accommodations like braille interfaces or audio assistance.

CHAPTER 3

SYSTEM DESIGN

The design of a voting management system encompasses the overall structure, functionality, and user interface of a software application that facilitates the electoral process. A well-designed voting management system ensures transparency, efficiency, and security throughout the entire voting process.

3.1 Algorithm

The algorithm for the voting management system is given below:

Step 1: Start.

Step 2: Print all options and select one:

- If 1, go to step 3.
- If 2, go to step 4.
- If 3, go to step 7.

Step 3: Login as an administrator:

- Enter username and password.
- Print admin options and select one:
 - If 1, go to step 5.
 - If 2, go to step 6.
 - If 3, go to step 2.

Step 4: Register a voter:

- Enter voter name and voter ID.
- Store details in a file.
- Update votes casted in respective files.
- Close the file and go to step 2.

Step 5: Display the results:

- Open the file.
- If the file does not open, print an error message, and go to step 7.

- Search for specific details and print the results. Close the file and go to step 3.

Step 6: Display voter details:

- Open the file.
- If the file does not open, print an error message and go to step 7.
- Print all details.
- Close the file and go to step 3.

Step 7: Stop.

3.2 Flowchart

A flowchart is a graphical representation of a process, using various symbols to depict the sequence of steps and decision points. Here is the flowchart for the voting management system.

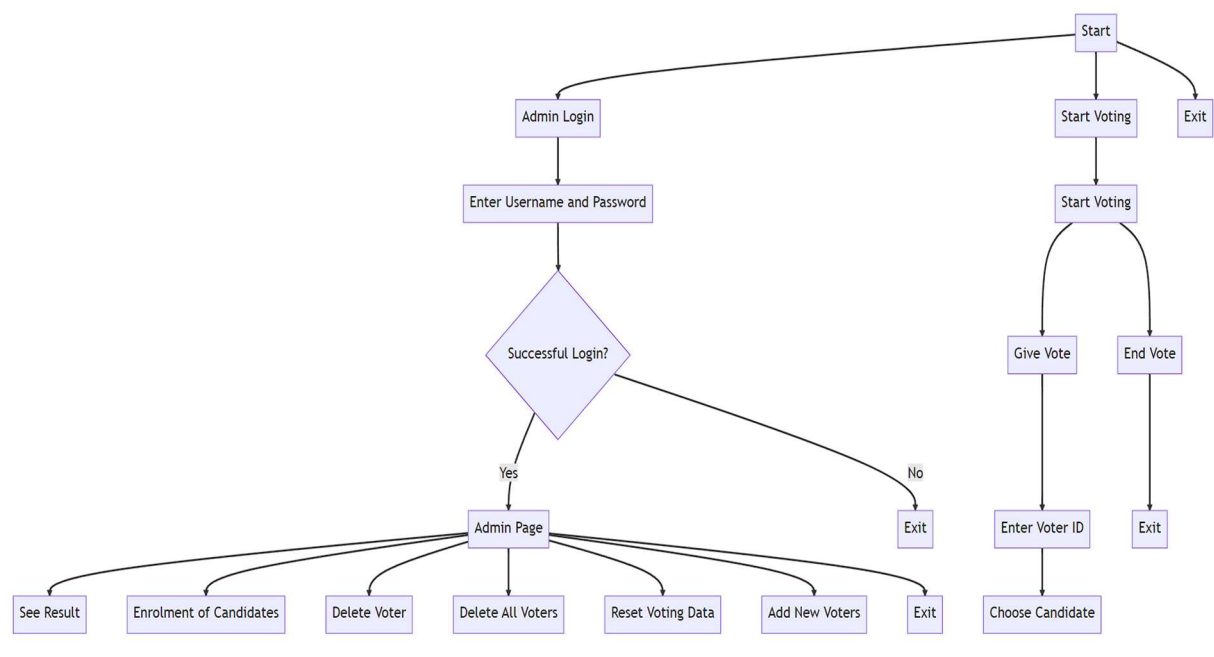


Figure 3.2 Flowchart of Voting Management System

Figure 3.2 shows the flowchart for a voting management system. A flowchart is a graphical representation of a process or system, using various symbols to depict the sequence of steps and decision points. Here is a description of a flowchart for a voting management system:

Start: The flowchart starts with this symbol, indicating the beginning of the process.

User Selection: The flowchart progresses to a decision symbol where users are presented with options. For example, they may choose between administrator login, voter registration, displaying results, or displaying voter details.

Admin Login: If the user selects the administrator login option, the flowchart moves to the symbol representing the login process. This step typically involves verifying the administrator's credentials and granting access to administrative functions.

Display Results: If the user selects the option to display results, the flowchart moves to the symbol representing the result display process. This step typically involves retrieving and presenting the voting results to the user.

Enrolment of Candidate: The process of enrolling new candidates in a voting management system plays a crucial role in ensuring the integrity and transparency of the democratic process. It involves collecting and verifying information about individuals who intend to run for various positions, such as local, regional, or national offices.

Delete Voters: It is a functionality in a voting management system that allows authorized users to remove a registered voter's information from the system's database. This feature is typically designed to be used in situations where a voter's registration needs to be invalidated or removed permanently for various reasons.

Voter Registration: If the user chooses to register a voter, the flowchart moves to the symbol representing the registration process. This step involves collecting voter information, such as name and ID, and storing it securely in the system.

Display Voter Details: If the user chooses to display voter details, the flowchart moves to the symbol representing the voter details display process. This step typically involves retrieving and presenting voter information, such as their name and voting history.

Enter Voter ID: Entering a voter ID refers to the process of providing and inputting the necessary information related to a voter identification card or document. A voter ID is a unique identification issued by an election commission or relevant authority to confirm an individual's eligibility to vote in an election or participate in the electoral process.

Give vote: It refers to the action or process through which an eligible individual casts their vote for a specific candidate or proposition in an election.

End Vote: It refers to the final step in a voting management system where the voting process is officially concluded. It signifies the end of the designated voting period and ensures that no further votes can be cast or altered.

End: The flowchart concludes with an end symbol, indicating the end of the process.

3.3 Use Case Diagram

A use case diagram for a voting management system provides a visual representation of the different interactions between the system and its users, showcasing the various actions or operations that can be performed within the system. It helps to identify the different roles or actors involved and illustrates how they interact with the system to achieve specific goals.

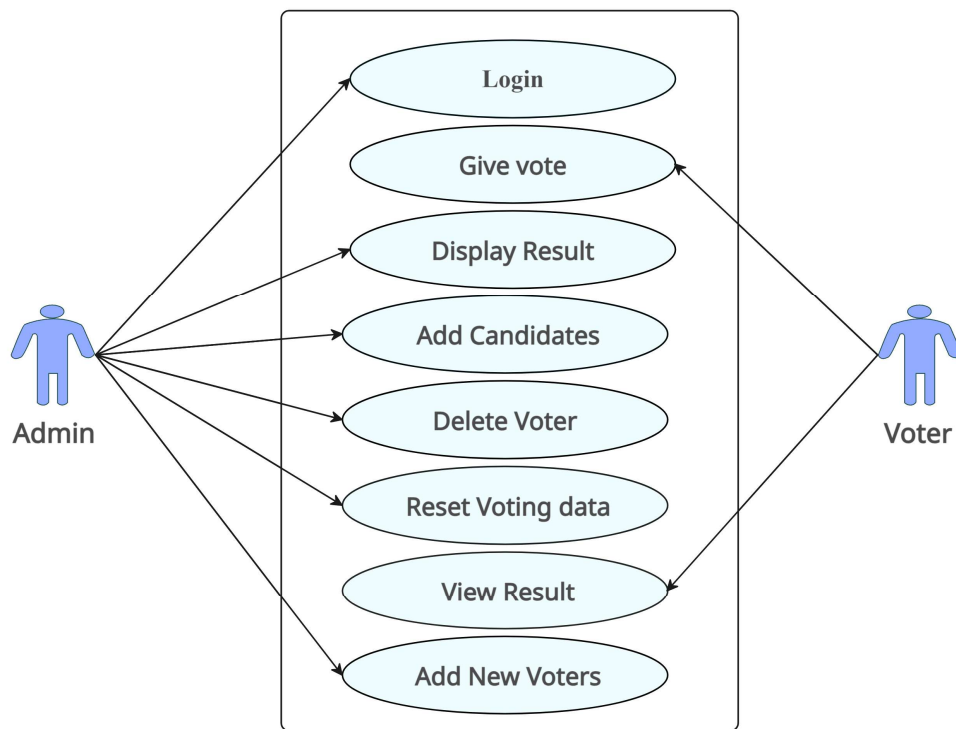


Figure 3.3 Use case diagram of voting management system

1. Actors

Administrator: The system user responsible for managing and overseeing the voting process.

Voter: An individual eligible to cast their vote in the election.

2. Use Cases

Login: The administrator and voters can log into the system using their credentials to access their respective functionalities.

Give Vote: The voter interacts with the system to cast their vote for a particular candidate or choice.

Display Results: The administrator can view and display the results of the election, showing the vote count for each candidate or choice.

Add Candidates: The administrator adds candidates to the system, providing necessary details such as name, party affiliation, and other relevant information.

Delete Voter: The administrator removes a registered voter from the system, typically done in exceptional cases or due to specific reasons.

Reset Voting Data: The administrator initiates this use case to reset the voting data, clearing all previous votes and preparing for a new election or restart.

View Result: The administrator and voters can view the current results of the ongoing or completed election.

Add New Voters: The administrator adds new voters to the system, registering them with their relevant information.

3. Relationship

Association: Both the administrator and voter actors are associated with all the use cases since they are involved in the corresponding actions.

Include: The "Give Vote" use case includes the "Login" use case since the voter needs to log in before casting their vote.

Extend: The "Display Results" use case can be extended by the "View Result" use case, allowing both the administrator and voters to view the election results.

CHAPTER 4

IMPLEMENTATION

Implementation is defined as specific set of activities designed to put into practice an activity or program of known dimensions. Implementation processes are purposeful and are described insufficient details such that independent can detect the presence and strength of the “specific set of activities” related to implementation.

4.1 Details of the language

The term programming languages usually refers to the high-level languages. Each programming languages has a unique set of keywords and special syntax for organizing program instructions. Programming languages is a set of commands, instructions and other syntax use to create software programs. Our project is implemented using Java programming language.

4.2 User defined functions

In a voting management system, user-defined functions play a crucial role in providing customized features and functionality tailored to the specific needs of the system. Here are some examples of user-defined functions that could be implemented:

- **User Registration:** This function allows new voters to register themselves by providing their personal information, such as name, address, and identification details. It verifies the eligibility of the voter and adds them to the voter database.
- **Ballot Creation:** This function enables administrators to create the ballot for an election. It allows them to define the candidates or options for each position or measure being voted upon.
- **Voter Authentication:** This function verifies the identity of the voters during the login process. It may include various authentication methods, such as username and password, biometric authentication, or two-factor authentication, to ensure secure access to the system.
- **Vote Casting:** This function allows registered voters to cast their votes. It presents the ballot options to the voter, validates their selections, and records their votes securely in the system.
- **Results Calculation:** This function calculates the results of the election based on the votes cast. It applies the appropriate counting method, such as plurality, majority, or proportional representation, to determine the winners for each position or measure.

- **Data Analytics:** This function provides analytical capabilities to analyze voting patterns, demographics, and trends. It generates reports and visualizations that help administrators gain insights into the election results and make informed decisions.
- **System Configuration:** This function allows administrators to configure various system settings, such as election dates, voting hours, maximum number of votes per voter, and other parameters, to customize the system according to specific requirements.
- **Voter Communication:** This function facilitates communication between the system and the voters. It can send notifications, reminders, and updates about upcoming elections, voting procedures, and important deadlines.

This code represents a simple voting management system. Let's go through the code in detail.

Adding Header Files

The code begins with several header file inclusions:

```
#include<iostream>
```

```
#include<fstream>
```

```
#include<conio.h>
```

- `iostream` is included to handle input and output streams.
- `fstream` is included to handle file operations.
- `conio.h` is included for using the `getch()` function, which reads a character directly from the console without echoing it.

Menu() Function

This function displays a menu to the user and takes their choice as input. It returns the choice.

The `menu()` function is defined as follows:

```
int menu()
{
    int a;
    cout<<"USER PANEL";
    cout << "\n\n[1] Admin \n\n";
    cout << "[2] Voter \n\n";
    cout << "[3] Exit \n\n";
```

```
    cout << " Your Choice: ";  
    cin >> a;  
    return a;  
}
```

Login() Function

This function prompts the user to enter a username and password. If the entered username is "admin" and the password is "12345", it displays a login successful message and calls the panel_1() function. Otherwise, it displays an error message and recursively calls itself to allow the user to try again. The login() function is defined as follows:

```
void login()  
{  
    string uName;  
    string uPass;  
    system("cls");  
    cout << "\n";  
    cout << "-----\n";  
    cout << "\tLOGIN\n";  
    cout << "-----\n\n";  
    cout << "Enter Username : ";  
    cin>>uName;  
    if(uName=="admin")  
    {  
        cout << "\nEnter Password : ";  
        cin>>uPass;  
        if(uPass=="12345")  
        {  
            cout<<"\nLogin successful.\n\n";  
            panel_1();  
        }  
    }  
    else{  
        cout<<"\nInvalid Password!";  
        login();  
    }  
}
```

```
        login();
    }
}
else{
    cout<<"\nInvalid Username.";
    login();
}
}
```

Voter() Function

This function handles the voting process. It prompts the user to enter their name and ID. It also performs validation to ensure that the ID is within a specific range (5555 to 5755). Then, the user is asked to choose a party for voting. The vote count for the chosen party is incremented in a file corresponding to that party. The voter's details (name, ID, and party) are written to a file called "details.txt". Finally, a success message is displayed. The voter() function is defined as follows:

```
void giveVote()
{
    cout << "    Give Vote" << endl;
    cout << "-----" << endl;
    string voterID;
    cout << "Enter Voter ID: ";
    cin >> voterID;
    ifstream voterFile(VOTER_FILE);
    string line;
    bool isVoterFound = false;
    while (getline(voterFile, line))
    {
        if (line.find(voterID) != string::npos)
        {
            isVoterFound = true;
            break;
        }
    }
}
```

```
    }  
}  
voterFile.close();
```

Admin() Function

This function displays a menu for the admin and returns the admin's choice. The admin() function is defined as follows:

```
void adminSection()  
{  
    cout << "Admin Section" << endl;  
    cout << "-----" << endl;  
    int choice;  
    while (true)  
    {  
        cout << endl;  
        cout << "ADMIN MENU" << endl;  
        cout << "-----" << endl;  
        cout << "1. See Result" << endl;  
        cout << "2. Enrollment of Candidates" << endl;  
        cout << "3. Delete Voter" << endl;  
        cout << "4. Delete All Voters" << endl;  
        cout << "5. Reset Voting Data" << endl;  
        cout << "6. Add New Voters" << endl;  
        cout << "7. Exit Admin Section" << endl;  
        cout << "-----" << endl;  
        cout << "Enter your choice (1-7): ";  
        cin >> choice;  
        clearScreen();  
        adminFile.close();  
    }  
}
```

Result() Function

This function reads the vote counts for each party from their respective files ("BJP.txt", "CONGRESS.txt", and "JDS.txt"). It then displays the vote counts for each party. The result() function is defined as follows:

```
void seeResult()
{
    cout << "*****" << endl;
    cout << "    This is the final result of the election " << endl;
    vector<Candidate> candidates = loadCandidates();
    ifstream partyVotesFile(PARTY_VOTES_FILE);
    while (getline(partyVotesFile, line)) {
        vector<string> partyVotesData = split(line, '|');
        string party = partyVotesData[0];
        int votes = stoi(partyVotesData[1]);
        cout << party << ": " << votes << " vote(s)" << endl;
    }
    partyVotesFile.close();
    cout << "-----" << endl;
    cout << "Candidates" << endl;
    cout << "-----" << endl;
    for (const Candidate& candidate : candidates)
    {
        cout << candidate.name << " (" << candidate.party << "): " << candidate.votes << " vote(s)" << endl;
    }
    cout << "-----" << endl;
}
```

Main() Function

This function displays the program title and then calls the panel() function to start the voting management system. The main() function is defined as follows:

```
int main()
{
    int choice;
```

```
while (true)
{
    cout << "-----\n";
    cout << "          V O T I N G   M A N A G E M E N T          \n";
    cout << "                                                                \n";
    cout << "                      S Y S T E M                      \n";
    cout << "-----\n";
    cout << "    1.Admin Page                                2.Start Voting    \n";
    cout << "                                                                \n";
    cout << "                                3.Exit the Page                    \n";
    cout << "-----\n";
    cout << "Enter your choice: ";
    cin >> choice;
    clearScreen();
    switch (choice)
    {
        case 1:
            adminSection();
            break;
        case 2:
            startVoting();
            break;
        case 3:
            cout << "Exiting the program. Goodbye!" << endl;
            return 0;
        default:
            cout << "Invalid choice!" << endl;
    }
}
```

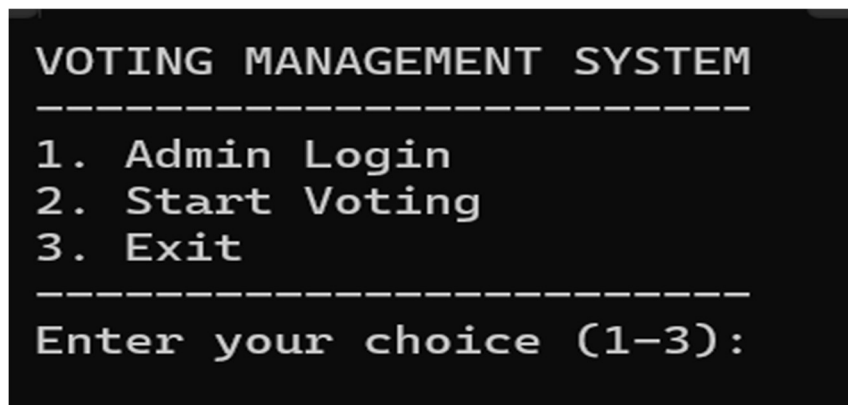
CHAPTER 5

RESULTS

Results refer to the outcome or summary of a voting process. It encompasses the data and information related to the votes cast by individuals, which are collected, analysed, and presented to determine the overall outcome of the election or poll. It provides a comprehensive picture of the voting process.

5.1.1 Home Page

Home page is the main page of this mini project. This page displays three options : Admin, Voter and Exit options.

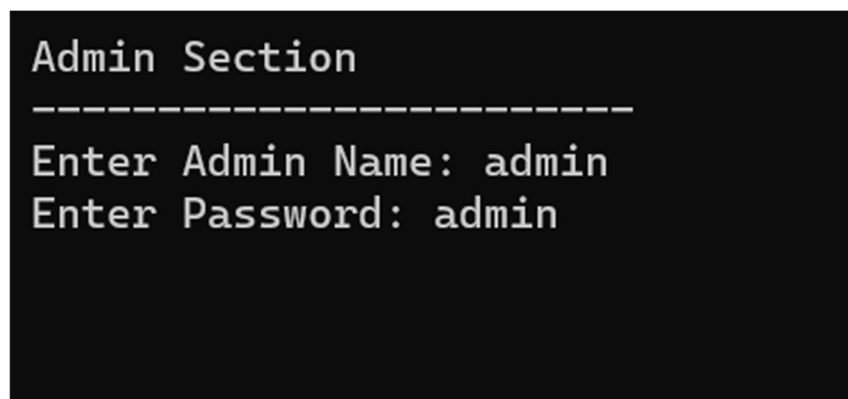
A screenshot of a terminal window displaying the 'VOTING MANAGEMENT SYSTEM' menu. The text is in a monospaced font on a black background. The menu lists three options: '1. Admin Login', '2. Start Voting', and '3. Exit'. Below the list, it prompts the user to 'Enter your choice (1-3):'.

```
VOTING MANAGEMENT SYSTEM
-----
1. Admin Login
2. Start Voting
3. Exit
-----
Enter your choice (1-3):
```

Figure 5.1.1: Home page

5.1.2 Admin Login Page

A Admin Login page is an entry page that requires user identification and authentication, regularly performed by entering a username and password combination.

A screenshot of a terminal window displaying the 'Admin Section' login prompt. The text is in a monospaced font on a black background. It shows the prompt 'Admin Section' followed by a separator line. Below, it shows the user has entered 'admin' for both the 'Admin Name' and 'Password' fields.

```
Admin Section
-----
Enter Admin Name: admin
Enter Password: admin
```

Figure 5.1.2: Admin Login Page

5.1.3 Admin Main Page

The Voting Management System encompasses several functionalities. The system allows the administrator to view voting results stored in a file, enrol new candidates by storing their details in a separate file, delete specific voters by removing their information from the voter file, delete all voters by emptying the voter file, and reset the system or add new voters by modifying the voter data file.

```
ADMIN MENU
-----
1. See Result
2. Enrollment of Candidates
3. Delete Voter
4. Delete All Voters
5. Reset Voting Data
6. Add New Voters
7. Exit Admin Section
-----
Enter your choice (1-7):
```

Figure 5.1.3: Admin main page

5.1.4 View Result Page

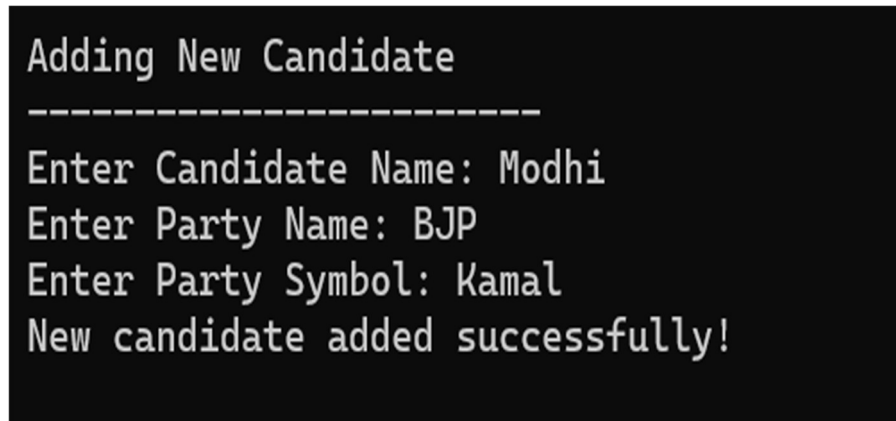
The See Result page in the Voting Management System allows administrators to view the voting results. By accessing a specific file, the page displays the data related to candidate names and their corresponding vote counts. This feature provides a quick and convenient way for administrators to monitor and analyse the outcome of the voting process.

```
C:\Users\hp\Desktop\allfs\du  X  +  v
Voting Result
-----
vongress: 1 vote(s)
BJP: 1 vote(s)
BJP: 1 vote(s)
-----
Candidates
-----
gandhi (vongress): 1 vote(s)
Modhi (BJP): 2 vote(s)
Others (AAP): 0 vote(s)
Kumarswami (JDS): 0 vote(s)
-----
```

Figure 5.1.4: View result Page

5.1.5 Candidate Enrolment Page

The enrolment of candidates feature allows the administrator to add new candidates to the Voting Management System. The candidate's details, such as name and party affiliation, are stored in a separate file for efficient organization. This functionality streamlines the process of managing and tracking candidates within the system.

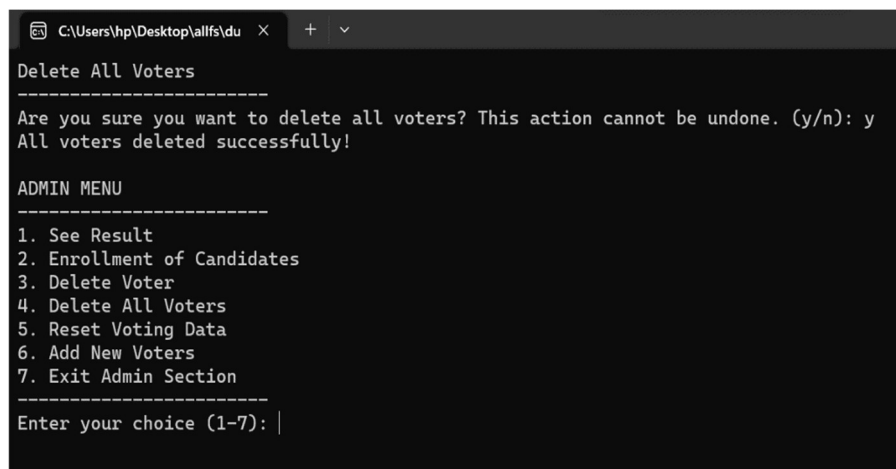


```
Adding New Candidate
-----
Enter Candidate Name: Modhi
Enter Party Name: BJP
Enter Party Symbol: Kamal
New candidate added successfully!
```

Figure 5.1.5: Candidate Enrolment Page

5.1.6 Delete all Voters Page

The Delete All Voters section in the Voting Management System's Admin page allows the administrator to remove all registered voters from the system. This functionality clears the entire voter database by emptying the voter file, ensuring a fresh start for managing the voting process. It provides a convenient option for resetting the system and starting a new.



```
C:\Users\hp\Desktop\allfs\du x + v
Delete All Voters
-----
Are you sure you want to delete all voters? This action cannot be undone. (y/n): y
All voters deleted successfully!

ADMIN MENU
-----
1. See Result
2. Enrollment of Candidates
3. Delete Voter
4. Delete All Voters
5. Reset Voting Data
6. Add New Voters
7. Exit Admin Section
-----
Enter your choice (1-7): |
```

Figure 5.1.6: Delete All Voters Page

5.1.7 Reset() Page

The Reset() section in the Voting Management System's Admin page allows the administrator to reset the system to its initial state. This functionality clears all data related to candidates, voters, and voting results, effectively restoring the system to its original configuration. It ensures a fresh start for the voting process, enabling the administrator to manage and conduct elections with updated and accurate information.

```
Reset Voting Data
-----
Are you sure you want to reset all voting data? This action cannot be undone. (y/n): y
Voting data reset successfully!

ADMIN MENU
-----
1. See Result
2. Enrollment of Candidates
3. Delete Voter
4. Delete All Voters
5. Reset Voting Data
6. Add New Voters
7. Exit Admin Section
-----
Enter your choice (1-7):
```

Figure 5.1.7: Reset Page

5.1.8 Add New Voters Page

The Add New Voters section in the admin page of the Voting Management System allows the administrator to add new voters to the system. This functionality involves modifying the voter data file, enabling the administrator to input and store details of new voters such as their names, identification numbers, and any other necessary information. This feature ensures the continuous expansion and update of the voter database within the file structure of the system.

```
Add New Voters
-----
Enter the number of voters to add: 2
Enter the Name of the Voter 1: Hrithik
Enter Voter ID of Hrithik: 1000
Enter Birthdate (dd/mm/yyyy) for Voter 1: 09/10/2001
Voter with ID 1000 added successfully!
Enter the Name of the Voter 2: Abhiram
Enter Voter ID of Abhiram: 2345
Enter Birthdate (dd/mm/yyyy) for Voter 2: 18/10/2002
Voter with ID 2345 added successfully!
New voters added successfully!
```

Figure 5.1.8: Add New voters Page

5.1.9 Start Voting Page

The Start Voting section enables the administrator to initiate the voting process. It includes functionalities such as activating the voting system, setting the start and end times for voting, and ensuring a secure and fair environment for voters. This section plays a crucial role in managing the overall flow and timing of the voting process within the Voting Management System.

```
Start Voting
-----

Start Voting Menu
-----
1. Give Vote
2. End Vote
-----
Enter your choice (1-2):
```

Figure 5.1.9: Start Voting Page

5.2.0 Give Vote Page

The Give Vote section enables users to cast their votes in the Voting Management System. Users can select their preferred candidate from the available options, and the system records their vote, updating the candidate's vote count in the file structure. This feature ensures a streamlined and user-friendly voting experience within the system.

```
Give Vote
-----

Enter Voter ID: 3444
Available Candidates:
1. gandhi (vongress)
2. Modhi (BJP)
Enter your choice (1-2): 2
Voter with ID 3444 has voted successfully for Modhi (BJP)
```

Figure 5.2.0: Give Vote Page

5.2.1 Already Voted Page

If the same voter ID is provided for a second vote, the system will display a message indicating that the voter has already cast their vote. This prevents individuals from voting multiple times and ensures the integrity of the voting process by enforcing a one-vote-per-voter policy.

```
Enter your Voter-ID: 12345
Voter Id: 12345, Name: Abhiram, DOB: 18-10-2002
You have already voted.
Press any key to continue..[]
```

Figure 5.2.1: Already Voted Page

5.2.2 Age limit page

When the user enters a birthdate, the system checks if the age derived from it is below 18 years. If so, the voter is skipped, and their voter ID is not included in the records. This ensures that only individuals who are 18 years or older are considered as eligible voters. The system effectively filters out underage voters, maintaining the integrity and compliance of the voter ID database. This process helps enforce age-based eligibility requirements for voting and ensures that only eligible individuals are included in the final list of voters.

```
C:\Users\hp\Desktop\allfs\du X + v
Add New Voters
-----
Enter the number of voters to add: 2
Enter the Name of the Voter 1: akil
Enter Voter ID of akil: 8889
Enter Birthdate (dd/mm/yyyy) for Voter 1: 18/06/2023
Voter with ID 8889 is not eligible to vote (age < 18). Skipping...
Enter the Name of the Voter 2: anil
Enter Voter ID of anil: 3445
Enter Birthdate (dd/mm/yyyy) for Voter 2: 32/13/2040
Invalid birthdate format! Skipping voter...
```

Figure 5.2.2: Age limit Page

5.2.3 Delete Voter

The delete voter section is a crucial component of a system that enables the removal of a voter's information based on their unique voter ID. This section interacts with a text file where voter data is stored, allowing users to conveniently delete voters from the system. By prompting the user to input the voter ID, the program performs a search within the text file to locate the corresponding voter entry.

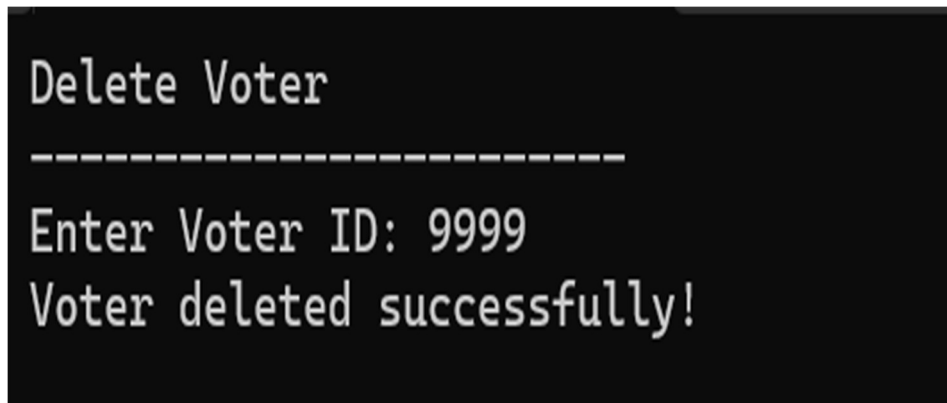


Figure 5.2.3: Delete Voter Page

5.2.4 Voters Text File

The voters text file is a file that stores information about voters. It contains data such as names, voter IDs, and ages of eligible individuals. Each line in the file represents a single voter and includes their relevant details separated by a delimiter. The file's format may vary, but it typically includes fields for full names, unique voter IDs, and ages. This file serves as a database for managing voter information and is used in election processes and related activities.

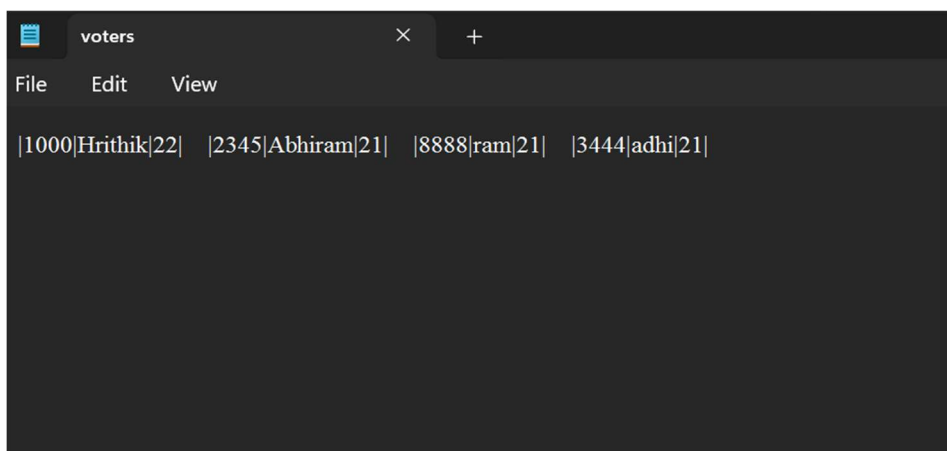


Figure 5.2.4: Voters text file

5.2.5 Votes Text File

The votes text file is a file format commonly used for storing voting records. It contains crucial information related to the electoral process, such as the voter's ID, the candidate's name, and the party name. Each line in the file represents a single vote and includes these three fields. The voter ID serves as a unique identifier for each voter, enabling proper tracking and identification of individual votes.

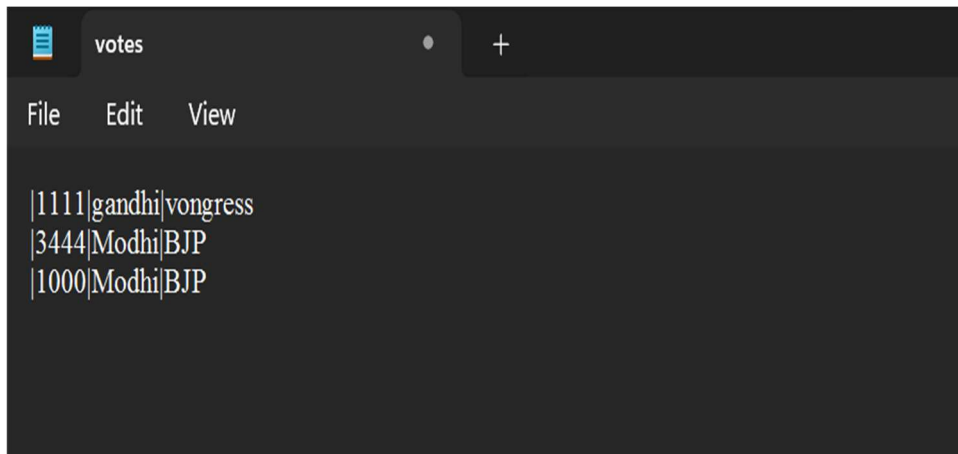


Figure 5.2.5: Votes text file

5.2.6 Candidates Text File

The candidates.txt file is a text file that stores information about voter candidates. Each line in the file represents a candidate and contains their name, party name, party symbol, and count of votes received. The information is separated by the "|" delimiter, allowing for easy parsing and organization of the data. This file structure enables efficient storage, retrieval, and analysis of candidate information during elections.

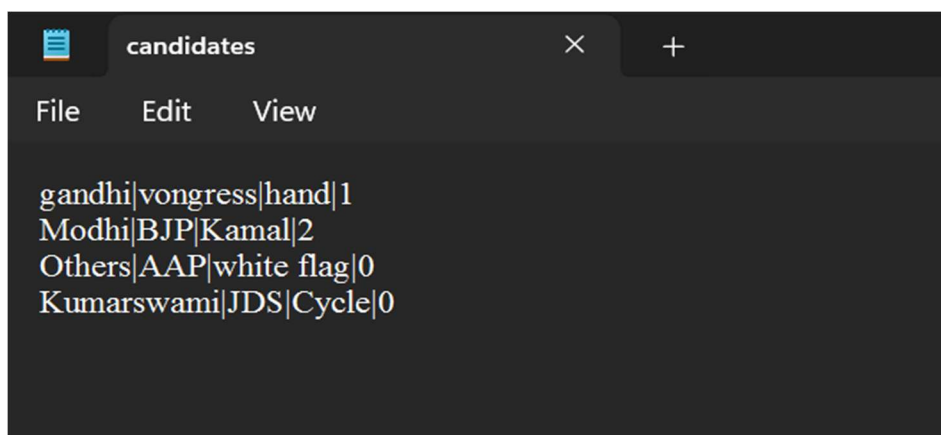


Figure 5.2.6: Candidates text file

CHAPTER 6

CONCLUSION AND FUTURE WORKS

In conclusion, the design of a voting management system is crucial for fair and efficient elections. However, the work does not end there. Future efforts should focus on improving security measures, enhancing accessibility and usability, exploring emerging technologies like blockchain, and fostering collaboration with stakeholders. Strengthening the system's security through continuous research and development is vital to stay ahead of potential threats. Improving accessibility and usability through user feedback and usability studies can make the system more inclusive. Exploring the integration of blockchain can enhance security, transparency, and verifiability. Collaboration with election authorities, cybersecurity experts, and stakeholders is essential for ongoing refinement and adaptation to changing needs and challenges. By prioritizing these future works, voting management systems can ensure elections remain fair, transparent, and trustworthy in an increasingly digital era.

REFERENCES

- [1] R. Krimmer, A. Ehringfeld, and M. Traxl, "The Use of E-Voting in the Austria Federation of Students Elections 2009,"
Internet: <https://pdfs.semanticscholar.org/6b8f/34a5bd3e7eabc7e3a9a3f008187e4415e26a.pdf>
[Nov. 26, 2018]
- [2] Tadayoshi kohno, Adam stubblefield, "Analysis of an Electronic Voting System", IEEE computer society press, July 2003.
- [3] A VISIONARY APPROACH TO SMART VOTING SYSTEM Rohit Srooa , Priyanshu Sinhaa , Ritwik Sharmaa , Parth Rustagia, and Moolchand Sharmab* a Student, Computer Science and Engineering, Maharaja Agrasen Institute of Technology, New Delhi, India
- [4] Z. A. Usmani, K. Patanwala, M. Panigrahi and A. Nair, "Multi-purpose platform independent online voting system," *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, India, 2017, pp. 1-5, doi: 10.1109/ICIIECS.2017.8276077.
- [5] <https://www.geeksforgeeks.org>
- [6] <https://youtu.be/wnb4laGrXug>
- [7] <https://youtu.be/Ma0P9T4nTDA>
- [8] https://en.wikipedia.org/wiki/Electronic_voting
- [9] <https://projectsgeek.com>